Notes on Numerical Fluid Mechanics
and Multidisciplinary Design 128

Norbert Kroll · Charles Hirsch
Francesco Bassi · Craig Johnston
Koen Hillewaert *Editors*

IDIHOM: Industrialization of High-Order Methods – A Top-Down Approach

Results of a Collaborative Research Project Funded by the European Union, 2010 – 2014

Springer

Notes on Numerical Fluid Mechanics and Multidisciplinary Design

Volume 128

About this Series

Notes on Numerical Fluid Mechanics and Multidisciplinary Design publishes state-of-art methods (including high performance methods) for numerical fluid mechanics, numerical simulation and multidisciplinary design optimization. The series includes proceedings of specialized conferences and workshops, as well as relevant project reports and monographs.

More information about this series at http://www.springer.com/series/4629

Norbert Kroll · Charles Hirsch
Francesco Bassi · Craig Johnston
Koen Hillewaert
Editors

IDIHOM: Industrialization of High-Order Methods – A Top-Down Approach

Results of a Collaborative Research Project Funded by the European Union, 2010 – 2014

Springer

Editors
Norbert Kroll
Institute of Aerodynamics
and Flow Technology
German Aerospace Center (DLR)
Braunschweig
Germany

Charles Hirsch
NUMECA International S.A
Brussels
Belgium

Francesco Bassi
Department of Engineering
University of Bergamo
Dalmine
Italy

Craig Johnston
Aircraft Research Association Limited
Bedford
United Kingdom

Koen Hillewaert
Cenaero ASBL
Gosselies
Belgium

ISSN 1612-2909 ISSN 1860-0824 (electronic)
Notes on Numerical Fluid Mechanics and Multidisciplinary Design
ISBN 978-3-319-12885-6 ISBN 978-3-319-12886-3 (eBook)
DOI 10.1007/978-3-319-12886-3

Library of Congress Control Number: 2014953524

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

Over the last two decades, computational fluid dynamics (CFD) has established itself to a point where it is widely accepted as an analysis and design tool complementary to theoretical considerations and experimental investigations. In aerospace, methods to solve the Navier-Stokes equations have matured from specialized research techniques to practical engineering tools being used on a routine basis in the industrial design process. Thus, numerical simulation has become one of the driving technologies for both, scientific discovery and industrial product development. Continuous development of physical models and numerical methods as well as the availability of increasingly powerful computers suggest using numerical simulation to a much greater extent than before, radically changing the way aircraft will be designed in the future. In order to meet the ambitious goals for air traffic of the next decades, however, significant investment in enhancing the capabilities and tools of numerical simulations in various aspects is required.

Within the 7th European Research Framework Programme, the collaborative target research project IDIHOM was initiated. The overall objective of this project was to enhance and mature adaptive high-order simulation capabilities for large scale industrial applications. Compared to its low-order counterparts, high-order methods have shown large potential to either increase the predictive accuracy related to the discretization error at given costs or to significantly reduce computational expenses for a prescribed accuracy. The IDIHOM project was driven by a top-down approach, in which dedicated enhancements and improvements of the complete high-order simulation framework, including grid generation, flow solver and visualization, were led by a suite of underlying and challenging test cases. The project gathered 21 partners from industry, research organizations and universities with well-proven expertise in high-order methods. It started at the end of October 2010 and finished in March 2014.

After a general project overview (chapter 1), this volume compiles technical papers of the IDIHOM partners, in which the major research activities and achievements are discussed (chapter 2). Research areas covered are the development and improvement of high-order grid generation techniques, high-order solution methods mainly based on continuous and discontinuous higher-order finite element methods and the extension of visualization techniques for high-order solutions. The final chapter presents the assessment of the high-order simulation capabilities developed within IDIHOM based on industrial application challenges and underlying test cases.

The editors would like to express their particular thanks to Dr. D. Knörzer, the European Commission's scientific officer of the IDIHOM project, for his constant technical interest and administrative help.

Thanks are due to all partners who have contributed in the context of the IDIHOM project in a very open and collaborative manner. The knowledge and

engagement of each individual contributed to the success and world wide appreciation of the IDIHOM project.
Finally, the editors would like to express gratitude to Dr. A. Michler for technical support in compiling this book. Acknowledgements are due to Prof. Dr. W. Schröder, the general editor of the Springer Series "Notes on Numerical Fluid Mechanics and Multidisciplinary Design", and to his colleague A. Hartmann for their help and editorial advice.

August 2014

Norbert Kroll
Charles Hirsch
Francesco Bassi
Craig Johnston
Koen Hillewaert

Table of Contents

Part I

The IDIHOM Project

The IDIHOM Project - Objectives, Project Structure and Research Activities

N. Kroll

German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7, D-38108 Braunschweig, Germany
Norbert.Kroll@dlr.de

Abstract. In aeronautical industry numerical flow simulation has become a key element in the aerodynamic design process. However, in order to meet the ambitious goals for air traffic of the next decades, significant investment in enhancing the capabilities and tools of numerical simulations in various aspects is required. Within the 7^{th} European Research Framework Programme, the collaborative target research project IDIHOM was initiated. The overall objective of this project was to enhance and mature adaptive high-order simulation methods for large scale applications. Compared to its low-order counterparts, high-order methods have shown large potential to either increase the predictive accuracy related to the discretization error at given costs or to significantly reduce computational expenses for a prescribed accuracy. The IDIHOM project was driven by a top-down approach, in which dedicated enhancements and improvements of the complete high-order simulation framework, including grid generation, flow solver and visualization, were led by a suite of underlying and challenging test cases. The project gathered 21 partners from industry, research organizations and universities with well-proven expertise in high-order methods. It started end October 2010 and finished March 2014.

1 Introduction

Over the last two decades, computational fluid dynamics (CFD) has established itself to a point where it is widely accepted as an analysis and design tool complementary to theoretical considerations and experimental investigations. In aerospace, methods to solve the Navier-Stokes equations have matured from specialized research techniques to practical engineering tools being used on a routine basis in the industrial design process. Thus, numerical simulation has become one of the driving technologies for both, scientific discovery and industrial product development. Continuous development of physical models and numerical methods as well as the availability of increasingly powerful computers suggest using numerical simulation to a much greater extent than before, radically changing the way aircraft will be designed in the future.

On the other hand, despite the progress made, large aerodynamic simulations of viscous high Reynolds number flows around complex aircraft configurations

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_1

are still by far too expensive in terms of turn-around time and computational resources. The requirement to achieve results at an engineering level of accuracy within wall-clock times that can be handled in the day-to-day procedure poses severe constraints on the application of CFD to aerodynamic design and data production. Improved flow physics modelling and the enormous increase in computer power is only one part of the solution. The other part is evidently dedicated to a necessary enhancement of the numerical methods themselves. The development of advanced CFD algorithms and their integration in an industrial environment to support multidisciplinary simulations and optimization procedures and at the same time achieving high predictive accuracy will significantly reduce design cycle and cost and is therefore indispensable for industry.

The majority of the aerodynamic simulation tools currently used in the aeronautical industry for routine applications are mainly based on finite volume methods. Being bound in most of the cases to second order discretization of the underlying governing equations, real-life applications require tens or hundreds of million mesh points to enable accurate solutions and to provide deep insight into complex flow features. In recent years there has been worldwide an ever increasing effort in the development of high-order CFD methods because of their potential in delivering either improved predictive accuracy or reduced computational cost compared to their low-order counterparts. Many different strategies have been followed and their pros and cons have been evaluated for a diverse range of academic problems ([3], [4]).

In Europe, the specific target research project ADIGMA (Adaptive Higher-Order Variational Methods for Aerodynamic Applications in Industry) was initiated within the 6th European Research Framework Programme [2], in order to add a major step towards the development of next generation CFD tools with significant improvements in accuracy and efficiency for turbulent aerodynamic flows. The main objective of the ADIGMA project (2006-2009) was the development of innovative high-order methods for the compressible flow equations in combination with reliable adaptive solution strategies, enabling mesh independent numerical simulations for aerodynamic applications. The project concentrated on technologies showing the highest potential for efficient high-order discretization on unstructured meshes, namely the Discontinuous Galerkin (DG) methods and the Continuous Residual Distribution (CRD) schemes. Although significant progress has been made in ADIGMA, it was finally concluded that many of the achievements in high-order simulation methods are still far away from industrial maturity. In general, the high-order methods investigated within ADIGMA demonstrated the potential for reducing the size of the discrete system by a factor of about 5-10 for most of the considered aerodynamic test cases, however, the gain could not fully transferred to increased runtime performance as limited attention has been paid to efficient solvers for large scale applications. On a given mesh, high-order discretization involves more floating point operations than do their low-order counterparts. Strong, memory-efficient solution algorithms are required to outperform well-tuned low-order methods. Furthermore, the ability of generating coarse unstructured meshes suitable for

high-order methods has been identified as a major bottleneck for industrial type applications. At the end of the ADIGMA research project it was recommended that further dedicated research effort is required to fully exploit the high-order methods for routine applications in aerospace industry [2].

2 Objectives

In 2011 the follow-on project IDIHOM (Industrialization of High-Order Methods) was initiated within the 7th European Research Framework Programme. It was motivated by the increasing demand of the European aerospace industries to improve their CFD-aided design procedure and analysis by using accurate and fast numerical methods. IDIHOM follows a top-down approach. A comprehensive suite of challenging application cases proposed by industry was set up prior to the project to direct dedicated development and improvement of high-order solvers at hand. The test case suite includes turbulent steady and unsteady aerodynamic flows, covering external and internal aerodynamics as well as aero-elastic and aero-acoustic applications. The complete process chain of the high-order flow simulation capability was addressed, with focus on flow solver efficiency and mesh generation capabilities, but including also visualization and coupling of CFD to other disciplines for multi-disciplinary analysis. The challenging application cases defined by the industry formed the basis for the demonstration and assessment of the current status of high-order methods as a workhorse for industrial applications. IDIHOM was assigned to help to close the gap between current expectations of what high-order methods are capable of and their strongly required use for industrial real-world applications - reaching out for improved, more accurate and time-saving design processes.

The main objectives to be achieved by gathering expertise from European experts in the field of numerical analysis in aerodynamics and their application to the multidisciplinary designs in aero-elastics and aero-acoustics are summarized as follows:

- Advance the maturity of current high-order methods, in particular Discontinuous Galerkin and Continuous Residual-Based approaches, and apply them to complex flows which are of particular interest for the aeronautical industry.
- Demonstrate the capabilities of high-order methods in solving industrially relevant, challenging applications and achieving synergy effects by combining requirements from external and internal aerodynamic flows. Aerodynamic flows to be investigated will focus both on complex steady and unsteady applications characterized by separation, wakes, and vortex interaction.
- Extend high-order methods to multi-disciplinary applications as there are aero-acoustics and aero-elastics, improving future cost-effective industrial designs.
- For the adaptive high-order methods employed, advance the Software Technology Readiness Level from about 3 (status at the end of the ADIGMA project) to 5.

– Facilitate co-operation between European industries, research establishments and universities and foster co-operation between different industries as there are airframe, turbo-engines, helicopters, and ground transportation.

Reaching these objectives will enable industry to use high-order methods in their daily work. Improved accuracy coupled with a reduction of computation and hence turn-around time is a pre-requisite for a routine employment of novel methods in future aircraft design cycles.

3 Partner Consortium

The IDIHOM consortium was comprised of 21 organizations with well-proven expertise in higher-order methods or underlying technologies. The following partners from European aircraft manufactures, small and medium enterprises, major European research establishments and universities were involved: Dassault Aviation (France), Airbus Defense and Space (Germany), CENAERO (Belgium), NUMECA (Belgium), DLR (Germany), FOI (Sweden), INRIA (France), NLR (The Netherlands), ONERA (France), TSAGI (Russia), VKI (Belgium), ARTS (France), Imperial College London (United Kingdom), University of Bergamo (Italy), University of Brescia (Italy), University of Stuttgart (Germany), Poznan University of Technology (Poland), Warsaw University of Technology (Poland), University of Linköping (Sweden), Université catholique de Louvain (Belgium).

The project was coordinated by the German Aerospace Center (DLR).

4 Project Structure and Main Activities

As mentioned already, the IDIHOM project followed a top-down approach to enhance the maturity of current high-order methods for routine use in aeronautical industry. Enhancement and improvement activities were identified which enable a robust, efficient and accurate computation of the comprehensive set of application challenges. The advancement of the complete set of the high-order environment was addressed to fulfil the industrial requirements, including mesh generation and adaptation, solver robustness and convergence acceleration, parallelization strategies for advanced computer hardware as well as flow visualization.

The technical work in IDIHOM was structured in three main work packages, all of them split into specific work tasks. Figure 1 sketches the project structure and exhibits the dependence of the project on the industrial test cases.

Work package 2 (WP2) was the key driver of the project, as it demonstrated and steered the top-down approach of IDIHOM. In a first task the pre-selected test cases were further specified in terms of geometry, flow and boundary conditions. Reference solutions using second-order finite volume methods and baseline solutions employing high-order methods available at project start were computed. These results were then used in a second task for detailed comparisons with solutions obtained with the improved high-order methodologies available at

the end of the project. Common test case templates were defined and issued by nominated test case coordinators, ensuring that reference, baseline and final results are properly compared, allowing a thorough assessment of the achievements and advances gained through the IDIHOM project.

The objective of **work package 3** (WP3) was to improve the high-order solvers with regard to the IDIHOM application challenges and the underlying test cases. After successful completion of the precursor project ADIGMA, focus was not put on the general development of high-order approaches, but more on the maturity of the implementation of codes at hand, thus extending the available capabilities of the solvers both with respect to physical modelling and the applicability to large-scale application cases. Five different areas for further enhancement and improvement were identified.

The first task dealt with improved physical modelling capabilities in the framework of high-order methods. The main aspect was not the development or assessment of an accurate and well-behaving turbulence model itself but rather an appropriate implementation of an explicit algebraic Reynolds stress turbulence model (EARSM) model in the high-order context. Reynolds stress turbulence models seem to be more reliable when being used for different applications, serving by this as a standard turbulence model. EARSMs were selected because of the promising compromise between accurate modelling of turbulence and moderate cost compared to both simpler models like Wilcox $k-\omega$ and more involved ones like differential Reynolds stress models.

The second task addressed extensions and improvements of selected high-order methods to unsteady turbulent flows guaranteeing both the accuracy and the overall performance of the solver. In the third task the assessment of the applicability of time accurate high-order methods in the context of scale resolving simulations was addressed. A major interest was put on hybrid RANS/LES methods which are seen to close the gap between RANS methods on one side and still expensive large eddy simulations (LES) methods on the other side. The work was based on results stemming from the European research project DESider [1], which was dealing with hybrid RANS/LES approaches in detail. It was well recognized that LES give rise to accurate simulation of vortical structures but rely on well-defined computational meshes as well as low-dissipation and low-dispersion schemes. High-order methods might be well suited for this type of applications.

The objectives of the last two activities were related to the improvement of solver efficiency for large-scale applications. At the beginning of IDIHOM, most of the high-order codes were either extremely memory consuming or CPU intensive. Thus activities were initiated to develop, implement and test various solution strategies (i.e. implicit schemes with appropriate preconditioners, h- and p-multigrid) that offer a reasonable compromise between reduced CPU usage and memory efficiency, thus enabling to run industrially relevant complex application cases on today's computer hardware. This also involved the investigation and implementation of parallelization strategies adapted to the specific features of high-order methods.

In **work package 4** (WP4) the enhancement of underlying technologies of high-order methods were addressed. Basically three aspects are of concern, in which high-order methods have special requirements. These are mesh generation, mesh adaptation and flow visualization. As high-order methods use relatively coarse meshes, straight-sided meshes, as typically used by second-order methods, are no longer able to represent smooth surfaces and obviously lack resolution of geometrical features. Therefore, high-order representation of the geometrical boundaries is required. Furthermore, for stretched elements, often encountered close to solid wall boundaries in aerospace applications, the volume mesh needs to be considered as curvilinear as well, at least close to the curved surface. For relevant aeronautical applications support for generation of high-order meshes is limited or not available at all in commercial mesh generation software. Therefore, specific development actions had been addressed, extending existing grid generation systems to provide suitable high-order meshes for the test cases considered in the project.

For high-order methods grid adaptation is an essential ingredient to become competitive with standard solvers. In particular for finite element based methods there is a strong theoretical background which can be exploited in order to derive local error estimates, which can in turn be used to drive a solution dependent mesh adaptation algorithm. To properly exploit this feature, several components have to be provided, among which are the provision of appropriate error indicators, the implementation of element subdivision or mesh regeneration for high-order meshes and the capability to recover the true geometry under mesh refinement. Based on the achievements gained in the ADIGMA project [2], necessary improvements were addressed to allow the final computations for some of the application challenges on flow-adapted meshes.

Visualizing solutions obtained with high-order methods using curved meshes is increasingly demanding, as the order of the local polynomial approximation spaces increases beyond the simple linear case. Hence, in order to close the process chain of high-order methods, limited effort was put on the investigations of high-order visualization techniques.

Test Cases

The test case suite selected for the IDHOM project can be split into four application areas: external and internal aerodynamics, aero-elastics and aero-acoustics. Two different types of test cases were defined. So called application challenges were mainly defined by industry and they are characterized by complex three-dimensional flows around complex geometries. A few underlying test cases were provided with moderate geometrical and/or flow complexity to speed-up the process of method enhancement in the first part of the project.

The test case suite was selected prior to project start. During the course of the project, some modifications and adaptations were made due to lack of reference results and detailed geometry definitions. Figures 2 to 5 list the test cases used in the IDIHOM project.

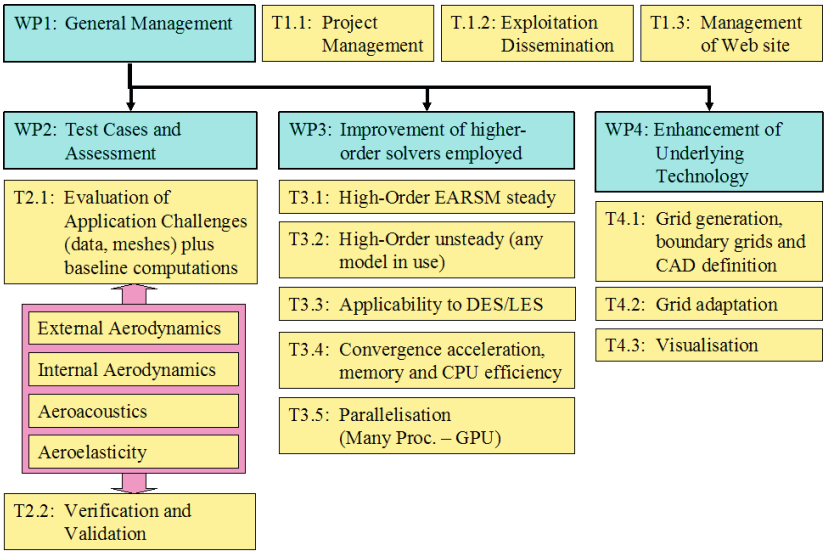
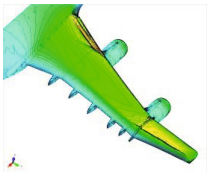
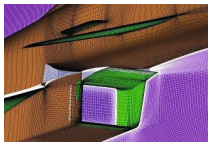


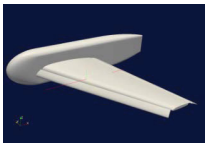
Fig. 1. Work package structure of the IDIHOM project



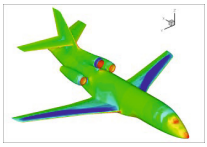
(a) **A.1** - Transport aircraft in cruise conditions (CleanSky)



(b) **A.2** - Fighter configuration in subsonic and transonic flow

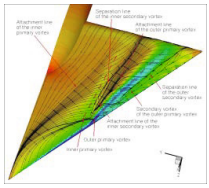


(c) **A.3** - AIAA high-lift prediction workshop test case (Trap wing)

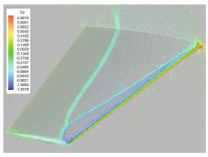


(d) **A.4** - Business jet in transonic flow

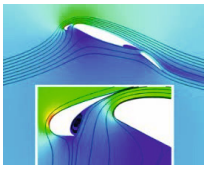
(e) **A.5** - High speed train head



(f) **U.1** - VFE-2 delta wing, high angle of attack



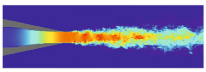
(g) **U.2** - ONERA M6 wing in subsonic and transonic flow



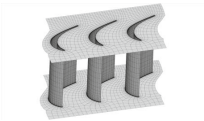
(h) **U.3** - L1T2 2D high-lift configuration

Fig. 2. External aerodynamic test cases (A.x: challenging applications, U.x : underlying test cases)

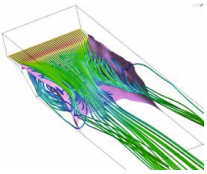
(a) **A.7** - NASA rotor 37



(b) **A.8** - Jean nozzle

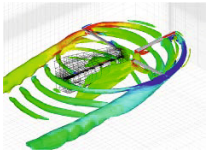


(c) **A.9** - MTU cascade

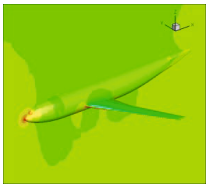


(d) **A.11** - DESider bump

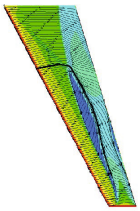
Fig. 3. Internal aerodynamic test cases



(a) **A.12** - HART II helicopter rotor

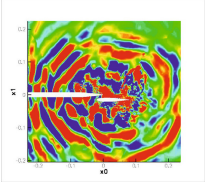


(b) **A.13** - DLR-F6 wing/body configuration

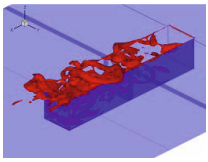


(c) **U.4** - LANN wing

Fig. 4. Aero-elastic test cases



(a) **A.14** - VALLIANT wing/flap interaction



(b) **A.15** - M219 cavity

Fig. 5. Aero-acoustic test cases

5 Conclusion

Within the 7$^{\text{th}}$ European Framework Programme the collaborative research project IDIHOM was set up to improve and demonstrate capabilities of high-order methods for large-scale applications in aeronautical industry. It followed a top-down approach, in which dedicated development and improvement activities for adaptive high-order methods were led by a comprehensive set of challenging and underlying test cases relevant for aeronautical industry. The complete process chain of the high-order flow simulation capability from grid generation,

via flow solver to visualization was addressed. The project gathered 21 partners from industry, research organizations and universities with well-proven expertise in development and utilization of high-order methods. It started end of 2010 and lasted 46 months.

In the following chapters of the book the developments and achievements obtained by the individual partners are presented in detail. Furthermore, for each IDIHOM test case the performance of the high-order methods is discussed. A final assessment of the IDIHOM project is given at the end.

References

1. Haase, W., Braza, M., Revell, A.: DESider – A European Effort on Hybrid RANS-LES Modelling. NNFM, vol. 103. Springer, Heidelberg (2009)
2. Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.): ADIGMA. NNFM, vol. 113. Springer, Heidelberg (2010)
3. Wang, Z.J.: High-Order Methods for the Euler and Navier-Stokes Equations on Unstructured Girds. Progress in Aerospace Science 43(1-3), 1–41 (2007)
4. Wang, Z.J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., Kroll, N., May, G., Persson, P.-O., van Leer, B., Visbal, M.: Visbal. High-order CFD methods: Current Status and Perspective. International Journal for Numerical Methods in Fluids 72(8), 811–845 (2013)

Part II

Research Activities

The Generation of Valid Curvilinear Meshes

C. Geuzaine[1], A. Johnen[1],
J. Lambrechts[2,3], J.-F. Remacle[2], and T. Toulorge[2,3]

[1] Université de Liège, Dept. of Electrical Engineering and Computer Science, B-4000 Liège, Belgium
{a.johnen,cgeuzaine}@ulg.ac.be
[2] Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), B-1348 Louvain-la-Neuve, Belgium
{thomas.toulorge,jonathan.lambrechts,jean-francois.remacle}@uclouvain.be
[3] Fonds National de la Recherche Scientifique, B-1000 Brussels, Belgium

Abstract. It is now well-known that a curvilinear discretization of the geometry is most often required to benefit from the computational efficiency of high-order numerical schemes in simulations. In this article, we explain how appropriate curvilinear meshes can be generated. We pay particular attention to the problem of invalid (tangled) mesh parts created by curving the domain boundaries. An efficient technique that computes provable bounds on the element Jacobian determinant is used to characterize the mesh validity, and we describe fast and robust techniques to regularize the mesh. The methods presented in this article are thoroughly discussed in Ref. [1, 2], and implemented in the free mesh generation software Gmsh [4, 12].

Keywords: High-order mesh, curvilinear mesh, geometry discretization, mesh validity, element Jacobian.

1 Introduction

There is a growing consensus in the computational mechanics community that state of the art solver technology requires, and will continue to require too extensive computational resources to provide the necessary resolution for a broad range of demanding applications, even at the rate that computational power increases. The requirement for high resolution naturally leads us to consider methods which have a higher order of grid convergence than the classical (formal) 2^{nd} order provided by most industrial grade codes. This indicates that higher-order discretization methods will replace at some point the current finite volume and finite element solvers, at least for part of their applications.

The development of high-order numerical technologies for engineering analysis has been underway for many years now. For example, Discontinuous Galerkin methods (DGM) have been thoroughly studied in the literature, initially in a theoretical context [5], and now from the application point of view [6]. In many contributions, it is shown that the accuracy of the method strongly depends on the accuracy of the geometrical discretization (see for instance Ref. [7]). It

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_2

is thus necessary to address the problem of generating high-order, curvilinear meshes. This article focuses on unstructured meshes, that make it possible to deal with the complex geometries involved industrial-grade problems. It is mainly dedicated to the theoretical and practical aspects of high-order mesh generation as they are implemented in the Gmsh mesh generator [4, 12], that is distributed as free software. An in-depth discussion of the techniques described hereafter, as well as numerous examples and applications, can be found in Ref. [1, 2].

The rest of the article is organized as follows. Section 3 gives an overview of high-order mesh generation methods available in the literature. In Section 3, we describe how high-order nodes are created, ordered in elements and placed on the curved boundaries. In Section 4, we address the topic of mesh validity and the reliable evaluation of element Jacobian determinants. Section 5 details the mesh regularization methods available in Gmsh. Finally, we shortly draw conclusions in Section 5.

2 Overview of High-Order Mesh Generation Methods

Although high-order numerical schemes have been a subject of intense research for many years, a comparatively small number of research groups have worked on high-order mesh generation methods, despite the necessity pointed out above. So far, efforts in this field have been focused on the problem of creating curvilinear meshes (i.e. meshes that follow the curvature of the domain boundaries defined by CAD models) that fulfill the validity requirements imposed by the dominant numerical methods (namely finite volumes and finite elements).

Two approaches can be considered for the generation of high-order meshes [8]: one can either try to create directly a valid high-order mesh through a high-order version of a classical meshing algorithm, or generate a first-order (i.e. straight-sided) mesh that is subsequently curved and regularized if invalidities are found. Both approaches are actually related, as they share the necessity of detecting and repairing the invalid parts of the mesh.

The procedures for detection and regularization of invalidities are much more complex and computationally expensive in the high-order case than for usual straight-sided meshes, but the curvilinear character of a high-order mesh is usually localized to the vicinity of curved boundaries. This is why the direct approach, that involves operations on high-order meshes in the whole domain, has been disregarded by most if not all authors in favor of the indirect approach that takes advantage of existing technologies.

2.1 Curving

After generation of the first-order mesh, the following step in the indirect approach is to curve the boundaries of the mesh according to the geometry of the problem. Model entities like model faces or model edges are supposed to be smooth manifolds that are defined through a parametrization. In engineering applications, the geometry of model entities is most often defined by a CAD

model, but discrete geometries that are common in biomedical applications can also be smoothly parametrized.

The mesh elements (or cells in the finite volume terminology) are usually defined in a Lagrangian manner by a set of nodes (or vertices). Thus, the curving procedure normally comes down to placing the high-order nodes corresponding to boundary edges and boundary faces on the exact geometry.

This part of the high-order mesh generation process is generally not considered as a crucial point, and little information can be found about it in the literature. Simple techniques for obtaining high-order boundary nodes include interpolating them between the first-order boundary nodes in the parametrization describing the corresponding CAD entity [8, 9] and projecting them on the geometry from their location on the straight-sided element.

More sophisticated procedures have also been proposed. In Ref. [10], the high-order nodes on boundary edges are interpolated in the physical space through a numerical procedure involving either the CAD parametrization (in the case of a mesh edge corresponding to an edge of the geometric model), or an approximation of the geodesic connecting the two first-order vertices (in the case of an edge located on a 3D surface). Nodes located within surface elements are obtained through a more sophisticated version of this procedure. Instead of interpolation, Sherwin and Peiró [11] use a mechanical analogy with chains of springs in equilibrium that yields the optimal node distribution along geometric curves and geodesics for edge nodes. Two-dimensional nets of springs provide the optimal distribution of surface element nodes.

These procedures generally aim to obtain a given node distribution on the boundary of the computational domain. However, it may be more interesting to search for the node locations that minimize the error in the approximation of the exact geometric by the high-order mesh boundary, for instance in terms of the Hausdorff distance. To our knowledge, this topic point has not yet been explored in the context of high-order mesh generation for scientific computation.

The creation and placement of high-order nodes is treated in detail in Section 3.

2.2 Detection of Invalidities

As illustrated in Section 3, curving the mesh boundaries while leaving other mesh entities unchanged may result in an invalid mesh with tangled elements. From the point of view of numerical schemes, a mesh is generally considered as valid if it is possible to integrate a quantity over each element, and if elements do not overlap. Both requirements can be evaluated through the Jacobian determinant of a transformation between a canonical reference element and the actual element of the mesh, as in the finite element framework (see Section 4.1). The validity depends on the sign of the Jacobian.

In order to characterize the quality of a high-order mesh independently of the size of its elements, authors have used different *distortion* measures based on the Jacobian determinant. Most of them involve the ratio of minimum to maximum Jacobian in each element [8, 10–14], the ratio of the minimum Jacobian to the

Jacobian of the corresponding straight-sided element [1] or the integral of the Jacobian divided by the Jacobian of the straight-sided element [15].

It is important to note that the Jacobian determinant of an element is a higher-order polynomial of the coordinates in the reference space. As such, its oscillatory nature makes it difficult to evaluate its extrema (and thus its sign) by sampling. Several authors have developed procedures based on the convex hull properties of Béziers polynomials for the reliable evaluation of Jacobian bounds [1, 13, 16]. This point is the object of Section 4.

2.3 Regularization of Invalid Meshes

In case invalid elements are detected in a high-order mesh, a procedure to regularize the mesh (i.e. untangle the broken elements) has to be applied. This is arguably the most important and problematic part of high-order mesh generation. On one hand, the procedure has to be robust, because a single invalid element can make the mesh inappropriate for computation. On the other hand, the regularization procedure shall not be much more computationally intensive than the rest of the meshing process, in order to remain suitable for practical applications. Several types of methodologies have been proposed in the literature.

A first possibility is to apply local refinement to the mesh in the regions where the boundary curvature is considered as large enough to possibly create invalid elements [11]. It has also been proposed to move the high-order nodes along specific lines depending on geometric constraints and neighbouring elements [13]. More sophisticated methodologies combine a prescribed deformation of elements near curved boundaries (by moving the control points of their Bézier representation) with topological modifications in order to untangle all invalid elements [8, 16, 17].

A second class of regularization methods are based on mechanical analogies: the mesh is considered as embedded in an elastic medium, and the imposition on the boundaries of a displacement corresponding to the curving results in a problem that can be solved using finite elements. No topological modification is made. The simple linear elastic analogy has been employed [10], but it may prove unreliable. A more robust method may be obtained by using non-uniform material properties, that is by using stiffer material in small elements near curved boundaries. More successful strategies have been proposed, including a non-linear mechanics formulation [14] and an iterative method working on the Bézier control points of the elements instead of the Lagrangian nodes [12].

The last approach for regularization methods is based on optimization algorithms. The point is then to find the mesh node locations which minimize an objective function that characterizes the validity of the mesh, using the measures described in Section 2.2. In Ref. [9], such a method is applied with a extended distortion measure on triangular surface meshes. In a recent paper [2], we have taken advantage of the control allowed by this approach on the validity of the mesh to propose a robust untangling method, that is detailed in Section 5.2.

3 Creation of High-Order Meshes

3.1 High-Order Elements

When dealing with high order elements, it is important to define how element nodes are numbered for a given element shape and for a given order p. Our convention is very general with respect to both those parameters. For all mesh and post-processing purposes, reference elements are defined as in a classical finite element framework.

We start by considering linear elements (lines, triangles, quads, tets, prisms, hexes and pyramids), that contain only corner vertices. In 2D elements, the nodes are numbered in such a way that each edge connects two consecutive nodes, the normal to the oriented edge pointing outwards. In 3D elements, the edge-vertex connectivity does not follow a general rule. Instead, the node ordering is such that of the normal to each face of the element points outward.

The node ordering of a higher order (possibly curved) element is a generalization of the one used in the first-order element. We number nodes in the following order:

- the element corner vertices (i.e. the first-order vertices);
- the internal nodes for each edge;
- the internal nodes for each face;
- the volume internal nodes.

The numbering for internal nodes is recursive, ie. the numbering follows that of the nodes of an embedded edge/face/volume of lower order. The higher order nodes are assumed to be equispaced.

A visual description of the node ordering used in the Gmsh mesh generation software can be found in its documentation [4].

3.2 Generating High Order Meshes Based on CAD Data

Modern mesh generation procedures take as input CAD models composed of *model entities*: vertices G_i^0, edges G_i^1, faces G_i^2 or regions G_i^3. Each model entity G_i^d has a geometry (or shape) [12,18] for which solid modelers usually provide a parametrization, that is, a mapping $\boldsymbol{\xi} \in R^d \mapsto \boldsymbol{x} \in R^3$. There are also four kind of mesh entities M_i^d that are said to be classified on model entities. Each mesh entity is classified on the model entity of the smallest dimension that contains it. The way of building a high order mesh is to first generate a straight sided mesh. Then, mesh entities (edges, faces and regions) are curved according to the geometry of the CAD entity it is classified on.

In the p-version of finite elements, high order nodes are added to edges, faces and regions of the element with the aim of creating curvilinear elements with their shape based on high order (Lagrangian or not) polynomial bases (see Figure 1). Other authors [19,20] would rather use the exact mappings of the geometry and build a so-called isogeometric mesh.

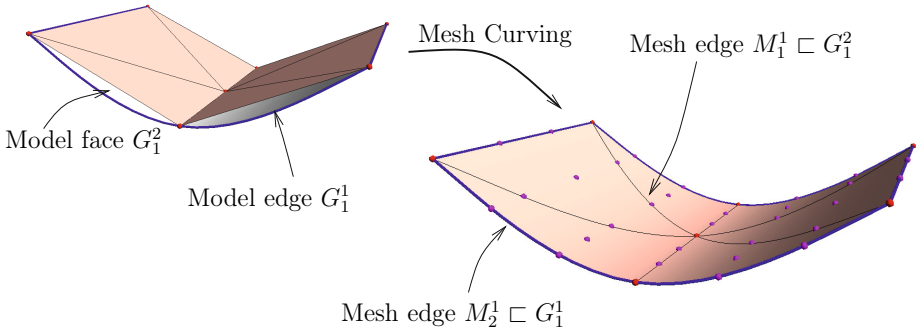


Fig. 1. Straight sided mesh (left) and curvilinear (cubic) mesh (right)

The naive curving procedures described above does not ensure that all the elements of the final curved mesh are valid. Figure 2 gives an illustration of this important issue: some of the curved triangles are tangled after having been curved. It is important to note that this problem is not related to the accuracy of the geometrical discretization: in Figure 2, the mesh would not be valid in the iso-geometric case i.e. if the curved edge was assigned the exact geometry (blue curve).

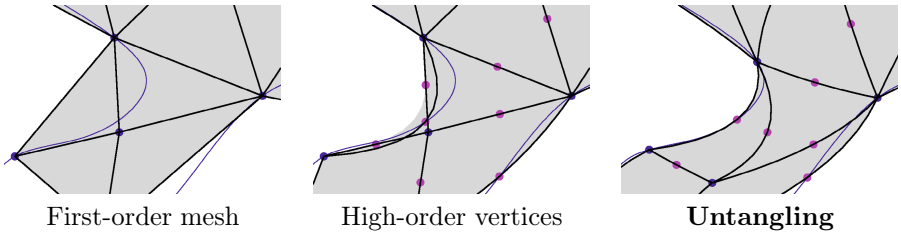


Fig. 2. Straight sided mesh (left) basic curvilinear (quadratic) mesh with tangled elements (center) and untangled mesh (right)

Gmsh implements the polynomial version of high order meshes. There, the choice of the position of high order nodes is a crucial parameter for the validity of the high order elements.

The position of high order nodes can be chosen by linear interpolation between the first-order nodes in the parametric space provided by the solid modeler for the corresponding model entity. This procedure has the advantage of being simple, fast and geometrically robust (no projection on surfaces is needed). Yet, ill conditioned mappings are very common in CAD systems. Such a procedure will generate very bad elements in the vicinity of singularities (e.g. the two poles of a sphere described using spherical coordinates).

Another option would be to place new nodes in such a way that the geometrical error, i.e. the distance between the CAD model and the mesh, is minimized. Note that the definition of such a distance is not trivial and that the non linear procedure involved in such a minimization is time consuming.

The way Gmsh generates high order nodes (order p) on a mesh made of simplices (triangles and tets) is the following one:

- For every model edge G_i^1, add $p-1$ high order nodes on mesh edges M_j^1 that are classified on G_i^1,
- For every model face G_i^2, add $p-1$ high order nodes on mesh edges M_j^1 that are classified on G_i^2,
- For every model region G_i^3, add $p-1$ high order nodes on mesh edges M_j^1 that are classified on G_i^3,
- For every model face G_i^2, add $(p-1)\times(p-2)/2$ high order nodes on mesh faces M_j^2 that are classified on G_i^2,
- For every model region G_i^3, add $(p-1)\times(p-2)/2$ high order nodes on mesh faces M_j^2 that are classified on G_i^3,
- For every model region G_i^3, add $(p-1)\times(p-2)\times(p-3)/6$ high order nodes on mesh regions M_j^3 that are classified on G_i^3.

When an edge going from $\boldsymbol{x}_a$ to $\boldsymbol{x}_b$ is saturated with its $p-1$ high order points $\boldsymbol{x}_i$, those are initially placed on a straight line segment between $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ in an equispaced manner. Then, points $\boldsymbol{x}_i$ are orthogonally projected on the real geometry, using the functions provided by CAD systems. When an internal edge of a region is saturated with high order points, no projection is needed.

When high order points are added on mesh faces classified on model faces, a different procedure is applied. An incomplete element that contains corner nodes and edge nodes is constructed. Those elements are sometimes called serendipity (or incomplete) elements [21]. Gmsh's way of numbering nodes allows us to easily generate such elements. Incomplete elements are not converging optimally to the exact geometry when the mesh is refined. Internal nodes have to be added to achieve optimal convergence. Shape functions of the incomplete elements are used to place the internal nodes on the geometry of the curvilinear incomplete element. Then, those nodes are projected on the model face. The same procedure is used for inserting mesh vertices inside high order regions. Yet, no projection is needed in this case. The use of incomplete elements to predict the position of high order nodes before projection has experimentally given better results than a simple projection. Yet, there is no guarantee that such a procedure produces valid high order meshes.

4 Validity of High-Order Meshes

4.1 Curvilinear Meshes, Distortion and Bounds on Jacobian Determinants

Let us consider a mesh that consists of a set of *straight-sided* elements of order p. Each element is defined geometrically through its nodes $\mathbf{x}_i$, $i = 1, \ldots, N_p$ and

a set of Lagrange shape functions $\mathcal{L}_i^{(p)}(\boldsymbol{\xi}),\ i = 1, \ldots, N_p$. The Lagrange shape functions (of order p) are based on the nodes $\mathbf{x}_i$ and allow to map a reference unit element onto the real one:

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{N_p} \mathcal{L}_i^{(p)}(\boldsymbol{\xi})\ \mathbf{x}_i. \tag{1}$$

The mapping $\mathbf{x}(\boldsymbol{\xi})$ should be injective, which means that it should admit an inverse. This implies that the Jacobian determinant $\det \mathbf{x}_{,\boldsymbol{\xi}}$ has to be strictly positive. In all what follows we will always assume that the straight-sided mesh is composed of well-shaped elements, so that the positivity of $\det \mathbf{x}_{,\boldsymbol{\xi}}$ is guaranteed. This standard setting is presented on Figure 3 (left) for the quadratic triangle.

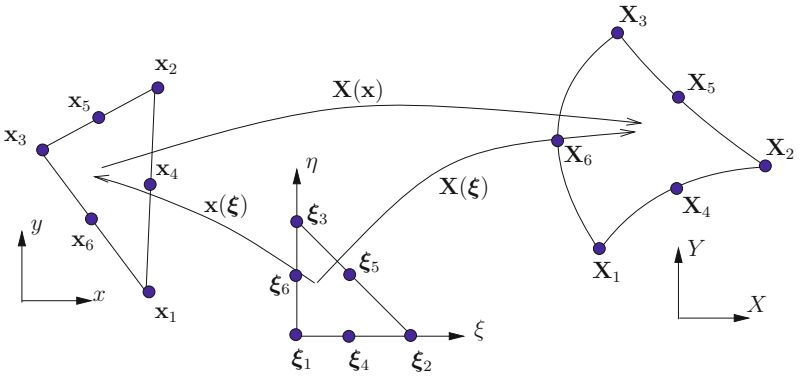


Fig. 3. Reference unit triangle in local coordinates $\boldsymbol{\xi} = (\xi, \eta)$ and the mappings $\mathbf{x}(\boldsymbol{\xi})$, $\mathbf{X}(\boldsymbol{\xi})$ and $\mathbf{X}(\mathbf{x})$

Let us now consider a *curved* element obtained after application of the curvilinear meshing procedure, i.e., after moving some or all of the nodes of the straight-sided element. The nodes of the deformed element are called $\mathbf{X}_i$, $i = 1 \ldots N_p$, and we have

$$\mathbf{X}(\boldsymbol{\xi}) = \sum_{i=1}^{N_p} \mathcal{L}_i^{(p)}(\boldsymbol{\xi})\ \mathbf{X}_i. \tag{2}$$

Again, the deformed element is assumed to be valid if and only if the Jacobian determinant $J(\boldsymbol{\xi}) := \det \mathbf{X}_{,\boldsymbol{\xi}}$ is strictly positive everywhere over the $\boldsymbol{\xi}$ reference domain. The Jacobian determinant J, however, is not constant over the reference domain, and computing $J_{\min} := \min_{\boldsymbol{\xi}} J(\boldsymbol{\xi})$ is necessary to ensure positivity.

The approach that is commonly used is to sample the Jacobian determinant on a very large number of points. Such a technique is however both expensive and not fully robust since we only get a necessary condition.

It is possible to follow a different approach: because the Jacobian determinant J is a polynomial in $\boldsymbol{\xi}$, J can be interpolated exactly as a linear combination of specific polynomial basis functions over the element. We would then like to obtain provable bounds on $J_{\min}$ by using the properties of these basis functions.

In addition to guaranteeing the geometrical validity of the curvilinear element, we are also interested in quantifying its *distortion*, i.e., the deformation induced by the curving. To this end, let us consider the transformation $\mathbf{X}(\mathbf{x})$ that maps straight sided elements onto curvilinear elements (see Figure 1). It is possible to write the determinant of this mapping in terms of the $\boldsymbol{\xi}$ coordinates as:

$$\det \mathbf{X}_{,\mathbf{x}} = \frac{\det \mathbf{X}_{,\boldsymbol{\xi}}}{\det \mathbf{x}_{,\boldsymbol{\xi}}} = \frac{J(\boldsymbol{\xi})}{\det \mathbf{x}_{,\boldsymbol{\xi}}}. \tag{3}$$

We call $\mathbf{X}(\mathbf{x})$ the distortion mapping and its determinant $\delta(\boldsymbol{\xi}) := \det \mathbf{X}_{,\mathbf{x}}$ the distortion. The distortion δ should be as close to $\delta = 1$ as possible in order not to degrade the quality of the straight sided element. Elements that have negative distortions are of course invalid but elements that have distortions $\delta \ll 1$ or $\delta \gg 1$ lead to some alteration of the conditioning of the finite element problem. In order to guarantee a reasonable distortion it is thus necessary to find a reliable bound on $J_{\min}$ and $J_{\max} := \max_{\boldsymbol{\xi}} J(\boldsymbol{\xi})$ over the whole element.

Note that many different quality measures can be defined based on the Jacobian determinant J. For example, one could look at the Jacobian determinant divided by its average over the element instead of looking at the distortion. Obtaining bounds on $J_{\min}$ and $J_{\max}$ is thus still the main underlying challenge. Other interesting choices are presented and analyzed in [15].

4.2 Adaptive Bounds for Arbitrary Curvilinear Finite Elements

In order to explain the adaptive bound computation let us first focus on the one-dimensional case, for "line" finite elements. Since Bézier functions can be generated for all types of common elements (triangles, quadrangles, tetrahedra, hexahedra and prisms), the generalization to 2D and 3D elements will be straightforward.

The One-Dimensional Case. In 1D the Bézier functions are the Bernstein polynomials:

$$\mathcal{B}_k^{(n)}(\xi) = \binom{n}{k} (1-\xi)^{n-k}\, \xi^k \qquad (\xi \in [0,1]\ ;\ k = 0, ..., n) \tag{4}$$

where $\binom{n}{k} = \frac{n!}{n!(n-k)!}$ is the binomial coefficient. The Bézier interpolation requires $n+1$ control values b_i. We have

$$J(\xi) = \sum_{k=0}^{N_n} \mathcal{B}_k^{(n)}(\xi)\ b_k. \tag{5}$$

Bernstein-Bézier functions have the nice following properties : (i) they form a partition of unity which means that $\sum_{k=0}^{n} \mathcal{B}_k^{(n)}(\xi) = 1$ for all $\xi \in [0,1]$ and (ii) they are positive which means that $\mathcal{B}_k^{(n)}(\xi) \geq 0$ for all $\xi \in [0,1]$. This leads to the well known property of Bézier interpolations:

$$\min_{\xi\in[0,1]} J(\xi) \geq b_{\min} = \min_i b_i \quad \text{and} \quad \max_{\xi\in[0,1]} J(\xi) \leq b_{\max} = \max_i b_i. \tag{6}$$

Moreover, the control values related to the corner nodes of the element are equal to the values of the interpolated function. In what follows we assume that these "corner" control values are always ordered at the K_f first indices. We then have

$$\min_{\xi\in[0,1]} J(\xi) \leq \min_{i<K_f} b_i \quad \text{and} \quad \max_{\xi\in[0,1]} J(\xi) \geq \max_{i<K_f} b_i. \tag{7}$$

Since Lagrange and Bézier functions span the same function space, computation of the Bézier values b_i from the nodal values J_i (and convertly) is done by a transformation matrix. The tranformation matrix $\boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}}$, which computes nodal values from control values, is created by evaluating Bézier functions at sampling points:

$$\boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}} = \begin{bmatrix} \mathcal{B}_0^{(n)}(\xi_0) & \dots & \mathcal{B}_n^{(n)}(\xi_0) \\ \mathcal{B}_0^{(n)}(\xi_1) & \dots & \mathcal{B}_n^{(n)}(\xi_1) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_0^{(n)}(\xi_n) & \dots & \mathcal{B}_n^{(n)}(\xi_n) \end{bmatrix}.$$

Those sampling points are taken uniformly, *i.e.* at the location of the nodes of the element of order n. The inverse transformation is $\boldsymbol{T}^{(n)}_{\mathcal{L}\to\mathcal{B}} = {\boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}}}^{-1}$ and from the expression of the interpolation of the Jacobian determinant (5), we can write

$$\begin{aligned} J &= \boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}}\, B \\ B &= \boldsymbol{T}^{(n)}_{\mathcal{L}\to\mathcal{B}}\, J\,, \end{aligned} \tag{8}$$

where B and J are the vectors containing respectively the b_i's and the J_i's.

Adaptive Subdivision. Let us assume that the domain $[0,1]$ is subdivided into Q parts. The interpolation $J^{[q]}(\xi^{[q]})$ on the q^{th} subdomain $[a,b]$ $(0 \leq a < b \leq 1)$ must verify

$$J^{[q]}(\xi^{[q]}) = \sum_{k=0}^{N_n} \mathcal{B}_k^{(n)}(\xi^{[q]})\; b_k^{[q]} = \sum_{k=0}^{N_n} \mathcal{B}_k^{(n)}(\xi(\xi^{[q]}))\; b_k \qquad (\xi^{[q]} \in [0,1]), \tag{9}$$

with $\xi(\xi^{[q]}) = a + (b-a)\;\xi^{[q]}$.

Considering the nodes $\xi_k^{[q]}$ such that $\xi_k^{[q]} = \xi_k$ $(k = 0, \ldots, n)$ (i.e., such that they are ordered like the sampling points), the expression (9) reads

$$\boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}}\, B^{[q]} \;=\; \begin{bmatrix} \mathcal{B}_0^{(n)}(a+(b-a)\,\xi_0) & \ldots & \mathcal{B}_n^{(n)}(a+(b-a)\,\xi_0) \\ \mathcal{B}_0^{(n)}(a+(b-a)\,\xi_1) & \ldots & \mathcal{B}_n^{(n)}(a+(b-a)\,\xi_1) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_0^{(n)}(a+(b-a)\,\xi_n) & \ldots & \mathcal{B}_n^{(n)}(a+(b-a)\,\xi_n) \end{bmatrix} B \;=\; \boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}}{}^{[q]}\, B,$$

where $B^{[q]}$ is the vector containing the control values of the q^{th} subdomain. This implies that

$$B^{[q]} = \left[\boldsymbol{T}^{(n)}_{\mathcal{L}\to\mathcal{B}}\, \boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}}{}^{[q]}\right] B = \boldsymbol{M}^{[q]} B. \tag{10}$$

Each set of new control values bounds the Jacobian determinant on its own subdomain and we have:

$$b'_{\min} = \min_{i,q} b_i^{[q]} \;\le\; J_{\min} \;\le\; \min_{i<K_f,q} b_i^{[q]} \tag{11}$$

and

$$\max_{i<K_f,q} b_i^{[q]} \;\le\; J_{\max} \;\le\; b'_{\max} = \max_{i,q} b_i^{[q]}. \tag{12}$$

If an estimate is not sufficiently sharp, we can thus simply subdivide the appropriate parts of the element. This leads to a simple adaptive algorithm, exemplified in Figure 4. In this particular case the original estimate (6)-(7) is not sharp enough ($J_{\min} \in [-3, 1]$). After one subdivision, the Jacobian determinant is proved to be positive on the second subdomain. The first subdomain is thus subdivided once more, which proves the validity. In practice, a few levels of refinement lead to the desired accuracy. The subdivision has quadratic speed of convergence [22, 23].

Note that in a practical implementation (with finite precision arithmetic), we must take care of a problematic situation. If the minimum of the Jacobian determinant is too close to zero but positive, then the upper bound is positive while the lower bound might never get positive. In order to avoid this situation, we limit the number of consecutive subdivisions that can be applied. The undetermined elements are then considered as invalid. Another way of getting rid of this issue is to relax the condition of rejection.

Extension to Higher Dimensions. The extension of the method to higher dimensions is straightforward, provided that Bézier functions can be generated and that a subdivision scheme is available. Jacobian determinants J are polynomials of ξ, η in 2D and of ξ, η, ζ in 3D.

For high order triangles, the Bézier triangular polynomials are defined as

$$\mathcal{T}^{(p)}_{i,j}(\xi,\eta) = \binom{p}{i}\binom{p-i}{j}\, \xi^i\, \eta^j\, (1-\xi-\eta)^{p-i-j} \qquad (i+j \le p).$$

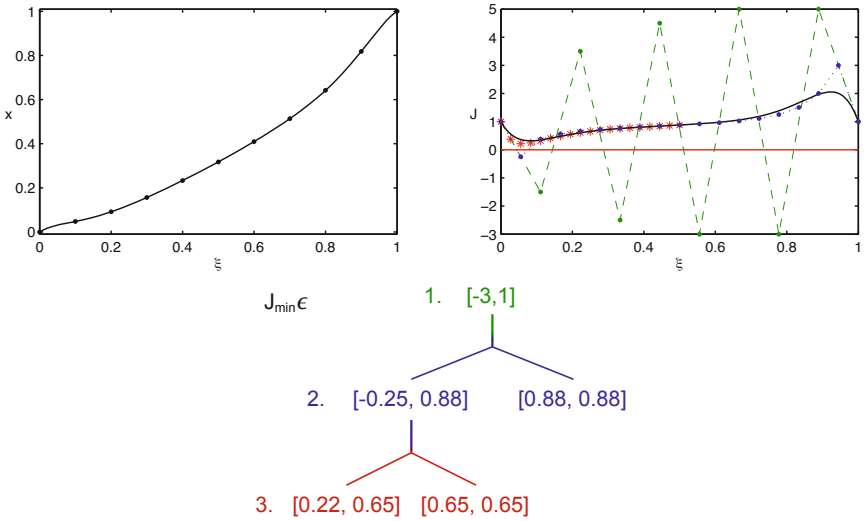


Fig. 4. Top left: One-dimensional element mapping $x(\xi)$. Top right: Exact Jacobian determinant $J(\xi)$ (black), control values on the original control points (green) and two adaptive subdivisions (blue and red). Bottom: Estimates of $J_{\min}$ at each step in the adaptive subdivision process.

It is possible to interpolate any polynomial function of order at most p on the unit triangle $\xi > 0, \eta > 0, \xi + \eta < 1$ as an expansion into Bézier triangular polynomials. Recalling that, for a triangle at order p, its Jacobian determinant $J(\xi, \eta)$ is a polynomial in ξ and η at order at most $n = 2(p-1)$, we can write

$$J(\xi, \eta) = \sum_{i+j \leq n} b_{ij} \mathcal{T}^{(n)}_{i,j}(\xi, \eta).$$

It is also possible to compute J in terms of Lagrange polynomials

$$J(\xi, \eta) = \sum_{i} J_i \mathcal{L}^{(n)}_{i}(\xi, \eta)$$

where the J_i are the Jacobian determinants calculated at Lagrange points. It is then easy to find a transformation matrix $T^n_{\mathcal{LB}}$ such that

$$B = T^n_{\mathcal{LB}} J,$$

where B and J are the vectors containing respectively the control values of the Jacobian determinant b_{ij} and the J_i's.

Other element shapes can be treated similarly. For quadrangles, tetrahedra, prisms and hexahedra, the Bézier are functions respectively:

$$\mathcal{Q}^{(p)}_{i,j}(\xi, \eta) = \mathcal{B}^{(p)}_{i}(\xi)\ \mathcal{B}^{(p)}_{j}(\eta) \qquad (i \leq p,\ j \leq p),$$

$$\mathcal{T}^{(p)}_{i,j,k}(\xi,\eta,\zeta) = \binom{p}{i}\binom{p-i}{j}\binom{p-i-j}{k}\ \xi^i\ \eta^j\ \zeta^k\ (1-\xi-\eta-\zeta)^{p-i-j-k} \qquad (i+j+k \le p),$$

$$\mathcal{P}^{(p)}_{i,j,k}(\xi,\eta,\zeta) = \mathcal{T}^{(p)}_{i,j}(\xi,\eta)\ \mathcal{B}^{(p)}_k(\zeta) \qquad (i+j \le p,\ k \le p)$$

and

$$\mathcal{H}^{(p)}_{i,j,k}(\xi,\eta,\zeta) = \mathcal{B}^{(p)}_i(\xi)\ \mathcal{B}^{(p)}_j(\eta)\ \mathcal{B}^{(p)}_k(\zeta) \qquad (i \le p,\ j \le p,\ k \le p).$$

Matrices of change of coordinates can then be computed inline for every polynomial order, and bounds of Jacobian determinants computed accordingly. Table 1 summarizes the order of the Jacobian determinant and the number of coefficients in its expansion for all common element types. In all cases the subdivision scheme works exactly in the same way as for lines. Figure 5 shows the first level of subdivision for a third-order triangle.

Table 1. Order of the Jacobian determinant and number of coefficients in the expansion for an element of order p

	Order (n) of J	Number of coefficients
Line	$p-1$	$n+1$
Triangle	$2(p-1)$	$(n+1)(n+2)/2$
Quadrangle	$2p-1$	$(n+1)^2$
Tetrahedron	$3(p-1)$	$(n+1)(n+2)(n+3)/6$
Prism	$3p-1$	$(n+1)^2(n+2)/2$
Hexahedron	$3p-1$	$(n+1)^3$

Implementation. As mentioned in Section 4.1, the bounds on the Jacobian determinant can be used to either make the distinction between valid and invalid elements with respect to a condition on $J_{\min}$, or to measure the quality of the elements by systematically computing $J_{\min}$ and $J_{\max}$ with a prescribed accuracy.

In both cases the same operations are executed on each element. First, the Jacobian determinant is sampled on a determined number of points N_s, equal to the dimension of the Jacobian determinant space, and so to the number of Bézier functions. Second, Bézier values are computed. Then adaptive subdivision is executed if necessary. Algorithms 1 and 2 show in pseudo-code the algorithm used to determine whether the Jacobian determinant of the element is everywhere positive or not.

Algorithm 2 could be further improved by optimzing the loop on line 5, by first selecting q for which we have the best chance to have a negative Jacobian determinant (line 4, algo 2). In practice this improvement is not significant since the only case for which we can save calculation is for invalid elements—and the proportion of them which require subdivision in order to be detected is usually

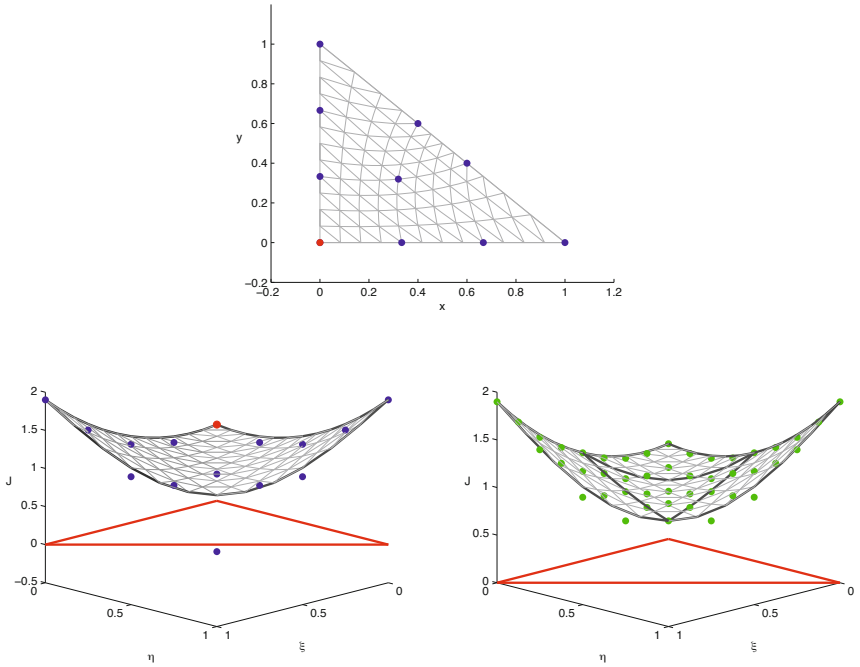


Fig. 5. Top: third-order planar triangle. Bottom left: exact Jacobian determinant and control values (dots) on the original control points; the validity of the element cannot be asserted. Bottom right: exact Jacobian determinant and control values (dots) after one subdivision; the element is provably correct.

small. Note that we may also want to find, for example, all the elements for which the Jacobian determinant is somewhere smaller than 20% of its average. We then just have to compute this average and replace the related lines (4 and 7 for algorithm 1).

Another possible improvement is to relax the condition of rejection. We could accept elements for which all control values are positive but reject an element as soon as we find a Jacobian determinant smaller than a defined percent of the average Jacobian determinant. The computational gain can be significant, since elements that were classified as good and which needed a lot of subdivisions (and have a Jacobian determinant close to zero) will be instead rapidly be detected as invalid.

More interestingly, the computation of sampled Jacobian determinants and the computation of Bézier control values in algorithm 1 can easily be executed for a whole groups of elements at the same time. This allows to use efficient BLAS 3 (matrix-matrix product) functions, which significantly speeds up the computations.

Algorithm 1. Check if an element is valid or invalid

Input: a pointer to an element.
Output: *true* if the element is valid, *false* if the element is invalid

set sampling points P_i, $i = 1, \ldots, N_s$;
compute Jacobian determinants J_i at points P_i;
for $i = 1$ **to** N_s **do**
    if $J_i <= 0$ **then return** *false*
compute Bézier coefficients b_i, $i = 1, \ldots, N_s$ using (8);
$i = 1$;
while $i \leq N_s$ **and** $b_i > 0$ **do**
    $i = i + 1$;
if $i > N_s$ **then return** *true*
call algorithm 2 with b_i as arguments and return *output*;

Algorithm 2. Compute the control values of the subdivisions

Input: Bézier coefficients b_i, $i = 1, \ldots, N_s$
Output: *true* if the Jacobian determinant on the domain is everywhere positive, *false* if not

compute new Bézier coefficients $b_i^{[q]}$, $q = 1, \ldots, Q$ as in equation (10);
for $q = 1$ **to** Q **do**
    for $i = 1$ **to** K_f **do**
        if $b_i^{[q]} <= 0$ **then return** *false*
for $q = 1$ **to** Q **do**
    $i = 1$;
    while $i \leq N_s$ **and** $b_i^{[q]} > 0$ **do**
        $i = i + 1$;
    if $i \leq N_s$ **then**
        call algorithm 2 with $b_i^{[q]}$ as arguments and store *output*;
        if *output* = *false* **then return** *false*
return *true*;

5 Untangling of High-Order Meshes

5.1 Strategy

Applying an untangling procedure to the entire mesh would result in a robust methodology. In most practical cases however, it represents an unnecessary expense of computational resources. Indeed, only the vertices and elements in the vicinity of the curved geometry really need to be modified in order to untangle the mesh, while the mesh entities located far from the boundary remain unchanged. In order to optimize the computational efficiency, the untangling procedure can thus be applied locally. Subdomains are first defined around each invalid element. Then, two strategies can be adopted in the framework of an optimization procedure.

Primary Subdomains. The first step in the untangling procedure consists in identifying all invalid elements in the mesh, for instance by means of the methods described in Section 4. A subdomain of N layers of elements can then be built around each of the invalid elements. Although these subdomains are usually appropriate in isotropic meshes, they may not be adequate for boundary layer-type meshes, where elements are highly anisotropic. In this case, the boundary curvature responsible for the mesh tangling is typically large compared to the normal size of the tangled element, but small compared to its tangential size (see Figure 6). Thus, untangling such a mesh requires several layers to be curved in the direction normal to the boundary, whereas there is no need to modify the elements that are adjacent in the tangential direction. Therefore, a geometrical criterion is introduced in the construction of the subdomains, as illustrated in Figure 6: among the elements contained in the N layers surrounding the invalid element, only those located within a certain distance are retained. This distance is defined by multiplying the distance between the straight-sided and high-order boundaries by a user-defined factor. The boundaries of the subdomain are fixed, in order to ensure that elements lying outside do not get invalidated.

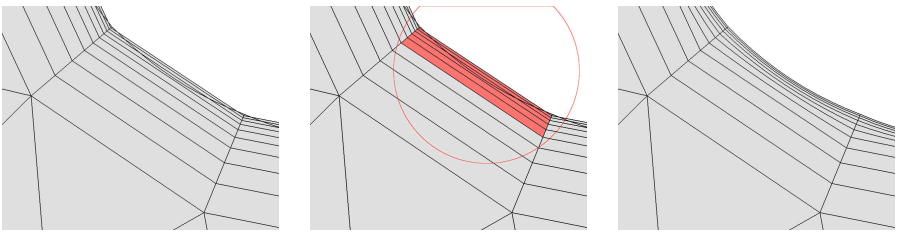


Fig. 6. Detail of a boundary-layer mesh on a curved geometry: tangled second-order mesh (left), subdomain definition with a circle representing the geometrical criterion (center), and untangled mesh (right)

The optimal values for the number of layers N and the distance factor are those that encompass the lowest number of elements while still leaving the untangling procedure enough freedom to untangle the mesh (or reach a target value for the Jacobian determinant). They are case-dependent. Note that in 3D, it is desirable to make sure that the subdomain includes at least one layer of elements around all faces of the invalid element, because the latter is likely to have tangled faces that cannot be repaired if they lie on the boundary of the subdomain. For the optimization procedure described in Section 5.2, a fair trade-off between computation cost and robustness can be obtained with $N = 6$ layers of quadrangles (or $N = 12$ layers of triangles) and a distance factor of 12 in many applications involving boundary-layer meshes.

In complex cases, a unique value of these parameters for the whole mesh may lead to subdomains that are too large for some invalid elements, but too small for others. It is then beneficial to use the untangling procedure in an adaptive loop where the subdomain size is progressively increased in case the procedure fails to untangle the mesh.

Overlapping Subdomains. A potential problem with the subdomains created as described above is that they may overlap. There is no guarantee that the untangling procedure can be efficiently applied to each of these subdomains. Let us consider the case of a subdomain built around an invalid element, that contains another invalid element close to its boundary: this subdomain may not provide the necessary degrees of freedom to fix the second invalid element, thus the untangling procedure can fail in this subdomain. In order to avoid such problems, two strategies can be applied:

Disjoint Subdomains. Overlapping subdomains are detected and merged. This strategy ensures that each invalid element in a subdomain has enough surrounding elements for the untangling procedure to be successful, provided the primary subdomains are large enough. It is also well-suited for a parallel setting in which each subdomain would be treated by a processor. However, it may lead to large subdomains, which can prove inappropriate if the cost of the untangling procedure does not scale well with the number of nodes and elements involved.

"One-by-one" Strategy. The untangling procedure is applied sequentially to each subdomain, with the objective of fixing only the invalid element around which it is built, while allowing other invalid elements in the same subdomain to remain broken. This strategy lets the untangling procedure work on small-size subdomains, but is less appropriate for parallel operation at subdomain level, because it is not possible to untangle concurrently subdomains that overlap.

5.2 Untangling through Optimization

We chose to apply to each subdomain an untangling procedure that preserves the mesh topology. Its role is to move the mesh vertices in such a way that the

tangled elements become valid, while the other elements do not become invalid. The measure of the scaled Jacobian determinant characterizes the element validity in a continuously derivable manner with respect to the position of the vertices, which allows us to use an optimization algorithm. In this section, we derive an appropriate objective function relying on the robust Jacobian evaluation technique presented in Section 4, and explain how to efficiently compute its gradient. We then describe the method used to solve the optimization problem.

Computation of Bézier Coefficients and Their Derivatives. The general goal of our method is to untangle both surface and volume meshes. To this end, we consider that all points have three-dimensional coordinates $\boldsymbol{x} = \{x, y, z\}$. We also use three local coordinates $\boldsymbol{\xi} = \{\xi, \eta, \zeta\}$. For surface meshes however, where only two parametric coordinates are defined, we assume that the vector $\mathbf{n} = \partial \boldsymbol{x}/\partial\zeta$ is the constant unit normal vector to the straight sided element, with the same orientation convention used for the low-order mesh. This assumption makes it possible to compute the Jacobian determinant at every Lagrange node $\boldsymbol{\xi}_k = (\xi_k, \eta_k, \zeta_k)$ at order q:

$$J_k = J(\boldsymbol{\xi}_k) = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} + \frac{\partial z}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} + \frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta}\frac{\partial x}{\partial \zeta} - \frac{\partial z}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial x}{\partial \zeta} - \frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta}. \tag{13}$$

As

$$x = \sum_{i=1}^{N_p} x_i^e \mathcal{L}_i^{(p)}(\boldsymbol{\xi}_k),$$

we can easily express the sensitivity $\partial J_k/\partial x_i^e$ of the Jacobian determinant at point k with respect to x_i^e. The quantities $\partial J_k/\partial y_i^e$ and $\partial J_k/\partial z_i^e$ can be calculated in the same manner. In practice, we compute for each element e the matrix $\mathbf{J}$ of size $N_q \times (3N_p + 1)$ as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial J_1}{\partial x_1^e} & \cdots & \frac{\partial J_1}{\partial x_{N_p}^e} & \frac{\partial J_1}{\partial y_1^e} & \cdots & \frac{\partial J_1}{\partial y_{N_p}^e} & \frac{\partial J_1}{\partial z_1^e} & \cdots & \frac{\partial J_1}{\partial z_{N_p}^e} & J_1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ \frac{\partial J_{N_q}}{\partial x_1^e} & \cdots & \frac{\partial J_{N_q}}{\partial x_{N_p}^e} & \frac{\partial J_{N_q}}{\partial y_1^e} & \cdots & \frac{\partial J_{N_q}}{\partial y_{N_p}^e} & \frac{\partial J_{N_q}}{\partial z_1^e} & \cdots & \frac{\partial J_{N_q}}{\partial z_{N_p}^e} & J_{N_q} \end{bmatrix}$$

Let $\mathbf{B}$ be the matrix

$$\mathbf{B} = \begin{bmatrix} \frac{\partial B_1}{\partial x_1^e} & \cdots & \frac{\partial B_1}{\partial x_{N_p}^e} & \frac{\partial B_1}{\partial y_1^e} & \cdots & \frac{\partial B_1}{\partial y_{N_p}^e} & \frac{\partial B_1}{\partial z_1^e} & \cdots & \frac{\partial B_1}{\partial z_{N_p}^e} & B_1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ \frac{\partial B_{N_q}}{\partial x_1^e} & \cdots & \frac{\partial B_{N_q}}{\partial x_{N_p}^e} & \frac{\partial B_{N_q}}{\partial y_1^e} & \cdots & \frac{\partial B_{N_q}}{\partial y_{N_p}^e} & \frac{\partial B_{N_q}}{\partial z_1^e} & \cdots & \frac{\partial B_{N_q}}{\partial z_{N_p}^e} & B_{N_q} \end{bmatrix}, \tag{14}$$

that contains both the Bézier coefficients B_l as well as their gradients with respect to the position of the nodes of element e. Defining the constant transformation matrix $T_{lk}^q = \mathcal{B}_l^{(q)}(\boldsymbol{\xi}_k)$ that gives the Bézier coefficients B_l from the

Lagrange coefficients J_k, $\mathbf{B}$ can be calculated through a single matrix-matrix product: $B_{lj} = T^q_{lk} J_{kj}$.

We can then make use of the B_i's and their gradients in a gradient-based optimization procedure. To this end, an objective function that allows us to control the quality of elements based on the coefficients B_i needs to be constructed.

Objective Function. In what follows, we detail the objective function $f(\boldsymbol{x}_i^e)$ that is employed in an unconstrained optimization procedure to untangle invalid curved elements. The function $f = \mathcal{E} + \mathcal{F}$ is composed of two parts $\mathcal{E}$ and $\mathcal{F}$.

The first part $\mathcal{E}$ is based on the assumption is that the method is provided with a straight-sided mesh of high quality. This mesh has potentially been defined to satisfy multiple criteria, such as a predetermined size field, or anisotropic adaptation. The conversion such meshes to high order is expected to preserve as much as possible all these features. Therefore, the nodes shall be kept as close as possible to their initial location in the straight sided mesh.

To this end, we define the function $\mathcal{E}$ by analogy with an *energy* associated with the displacement $\boldsymbol{x}_i^e - \boldsymbol{X}_i^e$ of the nodes, i.e. as a positive quadratic form that is a measure of the distance between the straight sided nodes $\boldsymbol{X}_i^e$ and their position $\boldsymbol{x}_i^e$ in the curved mesh:

$$\mathcal{E}(\boldsymbol{x}_i, \mathbf{K}) = \frac{1}{2} \sum_e \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} (\boldsymbol{x}_i^e - \boldsymbol{X}_i^e) \mathbf{K}_{ij}^e (\boldsymbol{x}_j^e - \boldsymbol{X}_j^e) \geq 0 \tag{15}$$

where $\mathbf{K}$ is a symmetric positive matrix of size $3n_v \times 3n_v$ and $\mathbf{K}_{ij}^e$ is of size 3×3. Here, we define $\mathbf{K}$ as the diagonal matrix with entries w_i^e / L^2, where w_i^e are user-defined weights used to set the balance between the two parts $\mathcal{E}$ and $\mathcal{F}$ of the objective function f, and L is a length scale representative of the problem. We choose L as the maximum distance, among all vertices of the initial tangled mesh, between a node and its counterpart in the straight-sided mesh. Regarding the non-dimensional weights w_i^e, we usually set $w_i^e = 1000$ if the node i of element e lies on the boundary, and $w_i^e = 1$ otherwise. In practice, we observe that the exact value of w_i^e has a limited influence on the convergence of the optimization method. The part $\mathcal{E}$ in f prevents the problem from being under-determined, and it steers the optimization procedure towards a solution that preserves the features of the straight-sided mesh, but the term $\mathcal{F}$ dominates for the most problematic (tangled) elements that drive the mesh deformation.

The second part $\mathcal{F}$ of the functional controls the positivity of the Jacobian determinant. A log barrier [24] prevents Jacobians from becoming too small, and a quadratic function is used to penalize Jacobians that are too large:

$$\mathcal{F}(\boldsymbol{x}_i, \epsilon) = \sum_{e=1}^{n_e} \sum_{l=1}^{N_q} F_l^e(\boldsymbol{x}_i^e, \epsilon)$$

where

$$F_l^e(\boldsymbol{x}_i^e, \epsilon) = \left[\log \left(\frac{B_l^e(\boldsymbol{x}_i^e) - \epsilon J_0^e}{J_0^e - \epsilon J_0^e} \right) \right]^2 + \left(\frac{B_l^e(\boldsymbol{x}_i^e)}{J_0^e} - 1 \right)^2. \tag{16}$$

The function $\mathcal{F}$ is designed to blow up when $B_l^e = \epsilon J_0^e$, but still vanishes whenever $B_l^e = J_0^e$. Barrier methods are one of the most successful class of algorithms used for general nonlinear optimization problems. These techniques are very robust with respect to the convexity characteristics of the objective function and constraints [25], so they converge to at least a local minimum in most cases. Figure 7 shows a plot of the barrier function in Equation (16) for $\epsilon = 0.2$.

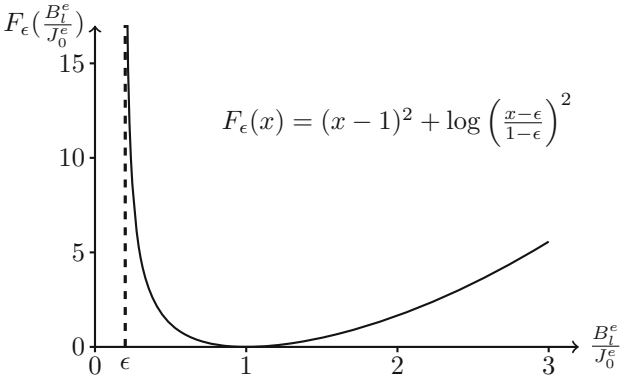


Fig. 7. Plot of the barrier function $F(J_l^e)$ for $\epsilon = 0.2$

As the objective function f is smooth, we can compute its gradient ∇f with respect to the positions of the element vertices $\boldsymbol{x}_i^e$. This gradient can then be used in a gradient-based optimization procedure. As explained in Section 3.2, each vertex is classified on a given model entity, to which it is geometrically linked. In order for the vertices to remain on their model entity, the sensitivity of f is computed with respect to the location of vertices expressed in the parametric space of the model entities rather than in the physical space.

A mesh vertex M_i^0 classified on a model edge G_j^1 can only be moved along G_j^1, i.e. its position only depends on one curve parameter t. The corresponding component of the gradient will thus be computed as

$$\frac{df}{dt} = \frac{\partial f}{\partial \boldsymbol{x}_i^e} \cdot \frac{d\boldsymbol{x}_i^e}{dt}$$

with $d\boldsymbol{x}_i^e/dt$ the tangent vector to the curve at point t, that is obtained from the CAD model.

A mesh vertex M_i^0 classified on a model face G_j^2 can only be moved along the surface. Two parameters u and v are associated to such a vertex, and the corresponding sensitivities $\partial f/\partial u$ and $\partial f/\partial v$ depend respectively on $\partial \boldsymbol{x}_i^e/\partial u$ and $\partial \boldsymbol{x}_i^e/\partial v$ the two tangent vectors to the surface at point (u, v). Those can be computed using the CAD model.

A vertex that is classified on a model region has complete freedom to move in every direction of the 3D space, in which case the physical coordinates $\{x, y, z\}$ are used. Finally, a mesh vertex that is classified on a model vertex has no freedom to move, it is thus excluded from the optimization problem.

Optimization Method. The general optimization procedure, that involves a moving barrier for the objective function, is detailed in Algorithm 3.

Algorithm 3. Optimization method

```
Define subdomains B_k, k = 1 ... N_B ;
for k = 1 to N_B do
    repeat
        compute κ = min_e min_l B_l^e / J_0^e,   e ∈ B_i,  l ∈ [1, N_q];
        if κ < ε̄ then
            set ε = κ − 0.1 |κ|;
            solve min_{x_i} f(x_i, K, ε) for all elements of subdomain B_k;
            recompute κ = min_e min_l B_l^e / J_0^e,   e ∈ B_i,  l ∈ [1, N_q];

    until κ ≥ ε̄;
```

The untangling problem can be formally defined as

$$\min_{\boldsymbol{x}_i} f(\boldsymbol{x}_i, \mathbf{K}, \epsilon), \quad i = 1, \ldots, n_v.$$

A broad range of methods can be used to solve such an unconstrained minimization problem. We have tested several alternatives: interior point methods implemented in the software package IPOPT [26], as well as L-BFGS [27] and conjugate gradients [28] algorithms provided by ALGLIB [29]. In our experience, the use of conjugate gradients seems to be the best choice in terms of computational efficiency.

A crucial aspect of the optimization procedure is the definition of an appropriate sequence of optimization problems. In a given subdomain, the value of the barrier ϵ must be lower than the worst scaled Jacobian determinant among all elements, so that the variables remain in the domain of definition of the barrier function $\mathcal{F}$. In particular, ϵ has to be negative for an initially tangled mesh. Therefore, we compute a sequence of optimization problems with "moving barriers": ϵ is increased between each optimization problem, until the desired barrier value $\bar{\epsilon}$ is reached. This procedure is illustrated in Figure 8.

In practical cases, it is most often necessary to apply preconditioning to the optimization problem, because the scale of the parametric or physical coordinates used for different mesh vertices can differ by orders of magnitude, depending on the model entities on which they are classified. We found that a simple diagonal preconditioner, based on the norm of the tangent vectors $d\boldsymbol{x}_i^e/dt$ for vertices classified on model edges (respectively $\partial \boldsymbol{x}_i^e/\partial u$ and $\partial \boldsymbol{x}_i^e/\partial v$ for vertices classified on model faces), and unity for vertices classified on model regions, allows the optimizer to converge in a fast and robust manner.

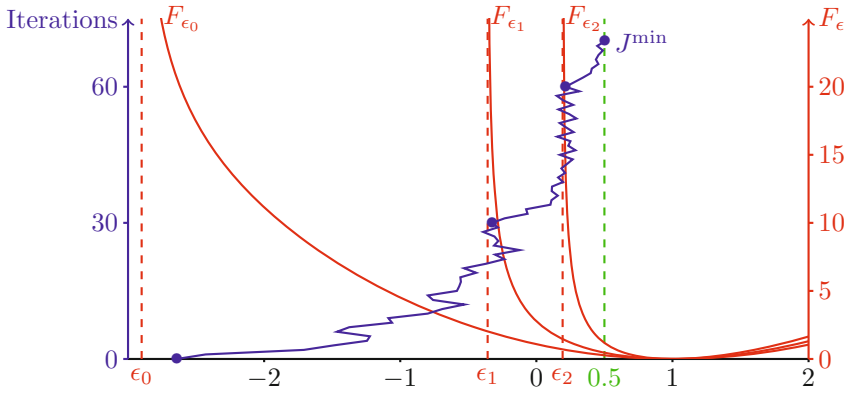


Fig. 8. Optimization procedure: three successive series of (maximum) 30 conjugate gradient iterations, with their respective log barriers

In practice, the value of J_0^e appearing in Algorithm 3 represents the Jacobian determinant of the straight-sided version of element e in the original mesh. It is computed at initialization and held constant during the optimization process, which saves computational time and favors curved elements that resemble their first-order counterpart in the original mesh. However, all vertices of element e are allowed to move, so the value of J_0^e after optimization is different from the original one. The measure of the minimum scaled Jacobian determinant in the untangled mesh may thus yield a lower value than the barrier $\bar{\epsilon}$. Nevertheless, the positivity of the Jacobian determinant is not threatened as long as the optimization procedure is successful.

5.3 Untangling through Analytical Methods

In boundary layers meshes used in CFD, where elements are highly stretched along the boundary, it is relatively clear that the curvature should "propagate" in the direction normal to the boundary in order to untangle the mesh. In this case, it may be possible to apply simple techniques to determine in an analytical fashion the displacement that high-order nodes shall undergo for the mesh to become valid.

Such a method is illustrated in Figure 9. In a boundary layer mesh, the subdomains created following the methodology described in Section 5.1 typically consist of a "stack" of elements above the tangled element. It is then possible to define a "meta-element" that includes the whole subdomain under consideration, based on the curved edge or face of the tangled mesh element. In 2D, the meta-element is a quadrangle created from the curved boundary edge. In 3D, the meta-element is either a prism created from a curved boundary triangle, or an hexahedron created from a curved boundary quadrangle.

Let us consider the mapping between the straight-sided version and the curved version of the meta-element, as described in Section 4.1. If the meta-element is sufficiently large in the direction normal to the boundary, its curved version is valid in the sense of Section 4.1 and the mapping is injective. All mesh vertices in the subdomain are located within the straight meta-element. We can then apply the straight-to-curved meta-element mapping to them: their images are located in the curved subdomain in a way that untangle the mesh by distributing the element curvature along the direction normal to the boundary.

The straight-to-curved meta-element transformation is defined by the composition $\mathbf{X} \circ \mathbf{x}^{-1}$, following the notation of Section 4.1 illustrated in Figure 3. As the straight-to-reference meta-element mapping $\mathbf{x}(\boldsymbol{\xi})$ is bilinear, its inversion is computationally inexpensive. So is the forward high-order transformation $\mathbf{X}(\boldsymbol{\xi})$ from the reference to the curved meta-element. This interpolation technique is thus much more efficient than the iterative optimization procedure presented in Section 5.2.

This method is particularly interesting in boundary layer-type meshes, where most elements are created from columns of vertices that are aligned in the direction normal to the boundary. In this case, the meta-element can be chosen to have the same boundaries as the column of elements above the invalid one. Thus, all the elements in the column undergo a consistent deformation, and the untangling is successful as long as the meta-element is valid. When vertices are not aligned, as in boundary layer meshes near complex geometric features or fully unstructured meshes, different vertices of a same element can belong to different meta-elements, which may not lead to a valid element. Even then, it can be beneficial to apply the analytical method to correct most invalid elements, before using the optimization method described in Section 5.2 to untangle the rest of the mesh in a robust manner.

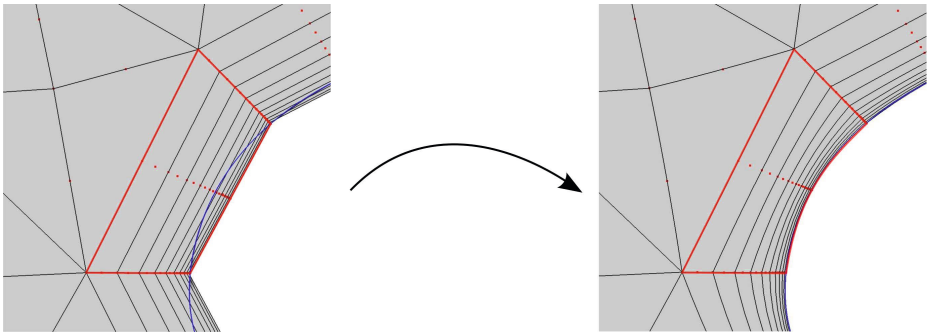


Fig. 9. Illustration of the analytical curving technique: the "meta-element" is shown with red lines and the high-order mesh nodes that are moved to untangle the mesh are marked in red

6 Conclusion

The techniques presented in this article prove valuable for the creation, diagnostic and regularization of high-order meshes. They can be used as the basis of a robust, efficient and versatile high-order mesh generation framework. They have already been demonstrated to produce valid meshes of relatively high order for a wide variety of industrial and scientific problems. For a more thorough discussion of these methods and examples of applications, the reader is referred to Ref. [1, 2]. An efficient and ready-to-use implementation can be found in the free software Gmsh [4, 12].

Nevertheless, two topics have not been addressed in this article. The first one is the geometrical accuracy of the curvilinear mesh: the quality of the approximation of the CAD geometry by the mesh depends on the location of the high-order nodes on the boundary. The second one is the numerical properties of high-order meshes beyond the mere validity: their curvilinear nature affects numerical schemes in terms of interpolation properties and conditioning. These aspects should ideally be taken into account in the mesh generation process, but have been the subject of little interest from the scientific community so far. They should be investigated more intensely in the future.

References

1. Johnen, A., Remacle, J.-F., Geuzaine, C.: Geometrical validity of curvilinear finite elements. Journal of Computational Physics 233, 359–372 (2013)
2. Toulorge, T., Geuzaine, C., Remacle, J.-F., Lambrechts, J.: Robust untangling of curvilinear meshes. Journal of Computational Physics 254, 8–26 (2013)
3. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)
4. Geuzaine, C., Remacle, J.-F.: Gmsh website (March 2014), `http://geuz.org/gmsh`
5. Cockburn, B., Karniadakis, G., Shu, C.-W. (eds.): Discontinuous Galerkin Methods. LNCSE, vol. 11. Springer, Berlin (2000)
6. Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.): ADIGMA. NNFM, vol. 113. Springer, Heidelberg (2010)
7. Bassi, F., Rebay, S.: High-order accurate discontinuous finite element solution of the 2D Euler equations. Journal of Computational Physics 138(2), 251–285 (1997)
8. Dey, S., O'Bara, R.M., Shephard, M.S.: Curvilinear mesh generation in 3D. In: Proceedings of the 8th International Meshing Roundtable, pp. 407–417. John Wiley & Sons (1999)
9. Gargallo-Peiró, A., Roca, X., Peraire, J., Sarrate, J.: High-order mesh generation on CAD geometries. In: Proceedings of the VI International Conference on Adaptive Modeling and Simulation (ADMOS 2013), CIMNE, Barcelona (2013)
10. Xie, Z.Q., Sevilla, R., Hassan, O., Morgan, K.: The generation of arbitrary order curved meshes for 3D finite element analysis. Computational Mechanics 51(3), 361–374 (2013)
11. Sherwin, S.J., Peiró, J.: Mesh generation in curvilinear domains using high-order elements. International Journal for Numerical Methods in Engineering 53(1), 207–223 (2002)

12. Abgrall, R., Dobrzynski, C., Froehly, A.: A method for computing curved 2D and 3D meshes via the linear elasticity analogy: preliminary results. Rapport de recherche RR-8061, INRIA (September 2012)
13. Lu, Q., Shephard, M., Tendulkar, S., Beall, M.: Parallel curved mesh adaptation for large scale high-order finite element simulations. In: Proceedings of the 21st International Meshing Roundtable, pp. 419–436. Springer, Heidelberg (2013)
14. Persson, P.-O., Peraire, J.: Curved mesh generation and mesh refinement using lagrangian solid mechanics. In: Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando (FL), USA, January 5-9 (2009)
15. Roca, X., Gargallo-Peiró, A., Sarrate, J.: Defining quality measures for high-order planar triangles and curved mesh generation. In: Quadros, W.R. (ed.) Proceedings of the 20th International Meshing Roundtable, pp. 365–383. Springer, Heidelberg (2012)
16. Luo, X.J., Shephard, M.S., O'Bara, R.M., Nastasia, R., Beall, M.W.: Automatic p-version mesh generation for curved domains. Engineering with Computers 20(3), 273–285 (2004)
17. Sahni, O., Luo, X.J., Jansen, K.E., Shephard, M.S.: Curved boundary layer meshing for adaptive viscous flow simulations. Finite Elements in Analysis and Design 46(1-2), 132–139 (2010)
18. Remacle, J.-F., Shephard, M.S.: An algorithm oriented mesh database. International Journal for Numerical Methods in Engineering 58(2), 349–374 (2003)
19. Dey, S., Shephard, M.S., Flaherty, J.E.: Geometry representation issues associated with p-version finite element computations. Computer Methods in Applied Mechanics and Engineering 150(1), 39–55 (1997)
20. Sevilla, R., Fernández-Méndez, S., Huerta, A.: NURBS-enhanced finite element method (NEFEM). International Journal for Numerical Methods in Engineering 76(1), 56–83 (2008)
21. Taylor, R.L.: On completeness of shape functions for finite element analysis. International Journal for Numerical Methods in Engineering 4(1), 17–22 (1972)
22. Lane, J.M., Riesenfeld, R.F.: A theoretical development for the computer generation and display of piecewise polynomial surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 2(1), 35–46 (1980)
23. Cohen, E., Schumacker, L.L.: Rates of convergence of control polygons. Computer Aided Geometric Design 2, 229–235 (1985)
24. Freitag, L., Knupp, P., Munson, T., Shontz, S.: A comparison of optimization software for mesh shape-quality improvement problems. Technical report, Argonne National Lab., IL, US (2002)
25. Fiacco, A.V., McCormick, G.P.: Nonlinear programming: sequential unconstrained minimization techniques, vol. 4. Society for Industrial Mathematics (1990)
26. Waechter, A., Laird, C., Margot, F., Kawajir, Y.: Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT (2009)
27. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical Programming 45(1), 503–528 (1989)
28. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. The Computer Journal 7(2), 149–154 (1964)
29. Bochkanov, S.: ALGLIB (December 2013), `http://www.alglib.net`

Curvilinear Mesh Generation for Boundary Layer Problems

D. Moxey, M. Hazan, S.J. Sherwin, and J. Peiro

Department of Aeronautics, Imperial College London
{d.moxey,menashe.hazan10,s.sherwin,j.peiro}@imperial.ac.uk

Abstract. In this article, we give an overview of a new technique for unstructured curvilinear boundary layer grid generation, which uses the isoparametric mappings that define elements in an existing coarse prismatic grid to produce a refined mesh capable of resolving arbitrarily thin boundary layers. We demonstrate that the technique always produces valid grids given an initially valid coarse mesh, and additionally show how this can be extended to convert hybrid meshes to meshes containing only simplicial elements.

1 Introduction

As the popularity of high order methods continues to increase in academia and industry alike, there is an increasing demand for new robust mesh generation strategies which are capable of generating meshes for complex three dimensional geometries. Industrial aeronautics applications, such as the test cases considered in the IDIHOM project, are particularly demanding as they usually require the generation of boundary layer grids, where the mesh is refined near walls in order to adequately resolve high-shear regions of flow. The size of elements in the wall-normal direction is dictated by a wall unit

$$y^+ = \frac{y}{L}\,\mathrm{Re}\sqrt{\frac{C_f}{2}},$$

where y dnotes the coordinate normal to the wall, Re is the Reynolds number, C_f is the skin friction coefficient and L is a reference length. Industrial requirements typically use very high Reynolds numbers and require meshes with a resolution of $y^+ = 1$ so that the flow in the viscous sublayer is adequately resolved. Since the skin friction coefficient scales as $O(\mathrm{Re}^{-2})$ or $O(\mathrm{Re}^{-3})$ [17], at the high Reynolds numbers which are required for aeronautical applications these grids are highly stretched in the wall-normal direction, so as to produce meshes which strike a balance between accuracy and computational efficiency. Such meshes typically have elements possessing stretching ratios of 1:1000 or even lower.

Whilst there are many well-established methods for generating linear boundary layer grids [8, 12, 15], the development of such techniques for unstructured curvilinear meshes has been limited due to the relatively recent nature of high order mesh generation strategies [2,19,19]. The most recent developments in curvilinear mesh generation methods rely on approaches that split a linear boundary

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_3

conforming mesh into high-order straight-sided elements, project the boundary nodes onto the curved surface and then deform the mesh to accommodate them [14, 21] or alternatively undertake an optimization procedure to untangle it [20]. Some work has been undertaken to investigate the applicability of these methods to boundary layer meshes [4,16]. However, the expensive nature of these techniques, particularly at high polynomial orders, and the uncertainty of the resolution that can be obtained when the mesh is deformed, means that their application to boundary layer grids remains to be investigated.

The common theme of these methods however is that they rely on the generation of a dense linear boundary layer grid, and then undertake an optimization procedure in order to deform it. In this chapter, we given an overview of an alternative method proposed in [9,10] and developed during the IDIHOM project that aims to address the problem of generating high-order meshes for high Reynolds number flows. The method is conceptually simple, cheap to implement and does not require a dense linear boundary-layer mesh. It is based on the use of an isoparametric [22] or, in general, a transfinite interpolation [6] where a high-order coarse boundary-layer mesh is subdivided using the same mapping that define these high-order elements. The procedure is also very versatile as it permits meshes with different distributions of y^+ to be generated with ease and, further, the validity of these meshes is guaranteed if the initial mesh is valid and the polynomial space is chosen appropriately.

An overview of this process can be seen for a representative quadrilateral element in figure 1, where we assume the bottom edge of the element is attached to the wall, and therefore require extra resolution in the vertical direction. The top row shows how the standard quadrilateral element Ω_{st} is deformed under the mapping χ to produce a curved element Ω. To refine the element in physical space, we first split Ω_{st} into a series of smaller elements as shown in bottom left of the figure. Applying the mapping χ to these subelements of the standard region leads to the production of curved subelements of the physical element as desired.

This chapter is organized in three main sections. Section 2 describes the various procedures involved in generating arbitrarily high-order boundary layer meshes suitable for these flows in three dimensions: a process for generating the initial coarse mesh comprising of a prismatic boundary layer; the splitting of the prismatic elements into finer prisms; and the spacing distributions used to define the size of the finer subelements. We also demonstrate how the method can be adapted to produce a curvilinear mesh containing only simplicial elements. The procedure for splitting the coarse boundary-layer mesh into a fine prismatic mesh whilst ensuring its validity and the mathematical framework for a generalisation of the method to other element types are described in section 3. Section 4 presents examples of application of the methodology to the generation of a high-order mesh for the ONERA M6 wing. A final section on conclusions follows.

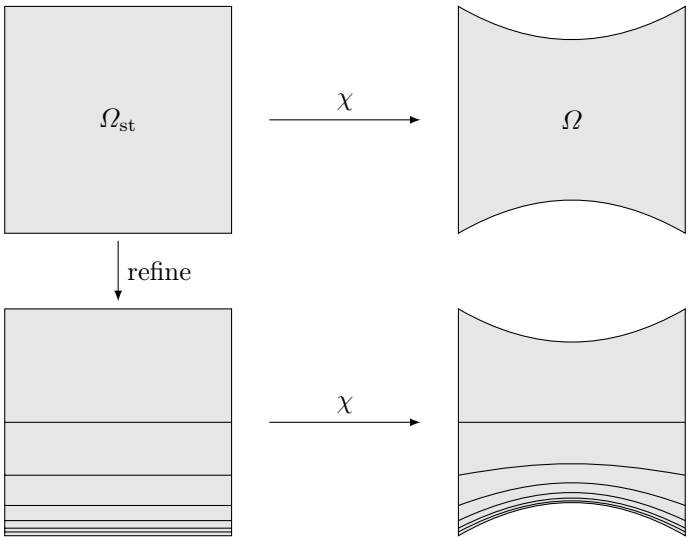


Fig. 1. Overview of boundary layer refinement technique

2 Mesh Generation Strategy

Generating meshes for simulations of turbulent viscous flows is very challenging due to the presence of a extremely thin boundary layer close to walls. In these regions the flow posses a high level of shear, and so the accurate representation of flow features typically requires far greater mesh resolution than other areas of the domain. However, since this level of accuracy is not usually required in transverse directions, computational efficiency necessarily dictates the use of elements possessing stretching ratios of the order of 1:1000 or below. This resolution is of critical importance for turbulence simulations, as many of the structures that lead to the development of turbulence form inside or near the boundary layer. This section describes a method to generate high-order meshes meeting this requirement. Given a high-order coarse mesh consisting of prisms in the boundary layer and tetrahedra elsewhere, a finer mesh is obtained by splitting the coarse prismatic elements (referred to in the following as 'macro-elements') into finer elements, utilising the mapping that defines the high-order prisms to insert the necessary curvature into subelements. The various steps involved are described in the following sections.

2.1 Generation of the Coarse Hybrid Mesh

We begin by outlining a procedure for generating the coarse hybrid mesh. This procedure is essentially the one presented in [19] and will be briefly summarized

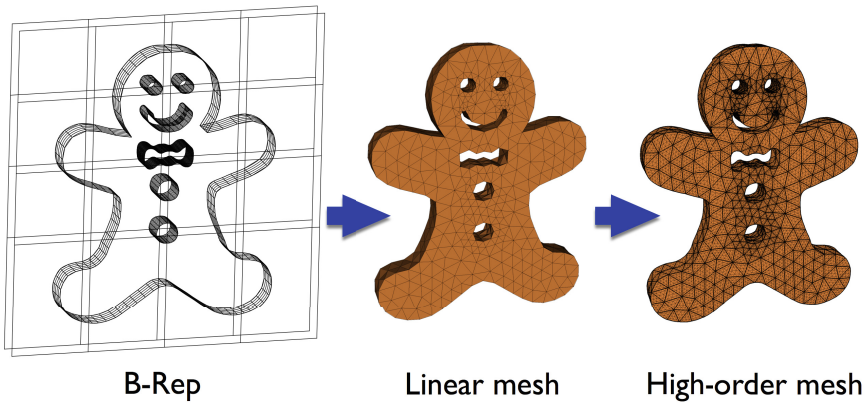


Fig. 2. High-order meshing strategy

here for completion. In broad terms, the high-order mesh generation steps are: the definition of a CAD boundary representation (B-Rep) of the computational domain; the generation of a hybrid mesh of linear prismatic and tetrahedral elements; and finally, the subdivision of this mesh into a mesh of high-order elements. The methodology is illustrated in figure 2. We emphasise however that this method is not the only one for generating a coarse grid, and the method we propose in the next section may equally well be applied to any valid coarse grid.

The geometry of the computational domain is represented by means of a CAD spline curves and surfaces to obtain a boundary representation (B-Rep) of the computational domain. The linear mesh is then generated using an implementation of the method of advancing normals [11] to generate a boundary-layer mesh of triangular prisms near the walls in the computational domain. The advancing front technique [13] is then used to generate a mesh of regular tetrahedra for the rest of the domain.

The final step in the generation process is to insert high-order curvature information to produce the high-order mesh. This follows the method presented in [19] where the generation of the additional degrees of freedom required to obtain a high-order mesh proceeds in a bottom-up fashion, starting with the edges, then the faces and finally the interior of the elements. One of the principal contributions of [19] is the development of a method for calculating the positions of the newly generated points by minimizing an energy of deformation that accounts for anisometries in the CAD surface mapping. This is the most troublesome aspect of the method as it could lead to the generation of invalid elements in the concave regions of the boundary of the computational domain. Figure 3 shows that when the prismatic boundary layer is generated using a fixed width, some elements are not sufficiently large to prevent a self-intersection of the element once curvature information is introduced to the element. In the following section, we describe a strategy to palliate this problem.

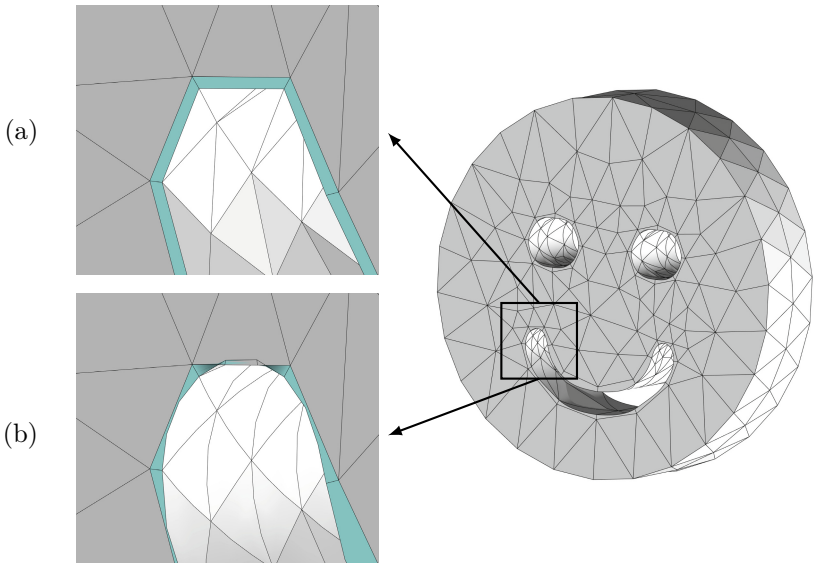


Fig. 3. Generation of invalid elements in concave regions of the computational domain. (a) depicts a valid linear mesh and (b) shows the self-intersection that often occurs when high-order information is introduced.

2.2 Determining a Suitable Height of the Macro-element

We use a criterion based on the curvature of the surface to determine a lower bound for the boundary layer thickness that aims to prevent the intersection of the element with the surface. The curvature of the surface is obtained through interrogation of the B-Rep of the computational domain.

Following the notation of figure 4 and considering a two-dimensional domain, we approximate a curve by its osculating circle [1]. Assuming that the radius of the osculating circle is R, and the size of the element along the curve is represented by the chord c, then the minimum length in the direction normal to the curve required to obtain a valid prismatic element, represented by δ, is given by

$$\left(\frac{c}{2R}\right)^2 + \left(\frac{R}{R+\delta}\right)^2 = 1 \quad \Rightarrow \quad \delta \geq -1 + \left[1 - \left(\frac{c}{2R}\right)^2\right]^{-\frac{1}{2}}.$$

If $c \ll R$, this requirement can be approximated by

$$\frac{\delta}{R} \geq \frac{c^2}{8R^2}.$$

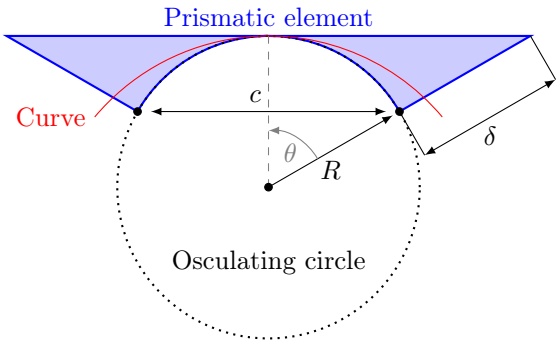


Fig. 4. Notation used for the calculation of the minimum height δ of a valid prismatic macro-element

To extend this method to surfaces in three dimensions, we simply interpret R as the smallest of the two principal radii of curvature of the surface and use the same criterion. In the following section, we describe how a prismatic boundary layer mesh generated using this criterion is refined in an isoparametric sense.

2.3 The Isoparametric Approach

The main idea of the splitting method is to interpret the prismatic macro-element as a mapping, $\chi^e : \Omega_{\text{st}} \to \Omega^e$, from a reference element Ω_{st} to obtain the physical element Ω^e. This mapping is illustrated in figure 5.

There are many ways to define this mapping. Here we adopt an approach in which χ^e isoparametrically projects coordinates $\xi = (\xi_1, \xi_2, \xi_3)$ in a reference element $\Omega_{\text{st}} = \{(\xi_1, \xi_2, \xi_3) \mid \xi_1, \xi_2 \in [-1, 1], \xi_1 + \xi_3 \leq 1\}$ onto the Cartesian coordinates $(x_1, x_2, x_3) \in \Omega^e$. Given polynomial orders P, Q and R for each component ξ_i, we represent each component of $\chi^e = (\chi^e_1, \chi^e_2, \chi^e_3)$ as a tensor product expansion of one-dimensional hierarchical modal functions, so that

$$\chi^e_i(\xi_1, \xi_2, \xi_3) = \sum_{p=0}^{P} \sum_{q=0}^{Q} \sum_{r=0}^{R-p} (\hat{\chi}_i)_{pqr} \psi^a_p(\eta_1) \psi^a_q(\xi_2) \psi^b_{pr}(\eta_3). \tag{1}$$

In this expansion we assume $P \leq R$, and map the standard prism Ω_{st} onto a reference hexahedron through a collapsed coordinate system $(\eta_1, \eta_3) \in [-1, 1]^2$ defined by

$$\eta_1 = 2\frac{1 + \xi_1}{1 - \xi_3} - 1, \qquad \eta_3 = \xi_3.$$

The use of the modified hierarchical expansion functions ψ^a and ψ^b, defined in [7], is useful in the mesh generation process as it permits a decomposition of the element into modes which have non-zero support only on designated vertices, edges and faces of the element. If only edge-interior or face-interior curvature is provided for an element, then upon a transformation from the coefficient

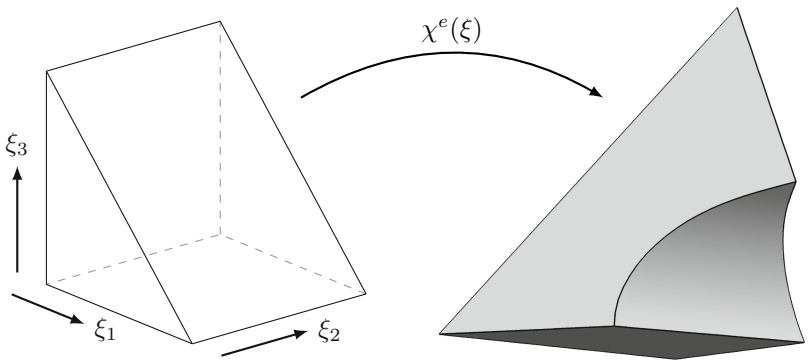


Fig. 5. High-order element mapping from a standard prism Ω_{st} to a corresponding prism Ω^e in Cartesian space

space $(\hat{\chi}_i)_{pqr}$ to an arbitrary set of quadrature points, the Cartesian positions of any volume-interior (or face-interior) points are blended in a linear manner from the curvature information that is supplied. Therefore, our existing mesh generation methodology only derives face-interior curvature information on a set of electrostatically-distributed triangular nodal points for faces of prisms representing the geometric surface. We assume without loss of generality that this face lies in the plane $\xi_2 = -1$ with respect to Ω_{st}. It should be noted therefore that in the ξ_2 direction, only a linear expansion is required so that $Q = 1$ in (1). This curvature information is transformed to the coefficient space through a Galerkin projection to obtain $(\hat{\chi}_i)_{pqr}$.

2.4 Splitting the Boundary-Layer Mesh

The goal of the splitting process is to refine each macro-element in the wall-normal direction ξ_2, so that at the wall surface, the flattest prism is correctly sized to resolve a desired boundary layer thickness. Where the Reynolds number is high, extremely thin elements are required in order to resolve the high shear of the flow, and therefore the method must be capable of generating valid elements of large aspect ratio, even where the curvature of the surface is large. To guarantee the validity of the refined elements, we pick a refinement strategy which focuses on first splitting the standard prismatic element into a series of n smaller elements distributed by a spacing Δ_k for $1 \leq k \leq n$. We then utilise χ^e in order to determine curvature information for each sub-element of the refinement. This projects the subelements back to Cartesian space, thereby splitting the original prism into curved prismatic subelements. This is depicted in figure 6.

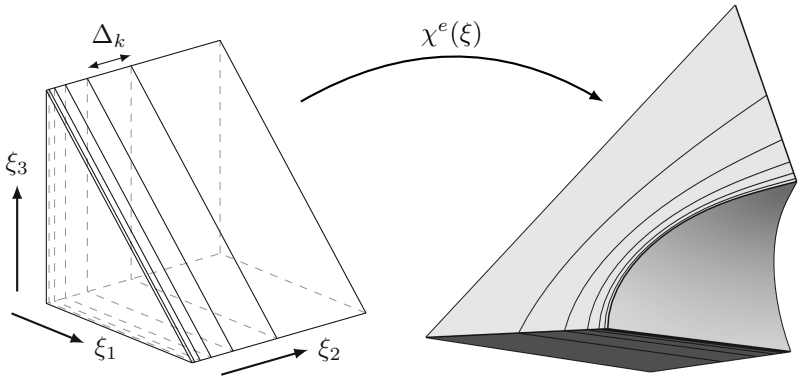


Fig. 6. Splitting Ω_{st} and applying the mapping χ^e to obtain a high-order layer of prisms from the macro-element of figure 5

Assuming that we wish to produce n subelements for each boundary layer prism, we select a partition of $[-1, 1]$ defined as $-1 = x_0 < x_1 < \cdots < x_n = 1$, for every element to be refined. This gives rise to a spacing distribution $\Delta_k = x_k - x_{k-1}$ which denotes the width of each sub-element in the reference element as shown in figure 6. The spacing of these points is unimportant in the following method. In order for the mesh to be conformal, we assume that n and Δ_k are fixed across all of the prismatic macro-elements. We describe in more detail a method for defining Δ_k in section 2.6, and how the distribution of points can be varied between elements if we relax the guarantee of mesh conformity.

Each prism is split by iterating over the number of subelements and constructing each one in a bottom-up fashion; that is, vertices are derived first, followed by edge-interior points and finally face-interior points. We calculate the position of these points inside Ω_{st}, and then apply the mapping χ^e at each point in order to compute the Cartesian coordinates. We note that edge and face interior points are only calculated for the two triangular faces of the sub-element, since the use of a linear expansion in the ξ_2 direction of the original prism means high-order information is not required.

If the high-order prismatic macro-element is valid, we can argue that the mesh is also valid after subdivision as follows. We will assume that the sides of the elements in the direction normal to the wall (ξ_2) are straight lines. For prismatic elements, the process of subdivision is thus equivalent to an affine transformation that will preserve the sign of the Jacobian of the mapping and therefore the validity of the mesh. Further, the subelements span the same polynomial space as the original macro-element, therefore the use of the macro-element mapping to define the sub-element mappings means they span an identical space and so have identical properties to the original element.

2.5 Fully Tetrahedral Meshes

Many solvers do not have the capability of producing solutions on hybrid meshes, instead preferring to consider meshes containing only simplicial elements (i.e. triangles and tetrahedra). This poses a considerable challenge for high-order mesh generation, particularly when extremely fine boundary layer elements are required.

One strategy is to adapt the prismatic boundary layer mesh by splitting each prism into three tetrahedra. Techniques such as the one presented in [3] are straightforward to implement and can be used to reliably split prismatic elements in a conformal manner. However these techniques are designed for linear finite element meshes. When curvature is introduced to the face of the tetrahedron lying on the surface of the geometry, and the prismatic element is sufficiently thin, then the lack of curvature on other faces causes the tetrahedron to self-intersect. This is illustrated in the lower half of figure 7.

To bypass this problem, we demonstrate how the refinement in the section above can be extended to introduce curvature into each of the tetrahedral elements in such a way that the elements are valid, as depicted in the upper half of figure 7. We begin by constructing the mapping χ^e defined above for each prism in the hybrid mesh. For the tetrahedral elements which are to be generated, we define a reference tetrahedron $\Omega_{\text{st}}^{\text{tet}} = \{(\xi_1, \xi_2, \xi_3) \mid \xi_1 \in [-1, 1], \xi_1 + \xi_2 \leq 1, \xi_1 + \xi_2 + \xi_3 \leq 1\}$ and a mapping ζ^e, which for each tetrahedron is represented by an expansion of modal functions

$$\zeta_i^e(\xi_1, \xi_2, \xi_3) = \sum_{p=0}^{P} \sum_{q=0}^{Q-p} \sum_{r=0}^{R-p-q} (\hat{\chi}_i)_{pqr} \psi_p^a(\eta_1) \psi_{pq}^b(\eta_2) \psi_{pqr}^c(\eta_3), \tag{2}$$

where $P \leq Q \leq R$. We again utilise a collapsed coordinate system $(\eta_1, \eta_2, \eta_3) \in [-1, 1]^3$ with

$$\eta_1 = \frac{-2(1+\xi_1)}{\xi_2 + \xi_3} - 1, \quad \eta_2 = \frac{2(1+\xi_2)}{1-\xi_3} - 1, \quad \eta_3 = \xi_3,$$

and the additional modified hierarchical basis function ψ^c is defined in [7].

We then proceed in a similar fashion to the prismatic splitting algorithm by first splitting the reference element of the prism, as opposed to the prism in Cartesian space. For each prismatic element in the domain, which we assume has been generated by the procedure in the previous section, we construct the isoparametric mapping χ^e. Assuming that each vertex in the mesh has a unique identifying number, we split $\Omega_{\text{st}}^{\text{prism}}$ into three tetrahedra according to the criterion defined in [3]. This ensures that where two prisms are connected by a quadrilateral face, the face is split in a consistent manner and a conformal mesh is produced. High-order information is then inserted into each tetrahedron in a bottom-up fashion by evaluating χ^e at positions in the standard prismatic element which correspond to the edge- and face-interior points. The insertion of this curvature information prevents each tetrahedron self-intersecting, leading to the generation of a valid mesh.

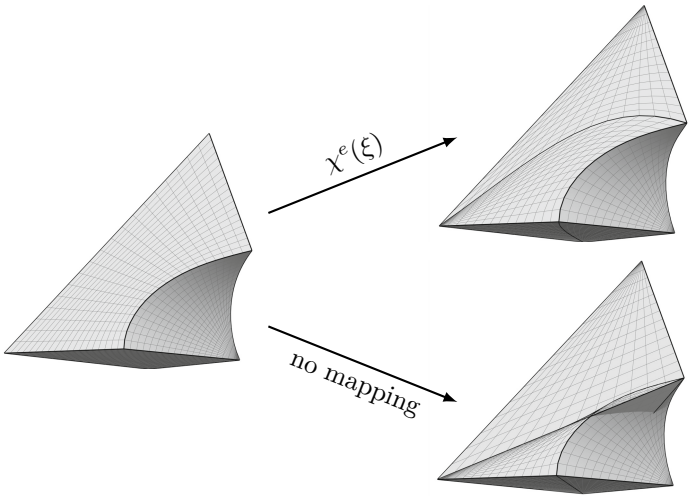


Fig. 7. Splitting a valid prismatic element at polynomial order $P = 20$ into tetrahedra. Applying the mapping χ^e leads to three valid tetrahedra (above), whereas preserving curvature on only a single face can lead to self-intersection of one or more tetrahedra (below).

An examination of the conditions under which the subdivision process produces valid elements, not only for the refinement procedure outlined here, but for more generic transformations of the standard element is presented in Section 3.

2.6 Normal Mesh Spacing Following a Geometric Progression

In this section we describe the form of the spacing function which determines the height Δ_k of each sub-element of the refined mesh. Recall that given a partition $-1 = x_0 < \cdots < x_n = 1$ of $[-1, 1]$, we set $\Delta_k = x_k - x_{k-1}$ for $1 \leq k \leq n$. To generate a spacing which gradually refines the elements towards the boundary lying at $\xi_2 = -1$, we utilise a geometric progression

$$x_k = x_{k-1} + ar^k, \quad a = \frac{2(1-r)}{1-r^{n+1}}$$

for $1 \leq k < n$ with $x_0 = -1$ and fix $x_n = 1$. The parameter r controls the relative sizes of elements, and $\Delta_1 = ar - 1$ is the height of the element closest to the boundary. Since in this formulation a is a function of r, we note that given a desired physical thickness (for example, a requirement for the element closest to the wall to have a thickness corresponding to $y^+ = 1$), one may derive the necessary value of r given n or vice versa. Figure 8 illustrates a sequence of meshes with various values of r (and thus y^+) that can be obtained using this

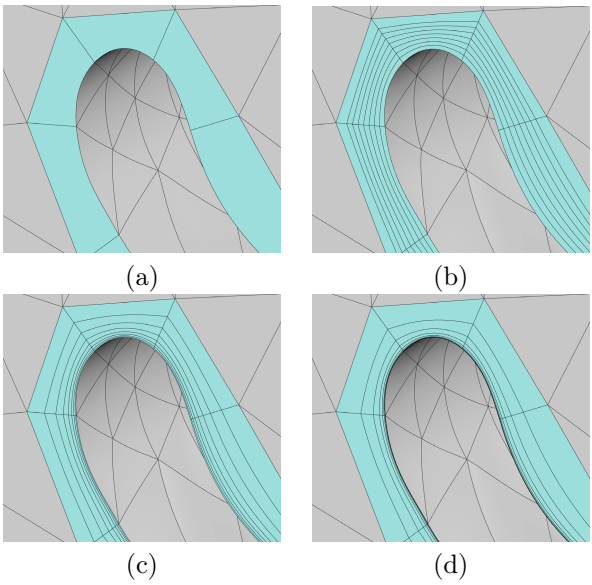


Fig. 8. A sequence of meshes obtained by splitting macro-elements into $n = 8$ elements using a geometric progression and various values of r. (a) The macro-element mesh; (b) $r = 1$; (c) $r = \frac{3}{2}$; (d) $r = 2$.

process. Additionally we note that if $r = 1$ as in figure 8(b), then a is ill-defined. Since $r = 1$ corresponds to the case where all elements are of an equal size, we assume a uniform spacing so that $\Delta_k = \frac{1}{n}$.

A further point to note is that a skin friction obtained from an empirical relation $C_f = F(\mathrm{Re})$ is usually derived using flat-plate calculations or regression based on experimental measurements. However, for complex geometries, there is usually no way to predetermine an exact value for C_f, and therefore a simulation is required to calculate an accurate value. An empirical relation gives an estimate for an initial simulation, and based on a series of experiments where the mesh is further refined (or coarsened) an accurate value of C_f can be obtained.

In addition, one can also smoothly vary the geometric factor as a function of spatial position so that different areas of the mesh receive varying levels of refinement. For each macro-element Ω^e, we consider an elementally-varying spacing distribution Δ_k^e. In order for the mesh to be conformal, it is clear that the splitting procedure must be altered slightly, so that the subelements which are produced connect correctly through quadrilateral faces. We consider a simple procedure where, for each of the three edges which connect the two triangular faces of a prism, we arbitrarily choose a spacing function from any of the prismatic macro-elements elements which share this edge. When each sub-element is

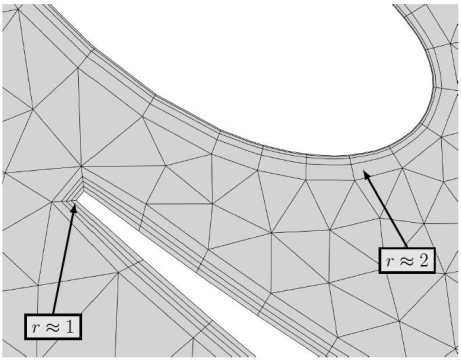


Fig. 9. Example of spatially-varying distribution of ratio of element size, r, for a staggered wing configuration

constructed, we use the distribution of points previously defined on these three edges. The method then proceeds in an identical fashion. By adopting this simple technique we assume that Δ_k^e varies smoothly so that the jump between different spacing distributions is not 'too large', or else the addition of high-order curvature may cause subelements to self-intersect.

We consider an example of this spatially-varying distribution in figure 9, which depicts a staggered wing configuration. The thin trailing edge of each blade dictates that extremely small elements are required. When the mesh is split with a constant distribution of points, any solvers which have a CFL constraint are therefore forced to use a prohibitively small timestep in order to maintain a stable simulation. This problem can be solved without varying the spacing distribution and by increasing the height δ of the macro-elements, but the close proximity of the two wings can cause issues when generating the volume mesh of tetrahedra in the space between the wings. In this case then, the use of a spatially-varying height distribution is essential, and we smoothly vary the ratio of element sizes so that at the leading edge $r \approx 2$ whereas at the trailing edge $r \approx 1$.

3 Subdivision Strategies and Analysis of Element Validity

The purpose of this section is to frame the subdivision technique in the context of a more general mathematical framework, to investigate the range of conditions under which the resulting meshes are valid, and to demonstrate how it can be utilised to subdivide a broader range of elemental types in both two and three dimensions. We note that in general, the subdivision of elements in this manner

often requires the enrichment of the polynomial space so that the subdivided elements capture all curvature of the original element.

3.1 Mathematical Framework

We begin by considering a finite element Ω^e, which in general belongs to a mesh arising from the tesselation $\mathcal{T}(\Theta) = \{\Omega^1, \ldots, \Omega^{N_{\text{el}}}\}$ of some domain $\Theta \subset \mathbb{R}^n$ with $n = 2, 3$ of N_{el} elements, so that

$$\Theta = \bigcup_{e=1}^{N_{\text{el}}} \Omega^e, \qquad \Omega^e \cap \Omega^f = \emptyset \text{ if } e \neq f.$$

In two dimensions, we consider quadrilateral and triangular elements, and in three dimensions tetrahedral, prismatic and hexahedral elements. In order to introduce curvature into an element Ω (where we drop the superscript e for convenience), we assume there exists a mapping $\chi : \Omega_{\text{st}} \to \Omega$ which projects a canonical standard element Ω_{st} into the Cartesian coordinates defining an element. In this work we define reference elements to be

$$\begin{aligned}
\Omega_{\text{st}}^{\text{quad}} &= \{(\xi_1, \xi_2) \quad\quad | \; -1 \leq \xi_1, \xi_2 \leq 1\}, \\
\Omega_{\text{st}}^{\text{tri}} &= \{(\xi_1, \xi_2) \quad\quad | \; -1 \leq \xi_1 + \xi_2 \leq 1\}, \\
\Omega_{\text{st}}^{\text{hex}} &= \{(\xi_1, \xi_2, \xi_3) \; | \; -1 \leq \xi_1, \xi_2, \xi_3 \leq 1\}, \\
\Omega_{\text{st}}^{\text{pri}} &= \{(\xi_1, \xi_2, \xi_3) \; | \; -1 \leq \xi_1 + \xi_3 \leq 1, -1 \leq \xi_2 \leq 1\}, \\
\Omega_{\text{st}}^{\text{tet}} &= \{(\xi_1, \xi_2, \xi_3) \; | \; -1 \leq \xi_1 + \xi_2 + \xi_3 \leq 1\},
\end{aligned}$$

respectively. Inside the standard elements we define a polynomial space in terms of the reference coordinates $\xi = (\xi_1, \xi_2, \xi_3)$ from which an expansion basis is selected. Assuming that we select a polynomial order P, Q and R for each coordinate direction, the polynomial spaces take the form

$$\mathcal{P}(\Omega_{\text{st}}) = \text{span}\{\xi_1^p \xi_2^q \xi_3^r \mid (pqr) \in \mathcal{I}\}$$

where $\mathcal{I}$ represents an indexing set, defined for each element as

$$\begin{aligned}
\mathcal{I}^{\text{quad}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq q \leq Q,\ r = 0\} \\
\mathcal{I}^{\text{tri}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq p + q \leq Q,\ r = 0,\ P \leq Q\} \\
\mathcal{I}^{\text{hex}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq q \leq Q,\ 0 \leq r \leq R\} \\
\mathcal{I}^{\text{pri}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq q \leq Q,\ 0 \leq p + r \leq P,\ P \leq R\} \\
\mathcal{I}^{\text{tet}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq p + q \leq Q,\ 0 \leq p + q + r \leq R,\ P \leq Q \leq R\}
\end{aligned}$$

In order to preserve the positivity of discretised spatial operators, we insist that given the components of $\chi = (\chi_1, \ldots, \chi_n)$ the determinant of the Jacobian matrix

$$[J_\chi(\xi)]_{ij} = \frac{\partial \chi_i(\xi)}{\partial \xi_j}, \qquad i, j = 1, \ldots, n$$

is positive for all $\xi \in \Omega_{\text{st}}$, so that χ preserves orientation and is invertible. Furthermore we consider an isoparametric representation of χ in terms of a set of shape functions ϕ_{pqr}, so that

$$\chi_i(\xi) = \sum_{(pqr)\in\mathcal{I}} (\hat{\chi}_i)_{pqr} \phi_{pqr}(\xi)$$

In the numerical examples below we consider an expansion in terms of a tensor product of modified hierarchical modal functions which permits a boundary-interior decomposition [7]. We note however that in this setting the choice of shape function is relatively unimportant, so long as they span the polynomial space of the element. However, as we will demonstrate later, this choice of basis is useful for certain types of elemental subdivisions as it permits fewer restrictions on the resulting subelement polynomial spaces.

3.2 Subdivision into the Same Element Type

In this section we demonstrate how the isoparametric mapping χ, which we assume has positive Jacobian for all $\xi \in \Omega_{\text{st}}$, can be used to subdivide an element into smaller elements of the same type. The goal of the subdivision process is to obtain a mapping $\zeta : \Omega_{\text{st}} \to \widetilde{\Omega}$ where $\widetilde{\Omega} \subset \Omega$ and $\det J_\zeta(\xi) > 0$ for all $\xi \in \Omega_{\text{st}}$.

In the isoparametric approach we adopt here, instead of attempting to determine the exact subdomain $\widetilde{\Omega}$ of the physical element Ω, we select a subdomain of the standard region, $\widetilde{\Omega_{\text{st}}}$, and construct an invertible mapping $f : \Omega_{\text{st}} \to \widetilde{\Omega_{\text{st}}}$ with $\det J_f(\xi) > 0$. Initially, we also assume that the polynomial expansion in each direction is equal so that $P = Q = R$. Setting ζ as the composition $\chi \circ f$ we then obtain a subelement $\widetilde{\Omega} = \zeta(\Omega_{\text{st}})$.

The justification for the validity of ζ, and moreover the resulting element $\widetilde{\Omega}$ under the restriction of equal polynomial order is as follows. Firstly, it is clear that the determinant of the Jacobian of ζ is positive for any $\xi \in \Omega_{\text{st}}$, since through an application of the chain rule we have that

$$\det J_\zeta(\xi) = \det J_\chi(f(\xi)) \det J_f(\xi) > 0. \tag{3}$$

Let us assume that each component of χ lies in the polynomial space $\mathcal{P}(\Omega_{\text{st}})$. In order for ζ to retain the isoparametric representation of the subelements, we note in turn that each of its components must be defined in a polynomial space $\mathcal{P}'(\Omega_{\text{st}})$ where in the most general case, $\mathcal{P}(\Omega_{\text{st}}) \subset \mathcal{P}'(\Omega_{\text{st}})$. A consequence of subdivision therefore is that the subdivided elements may have a higher polynomial order than the parent element depending on the choice of f.

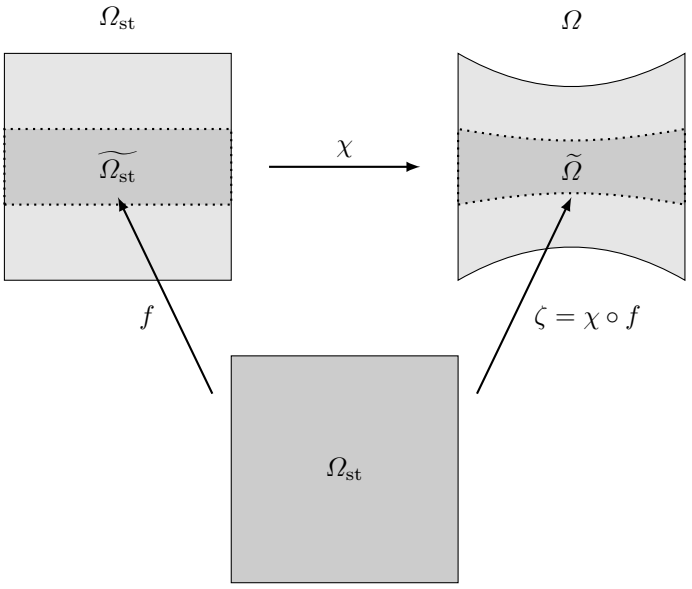


Fig. 10. Construction of the mapping ζ for the subdivision of a quadrilateral element

Figure 10 shows a simple application of this subdivision strategy for a quadrilateral element. Here we choose for example an affine mapping $f(\xi_1, \xi_2) = (\xi_1, c\xi_2)$ for some $c \in (0, 1)$ so that the standard element is scaled in the ξ_2 direction. Applying the original χ mapping we obtain a new element $\widetilde{\Omega}$ which is appropriately scaled, and naturally introduces curvature into the resulting subelement. In this case, any polynomial term $\xi_1^p \xi_2^q$ is mapped under f to the term $c^q \xi_1^p \xi_2^q$ which clearly lies in $\mathcal{P}(\Omega_{\text{st}}^{\text{quad}})$, and indeed it is clear that by equation 3 that $\det_\zeta(\xi)$ is simply a scalar multiple of $\det_\chi(\xi)$. We may therefore choose $\mathcal{P}'(\Omega_{\text{st}}) = \mathcal{P}(\Omega_{\text{st}})$ and the order of the subelements may be the same as the parent element.

Since the restriction of equal polynomial order is somewhat restrictive, we now consider the case where the polynomial order in each direction is not equal. Whilst a similar argument to the previous explanation can be used in this case, more care must be taken either in the choice of the mapping f or in the order of the resulting subelements to ensure that the polynomial space is correctly spanned. For example, consider a quadrilateral element with expansion orders $P = 2$ and $Q = 1$ which has corresponding polynomial space $\mathcal{P}$, and suppose we choose to produce a trivially subdivided element by applying the transformation $f(\xi_1, \xi_2) = (-\xi_2, \xi_1)$. This map has positive Jacobian determinant and indeed is affine, as in the previous example. However, since ξ_1 and ξ_2 are permuted in the composition with f, the expansion has polynomial terms which lie outside of $\mathcal{P}$ leading to unpredictable element generation.

There are two solutions in this case. Firstly we may choose to obey the general condition $\mathcal{P}(\Omega_{\text{st}}) \subset \mathcal{P}'(\Omega_{\text{st}})$, and enrich the polynomial order of the subelement so that $P = Q = 2$. Alternatively however, we may permute the polynomial orders of the resulting subelements, so that $P = 1$ and $Q = 2$, to form a space $\mathcal{Q}$. We see in this instance that the resulting subelement is still valid as all of the terms of the original χ expansion are represented in ζ, but the previous condition is not held since $\mathcal{P} \not\subset \mathcal{Q}$. We therefore note that $\mathcal{P}(\Omega_{\text{st}}) \subset \mathcal{P}'(\Omega_{\text{st}})$ represents a sufficient, but not necessary condition on the validity of subelements in this case.

A similar warning also applies to the other element types, and in particular triangles, prisms and tetrahedra since additional conditions are placed on the summation of mode indices which must be observed. In the next section, we discuss a similar enrichment strategy to permit the subdivision of elements into different element types.

3.3 Subdivision into Different Element Types

Another possible strategy one may adopt when subdividing elements is to consider their division into elements of a different type; for instance, we may subdivide a quadrilateral into triangles in two dimensions, or alternatively hexahedra into prisms or prisms into tetrahedra in three dimensions. Such techniques are well understood for linear finite elements [3] but for curvilinear elements self-intersection may occur, as outlined in the previous section. In this section we demonstrate how the technique introduced in the previous section can be adapted in a more general way than simply prismatic to tetrahedral splitting in order to introduce curvature into the subelements in such a way as to prevent them becoming invalid.

We must adapt the previous argument above since now $f : \Omega'_{\text{st}} \to \widetilde{\Omega}$ where $\Omega'_{\text{st}} \subsetneq \Omega_{\text{st}}$ and so the polynomial spaces which span these standard elements obey the relation $\mathcal{P}(\Omega'_{\text{st}}) \subsetneq \mathcal{P}(\Omega_{\text{st}})$. In the same way that the technique needs an enrichment of the polynomial space if direction-dependent polynomial orders are used, if we naively apply the method then the polynomial expansion χ can contain terms which are not contained inside $\mathcal{P}(\Omega'_{\text{st}})$, and so the resulting mapping ζ may not produce valid elements.

To demonstrate this point, we first examine the problem of figure 11, which depicts an example where a quadrilateral is split along a diagonal edge in order to obtain two triangles. We may again utilise an affine mapping $f(\xi) = -\xi$ in order to map $\Omega_{\text{st}}^{\text{tri}}$ onto a subdomain $\widetilde{\Omega_{\text{st}}}$ of $\Omega_{\text{st}}^{\text{quad}}$. From our previous argument we see that each component of $\zeta = \chi \circ f$ has degree $2P$ in general if the original quadrilateral is of order P.

Since $\zeta \in [\mathcal{P}(\Omega_{\text{st}}^{\text{tri}})]^2$ we must select a sufficiently large polynomial order for the triangular space so that all terms of the expansion are represented in the resulting expansion. To guarantee this for a general quadrilateral-to-triangle split, given a quadrilateral of order P we must generate triangles of order $2P$. Then the space $\mathcal{P}(\Omega_{\text{st}}^{\text{quad}}) \subset \mathcal{P}(\Omega_{\text{st}}^{\text{tri}})$ and thus ζ captures all curvature of the original mapping. For a visual illustration of this, we may represent the polynomial spaces of the

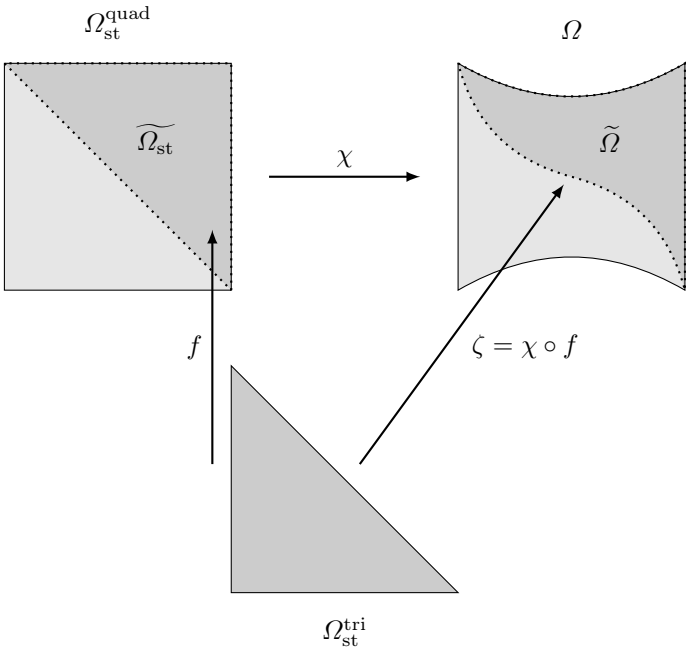


Fig. 11. Construction of the map ζ in the case of a quadrilateral being split into two triangles

triangular and quadrilateral elements in the form of a Pascal's triangle as shown in figure 12.

Figure 13 illustrates the problem of using triangular elements which are not sufficiently enriched. On the left, a second-order ($\mathbb{P}^2$) quadrilateral is split into two second-order triangles. Splitting the quadrilateral into two $\mathbb{P}^2$ triangles leads to the generation of degenerate elements. In this case, the symmetry of the deformed element coupled with the quadratic order of the triangles means that the diagonal edge which bisects the quadrilateral is forced to remain straight and thus causes a self-intersection. We note that in this example, the interior quadrilateral mode $\xi_1^2\xi_2^2$ is not energised since curvature is only introduced in one coordinate direction. We additionally note that this can be intuitively achieved by the choice of a boundary-interior hierarchical expansion in which edge and vertex degrees of freedom are decoupled from the interior. Other basis types, such as a nodal Lagrange scheme, will not in general have this property, although the use of the classical Gordon-Hall blending does have this property. Consulting the Pascal triangle of polynomial spaces we therefore see that only a $\mathbb{P}^3$ expansion is required for the triangular elements.

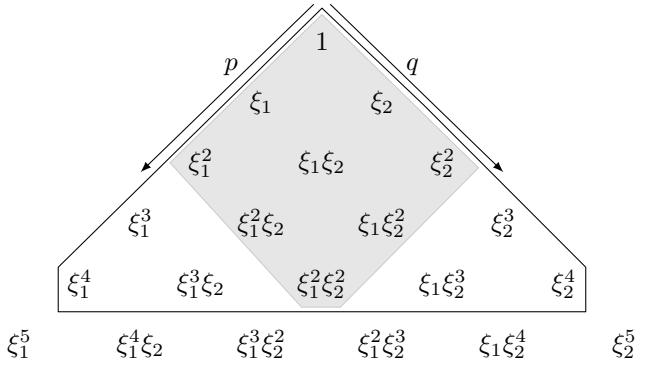


Fig. 12. Pascal's triangle representing the polynomial spaces of $\mathbb{P}^2$ quadrilateral (shaded grey) and $\mathbb{P}^4$ triangular (black outline) elements. The triangle shows that in order to split a general $\mathbb{P}^2$ quadrilateral we require $\mathbb{P}^4$ triangles so that all terms can be represented in the resulting mapping.

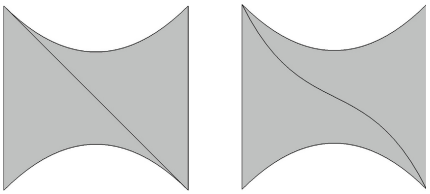


Fig. 13. Qualitative example of the necessary condition for subdivision. A $\mathbb{P}^2$ quadrilateral is split into $\mathbb{P}^2$ (left) and $\mathbb{P}^3$ (right) triangles. Since a $\mathbb{P}^2$ triangular expansion does not capture some of the terms of the original mapping, an additional order is required to produce valid elements.

The same logic can be used in the splitting of prismatic and hexahedra elements into tetrahedra. In general an order P prismatic or hexahedral element also requires enrichment so that the resulting tetrahedra have order $2P$ and $3P$ tetrahedra. However by applying the logic above, if curvature is introduced only into the triangular faces of the prisms, then it is only necessary to produce order $P + 1$ tetrahedra. Since visualisation of the Pascal's triangle structure is more difficult in three dimensions, this can alternatively be seen from a brief analysis of the prismatic and tetrahedral spaces. If a linear expansion is used in the homogeneous direction of the prismatic element (i.e. $Q = 1$) and $P = R$ then the resulting polynomial space is

$$\mathcal{P}^{\text{pri}}(\Omega_{\text{st}}) = \{\xi_1^p \xi_2^q \xi_3^r \mid 0 \leq p + r \leq P,\ q = 0, 1\}.$$

A tetrahedron with equal polynomial order P in each direction has the restriction on a triple (pqr) that $0 \leq p + q + r \leq P$. If $q = 1$ then we obtain the restriction $0 \leq p + r \leq P - 1$, and so the tetrahedral space at order P does not contain the prismatic space, leading to possible invalid elements. In order to guarantee validity of elements we therefore require tetrahedra of order $P + 1$.

In the following section, we give a demonstration of this prism-to-tetrahedron splitting and also highlight the application of the refinement method in boundary-layer problems.

4 Applications

This section demonstrates the usefulness of the subdivision method by showing how it can be used to generate three-dimensional meshes for challenging applications. Firstly we consider the subdivision of a coarse prismatic boundary-layer mesh into a series of progressively thinner elements as the distance to the wall decreases. We then show how the prismatic elements can be subdivided to obtain a boundary-layer mesh comprising only tetrahedra for use by solvers supporting only simplicial elements. Finally we produce boundary layer meshes for the ONERA M6 wing in order to demonstrate the applicability of the technique in aeronautical problems. For each case, we consider a high-order coarse discretisation at polynomial orders of $P \geq 10$ to show the viability of the method even at very high polynomial orders.

4.1 Boundary Layer Mesh Generation

In order to generate a sequence of n subelements which gradually become more slender towards the surface of the domain, we define a spacing distribution Δ_k for $1 \leq k \leq n$ following the procedure described in Section 2.6. Under the framework of section 3.1 then, we define a straightforward affine scaling function similar to that used in figure 10 which obeys the necessary conditions in order to generate valid subelements. We additionally note that as long as the same spacing distribution is used for all prismatic elements, the resulting mesh is conformal. One of the major advantages of this method for the generation of boundary layer meshes is that the resulting subelements are guaranteed to be valid, as shown in section 3.1, and thus we are able to produce boundary layers of arbitrary thickness. Figure 14 shows how the subdivision technique can be used to generate a boundary layer mesh for a simple geometry: a cylindrical duct.

Certain solvers only have support for meshes which are composed only of simplicial elements. For problems where boundary layers are required, this poses an additional problem for mesh generation software. In figure 14, we show how the same method can be used to split the prismatic elements of figure 14 into three tetrahedra. Firstly we note that in order for the resulting mesh to be

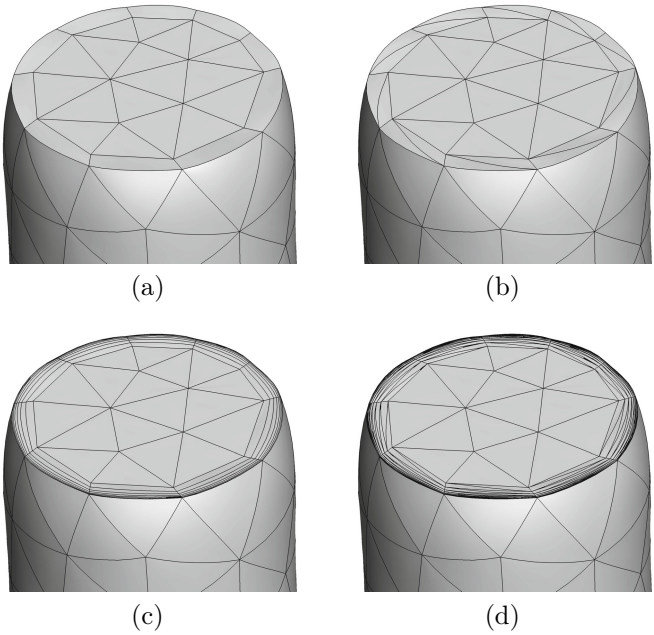


Fig. 14. Duct geometry: (a) Initial coarse mixed mesh with prisms in the boundary layer; (b) prism to tetrahedron split of the original mesh; (c) boundary layer refinement; and (d) we apply the prism to tetrahedron splitting after the boundary layer refinement has been performed.

conformal, we must employ a strategy so that the quadrilateral faces which connect prismatic elements are split in a consistent fashion, such as the one outlined by Dompierre et al. [3].

Once this strategy is applied, we may utilise the subdivision strategy to split the standard prismatic element into three tetrahedra by using an affine transformation similar to that used in figure 11. We note that in the specific case of figure 14, since the curvature of the original prisms is only imposed on the triangular surface, we may obtain valid tetrahedra by enriching the polynomial space by one order.

An important point to note is that whilst the validity of the resulting tetrahedra is guaranteed through our previous arguments, this method may lead to the production of tetrahedra which have suboptimal quality in terms of interior angles, depending on the curvature of the original prismatic elements. However, when tetrahedral boundary layers are required this is often unavoidable since the elements exhibit a large stretching ratio. In the very worst cases, the use of these meshes as a starting point for a mesh deformation procedure may lead to better quality elements. We suggest that the validity of the meshes produced here may lead to improved convergence speeds in such methods.

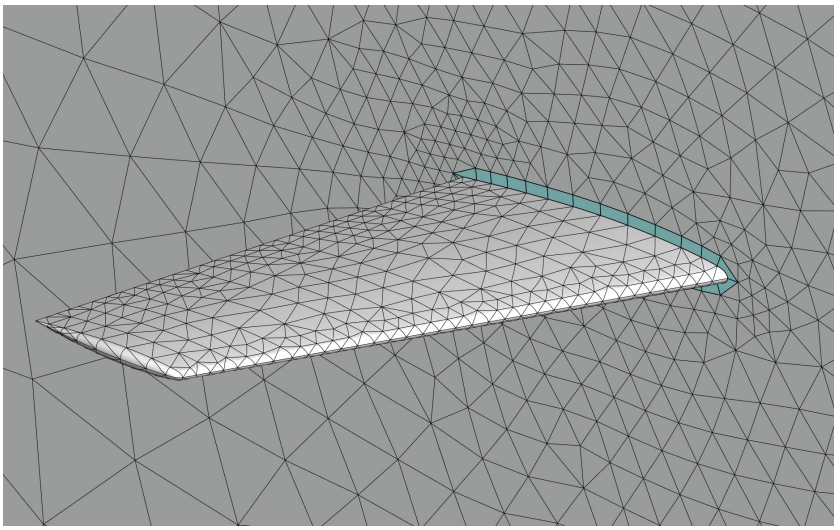


Fig. 15. Boundary of the initial coarse straight-sided mesh with prisms in the boundary layer (highlighted) and tetrahedra in the interior

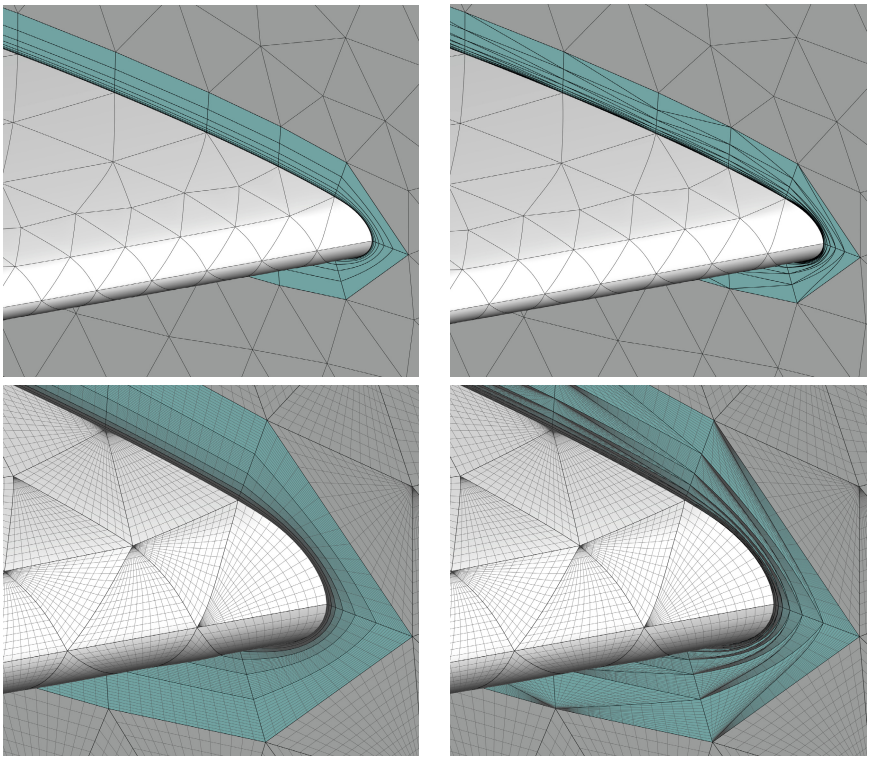
4.2 ONERA M6 Wing

The ONERA M6 wing is a classic CFD validation case for external transonic flow. The description of the geometry is given in [18]. Here we use this case as an illustration of the methodology for generating high-order meshes for high Reynolds number flows. In a typical simulation of transonic flow, the Reynolds number has a value of around 10^5, and therefore an extremely thin boundary layer is required in the wall-normal direction. However, in the transverse directions such high resolution is not necessarily required.

Figure 15 shows the initial coarse high-order surface mesh of the wing geometry coupled with a symmetry plane. The intersection of the symmetry plane with the boundary layer of the wing, composed of the prismatic macro-elements, is highlighted, with the remainder of the mesh comprising tetrahedral elements. In figure 16 we apply the splitting strategies from section 2 to obtain a mesh with $y^+ \approx 5$ at $\mathrm{Re} = 10^5$, where the skin friction C_f is approximated using the Schlichting formula [17]

$$C_f \approx (2\log(\mathrm{Re}) - 0.65)^{-2.3}\,.$$

Figure 16(a) shows the hybrid prismatic-tetrahedral mesh that is obtained after the splitting algorithm is applied. In figure 16(b) we apply the prism to tetrahedron splitting technique of section 2.5 to obtain a fully tetrahedral mesh. In all cases a valid mesh is generated, with no elements having negative Jacobian.



(a) (b)

Fig. 16. Enlargements of the initial coarse mesh demonstrating the splitting of the boundary layer into (a) prismatic and (b) tetrahedral elements near the leading edge at polynomial order $P = 15$. The lower picture highlights interior quadrature points of the elements.

5 Conclusions

We have presented a technique for generating highly stretched high-order boundary-layer meshes as required by current CFD solvers. The proposed technique is very effective and modular. Starting from a valid coarse prismatic boundary-layer mesh, our isoparametric approach permits the generation of a sequence of meshes with increased resolution with very little additional cost. This should prove very valuable for mesh convergence studies at high Reynolds numbers. We have established requirements of validity for modal elemental shape functions, but the same arguments are also applicable to guarantee the validity of the mesh when using nodal shape functions.

We have derived the mathematical conditions that are necessary for the subdivision of high-order isoparametric elements, and show how this technique can be applied to tackle challenges in high-order mesh generation. We posit that the

simplicity of the method outlined here will prove to be a valuable tool in improving both the efficiency and robustness of curvilinear mesh generation software, and particularly for the generation of meshes for high Reynolds number computational fluid dynamics problems or high Peclet number advection-diffusion problems.

The main limitation of the technique, as presented here, is the requirement that the subdivision of the prismatic mesh should be accomplished without affecting the rest of the mesh. Extending the method to other cases is not a difficult technical issue, but it will require the use of transition elements, such as pyramids. However, such implementation is beyond the scope of the work within the IDIHOM project.

References

1. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)
2. Dey, S., O'Bara, R.M., Shephard, M.: Curvilinear mesh generation in 3D. In: Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, California (1999)
3. Dompierre, J., Labbé, P., Vallet, M.-G., Camarero, R.: How to subdivide pyramids, prisms and hexahedra into tetrahedra. In: 8th International Meshing Roundtable, Lake Tahoe, California, October 10–13 (1999)
4. Gargallo-Peiró, A., Roca, X., Sarrate, J., Peraire, J.: Inserting curved boundary layers for viscous flow simulation with high-order tetrahedra. In: 22nd International Meshing Roundtable (2013)
5. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)
6. Gordon, W.J., Hall, C.A.: Construction of curvilinear co-ordinate systems and applications to mesh generation. International Journal for Numerical Methods in Engineering 7(4), 461–477 (1973)
7. Karniadakis, G.E., Sherwin, S.J.: Spectral/*hp* Element Methods for Computational Fluid Dynamics, 2nd edn. Oxford University Press (2005)
8. Mavriplis, D.J.: Unstructured grid techniques. Annual Review of Fluid Mechanics 29, 473–514 (1997)
9. Moxey, D., Hazan, M., Peiró, J., Sherwin, S.J.: On the generation of curvilinear meshes through subdivision of isoparametric elements. To Appear in Proceedings of Tetrahedron IV (2014)
10. Moxey, D., Hazan, M., Sherwin, S.J., Peiró, J.: An isoparametric approach to high-order curvilinear boundary-layer meshing. Under Review in Comp. Meth. App. Mech. Eng. (2014)
11. Peiró, J., Sayma, A.I.: A 3-D unstructured multigrid Navier-Stokes solver. In: Morton, K.W., Baines, M.J. (eds.) Numerical Methods for Fluid Dynamics V. Oxford University Press (1995)
12. Peraire, J., Morgan, K.: Unstructured mesh generation including directional refinement for aerodynamic flow simulation. Finite Elements in Analysis and Design 25, 343–356 (1997)

13. Peraire, J., Peiró, J., Morgan, K.: Multigrid solution of the 3-D compressible Euler equations on unstructured tetrahedral grids. International Journal for Numerical Methods in Engineering 36, 1029–1044 (1993)
14. Persson, P.-O., Peraire, J.: Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In: 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando (FL), USA, January 5–9. AIAA paper 2009–949 (2009)
15. Pirzadeh, S.: Three-dimensional unstructured viscous grids by the advancing-layers method. AIAA Journal 34(1), 43–49 (1996)
16. Sahni, O., Luo, X.J., Jansen, K.E., Shepard, M.S.: Curved boundary layer meshing for adaptive viscous flow simulations. Finite Elements in Analysis and Design 46(1-2), 132–139 (2010)
17. Schlichting, H.: Boundary-layer theory, 7th edn. McGraw-Hill (1979)
18. Schmitt, V., Charpin, F.: Pressure distributions on the ONERA-M6 wing at transonic mach numbers. In: Experimental Data Base for Computer Program Assessment, AGARD AR 138. Report of the Fluid Dynamics Panel Working Group 04 (May 1979)
19. Sherwin, S.J., Peiró, J.: Mesh generation in curvilinear domains using high-order elements. International Journal for Numerical Methods in Engineering 53(1), 207–223 (2002)
20. Toulorge, T., Geuzaine, C., Remacle, J.-F., Lambrechts, J.: Robust untangling of curvilinear meshes. Journal of Computational Physics 254, 8–26 (2013)
21. Xie, Z.Q., Sevilla, R., Hassan, O., Morgan, K.: The generation of arbitrary order curved meshes for 3D finite element analysis. Computational Mechanics 51(3), 361–374 (2013)
22. Zienkiewicz, O.C., Phillips, D.V.: An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co-ordinates. International Journal for Numerical Methods in Engineering 3(4), 519–528 (1971)

Development of High-Order Meshing for Industrial Aerospace Configurations

C. Johnston and S. Barnes

ARA Ltd, Manton Lane, Bedford, United Kingdom, MK41 7PF
{cjohnston,sbarnes}@ara.co.uk

Abstract. Within the IDIHOM project, ARA have worked to develop a high-order grid generation capability that allows the generation of meshes on three dimensional aircraft configurations typical of the complexity currently used in industrial finite volume simulations. Details of this capability are provided alongside examples of meshes generated using it and a discussion of its strengths and limitations. We conclude by considering the ways in which this existing capability may be further enhanced to provide a fully industrialised capability.

1 Introduction

One of the key limiting factors preventing the uptake of high-order methods by industry is the requirement to generate robustly quality boundary conforming unstructured curvilinear grids on complex configurations. Such grids are required to allow a significantly reduced number of grid elements compared to those used by industry standard (nominally) second-order accurate Reynolds-Averaged Navier-Stokes (RANS) flow solvers, thereby allowing the computationally expensive high-order simulations to be tenable. Equally importantly, however, there is evidence that using straight sided boundaries in a high-order discontinuous Galerkin simulation can yield physically incorrect results [1].

Although a number of promising tools for high-order grid generation already exist, see for example [2], there is still a significant gap between their capabilities and those of grid generators for producing finite volume RANS grids. In particular, there is a significant limitation in the ability of high-order grid generators to produce a high quality boundary layer grid on all but the simplest of geometries. Indeed, in most cases, they fail on complex configurations to even produce a grid satisfying the minimum criteria for it to be usable, that is, a conformal grid containing no elements with negative Jacobian or volume.

During IDIHOM, ARA have investigated one approach towards resolving the problems relating to high-order grid generation on complex aeropace configurations. This approach combines the well established linear mesh generator SOLAR [3–6] with an approach for generating curved elements developed in the IDIHOM predecessor project ADIGMA [7] and detailed in [8]. In Sect. 2 we briefly review the relevant aspects of the SOLAR grid generator and the work in [8]. In Sects. 3 and 4 we provide details of the approach developed in IDIHOM

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_4

and representative grids generated using this process. Finally in Sects. 5 and 6 we discuss the remaining issues still to be resolved and draw conclusions about the future direction of this work.

2 Background

2.1 The SOLAR Mesh Generator

The high-order meshing capability developed at ARA within IDIHOM builds upon the well-established SOLAR hybrid unstructured mesh generator which is used as part of a standard computational fluid dynamics (CFD) toolset by several major European aerospace manufactures, research institutes and academia. This mesh generator has been developed for over a decade and is designed to produce high-quality meshes for viscous finite volume flow simulations on complex aeronautical configurations. The near-field boundary layer grid is created using an advancing layer technique [9], illustrated in Fig. 1, to march away from an unstructured quadrilateral-dominant base surface mesh. Edge-collapsing and face-enrichment algorithms alter the topology of the layer to take into account automatically the underlying curvature of the region being meshed, with the rate of local layer growth being varied locally to achieve a smooth outer layer. In three dimensions, the near-field layer mesh is then coupled to either a Cartesian far-field mesh that uses cut cells to avoid overlap of the near and far-field meshes [3], or alternatively to a tetrahedral far-field mesh, generated using a Delaunay algorithm, via a buffer layer comprising pyramids and tetrahedra [5]. The latter of these approaches has the key benefit of avoiding the arbitrary polyhedra that are inevitably produced by a cut cell interface, instead generating a conformal grid containing hexahedra, prisms, pyramids and tetrahedra. Our work in IDIHOM uses meshes generated using this approach.

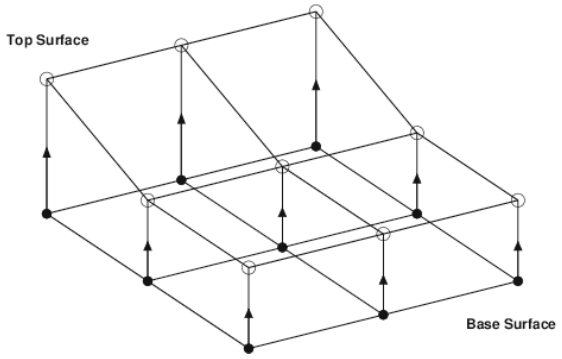


Fig. 1. Schematic illustration of the advancing-layer approach to grid generation. The base surface is projected out into the flow domain to obtain the top surface of the layer. The top surface is then used as the base surface for generation of the subsequent layer

2.2 The ADIGMA High-Order Grid Generator

The approach for high-order grid generation developed by ARA within ADIGMA was integrated fully within the SOLAR layer meshing code [8]. This was achieved by following the basic algorithm, designed to simplify the implementation by restricting the curved elements to the near-field buffer layer mesh, and illustrated schematically in Figs. 2 and 3:

- use the existing surface meshing functionality to produce a coarse surface mesh as an input to the main algorithm;
- use this mesh to provide the corner nodes defining the topology of the curvilinear boundary elements;
- use knowledge of the underlying CAD geometry plus the location of the corner nodes to identify where the additional nodes on element edges and faces should be inserted to obtain geometry conforming high-order boundary elements (Fig. 2);
- use a minor modification to the existing layer meshing algorithms to position the high order nodes on subsequent layers of the buffer layer mesh, a scaling factor being applied to reduce the layer growth of the high-order nodes to ensure a linear outer surface to the boundary layer mesh, enabling it to be conformally interfaced to linear elements in the buffer layer (Fig. 3).

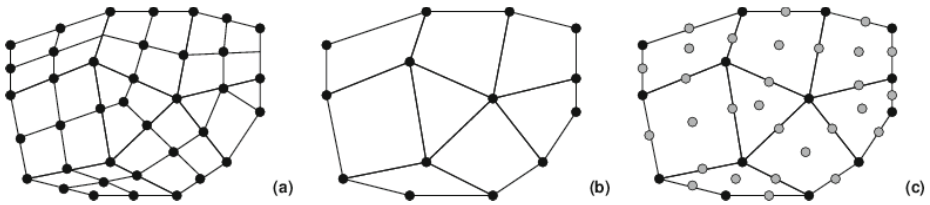


Fig. 2. Illustration of the process for generating a higher-order boundary representation. (a) illustrates the mesh that would be generated for use by a second order RANS solver. To obtain the higher-order boundary representation, the coarser mesh in (b) is first generated using the existing surface meshing process. This mesh contains the corner nodes (in black) used to define the extent of the element. In (c) the higher-order nodes (in gray) are then added to provide the higher-order boundary discretization

This approach was chosen since it required only relatively minor changes to the existing software: enabling the volume meshing routines access to the CAD geometry, previously used only by the surface mesher; adding code to insert the additional high-order nodes and store them consistently; and implementing export routines to allow the grid to be written in an appropriate format. Unfortunately, however, the approach of integrating the prototype high-order capability fully within SOLAR was found to be sub-optimal in terms of the future extensibility of the capability, imposing unacceptably adverse effects upon memory usage and run-time performance within the standard meshing process. In addition, inclusion of high-order elements prevented a number of the algorithms

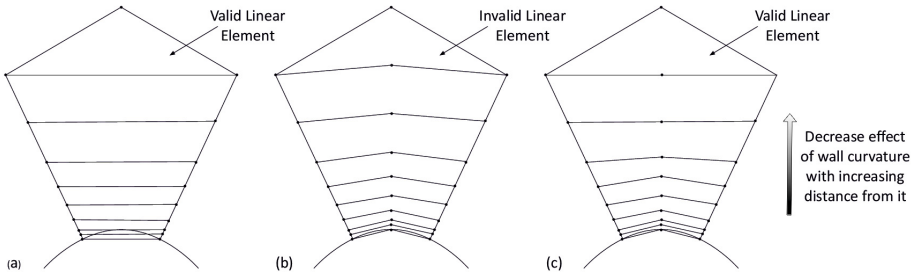


Fig. 3. Propagation of boundary curvature through the boundary layer grid. (a) represents the input linear grid containing valid linear elments. (b) illustrates the problems encountered at the outermost layer if a naive propagation approach is used. In this case the curvature in the outer surface of the boundary layer cannot be conformally interfaced to the linear buffer layer element. (c) illustrates the approach implemented by our algorithm, where the displacement of the high-order nodes is decreased with increasing distance from the wall, to obtain a linear element in the outer layer of the near-field grid.

used by SOLAR to ensure the quality of mesh elements from being applied without a significant effort in rewriting their implementation. Although in principle all these issues could have been addressed by a re-engineering of the SOLAR software architecture, within IDIHOM it was instead decided to decouple the high-order capability into a stand-alone post-processing stage. This enables all the capabilities of SOLAR to be fully utilised without constraint, enabling the high-order capability to benefit immediately from any relevant enhancements to SOLAR. Furthermore, it significantly simplifies the development of the code, allowing optimisation to be performed without any need for consideration of secondary impact elsewhere in the code base. Unlike the ADIGMA prototype, which works on a surface grid, this code instead takes as input a coarse linear volume grid, generated using SOLAR, and the CAD geometry used to generate this. The algorithms developed in ADIGMA are then used to curve the boundary and volume element using the steps described in Sect. 3.

3 High-Order Meshing Process

3.1 Basic Methodology

The basic algorithm developed for post processing an input linear grid to an equivalent high-order grid takes the following steps:

1. Input a linear volume grid, $\mathcal{M}$, that satisfies the necessary constraints for the overall process. These constraints are discussed fully in Sect. 3.4.
2. Obtain the dual graph of this mesh, $\mathcal{M}^*$.
3. Identify the set of surface elements within this mesh that lie on walls with a viscous boundary condition, $\mathcal{S}$.

4. For each member of $\mathcal{S}$:
   (a) Use $\mathcal{M}^*$ to identify the column (or stack) of hexahedral or prismatic elements above it. This column is determined by the opposite faces of the volume elements, and terminates when an element of different type is found, for example a pyramid element on top of a column of hexahedra.
   (b) Insert high-order nodes in the surface element in such a way that the underlying CAD geometry is captured.
   (c) Propagate the effects of curving the surface element through the column, in such a way that the validity of the volume elements is ensured, and the outermost element can be conformally matched to a linear element in the buffer layer between the boundary-layer and far-field tetrahedral grids.
5. Recombine the columns, buffer-layer and far-field grids to create the grid for the whole volume.
6. Output the volume grid in an appropriate format.

By applying this process the input linear grid can be converted to a high-order equivalent in the manner illustrated in Fig. 4. The key components of this process lie in the curving of the boundary elements, and propagating the effect of this through the near-field volume grid. These steps are considered in more detail in Sects. 3.2 and 3.3. Throughout this work we have only considered second-order (quadratic) elements, but there are no fundamental reasons why the approaches described could not be used to obtain higher-order elements.

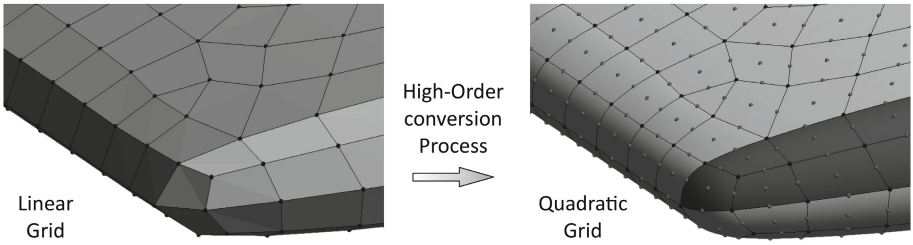


Fig. 4. Conversion of a coarse linear grid to high-order

3.2 Boundary Representation

In order to represent accurately the underlying CAD within the high-order grid we need to transform the linear surface elements, 3 vertex triangles and 4 vertex quadrilaterals, from the original grid into their high-order equivalents, 6 vertex triangles and 9 vertex quadrilaterals. These additional vertices must be added in such a way as to ensure an accurate capture of the underling CAD geometry.

To achieve this, two approaches were implemented; a relatively simple approach based upon edge subdivision and re-projection to the CAD surface; and an alternative approach which also includes the surface curvature in an attempt to capture better the underlying geometry. Regardless of which of these approaches is used to obtain the intermediate edge nodes, the face nodes were found by obtaining the mean of the edge node locations, then projecting this mean position onto the CAD surface.

In the first approach the intermediate high-order nodes on an edge are obtained simply by applying a point distribution along the linear edge and then projecting from these positions onto the CAD surface. In the current implementation a uniform point distribution is used, but there is no reason why an alternative choice, such as a Chebyshev-Lobatto point distribution, could not be used. The use of a uniform distribution is arguably not ideal, since it will fail to provide a good approximation to the underlying geometry should there be a significant change in curvature along the edge. In practice we have not found this to be a significant problem in terms of obtaining a valid high-order grid. The example grids shown in Sect. 4 were generated using this approach.

Our second approach to curving the boundary attempts to address some limitations of our first implementation. To curve an edge E we first identify an approximation to the geodesic between its two end points on the CAD. We obtain this approximation thus:

1. Find the vector $\boldsymbol{v}$ that is the mean of the normal vectors on the two faces adjacent to the edge.
2. Find the plane P containing the two end nodes of E that is parallel to $\boldsymbol{v}$.
3. Find the curvature of the geometry in this plane at the location of the two end nodes.
4. Use this curvature information and the locations of the end nodes to provide the boundary conditions used to obtain the constants in the parametric curve

$$\boldsymbol{r} = \boldsymbol{a}t^3 + \boldsymbol{b}t^2 + \boldsymbol{c}t + \boldsymbol{d} \ , \tag{1}$$

   where $t = 0$ and $t = 1$ correspond to the end nodes of the edge.

The position of the intermediate high-order nodes are then obtained by applying the desired distribution to the parameterization, obtaining the physical location of the corresponding point on the geodesic by substitution into (1) and then projecting from this location onto the CAD surface.

3.3 Layer Mesh Modification

The approach to curving elements within the layer mesh closely follows the algorithm developed in [8]. The intention here is to obtain columns that propagate the curvature of the boundary away from the wall, preventing problems of mesh cross over that could occur if the curved boundary representation is not accounted for (see Fig. 5), whilst also ensuring that the outermost face of the column is flat.

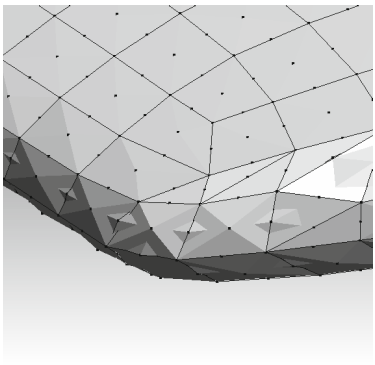


Fig. 5. Illustration of mesh cross over in the wing-tip region of the ONERA M6 geometry if the high-order boundary representation is not accounted for. The figure shows the surface of the first layer of the boundary layer mesh, through which the surface mesh can be seen to protrude.

As shown previously in Fig. 3, a naive algorithm for curving elements in a column would result in a non-conforming interface between near- and far-field, and hence a non-conservative system. To avoid this issue, an approach is introduced, by which the effect of wall curvature is systematically decreased with increasing distance from the wall. This is achieved in two stages. First, the displacements of the high-order nodes that would be used if the naive approach was followed are calculated. These displacements are obtained for the surface nodes by calculating the displacement vector from where they would be positioned if inserted parametrically on the corresponding edge in the linear grid. These surface node displacements are then mapped and applied to the corresponding nodes immediately above in the column, but scaled by a factor which reduces their effect as the layer number increases. In principle, any scaling function satisfying

$$0 \leq f(\text{layer}) \leq 1 \tag{2}$$

and

$$f(\text{layer} + 1) \leq f(\text{layer}) \tag{3}$$

can be used for this purpose, but in practice, the sigmoid function used in [8] was found to yield good results. This function takes the form illustrated in Fig. 6 and is described by the expression

$$f(x) = \left\{ 1 + \exp\left(\frac{(x - Np)}{\sigma} \right) \right\}^{-1} \quad , \tag{4}$$

where x defines the layer, N the total number of layers in the column, p controls the location of the step down from 1 to 0 and σ controls the width of this step. Since the height of layers in the near-field mesh is generally fixed for an initial number of layers and then increased according to a power series, this function

has the desired property of having minimal effect very near to the wall, only scaling back the effects of curvature in the larger elements away from the wall. Here the effect of this process upon element quality can be minimised. Other formulae were considered for this purpose but were found to have no benefit over (4). The developed capability allows for the user to change the values of the parameters p and σ in (4), but in practice default values of $p = 0.85$ and $\sigma = 1$ were found to perform well.

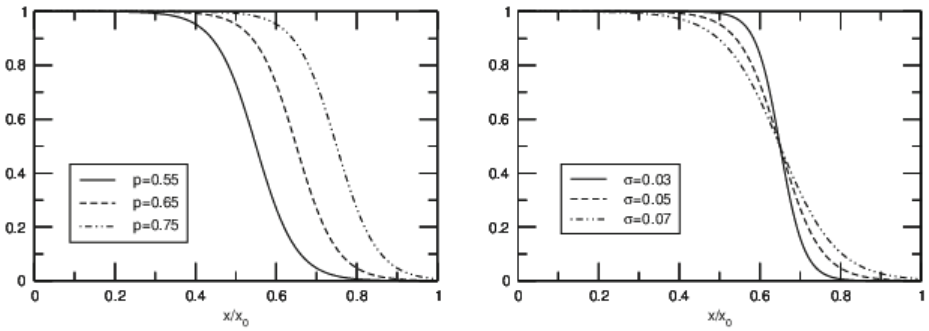


Fig. 6. Form of the sigmoid function used to scale displacements and the effect of varying the parameters

3.4 Constraints on the Input Grid

Although the capability developed at ARA takes as input a SOLAR grid, the process detailed in Sects. 3.1 to 3.3 can in principle be used to curve an arbitrary input grid, provided that it satisfies a number of necessary constraints. First, in its current form there is an implicit assumption that the linear grid possesses an O-type topology and that each of the columns are ultimately terminated by an element of different type before the far-field is reached. This would prevent application of the current algorithm to, for example, a block structured grid. The second of these constraints would be relatively easy to remove, simply by setting a maximum number of elements to grow away from the wall, but the first would require amendment to the algorithm to allow for the scenario of an element being the member of two columns when in a junction region.

Second, the input grid is required to be both conformal and contain no negative volume elements, since the algorithm performs no explicit checks regarding this. The production of non-conformal elements was a significant limitation for the process in [8] since existing post-processing tools that would remove such issues are incompatible with high-order elements. Since these tools can now be applied on the linear inputs to the process, this issue can be almost eliminated in the current approach. If any issues do remain, the method for robust untangling of curvilinear meshes developed by Remacle *et al.* [10] may provide a solution.

Finally, a non-conformal high-order grid will be produced if the height, in terms of element count, of neighbouring columns is inconsistent, a scenario that is likely to occur within SOLAR on complex configurations, where layer growth may be terminated in order to ensure mesh quality and validity. The reason for this problem is illustrated schematically in Fig. 7. If a column is terminated by a linear element and this is not reflected in neighbouring columns, then a non-conformality between neighbouring faces is introduced. Within our implementation this issue is dealt with by first identifying the minimum height of any column on a specific geometric entity and ensuring that any column grown from a face on this entity has no high-order elements beyond this layer. If, however, the number of high-order layers is too low, in the worst-case scenario it could be zero, then the overall algorithm is likely to fail to produce a valid grid.

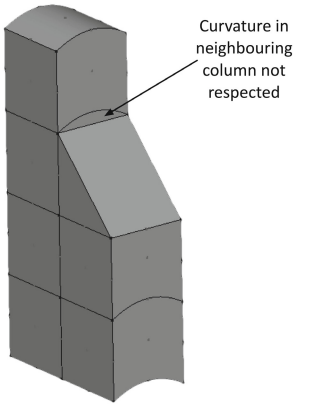


Fig. 7. Illustration of the issue encountered if the number of layers in the near-field mesh differs between adjacent columns

4 Examples

The high-order mesh generation capability detailed here was been developed on a variety of 3D geometries. For the purpose of illustration we concentrate here on three specific test cases, the ONERA M6 wing [11], the NASA trap wing geometry from the first American Institute of Aeronautics and Astronautics (AIAA) CFD high-lift prediction workshop [12] and a configuration used within the European Clean Sky programme [13], comprising a variant of the Airbus A340 aircraft with a modified outer wing to enable natural laminar flow conditions. These configurations are labelled, respectively, the IDIHOM test cases U2, A3 and A1, and are representative of the outputs that can be generated with the developed toolset. The grids generated for these test cases are illustrated in Figs. 8-10. For each of these, the input linear meshes were generated using SOLAR, and the high-order grids were visualised using Gmsh [2]. Corresponding details of grid sizes and timings for generation are given in Tables 1-3.

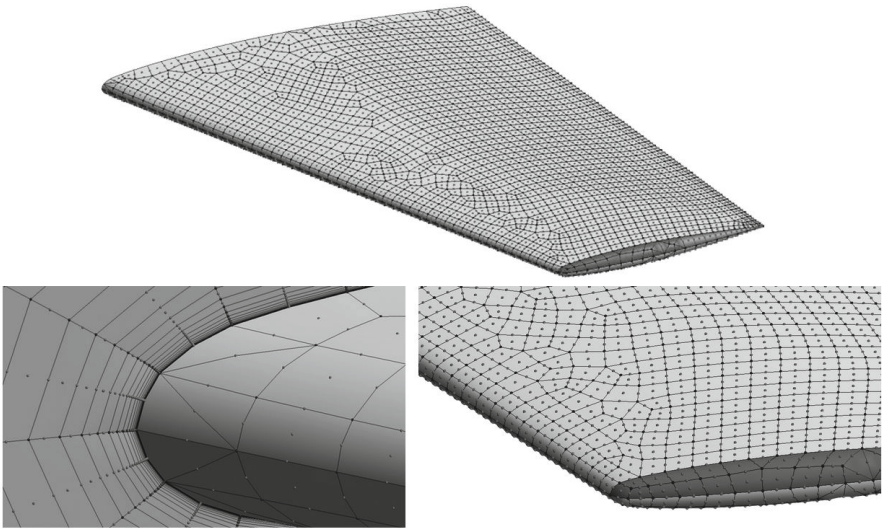


Fig. 8. Quadratic grid generated on the ONERA M6 wing

Table 1. Vertex counts for the example test cases

ONERA M6	NASA HLPW	Clean Sky
136704	3745601	16762238

Regarding the generated grids, those on the ONERA M6 and NASA HLPW geometries were found to be valid according to the curved mesh analysis tools provided by Gmsh [2], but the grid for the Clean Sky configuration contained 4185 invalid elements out of the total 4.12 million. These invalid elements, approximately 0.1% of the total number, are ultimately due to a failure of the input mesh to satisfy fully the criteria detailed in Sect. 3.4 and are considered further in Sect. 5.

In terms of computational constraints, the implemented algorithm does however perform as hoped, the results illustrated in Table 3 showing only a marginal increase in run-times for the high-order conversion compared to the initial generation of the linear grid. No performance optimisation has yet been attempted on the conversion algorithm, so it is expected that the current measure of performance could be significantly improved upon. In practice, however, when meshing industrial configurations, the main time penalty lies in the upstream preparation of the CAD and control inputs for the linear meshing stage. Since the current performance of the high-order conversion appears to be adequate for our needs, at present such performance optimisations are a relatively low priority.

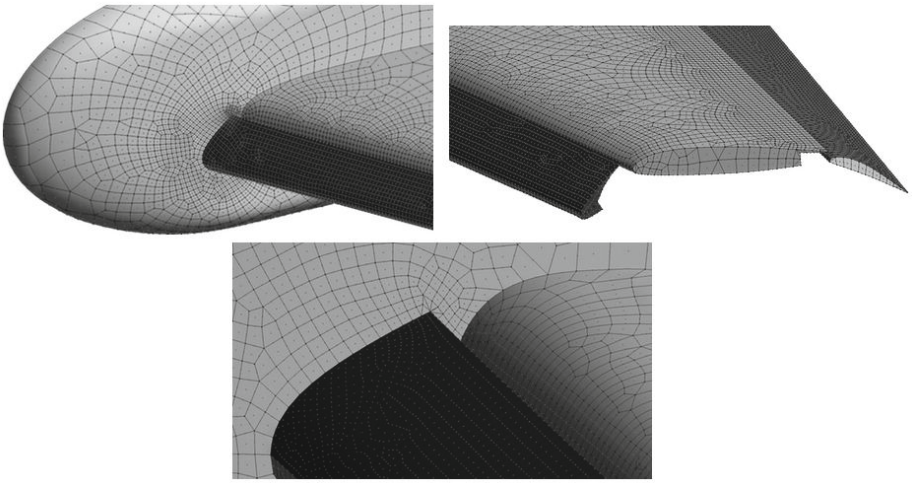


Fig. 9. Quadratic grid generated on the NASA high-lift workshop configuration

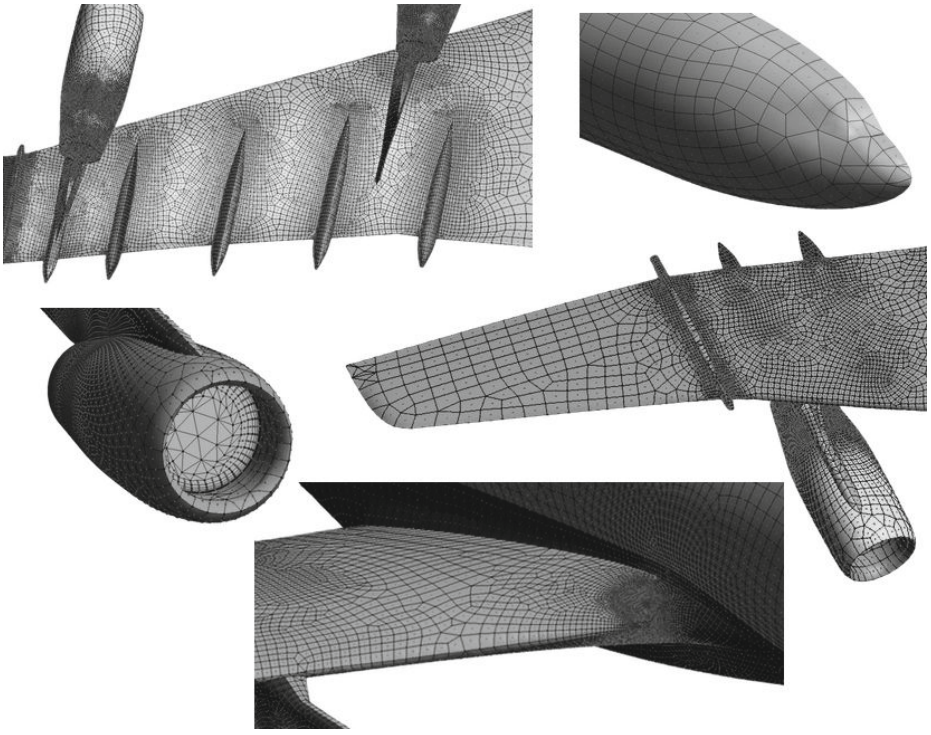


Fig. 10. Quadratic grid generated on the Clean Sky configuration

Table 2. Element counts for the example test cases

		ONERA M6	NASA HLPW	Clean Sky
Linear Elements	Quadrilaterals	-	282	-
	Triangles	3695	22005	16078
	Hexahedra	-	43039	-
	Prisms	-	880	-
	Pyramids	626	21591	84483
	Tetrahedra	96246	1930406	1881845
Quadratic elements	Quadrilaterals	1176	282	94679
	Triangles	79	22005	16078
	Hexahedra	13772	43039	1998102
	Prisms	1738	880	32506
		117331	2460600	4123771

Table 3. Timings (in seconds) for the high-order mesh generation process

Test Case	Linear mesh generation	Quadratic conversion
ONERA M6	2	2
NASA HLPW	43	57
Clean Sky Configuration	174	178

5 Issues Encountered

Considering the grids illustrated in Sect. 4, the most obvious issue lies in the failure to generate a valid high-order grid on the Clean Sky configuration, but as indicated in Table 2, all three meshes also have a total element count greater than would be ideal for use with high-order solvers. Both of these issues are ultimately due to difficulties in generating a sufficiently coarse linear grid as input. These difficulties arise as a result of design constraints imposed by the main use of SOLAR, namely, the generation of meshes for use with second-order finite volume flow-solvers. This imposes a requirement that the element sizes are graded in a smooth manner, between regions where fine elements are needed to capture accurately small-scale geometric features, and those where large elements may be used. In contrast, for the purposes of high-order grid generation a faster transition to coarse elements is desirable. As a result of this smooth grading the valid meshes contain more elements than desirable, whilst the invalid elements in the Clean Sky configuration grid are a result of difficulties in achieving an appropriate degree of coarseness. Indeed, the invalid elements are actually present in the linear mesh input to the high-order conversion process, meaning that this grid fails to satisfy the criteria detailed in Sect. 3.4. In all the

cases where we have successfully generated a linear input grid satisfying these constraints we have also been able to generate a valid quadratic grid.

Investigations into how the standard best practice for SOLAR mesh generation may be modified to achieve the desired grading were performed but trials were found to lack general robustness. This is unsurprising since we are, in fact, attempting to use the code outside its original design criteria. To achieve the desired coarsening, alternative algorithms will need to be added to SOLAR. Such modifications were not achievable within the timeframe of IDIHOM, but will form the immediate basis for future work on further industrialising the developed capability.

6 Conclusions

In the previous sections we have presented a new approach for obtaining high-order grids on complex aeronautical configurations. The presented algorithm appears to be fast and robust provided the inputs meet set requirements. The main remaining issue is the question of how to generate an appropriately coarse linear grid as input. Developing a capability to robustly generate such grids should be the priority of any future work.

Beyond this primary issue there do, however, remain additional constraints, namely a restriction of the current algorithm to meshes with an O-type topology and a need to ensure that the height of neighbouring columns of near-field elements are consistent with one another. These issues still need further investigation and would provide the secondary focus of activities once the issue of coarse linear mesh generation is resolved. Although our work in IDIHOM has provided definite progress towards the generation of an industrialised high-order meshing capability, it is clear that further work in this area is still needed.

References

1. Bassi, F., Rebay, S.: High-order accurate discontinuous finite element of the 2D Euler equations. Journal of Computational Physics 138(2), 251–285 (1997)
2. Geuzaine, C., Remacle, J.: Gmsh: a three-dimensional finite element mesh generator with build-in pre- and post-processing facilities. International Journal of Numerical Methods in Engineering 79, 1131–1309 (2009)
3. Leatham, M., Stokes, S., Shaw, J., Cooper, J., Appa, J., Blaylock, T.: Automatic Mesh Generation for Rapid Response Navier-Stokes Calculations. AIAA Paper 2000–2247 (June 2000)
4. Shaw, J., Stokes, S., Lucking, M.: The rapid and robust generation of efficient hybrid grids for RANS simulations over complete aircraft. International Journal of Numerical Methods in Fluids 43, 785–821 (2003)
5. Martineau, D., Stokes, S., Munday, S., Jackson, A., Gribben, B., Verhoeven, N.: Anisotropic Hybrid Mesh Generation for Industrial RANS Applications. AIAA Paper 2006–0534 (January 2006)
6. Munday, S., Martineau, D., Verhoeven, N.: Automated Hybrid Mesh Generation for Industrial High-Lift Applications. AIAA Paper 2007–268 (January 2007)

7. Kroll, N.: ADIGMA - A European Project on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications. AIAA Paper 2009–0176 (2009)
8. Johnston, C., Gould, J.: Investigation of issues relating to meshing for higher-order discretizations. In: Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.) ADIGMA. NNFM, vol. 113, pp. 181–193. Springer, Heidelberg (2010)
9. Pirzadeh, S.: Unstructured Viscous Grid Generation by Advancing Layers Method. AIAA Paper 93–3453–CP (1993)
10. Remacle, J.F., Toulorge, T., Lambrechts, J.: Robust untangling of curvilinear meshes. In: Jiao, X., Weill, J.C. (eds.) Proceedings of the 21st International Meshing Roundtable, pp. 71–83. Springer, Heidelberg (2013)
11. Schmitt, V., Charpin, F.: Pressure Distributions on the ONERA-M6-Wing at Transonic Mach numbers, Experimental Data Base for Computer Program Assessment. Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138 (May 1979)
12. Slotnick, J.P., Hannon, J.A., Chaffin, M.: Overview of the First AIAA CFD High Lift Prediction Workshop. In: 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. AIAA 2011–862 (January 2011)
13. Clean Sky Website, `http://www.cleansky.eu` (accessed: April 12, 2014)

High-Order 3D Anisotropic Hybrid Mesh Generation for High-Reynolds Number Flows

P. Szałtys, J. Majewski, S. Gepner, and J. Rokicki

Institute of Aeronautics and Applied Mechanics,
Warsaw University of Technology,
ul.Nowowiejska 24, 00-665 Warszawa
{pszaltys,jmajewsk,sgepner,jack}@meil.pw.edu.pl

Abstract. This paper considers a problem of generation of high-order anisotropic hybrid grids to be used for simulation of high-Reynolds number compressible turbulent flows around 3D geometries. The algorithm relies on generating a curvilinear structural grid in the boundary layer region, separately from the usual low-order unstructured grid in the rest of the computational domain. A grid deformator based on an elastic analogy is used in order to curved unstructured elements. The whole process is driven by the global spacing described in a form of a metric field. The presented method is verified for the Onera M6 wing and the L1T2 high lift testcases.

Keywords: CFD, high-order methods, anisotropic grid generation, high-Reynolds number flows.

1 Introduction

The High-Order Methods (HOM) became recently one of the most promising and dynamically developing branches of CFD. HOMs are considered an ideal candidates for highly resolved and accurate simulation methods. However they require a good quality High-Order (HO) curvilinear grids which conform to curvilinear boundaries. Such grids are quite difficult to generate as in particular within the boundary layer highly elongated curvilinear cells are very likely to appear. In case of traditional linear grids a few validity criteria could be distinguished, e.g., the positive volume, the limited ratio between radius of circumscribed sphere and the maximum edge length, the prescribed aspect ratio [1], [2]. For the HO grids this becomes much more complicated. The most commonly used criterion for the HO grids is positivity of Jacobian transformation [3], [4]. However, even such simple criterion is quite difficult even to check in the region of a boundary layer.

Typical strategies for generating a hybrid HO grid are based on modification of a linear grid. Linear elements are enriched with collocation nodes and subsequently curved using CAD description of the boundary. This method is rather simple in implementation but can result with negative Jacobians in case of thin elements. This problem could be avoided by applying a deformation only for wheel-shaped elements, in case of hybrid grids and only for unstructured elements far from the viscous walls. Nevertheless, the BL grid needs to be generated in the curvilinear form from the very beginning.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_5

In this paper the authors present, an algorithm which is based on this concept. The BL grid is generated as HO and the deformation is applied only for the external unstructured grid. The obtained generator is capable of generating high-order hybrid grids which are suitable for simulation of high-Reynolds number flows.

1.1 Overview of the Implementation of the High-Order Hybrid Grid Generator

The high-order hybrid grid generator consists of several separated modules:

- description of the geometry of the computational domain – this module (GeT) provides full information on all surface patches forming the wall (with the possibility to calculate 1st and 2nd derivatives of the parametrised surface) as well as the information about the connectivity of all the surface patches.
- description of the grid spacing (Control Space) – provides anisotropic spacing at an arbitrary point of the domain.
- anisotropic unstructured grid generator – a backbone of the hybrid grid generator which is able to form the complete unstructured grid in hierarchical manner starting from the lowest dimension entities (nodes) and ending at the 3D volume grid.
- boundary layer grid generator – used to form the thin grid necessary for proper discretization of the boundary layer zone
- grid deformator – used for curving an unstructured grid. The deformation algorithm is based on linear elastic analogy (with the assumption of small deformations).

The generation process starts by setting up the description of the domain by means of the GeT module. This allows to use the CAD model as an input (taking advantage of the STEP-file exchange format). Such a model consist of set of surface patches, which together form the closed boundary of the domain.

During the next step the surface grid for the boundary of the domain has to be generated. This is done using unstructured grid generator. The surface grids for the patches with the non-slip boundary condition, must be grouped into one open surface grid.

In the next step the surface grid is transformed into the high-order form. Firstly the collocation nodes are generated at each surface element. Subsequently these nodes are projected onto the curvilinear boundary. It is done according to the topological ownership, if the parent linear element is laying on the edge/face then the new collocation nodes are projected on the corresponding curve/surface. This high-order surface grid is used as a starting front in the boundary layer generator.

The generation of the boundary layer grid is based on the algebraic frontal algorithm. The BL grid generated in such a way consists of prisms which can be split into tetrahedra (if necessary). The generation of layers of the boundary grid cells is terminated either after predefined number is reached or another stopping criteria are met.

Once the BL grid is ready, the new boundary is extracted (of the domain not yet filled with the grid). Such boundary is formed using the last front of the

boundary layer as well as the surface grids of the remaining boundary patches. Once the boundary is extracted, the unstructured volume grid generator is used to fill the empty space with the linear elements (tetrahedra or triangles). In the next step the collocation nodes are added to the unstructured elements. At this point the unstructured grid does not fit the surface extracted from the last front of the BL grid. In order to alleviate this problem the unstructured elements need to be deformed.

After deformation is finalized, the BL grid and the unstructured grid are merged together.

During all phases of the generation process the generator is using information about the grid spacing provided by the Control Space. The Control Space can either be defined as an output from an error estimator or can be predefined by the user. Alternatively it can also be a mixture of both. Thanks to the fact that the same type of grid spacing description is used throughout the whole process of generation the continuity of the cell sizing is assured.

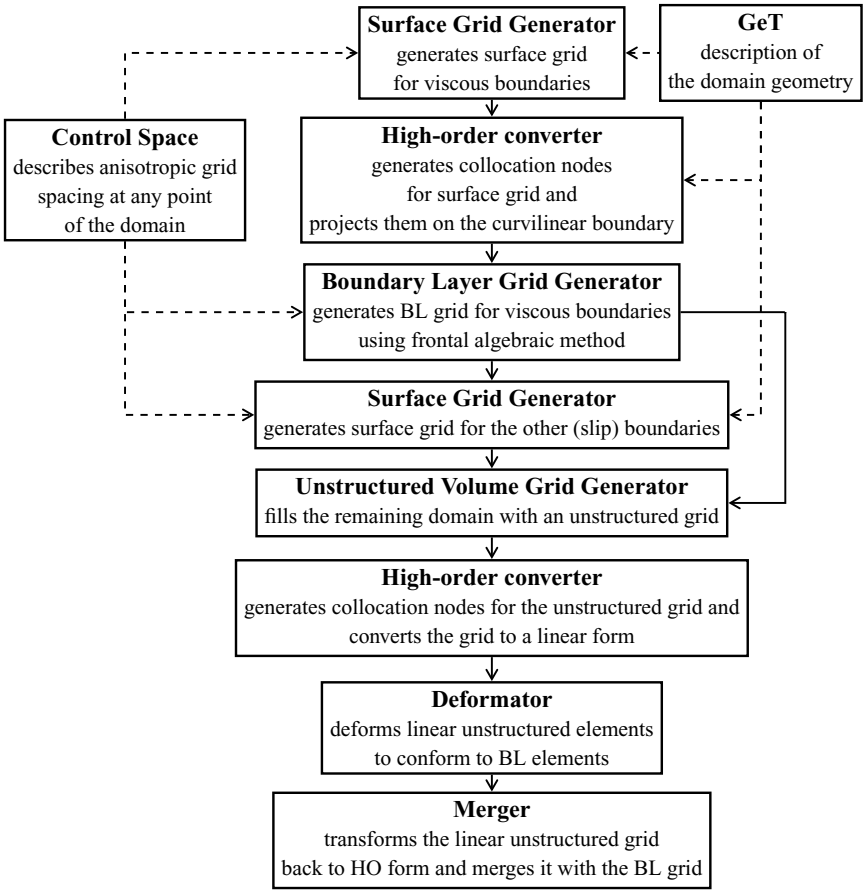


Fig. 1. Overview of the algorithm used in the high-order hybrid grid generator

2 Boundary Layer Grid Generation Algorithm

Simulations of viscous flows highly depends on the discretization within the boundary layer. Such grid is characterized by very thin and stretched cells. If such cells are formed by an unstructured grid generator there is no control on their shape. Especially the maximum dihedral angle for such cells can be very large (close to 180) which may cause excessive growth of error during simulations. Therefore, a much better choice is to use a specialized generator for boundary layer grids. Furthermore, such generator can create structured cells (e.g., prisms) which have much better properties from the point of view of the solver.

2.1 Overview of the Algorithm

The boundary layer grid generator is based on algebraic, frontal method [5], [6]. The grid is formed by extruding consecutive sublayers from the high-order surface mesh. High-order elements are created by assembling the sufficient number of sublayers (this number is determined by the predefined order of the mesh).

The height of the resulting elements is calculated according to the control space definition. The Laplace smoothing (with weights) is used in order to improve the quality of the final grid. The normal vectors in the proximity of sharp edges can be frozen in order to avoid problems related to the possibility of generation of invalid (negative volume) elements.

The boundary layer grid generation algorithm is based on the algebraic, frontal approach and consists of the following elements:

- The process of generation starts by taking the initial linear surface grid as the initial front (see Fig. 3a).
- The linear surface grid is transformed to the high-order (see Fig. 3b).
- For every base node and for every collocation node in the front, the normal vector as well as the cell height are calculated (see Fig. 2a).
- A single sublayer of the structured grid cells is generated and the front is moved to the external boundary of the sublayer (see Fig. 2b, c).
- When sufficient number of sublayers is obtained, the high-order structural elements are assembled from the sublayer slices (see Fig. 2d)
- The procedure is repeated until a stopping criterion is met.
- Once the generation is finished the remaining space is filled using the unstructured grid generator.

2.2 Cell Height Calculation

The height of the elements can be determined in three different ways. The first method is based on the user defined parameters such as the initial height and the predefined growth factor (Fig. 4a). The initial height is applied to all elements in the first layer and then the height of elements in every consecutive layer is obtained by multiplying the height from the previous layer by the growth factor.

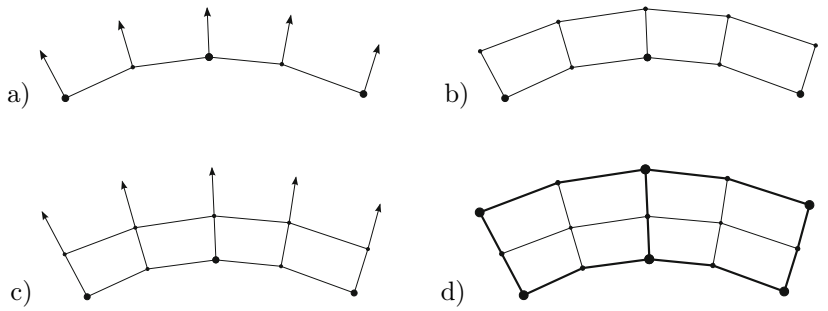


Fig. 2. Boundary layer grid generation using the frontal method.
a) Setting up normal vectors for the initial surface grid, which defines a starting position of the front
b) Creating a new sublayer by extruding grid cells along the normal vectors
c) Moving the front to the new position
d) Assembling high-order volume elements

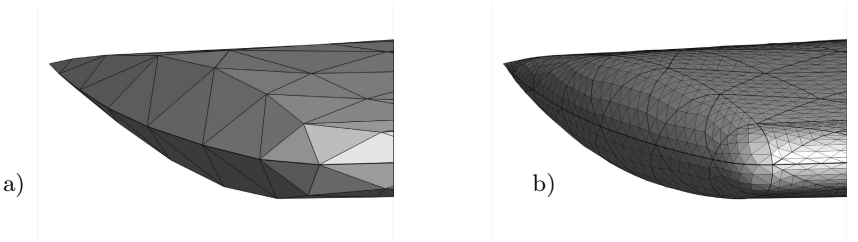


Fig. 3. Surface high-order transformation
a) Linear surface grid
b) Surface grid after the addition and projection step (of the collocation nodes on the surface)

The second approach uses the control space which can provide the generator with aniso-tropic spacing defined by a metric tensor $\mathcal{M}$ (Fig. 4b). For every node of the front the height is defined as the spacing in the direction of the normal to the front. Once the normal and the metric tensor $\mathcal{M}$ are known the height can be obtained from the formula:

$$h(\mathbf{x}) = \frac{1}{\sqrt{\mathbf{n}^T \cdot \mathcal{M}(\mathbf{x}) \cdot \mathbf{n}}}$$

where $\mathbf{n}$ denotes the normal vector while $\mathcal{M}(\mathbf{x})$ stands for the metric tensor at a point $\mathbf{x}$.

The last approach is a combination of the two. Firstly the height is calculated using the control space, then this height is limited using values resulting from the algebraic method (based on the user-defined growth factor). This approach allows to avoid large changes in the neighboring elements, which are likely to appear if solely the control space is used. The results of such limiting can be observed in Fig. 4b, c near the leading edge.

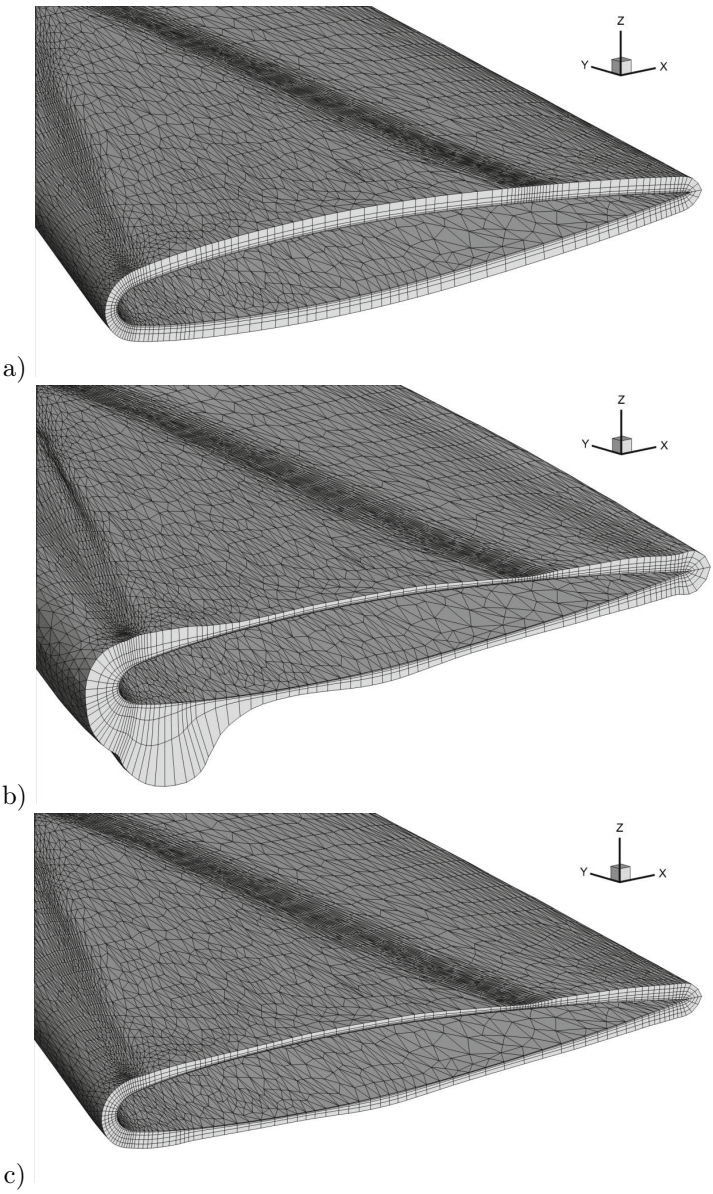


Fig. 4. Height of the boundary grid cells - different approaches:
a) Based on the user-defined growth factor
b) Based on the metric field
c) Based on the metric field with additional limiting

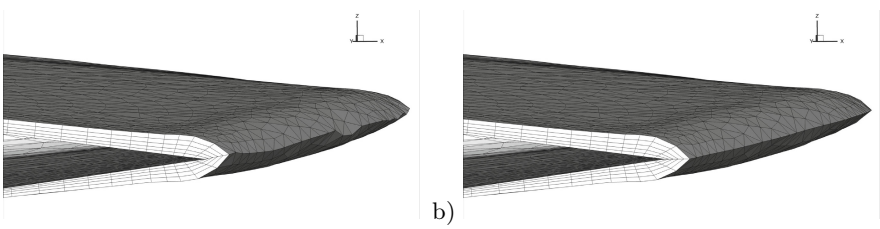
2.3 Normal Directions with Constrains

The generation of normal directions for the frontal nodes is one of the most important elements for boundary layer grid generator. The algorithm to calculate normal directions consists of several steps:

- The initial calculation of normal directions.
- Detection of possibly problematic regions, e.g., sharp edges/convex/concave and the calculation of corresponding symmetry surfaces (for edges) or vectors (for convex/concave).
- Projection onto symmetry surface/vector and freezing of normal vectors in the sharp edges/convex/concave regions.
- Calculation of visibility cones.
- Checking the visibility cone criterion.
- Smoothing the normal vectors according to the visibility criterion.
- Projection of the corresponding normal vectors onto symmetry/inlet/outlet boundary surfaces.

Initially for every frontal node, the normal vectors are calculated by averaging normals from the corresponding manifold panels. Unfortunately the normal directions obtained by this algorithm may result in negative volume elements. This problem can become even more serious after smoothing.

The irregularities in the distribution of the nodes in the proximity of sharp edges (e.g., trailing edges) could result in characteristic wavy shape of the extruded edge (see Fig. 5a). The problem can be alleviated by calculating normal vectors for the sharp-edge-nodes using geometry description of the joint surfaces and then by freezing the normals for the whole generation process (see Fig. 5b). This criterion can be weakened by calculating an average surface (from the geometry description of the joint surfaces) and allowing the normal vector to vary only at this surface. This approach does not introduce excessive stiffness to the system.



a) b)

Fig. 5. Freezing of normal vectors at trailing edge
a) Trailing edge region without freezing of normal directions
b) Trailing edge region with freezing of normal directions

Unfortunately this method does not guarantee a validity of the whole boundary layer grid. In order to fulfill the validity criterion an additional restriction for normal directions has to be introduced.

One of the most frequently used methods for this purpose is the so called *visibility cone* criterion (see Fig. 6). An algorithm is developed by the assumption that every frontal node should be *visible* from the surrounding panels:

$$\bigvee_{F_j \in M_i} \mathbf{N}_i \cdot \mathbf{N}_{F_j} > \mathbf{0}$$

where M_i stands for the manifold panels corresponding to the node $\mathbf{n}_i$ while F_j denotes jth face of the manifold i.

The algorithm starts from finding a pair of facets (F_1, F_2) which create the most acute angle:

$$\bigvee_{\substack{F_j, F_k \in M_i \\ j \neq k}} \mathbf{N}_{F_1} \cdot \mathbf{N}_{F_2} \leq \mathbf{N}_{F_j} \cdot \mathbf{N}_{F_k}$$

In the next step a bisection plane of the facets F_1, F_2 is formed. An axis of the unknown visibility cone lies on this plane. This feature implies that normal vector maximizes the minimum angle between itself and normal vectors of all faces from the manifold i. Moreover a half cone angle β_i is obtained which is used to form a validity criterion. Additionally with the normal vector $\mathbf{N}_i$ it defines a region where node $\mathbf{n}_i$ is *visible* for all surrounding nodes. After each modification of the normal vector, a check operation can performed, according to the formula:

$$\widetilde{\mathbf{N}}_i \cdot \mathbf{N}_i \leq \cos \beta_{\mathbf{i}}$$

where: $\widetilde{\mathbf{N}}_i$ - is normal vector after modification
More details can be found in the paper of Kallinderis and Ward [7].

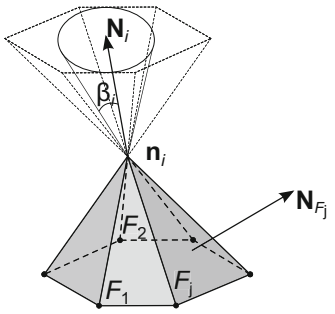


Fig. 6. The visibility cone formed for the for node $\mathbf{n}_i$

2.4 Smoothing

In order to improve the quality of the final grid, a simple Laplacian smoothing is applied to the normal vectors as well as to the heights of the consecutive layers [5]. The normal vectors for every node in the front are calculated locally by averaging

normals of neighbouring panels (frontal cells). This approach can result in strong "discontinuity" of the normal directions especially in the proximity of the trailing edges, be it either convex or concave as well as generally for each sharp edge of the boundary surface (Fig. 7a). It can also result in highly deformed elements and large differences in volume of the neighbouring elements. By applying simple Laplacian smoothing this problems can be eliminated in most of cases (Fig. 7b, c). Weights used in the smoothing process can be determined by, e.g., ratio of area/volume of neighbouring elements. The smoothing algorithm respects a visibility criterion and does not change normal directions for the frozen nodes.

The Laplacian smoothing of the normal vectors can be described as follows:

$$\mathbf{n}_i^{n+1} = (1-\omega)\mathbf{n}_i^n + \omega \frac{\sum_j w_j^n \mathbf{n}_j^n}{\sum_j w_j^n}$$

where:

$\mathbf{n}_i^{n+1}$ - the normal vector for the i^{th} node
$\mathbf{n}_i^n$ - the normal vector for the i^{th} node from previous iteration
w_j^n - the weight for the j^{th} neighbour
ω - underrelaxation factor

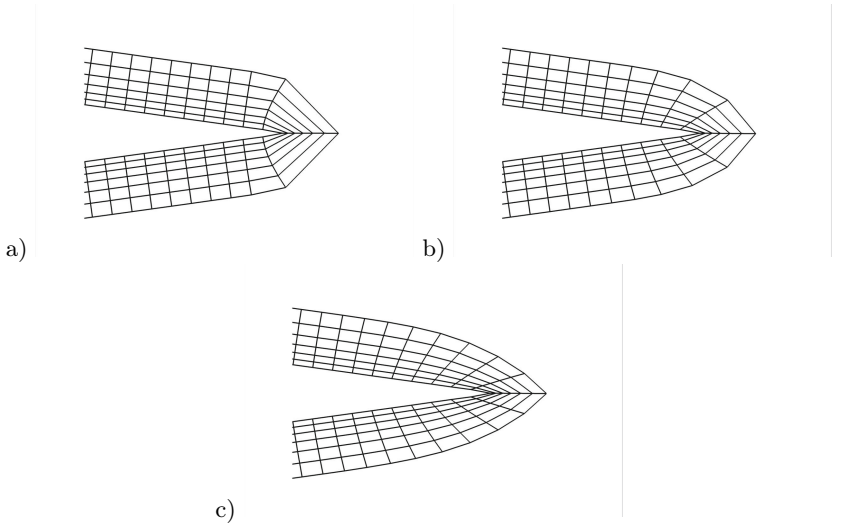


Fig. 7. Laplacian smoothing applied to normal vectors - trailing edge region.
a) No smoothing
b) Five iteration of smoothing
c) Twenty iteration of smoothing

At the final stage all nodes laying on symmetry/inlet/outlet boundary surfaces are projected on the normal direction (on the corresponding face). A complete example of high-order boundary layer grid can be observed in Fig. 8.

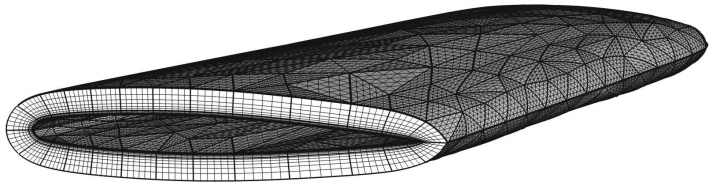


Fig. 8. P6 boundary layer grid for the Onera M6 wing

3 Anisotropic Unstructured Grid Generation Algorithm

The unstructured grid generator works in hierarchical manner. In order to generate 3D volume grid it is necessary to generate a grid on the boundary. Since the boundary is typically composed of many surface patches connected together, it means that the grid must be generated for every surface patch. This process starts with generation of the surface patch boundary composed of edge grids [9].

Summarizing, the generation process starts with the definition of the domain geometry and the corresponding topology. The generation of the 3D grid starts with the 1D grid bounded by the 0D vertices (logically they can also be treated as the 0D grids). Then the grid for each surface patch is created using the 1D grids to form the given patch boundary. The grids for the surface patches are used to form the 3D domain boundary and the volume generator can be used to form the final grid.

This method of generation can be called hierarchical because it follows, from the bottom to the top, following the hierarchy of the topological entities used for the definition of the domain. This can be best presented for the grid for the IRYDA testcase (see Fig. 9). Figure 9a shows the geometry of the boundary of I-22 airplane split into many connected surface patches. Figure 9b presents the first step where the 1D grids are generated for every 1D edge. These grids are used as a boundary for the 2D faces where the 2D surface grids are formed. All 2D grids forming the boundary for the 3D volume grid are shown in Figure 9c. Subsequently the volume grid generator is used in order to create the final 3D grid (see Fig. 9d).

The grids are generated for every nD topological entity, in the parametric space of the corresponding geometrical entity using algorithm based on anisotropic Delaunay triangulation with Boyer–Watson algorithm for incremental point insertion.

The general algorithm in nD (as used for 2D and 3D) can be summarized as follows:

1. Create a $(n-1)D$ grid which bounds the region for which a grid must be generated. Typically the bounding grid is composed using grids already generated for $(n-1)D$ topological entities given surface patch in 2D or 3D volume.

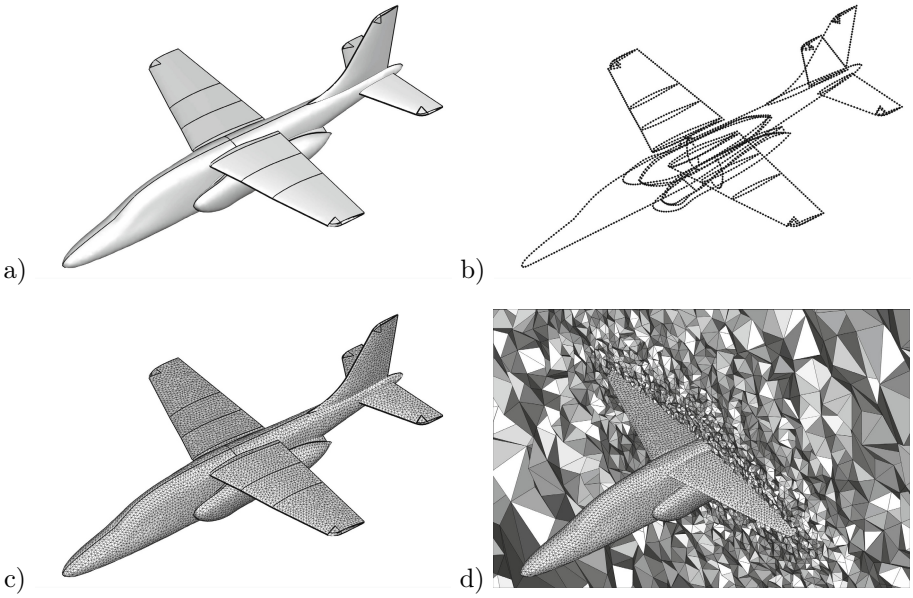


Fig. 9. I-22 unstructured grid - example of the hierarchical generation process
a) Boundary split into many connected surface patches
b) The 1D grids generated for each topological edge
c) The 2D grids for each topological face (surface patch)
d) A cut through the final 3D volume grid

2. Create a quad (cube in 3D) which contains all boundary grid nodes.
3. Split the quad in 2 triangles (cube into 5 or 6 tetrahedra). This will define initial Delauney triangulation.
4. Insert all boundary grid nodes into existing triangulation using the incremental algorithm of Boyer-Watson.
5. Recover boundary grid cells into the existing triangulation. This step is relatively easy in 2D however it poses quite a challenge in 3D especially for highly stretched anisotropic elements. During this step it may be necessary to introduce extra boundary points (Steiner points).
6. Flag all external cells. Thanks to the previous step there are no cells which are crossing the boundary.
7. Fill the domain with the internal nodes. New nodes are generated such that the grid spacing described by the *C*ontrol Space is satisfied. The new points are inserted into triangulation using anisotropic Boyer-Watson algorithm.
8. Remove all external cells.
9. Remove Steiner points added in step 5.
10. Perform the optimization by means of local topology operators, e.g., cell swaps, node movement, etc.

4 Unstructured Grid Deformation

The high-order elements introduced into the unstructured part of the grid are created based on initially linear elements, generated by the standard volume grid generator. Since elements are originally straight faced (low order) the resulting high-order grid does not conform well to the curved geometries. It is therefore necessary to relocate some of the nodes of the boundary elements.

A naive transition of the nodes to their desired positions is likely to produce ill conditioned elements and result in the unusable grid. Such an effect is illustrated in Fig.10.

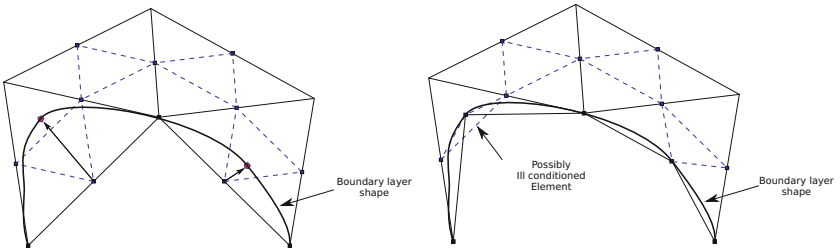


Fig. 10. Large nodal displacements result in generation of ill conditioned elements

To limit the generation of ill conditioned elements it is necessary to employ some mesh moving technique. Application of a hypothetical mesh moving technique is illustrated in Fig. 11. A driving mechanism to implement the movement of the internal nodes might be based on some grid optimization [3], smoothing, or a deformation by means of elastic analogy [9], [10]. The latter approach is used in the present work.

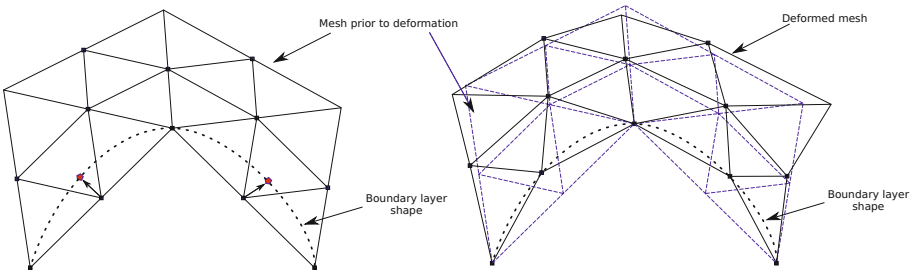


Fig. 11. Large nodal displacements are accommodated by appropriate relocation of the internal nodes

Due to the hybrid nature of the generation process the boundary layer is generated in a separate step. Therefore, the mesh movement needs to be applied only in the regions relatively far from boundaries. This greatly increases the robustness of the algorithm, especially in the proximity of the boundary, where elements are thin and highly stretched. Farther grid deformation is performed using the linear grid resulting from disassembly of higher-order elements into the linear ones.

4.1 High-Order Elements Splitting

Since the grid deformation module is capable to support only linear unstructured grids (triangle or tetrahedral) the high-order elements need to be converted to P1 form. In 2D a conversion is simple (see Fig. 12). Splitting is unique and the number of the linear elements is equal to N^2 where N denotes the order of the base element. In 3D splitting becomes more complicated. However, several characteristic *patterns* can be distinguished. The simplest splitting concerns P2 tetrahedron. It could be easily divided into four P1 tetrahedrons and one P1 octahedron (see. Fig. 13). Unfortunately there is no unique splitting of the octahedron without introducing the additional node in the centre of the mass. There are three different diagonal planes which can be used to construct pairs of pyramid elements (see. Fig. 14). Then for each pair of pyramids there are two different configurations of splitting into tetrahedrons defined by the diagonals of the pyramid base. The actual algorithm performs a quality check of resulting tetrahedrons and chooses a pattern which produces the best quality elements.

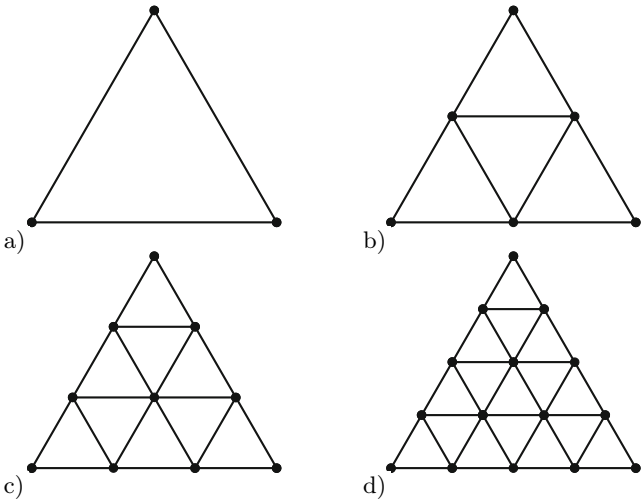


Fig. 12. The triangle element splitting schemes
a) Linear triangle
b) P2 triangle split into 4 P1 triangles
c) P3 triangle split into 9 P1 triangles
d) P4 triangle split into 16 P1 triangles

For the higher-order (more then P2) tetrahedrons another characteristic pattern can be observed. Fig. 15a shows a P3 tetrahedron which can be split into P2 tetrahedron and a P3 triangle frustum. Since the algorithm for splitting P2 tetrahedrons does not introduce any additional nodes, both figures can be treated separately. For clarity reasons a frustum case is shown in Fig. 16a as an equivalent 2D scheme. The presented scheme corresponds to the top view of the frustum. Solid lines represent the upper base while dashed lines represents the lower base. The P1 tetrahedrons can be formed by connecting a triangle from the lower base with a single node from the upper base (Fig. 16c, d) or in reverse conditions (Fig. 16e). The three hexagrams Fig. 16b (only upper hexagram is made bold for clarity reasons) correspond to three octahedrons. These octahedrons can be further split into the tetrahedrons according to the algorithm described above. These two different patterns (the tetrahedron type and the octahedron type) exhaust a collection of possible diagrams in a triangle frustum splitting.

Since the higher order tetrahedrons are formed with the consecutive triangle frustums of the decreasing order, the algorithm could be used recursively to split the tetrahedrons of any order.

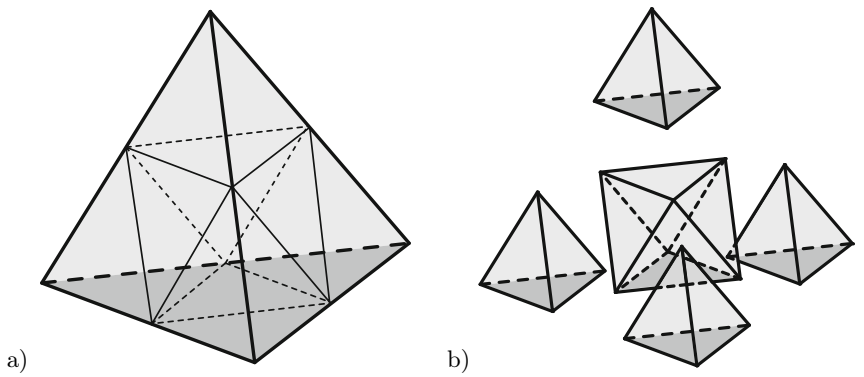


Fig. 13. Second order tetra splitting scheme
a) P2 tetra with division lines
b) P2 tetra after splitting

4.2 Elastic Analogy of Mesh Moving

The grid, undergoing movement of nodes, is treated as a solid body under deformation due to the external force $F(\mathbf{r})$. External force corresponds to the displacement of boundary nodes necessary for the grid to conform to the prescribed geometry. This results in displacement of each grid node by $\mathbf{u}_i$, in agreement with the force balance equation:

$$\nabla\sigma - F = 0 \tag{1}$$

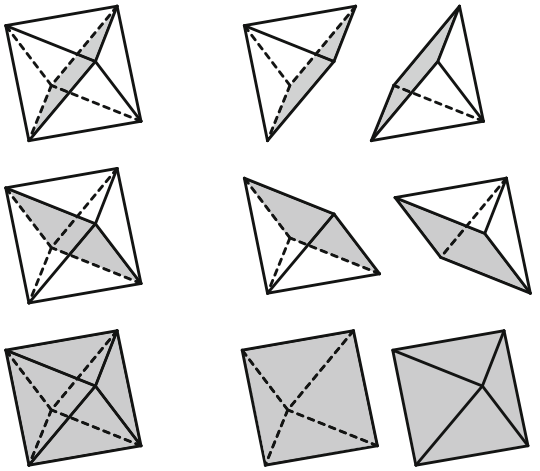


Fig. 14. Octahedron splitting scheme

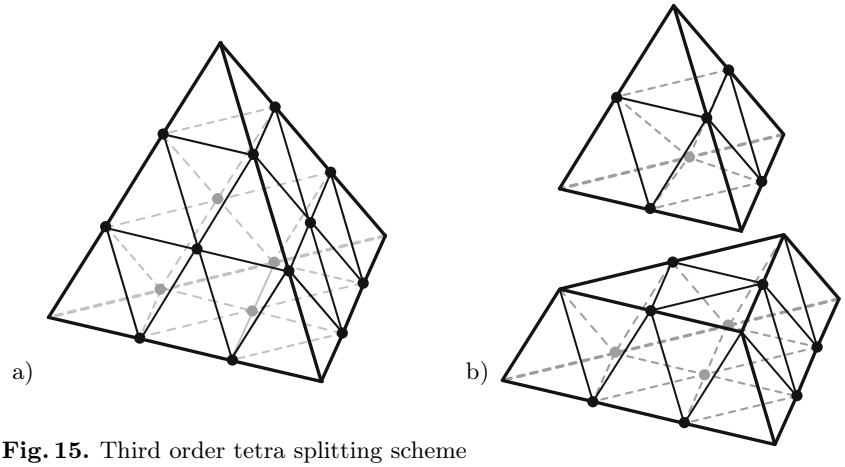


Fig. 15. Third order tetra splitting scheme
a) P3 tetra with collocation nodes and splitting lines at faces
b) P3 tetra split into P2 tetra and third order triangle frustum

The constitutive relations connect stress tensor σ, present in the deformed continuum, with the strain tensor ε. This work assumes an isotropic hetrogenic material represented by two Lame parameters $\lambda = \lambda(\mathbf{r})$ and $\mu = \mu(\mathbf{r})$ with σ depending on ε via:

$$\sigma = 2\mu\varepsilon + \lambda \operatorname{tr}(\varepsilon) I \tag{2}$$

where I is identity matrix, and $tr(\varepsilon)$ is a trace of ε.

In a general case the non-linear strain-deformation relationship can be used, as deformation of grid elements might be large. This would result in computationally expensive non-linear problem to be solved. Following [9] the current

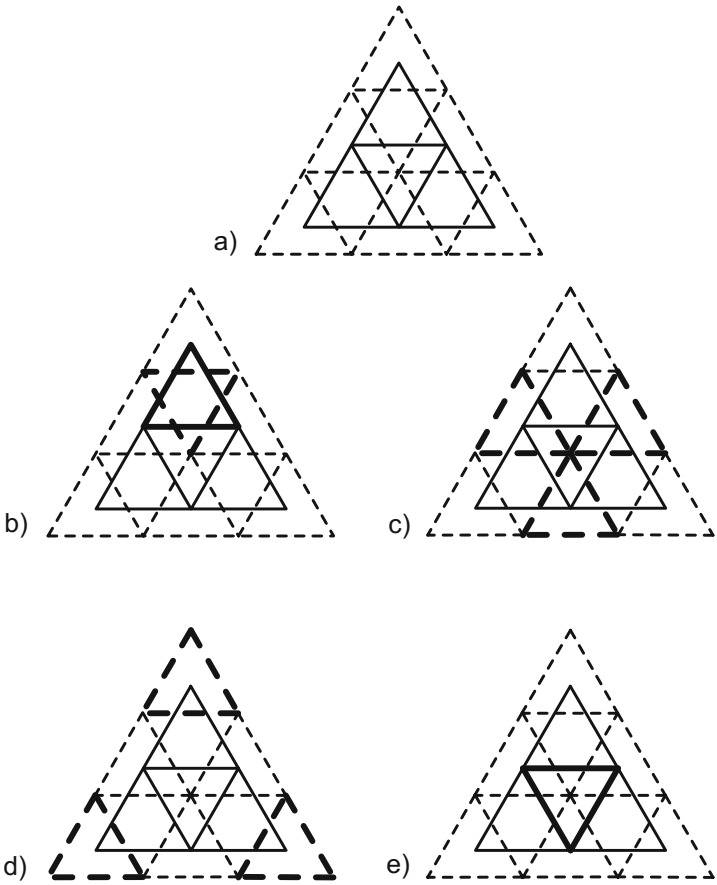


Fig. 16. Splitting patterns for p3 tetra slice
a) Top view, dashed lines - base P3 triangle, solid lines - P2 triangle
b) Octahedron type
c)-e) Tetrahedron type

approach disregards the non linear behaviour, in favour of a relatively inexpensive linearised formulation. The rationale here is the fact that the accuracy of an elastic computation is unimportant as long as boundary nodes conform to the assumed shape while the grid does not contain degenerated elements. Therefore the strain-deformation relation can be expressed in the form of linear relation:

$$\varepsilon = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \tag{3}$$

Finally to solve problem (1) the standard finite element approximation is used. The problem is discretized with the first order polynomials over elements of the grid being deformed. As a result the numerical problem reduces to the usual:

$$\mathbf{Ku} = \mathbf{f} \tag{4}$$

The solution of (4) yields nodal displacements, required for the grid to be appropriately deformed. The stiffness matrix $\mathbf{K}$ is symmetric and positive definite, $\mathbf{u}$ represents nodal displacement vector used for grid modification, while $\mathbf{f}$ is a vector of external forces.

Figure 17 shows fragments of higher-order grids generated with the use of the deformation algorithm for the Onera M6 wing geometry.

4.3 Robustness Issues

Since large nodal deformations are common, the robustness of the linearised elastic approach might be insufficient. This is manifested by generation of degenerated linear elements, and in consequence invalid HO grids.

To improve robustness of the deformation technique, some heuristic techniques are used. First of all, some of the nonlinear behaviour is reintroduced by performing the deformation as an iterative process. This results in smaller deformations calculated in a single iteration.

Another measure can be obtained by manipulation of fictional material parameters λ and μ, governing the constitutive relation. To improve stability of the method the values of these parameters are dependent on the element position, orientation, shape and size. This allows for hardening of close-to-degeneration elements, while relaxation of others. As a result the robustness of the method significantly increases.

5 High-Order Grid Merging

In the final step of grid generation the unstructured grid is connected with the boundary layer grid in order to produce the final hybrid grid. As an output the grid deformator returns a linear unstructured grid which needs to be converted back to high-order version prior to merging with the BL grid. The grid merger stores an information about the old (not deformed) high-order unstructured grid. Since the linearization and the deformation processes add no extra nodes and do not change numbering of the existing nodes, a conversion to high-order is very simple. It can be done through updating a position of every node in the not deformed high-order unstructured grid by using an information from the deformed linear unstructured grid. In the last step the algorithm deletes the doubled nodes at the common surfaces between the BL grid and the unstructured grid and renumbers the remaining nodes.

6 Numerical Results

6.1 High-Order Hybrid Grid for the L1T2 Profile

In this section the grid for the L1T2 high-lift testcase is presented [11]. The grids were generated using a control space obtained from the second order adaptation

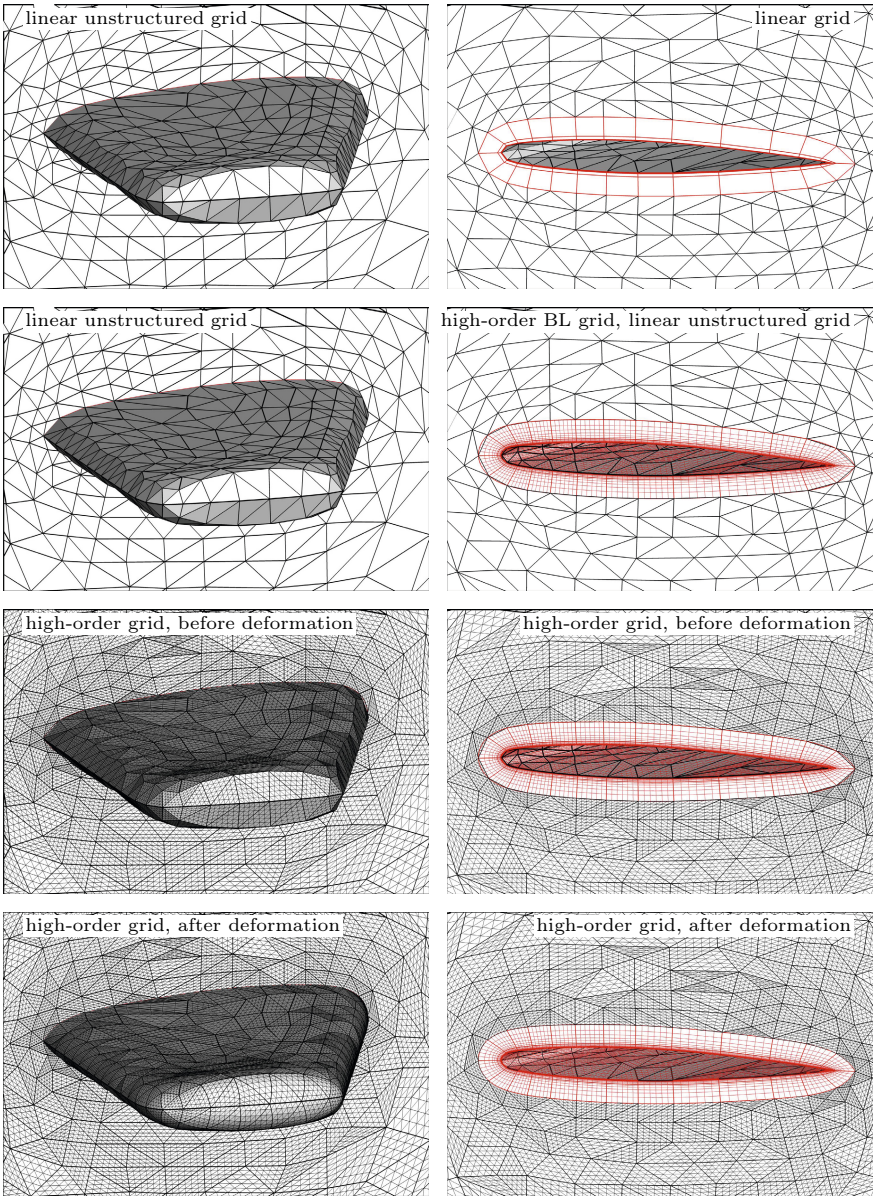


Fig. 17. Deformation steps applied to the Onera M6 wing

process (more details can be found in [4]). All grids are fully hybrid (quads and triangles). A general view of the example P4 grid is presented in Fig. 18. Details of the grids for various orders are shown in Fig. 19.

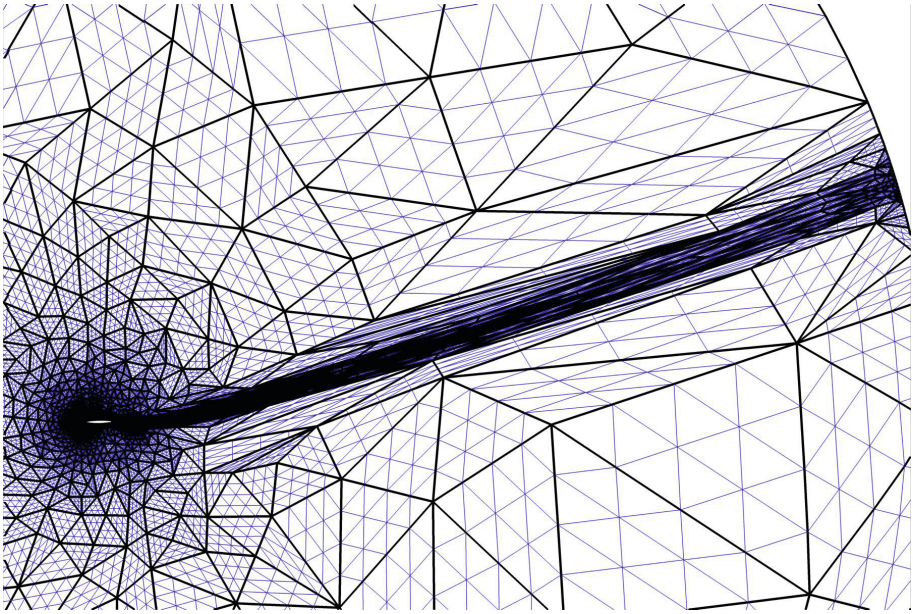


Fig. 18. The P4 grid generated from the metric field - details of the wake

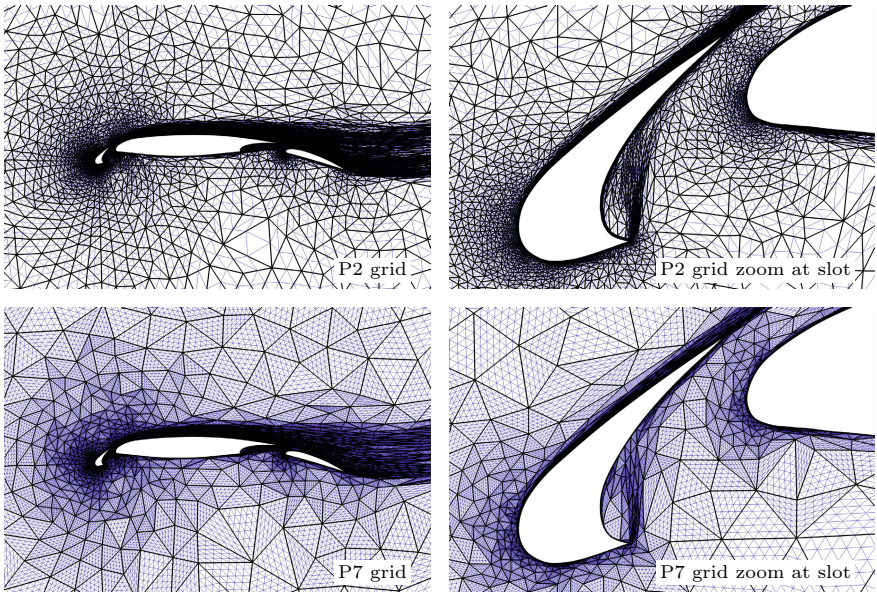


Fig. 19. High-order hybrid grids for the L1T2 high lift testcase

6.2 High-Order Hybrid Grid for Onera M6 Wing

For the Onera M6 wing [14] the grids were generated for the Euler solvers, with purely tetrahedral elements (see. Fig. 20) and for the Navier-Stokes solvers, with the mixed tetrahedral elements and prismatic elements in the BL region (see. Fig. 21). The Control Space used in the generation process of the unstructured grid was derived from the solution of simple Euler flow. The hight of the boundary layer elements was driven by the user defined progression.

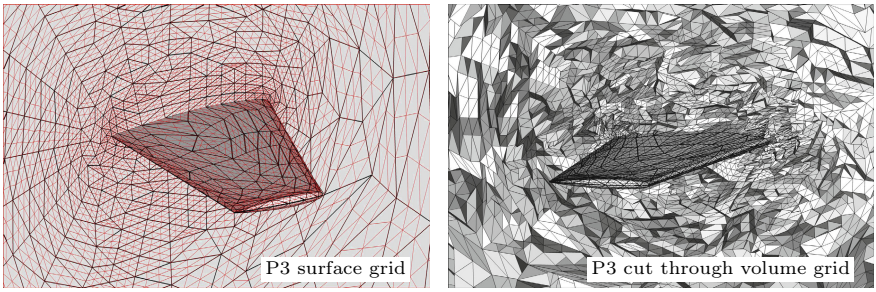


Fig. 20. High-order tetrahedral grids for the Onera M6 wing

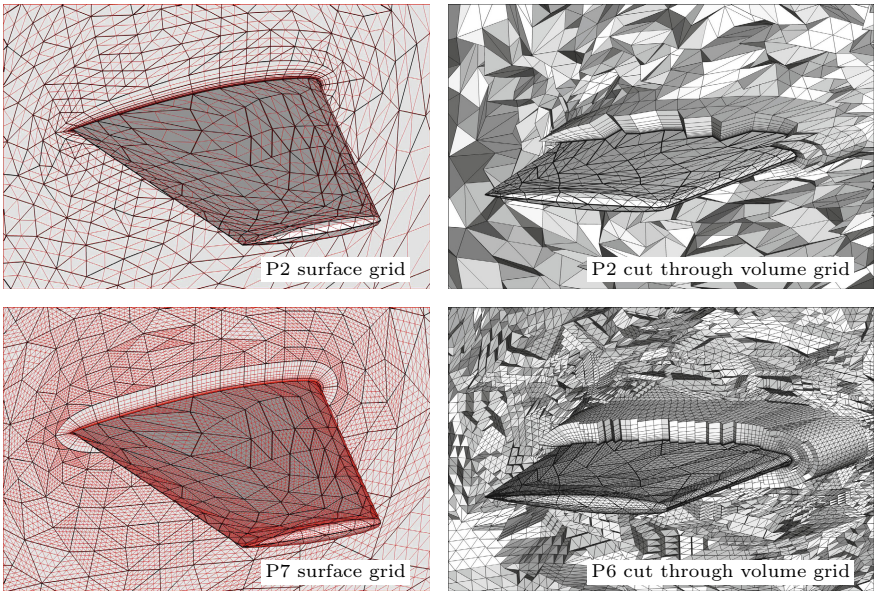


Fig. 21. High-order hybrid grids for the Onera M6 wing

7 Conclusions and the Future Work

In this paper the authors presented a pragmatic yet an effective solution for the high-order grid generation problem. The generator is capable to create high-order grids (no order limitation in theory) suitable for simulation of viscous compressible hight-Reynolds number flows. The generator was tested for 2D and 3D testcases (L1T2 high-lift configuration, Onera M6 wing). The obtained grids were checked for the validity of elements (positive volume and the Jacobian criterion).

The investigation has identified the most critical elements of the generation of the high-order grids especially for complex geometries. In particular such elements are the algorithm to calculate the normal directions, as well as the smoothing algorithm. Some elements of the algorithm require further work, in particular the local optimization of elements, in order to obtain their better quality.

Acknowledgments. The research presented here was done in the frame of the IDIHOM project (*Industrialisation of High-Order Methods - a Top-Down Approach*, FP7-AAT-2010-RTD-1, Grant agreement no.: 265780).

References

1. Weatherill, N.P., Hassan, O., Marcum, D.L., Marchant, M.J.: Grid generation by the delaunay triangulation. In: Lecture Series on Grid Generation (LS1994-02), vol. 2. von Karman Institute for Fluid Dynamics (1994)
2. Shewchuk, J.R.: What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures. Technical report. In: Proc. of the 11th International Meshing Roundtable (2002)
3. Johnen, A., Remacle, J.F., Geuzaine, C.: Geometrical validity of curvilinear finite elements. Journal of Computational Physics 233, 359–372 (2013)
4. Roca, X., Gargallo-Peiró, A., Sarrate, J.: Defining quality measures for high-order planar triangles and curved mesh generation. In: Quadros, W. (ed.) Proceedings of the 20th International Meshing Roundtable, pp. 365–383. Springer, Heidelberg (2012)
5. Athanasiadis, A.: Three-Dimensional Hybrid Grid Generation with Application to High Reynolds Number Viscous Flows. PhD thesis, Université Libre de Bruxelles, Chaussée de Waterloo, 72, 1640 Rhode-St-Genèse, Belgium (2004)
6. Athanasiadis, A.N., Deconinck, H.: A folding/unfolding algorithm for the construction of semi-structured layers in hybrid grid generation. Computer Methods in Applied Mechanics and Engineering 194(48-49), 5051–5067 (2005)
7. Kallinderis, Y., Ward, S.: Prismatic grid generation for three-dimensional complex geometries. Aiaa Journal 31, 1850–1856 (1993)
8. Majewski, J., Rokicki, J.: Anisotropic mesh adaptation in the presence of complex boundaries. In: Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.) ADIGMA. NNFM, vol. 113, pp. 441–453. Springer, Heidelberg (2010)
9. Dwight, R.: Robust mesh deformation using the linear elasticity equations. In: Deconinck, H., Dick, E. (eds.) Computational Fluid Dynamics 2006, pp. 401–406. Springer, Heidelberg (2009)

10. Gepner, S.: Grid Adaptation and Parallel Processing in Transsonic Flows Simulation. PhD thesis, Warsaw University of Technology (in print)
11. A Selection of Experimental Test Cases for the Validation of CFD Codes. Technical Report AR-303, AGARD (1994)
12. Majewski, J.: Anisotropic adaptation for flow simulations in complex geometries. In: 36th Lecture Series on Computational Fluid Dynamics / ADIGMA course on hp-adaptive and hp-multigrid methods. Von Karman Institute for Fluid Dynamics (2009)
13. NPARC Alliance Verification and Validation Archive – Onera M6 wing, `http://www.grc.nasa.gOV/WWW/wind/valid/m6wing/m6wing.html`

Anisotropic Adaptation for Simulation of High-Reynolds Number Flows Past Complex 3D Geometries

J. Majewski, P. Szałtys, and J. Rokicki

Institute of Aeronautics and Applied Mechanics,
Warsaw University of Technology,
ul.Nowowiejska 24, 00-665 Warszawa
{jmajewsk,pszaltys,jack}@meil.pw.edu.pl

Abstract. The paper presents the anisotropic adaptation algorithm applied to simulations of high-Reynolds turbulent compressible flows past complex 3D geometries. The adaptive algorithm relies on anisotropic grid-cell spacing definition provided by the error estimator in a form of a metric field. The error estimator is based on the Hessian of the solution with additional terms used to improve the grid spacing in the regions of high viscous shear forces. The adaptive algorithm is used for the two testcases the Onera M6 wing and the High Lift Prediction Workshop 1 trap wing.

Keywords: CFD, turbulence modeling, mesh adaptation, anisotropic error estimation.

1 Introduction

The quality of the numerical simulations strongly depends on discretization of the domain. The adaptive mesh refinement is a well-known strategy for minimization of the computational costs while improving resolution of the solution in critical regions. Additional gain can be obtained by using adaptation which is based on anisotropic definition of the grid spacing. This approach allows for generation of elongated cells with the shape conforming to the discretized phenomena. In particular cells in a region of a shock-wave are thin in direction normal to the shock-wave and relatively elongated in the tangent directions. Thanks to this, in comparison to the isotropic approach, smaller number of cells are needed to resolve the flow around the shock-wave.

The anisotropic adaptation approach has already been successfully applied for flow simulations. However, the simulation of the viscous, turbulent 3D problems still poses numerous challenges. The two main problems are: (i) grid generator/adapter should be robust enough in order to handle complex 3D domains and (ii) the error estimator should be capable to resolve properly the shear layers, e.g., wake behind the wing. This paper will address the latter issue proposing modifications, which should improve the quality of the grid. The results of the

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_6

simulations using the developed adaptation technique will be presented for the Onera M6 wing as well as for the HLPW-1 trap wing.

2 Adaptation Algorithm

The presented approach to adaptation relies on full grid regeneration. The grids are generated according to the anisotropic spacing provided by the *Control Space* (more details about the *Control Space* will be presented in Section 3). The *Control Space* is generated either by the user (e.g., for the initial grid) or by the error estimator (in the process of adaptation). The new grid is used by the solver to obtain the next generation flow solution. This sequence is executed a predefined number of times or until some convergence criterion is reached. The adaptation algorithm is shown in Figure 1.

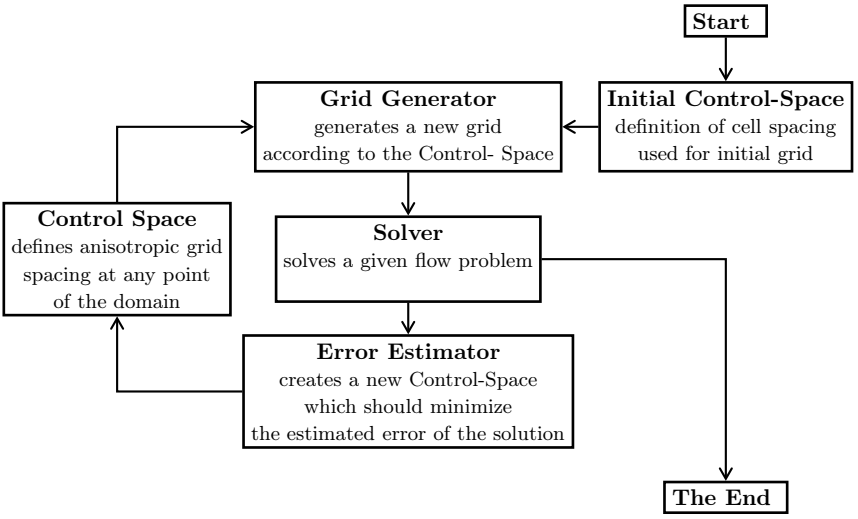


Fig. 1. Overview of the adaptation algorithm

This approach is quite challenging to the grid generator, which must be capable of producing anisotropic grid for complex 3D geometries refined according to the error estimator, but also must take into account the spacing needed for proper discretization of the boundary layer (the representative grids can be found in Fig. 6 and 7).

3 Anisotropic Spacing and the *Control-Space*

The main element of the anisotropic adaptation algorithm is the definition of grid spacing. If the grid is generated in a domain Ω, then at every point $P \in \Omega$

a preferred size of a cell must be known. This size is defined not by a scalar (as in the isotropic approach) but by the metric tensor $\mathcal{M}$ which is used to calculate distance between points of the domain.

The rank of the SPD tensor $\mathcal{M}$ must be equal to the dimension of the domain Ω. The metric $\mathcal{M}$ can be decomposed as follows:

$$\mathcal{M} = R \cdot \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \cdot R^{-1} = R \cdot \begin{bmatrix} \frac{1}{h_1^2} & 0 \\ 0 & \frac{1}{h_2^2} \end{bmatrix} \cdot R^{-1}$$

where R is a matrix which columns define unit directions of ellipse main axes and h_i is the length of the i-th main axes (see Fig. 2). The preferred cell is then such, that can be inscribed into the ellipse defined by $\mathcal{M}$.

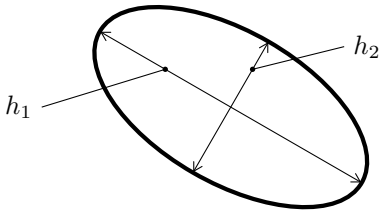


Fig. 2. Metric spacing definition presented as an ellipse

The size of the cells can depend on the user definition, error estimation or properties of the domain Ω. Thus it is necessary to provide the grid generator which uses this information in a consistent way. For this purpose the *Control Space* (*CS*) concept is used (see also [1]).

From the programming point of view, a continuous functions is needed which provides a value of metric tensor $\mathcal{M}$ for every point $P \in \Omega$.

4 Error Estimator

The detailed description of the standard approach to anisotropic error estimation based on interpolation error can be found in [2], [3], [1], [4], [5]. The anisotropic adaptation can also be extended to goal-oriented approach [6], [7]. The Hessian based approach to error estimation works well for inviscid problems, however it has limitation when applied directly to high Reynolds turbulent flow simulations. The problem was already recognized by the authors within the ADIGMA project, in which improvements were proposed and tested for 2D testcases [8], [9]. In the frame of the IDIHOM project the estimator was further extended and applied for fully turbulent complex 3D simulations.

4.1 Calculation of the Metric - Hessian Based Approach

The Hessian based approach is based on error estimator which tries to minimize the local interpolation error at the same time allowing to obtain information on the anisotropy of the solution [2], [1], [4], [5], [6], [7]

Assume now that E denotes a grid cell inside which a function u is being interpolated and $\mathbf{x}_c$ is the center of E. Then after dropping terms of higher order, the interpolation error for E can be estimated [1] as:

$$\varepsilon_E \leq \max_{\mathbf{x}\in E} \left(\mathbf{x}-\mathbf{x}_c\right)^T |\mathcal{H}| (\mathbf{x}-\mathbf{x}_c) \tag{1}$$

where $\mathcal{H}$ is a Hessian of u:

$$\mathcal{H} = \nabla\nabla u = R \cdot \Lambda \cdot R^{-1} \tag{2}$$

The Hessian is symmetric, therefore it has real eigenvalues λ_1, λ_2 and λ_3. The matrix R consist of columns being right eigenvectors of $\mathcal{H}$ while Λ is a diagonal matrix consisting of corresponding eigenvalues λ_i. Consequently $|\mathcal{H}|$ can be defined as:

$$|\mathcal{H}| = R \cdot |\Lambda| \cdot R^{-1} \tag{3}$$

The interpolation error in a given direction defined by the unit vector $\mathbf{w}$, is proportional to a constant C calculated as follows:

$$C = h^2 \ \mathbf{w}^T |\mathcal{H}| \mathbf{w} \tag{4}$$

where h denotes a length of a cell in the direction of unit vector $\mathbf{w}$. In the present approach $\mathbf{w}$ becomes a direction of a given edge and h becomes an edge length. The construction of an optimal grid is based on the equidistribution of the interpolation error. This is equivalent to assuming that for every edge e_i, the constant C_i is equal to a global C. We can introduce now a scaled metric $\mathcal{M}$:

$$\mathcal{M} = C^{-1} \ |\mathcal{H}| \tag{5}$$

This approach works well for inviscid flow simulations, however when applied directly to viscous (especially high Reynolds number) flows, it tends to underestimate viscous shear layers. This is understandable as in the regions of such layers, the gradient of velocity is large while hessian is quite modest. The grid obtained by this method will have unresolved regions close to the viscous walls and in the middle of shear layers (e.g., wakes). This effect can be observed in the Figure 3a. Additional modifications are necessary to alleviate such problems.

4.2 Calculation of the Metric - Gradient Based Approach

The first modification was already proposed by the authors within the ADIGMA project [9] and this modification was relying on a metric tensor formed using gradient of the magnitude of the fluid velocity $u = |\mathbf{v}|$.

$$\mathbf{w} = \nabla u \tag{6}$$

$$\mathcal{M} = \mathbf{w} \otimes \mathbf{w} \tag{7}$$

$$\mathcal{M} = R \cdot \Lambda \cdot R^{-1} \tag{8}$$

$$\Lambda = diag(\mathbf{w} \cdot \mathbf{w}, 0, 0) \tag{9}$$

The gradient and the Hessian metrics were mixed using the metric intersection approach (described in Section 4.4). Only then the adaptation algorithm was capable to resolve both, the regions dominated by viscous stresses as well as the regions indicated by the Hessian based estimation.

4.3 Calculation of the Metric - Viscous Stress Tensor Approach

Additional improvement of the anisotropic error estimator is proposed, which is based directly on the velocity gradient decomposed into the rate of strain tensor and the vorticity tensor.

Rate of Strain. The rate of strain tensor is used as the indicator of the shear layers. The tensor is defined by:

$$\mathcal{T}_S = \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right) \tag{10}$$

The tensor can also be defined as in the definition of the viscous stress tensor for compressible flows:

$$\mathcal{T}_S = \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right) - I\,\frac{2}{3}\,\nabla \cdot \mathbf{v} \tag{11}$$

In order to obtain the estimation of the strength and the normal direction of the shear layer (in a form of vector $\mathbf{w}$) the tensor $\mathcal{T}$ should be multiplied by normalized vector of the velocity.

$$\mathbf{w} = \mathcal{T}_S \cdot \frac{\mathbf{v}}{|\mathbf{v}|} \tag{12}$$

The final metric can be assembled using (7). This metric is mixed with Hessian metric as it was done for gradient based method.

Vorticity. The last possibility for modification of the metric tensor is based on vorticity. This time the $\mathcal{T}_\Omega$ is used for detection of shear layer:

$$\mathcal{T}_\Omega = \left(\nabla \mathbf{v} - (\nabla \mathbf{v})^T\right) \tag{13}$$

Vector $\mathbf{w}$ is defined similarly to (12):

$$\mathbf{w} = \mathcal{T}_\Omega \cdot \frac{\mathbf{v}}{|\mathbf{v}|} \tag{14}$$

The final metric is evaluated using (7).

4.4 Metric Intersection

During adaptive simulation the error estimator can be applied to different scalar fields (e.g., velocity magnitude, Mach number, pressure, etc.). As a result, different metric fields are obtained. Additionally, for viscous simulations the metric corresponding to the shear layer must also be taken into account. It is then necessary to merge or blend all the metric fields in order to obtain a single metric field which satisfies cell spacing requirements defined by all partial metric fields. This process is done at every node of the background grid where all metric tensors are defined. The new metric can be obtained by means of a metric intersection (see [10] for the notation):

$$\mathcal{M} = \bigcap_{i=1}^{n} \mathcal{M}_i \tag{15}$$

The algorithm is based on the method which allows to find intersection of two metrics and in order to obtain common metric for multiple partial metrics it must be executed in an iterative manner. Since the algorithm is not commutative, it is also important to apply it in an appropriate order, i.e., starting with Hessian based metric and then adding components related to gradient based metric.

The detailed description of the algorithm used for finding metric intersection can be found in [11], [12] and [10].

4.5 Metric Limiting

The process of limiting is necessary to avoid unrealistic shape/size of the grid cells. Assuming that values of h_{min} and h_{max} (which are provided by the user) define the maximum and minimum of the acceptable grid spacing the following modification are introduced:

$$\hat{\lambda}_i = (\,\max(\,h_{min},\ \min(\,h_{max},\ h_i\,)\,)\,)^{-2} \tag{16}$$

where $h_i = 1/\sqrt{\lambda_i}$. The final metric is taken as:

$$\mathcal{M} = R \cdot \hat{\Lambda} \cdot R^{-1}$$

The metric $\mathcal{M}$ is subsequently calculated for every node of the old grid and by means of interpolation is extended to form a continuous metric field. The new grid is generated in the Riemann space defined by the metric field in such a way that all edges have approximately unit length.

The final metric field can be further improved by using additional techniques like smoothing, gradation control [13], etc.

5 Comparison of the Results by Different Error Estimators

In order to compare all proposed modifications of the error estimator, the grids were generated using the same solution (NACA-0012, $M = 0.5$, $Re = 5000$,

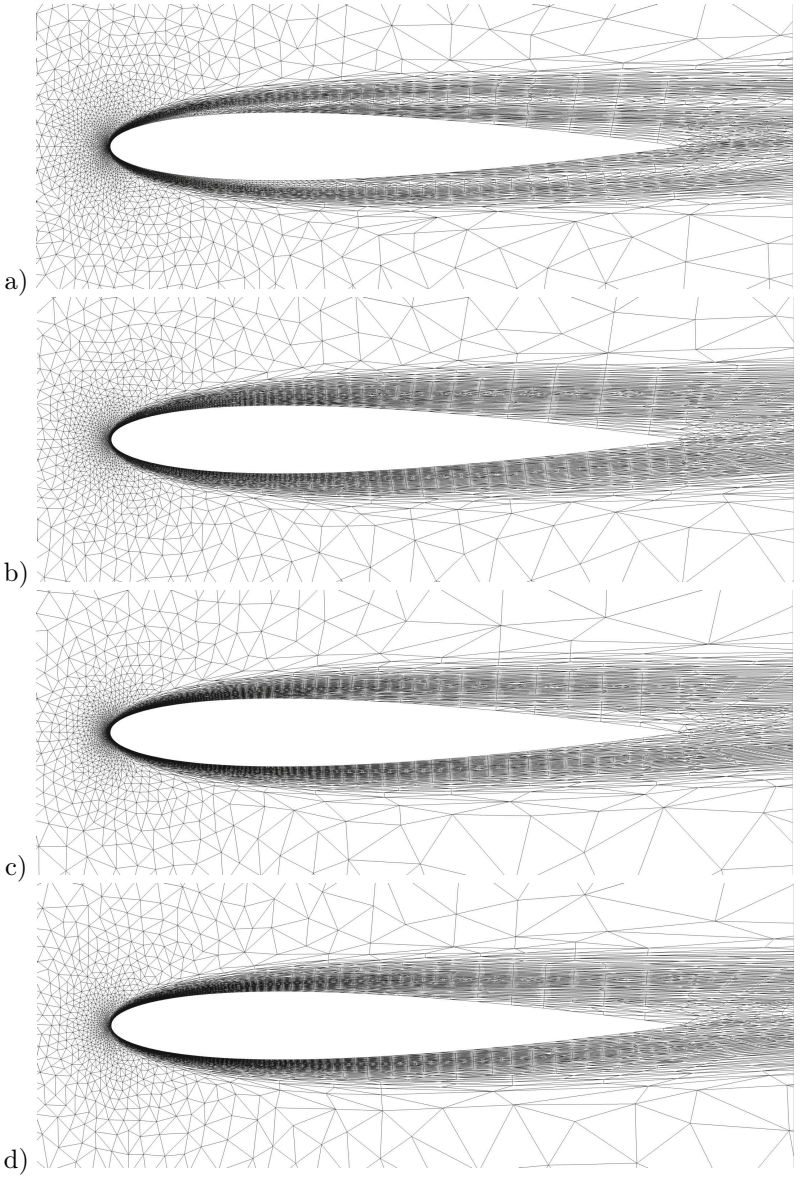


Fig. 3. Comparison of two approaches to the anisotropic error estimation applied for viscous flows (NACA-0012 in laminar flow: $M = 0.5$, $Re = 5000$, $\alpha = 2^\circ$).
a) Hessian of velocity module
b) Hessian of velocity module with velocity gradient component
c) Hessian of velocity module with strain tensor component
d) Hessian of velocity module with vorticity tensor component

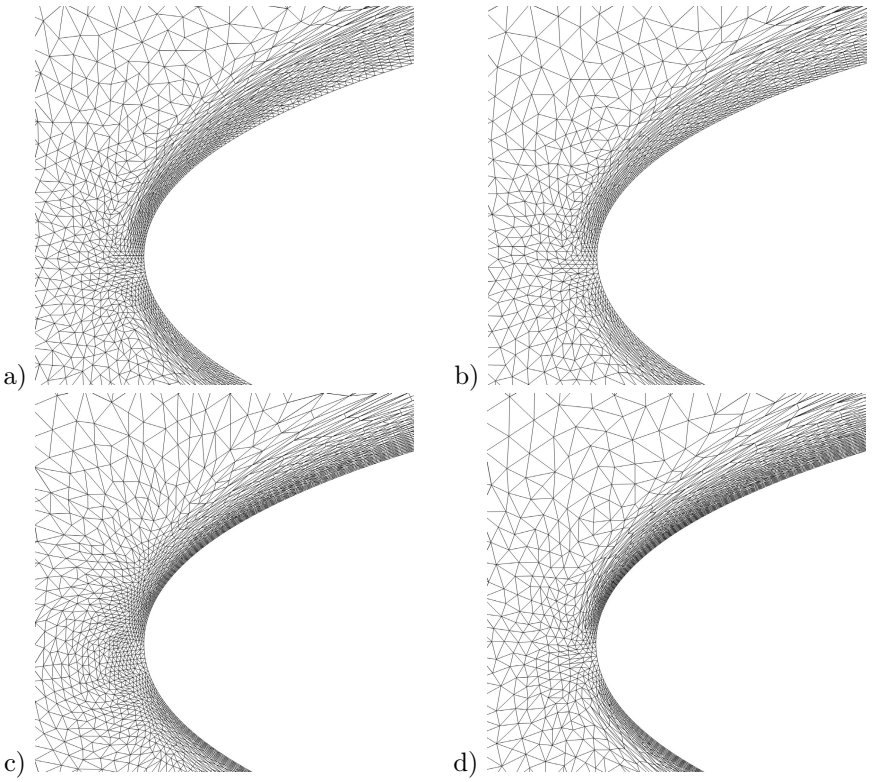


Fig. 4. Comparison of two approaches to the anisotropic error estimation applied for viscous flows. Details of the grid near the leadning edge. (NACA-0012 in laminar flow: $M = 0.5$, $Re = 5000$, $\alpha = 2^\circ$)
a) Hessian of velocity module
b) Hessian of velocity module with velocity gradient component
c) Hessian of velocity module with strain tensor component
d) Hessian of velocity module with vorticity tensor component

$\alpha = 2^\circ$) as a base input. In this comparison the estimators were set in such a way that all new grids had approximately the same number of nodes.

Figure 3a presents grids around the airfoil for all the methods. It can be clearly seen that the grid generated for pure Hessian based estimator is not satisfactory for viscous simulations. This method underestimates the area dominated with high shear (e.g., near the viscous wall), which should be refined in order to properly simulate viscous flows. The remaining three methods (Fig. 3b,c and d) are generating much better results. The differences between them can be established by inspecting the enlargement of the area of the upper surface near the stagnation point (Fig. 4).

Figure 4a demonstrates that Hessian approach fails to properly detect demanded spacing near the viscous wall. The gradient approach (see Fig. 4b) is

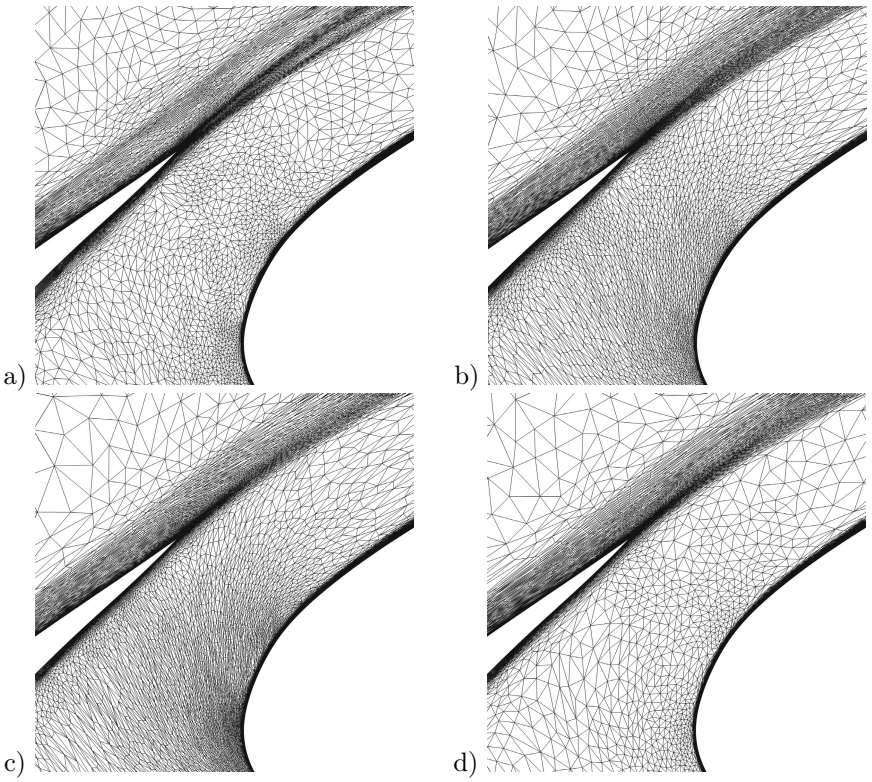


Fig. 5. Comparison of two approaches to the anisotropic error estimation applied for viscous flows. Details of the grid in the region between the slat and the main body (L1T2 in turbulent flow: $M = 0.197$, $Re = 3.52 \times 10^6$, $\alpha = 20.18^\circ$)
a) Hessian of velocity module
b) Hessian of velocity module with velocity gradient component
c) Hessian of velocity module with strain tensor component
d) Hessian of velocity module with vorticity tensor component

much better, nevertheless in the region of maximum friction force the grid is still too coarse. Both methods based on velocity tensor perform much better. The comparison between the methods based on the rate of strain and the vorticity shows that the first one (see Fig. 4c) refines also the region near the stagnation point while the second one (see Fig. 4d) detects only shear layers.

Another comparison was done for the turbulent flow around the multi-element airfoil L1T2. The enlargement of the area between slat and the main body for grids generated for different methods is shown in the Fig. 5. Comparing the grids for the last two methods one can find that the method based on the rate of strain tensor (Fig 5c) is over-refining the grid in this area. The approach based on vorticity tensor (Fig 5d) once again detects only strong shear layers and the final grid seems to be the best of the four.

6 Numerical Results

The anisotropic adaptation technique presented in this paper was used in simulations of two testcases. The first one is a transonic turbulent flow past Onera M6 and the second one is a subsonic turbulent flow past HLPW-1 trap wing. The numerical solution in both cases was computed using an in-house solver RED developed at the Warsaw University of Technology. The solver is based on Residual Distribution Scheme of second order with Spalart-Allmaras turbulence model. The grid generator used in the adaptation loop was also developed at the Warsaw University of Technology (description of the high-order version of this generator can be found in the present book in the chapter "High-order 3D anisotropic hybrid mesh generation for high-Reynolds number flows").

6.1 Onera M6 Wing in Turbulent Transonic Flow (U.2 Testcase)

The geometry of the U.2 testcase is relatively simple, nevertheless, the simulations are quite demanding due to the transonic regime of the flow. For such conditions two shock-waves appear on the upper surface of the wing. After merging into a single discontinuity, the shockwave is strong enough to induce a separation bubble in the boundary layer.

The challenges posed to the numerical simulations can be summarized in following points:

- the shock-waves must be properly resolved by the simulation - the grid must be sufficiently fine.
- the boundary layer must be properly resolved - if no wall functions are used the boundary layer grid must be adequately thin.
- separated flow around the wing tip must be properly resolved.
- the solver must be able to compute oscillation free solution in the vicinity of the shock-waves.

The first three points are difficult to meet by a typical simulation without adaptation, if the number of nodes in the grid is kept at a reasonable level.

Summary of the Testcase. The details of the geometry of the wing can be found in [14]. The flow conditions are: $M = 0.8395$, $Re = 11.72 \cdot 10^6$, $\alpha = 3.06°$. Four adaptation steps were executed. The grids used in the simulation were fully tetrahedral. The structural grid for the turbulent boundary layer was created using prismatic elements, which at the end of the generation process were split into tetrahedra. The thickness of the first layer of cells in the boundary layer grid and the growth ratio were defined by the user.

Grids used in the adaptation loop and the corresponding lift and drag force coefficients:

Adaptation step	N nodes	N cells	C_D	C_L
Initial	11104	60669	0.047331	0.27576
1st	143974	823667	0.022973	0.28342
2nd	280108	1618638	0.018840	0.28071
3rd	801498	4651212	0.018250	0.28023
4th	2690992	15820448	0.017486	0.28194

Comments on the Results. First, some details of the grid generated for the 4^{th} adaptation step are shown in the Figures 6 and 7. Figure 6 presents details of the grid at the leading edge where the aspect ratio of cells is high. This is also true for the surface cells which typically can pose a substantial problem for grid generator to deal with. As can be seen the hybrid generator of WUT is able to form a valid grid for such conditions. This picture can also demonstrate the advantage of anisotropic approach for 3D grids. If the cells were isotropic (aspect ratio of 1) then in order to discretize the flow features with the same resolution the grid would have to have much more grid nodes. Figure 7 shows the adapted grid at the tip of the wing. The trace of the shock-wave and the separation at the trailing edge can be easily found on the adapted grid.

Figure 8 shows for the selected slices the Mach number distrbution and the comparison of the c_p distribution of the adapted solution with the results of the simulation using WIND solver [14]. The experimental results are also shown as the reference. The plots show two adapted solutions: the first one is after 2^{nd} adaptation (the grid has 280108 nodes) and the second one is the final result (the grid has 2690992 nodes).

The reference results from the WIND solver were computed on a structured grid which had 316932 nodes. The WIND solver is based on node-centered finite volume method with decoupled Spalart-Allmaras turbulence equation [15]. The grid used by WIND solver has thickness of the first layer of the cells equivalent to $y^+ = 30$, thus the wall functions were used. The solver of WUT (used in the adaptive simulations) does not take advantage of wall functions so the first layer of cells in the grid must be sufficiently thin in order to resolve the viscous sublayer. This significantly increases the number of grid nodes that are needed for simulation.

Comparing the adapted results of WUT after the 2^{nd} step of adaptation with the WIND simulation (the grid of WUT is smaller than the one used by WIND) the advantage of adaptation becomes evident. Especially, this is the case for section $y/b = 0.80$ where the WIND solution cannot resolve both shockwaves while the adapted solution of WUT can. Similarly for remaining sections the WUT solution is of much better quality even though it is still of lower quality than the solution after the final adaptation step.

The results of the last adaptation show good agreement with experiment. Especially the last section $y/b = 0.99$ which typically causes most problems here, thanks to adaptation, was resolved reasonably well. Some discrepancies in

the sections near the symmetry plane are caused by the fact that in experiment a finite plate was used to separate wing from the wall. It was not present in the simulated geometry.

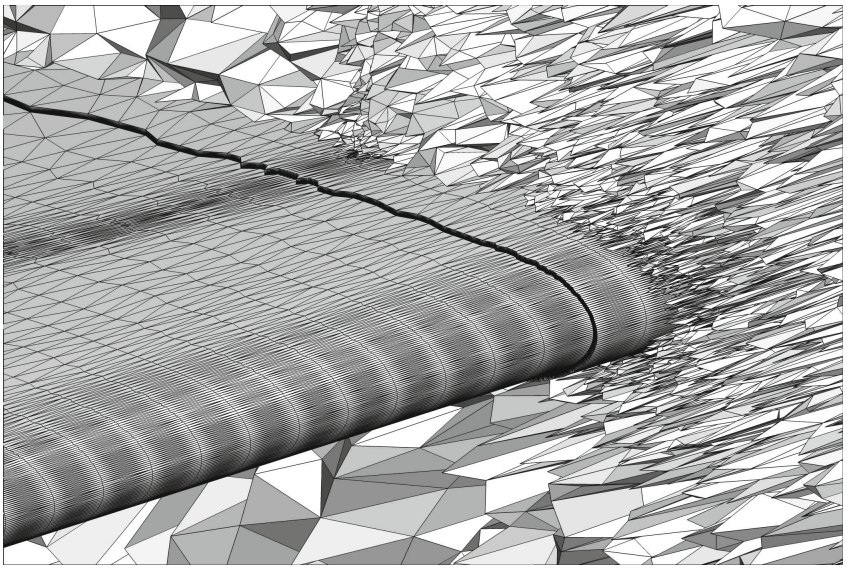


Fig. 6. The hybrid grid used for simulation in 4^{th} adaptation step - details near the leading edge

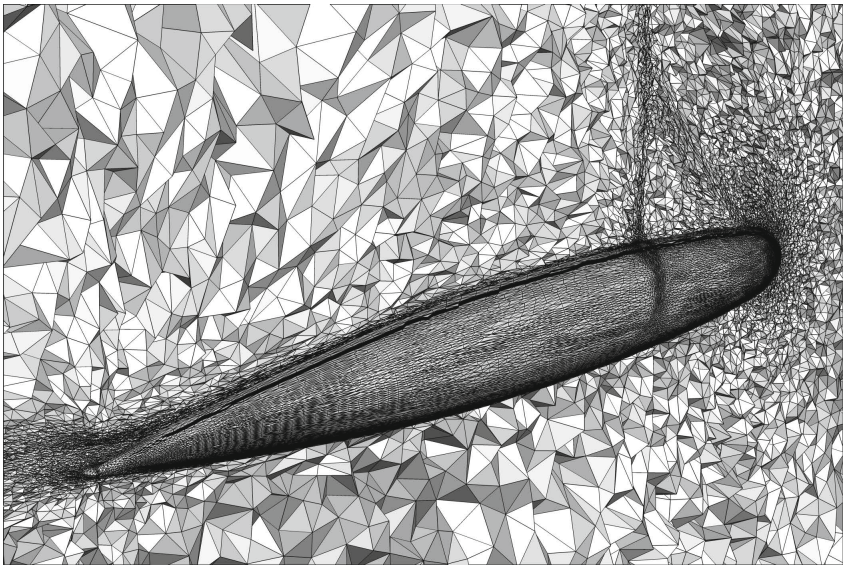


Fig. 7. The hybrid grid used for simulation in 4^{th} adaptation step - details near the wing tip

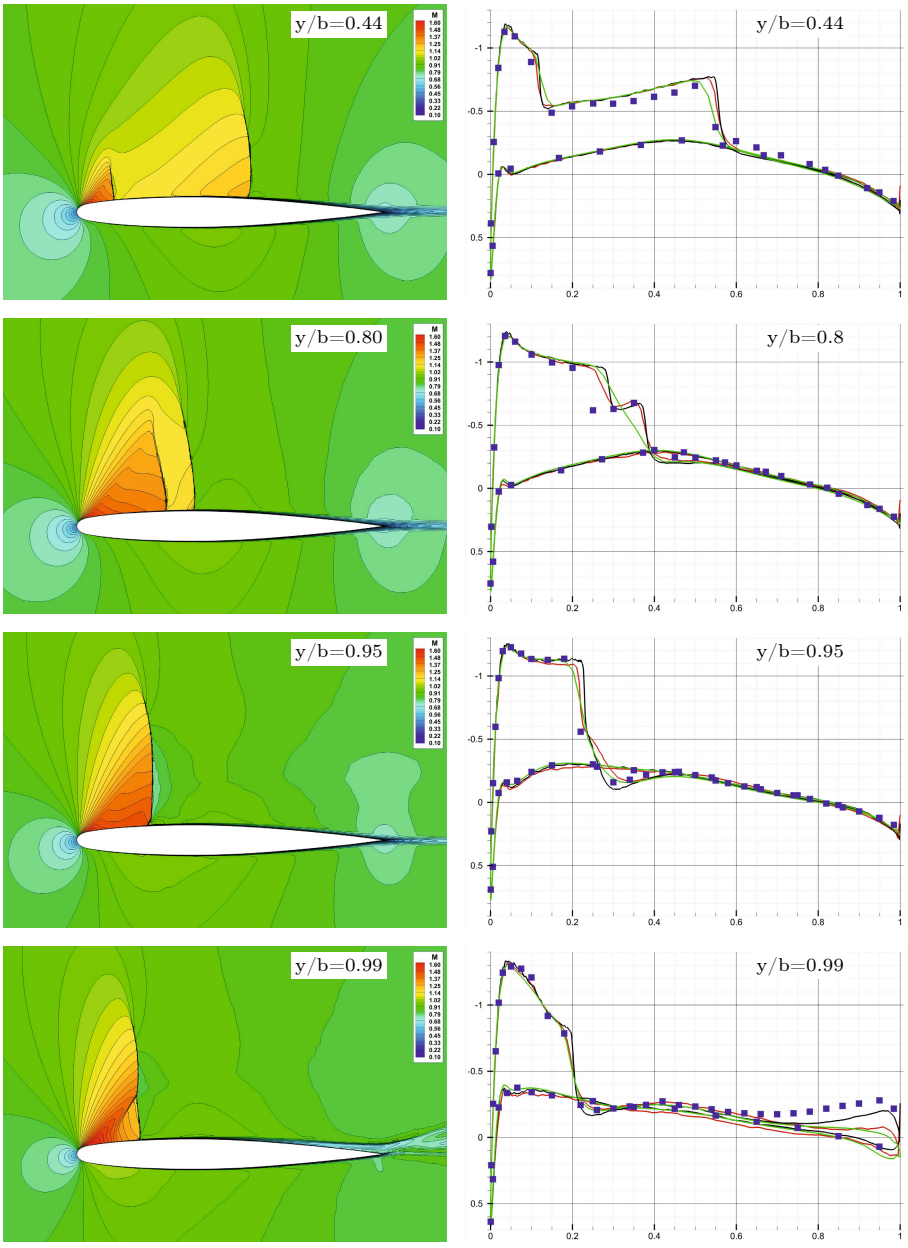


Fig. 8. Mach number field and corresponding c_p distribution Onera M6 wing — 2^{nd} adaptation, — 4^{th} adaptation, — 3^{rd} WIND, ■ - experiment

6.2 High Lift Prediction Workshop 1 Trap Wing (A.3 Testcase)

The HLPW-1 testcase geometry is a wing-body with extended slat and defected flap which, in comparison with Onera M6 wing is much more complicated. The flow conditions at the high angle of attack pose further challenge to the simulation process. In addition, a relatively small Mach number (0.2) can have adverse influence on the quality of the results obtained with a compressible solver. Some improvements for the force coefficients (especially for $\alpha = 28°$) can be achieved by using the far field correction.

Summary of the Testcase. The testcase is defined in HLPW-1 [16] as a Trap Wing Config 1 (Slat 30, Flap 25) with the following flow conditions: $M = 0.2$, $Re = 4.3 \cdot 10^6$. Simulations were carried out for two angles of attack: $\alpha = 13°$ roughly in the middle of the polar curve and $\alpha = 28°$, close to the maximum lift conditions.

Three adaptation steps were executed for both cases. The structural cells in the grid for boundary layer were split into tetrahedra to have fully tetrahedral mesh. Thickness of the first layer of the cells in the boundary layer grid and the growth ratio were defined by the user.
Grids generated and used in the adaptation loop:

Adaptation step	$\alpha = 13°$		$\alpha = 28°$	
	N nodes	**N cells**	**N nodes**	**N cells**
Initial	416963	2423746	416963	2423746
1st	1605833	9339125	3396003	19989680
2nd	3412794	20003949	5116431	30132718
3rd	6891914	40590444	11415558	67406180

The lift and drag force coefficients computed for consecutive adaptation steps:

Adaptation step	$\alpha = 13°$		$\alpha = 28°$	
	C_D	C_L	C_D	C_L
Initial	0.344649	1.99758	0.663004	2.85703
1st	0.326860	1.99497	0.661541	2.85379
2nd	0.323620	2.00434	0.661609	2.86373
3rd (prior to SOB)	0.324047	2.00905	0.665644	2.88021
3rd (after SOB)	0.318835	1.97697	-	-
Experiment	0.3330	2.0468	0.6776	2.8952

Comments on the Results. Presentation of the results of the simulation starts with Figures 9, 10 which show details of the 3^{rd} adapted grid for $\alpha = 13°$ in the region of the wake as well as around the the tip vortex over the flap. Figures 11 and 12 show grids for case $\alpha = 28°$ in the region of the wake as well as around

the the tip vortex. The overview of the structure of the flow for both angles of attack can be seen in the Figures 13 and 14. Interesting details of the flow near the tip can be found in Figures 15 and 16. As can be seen the tip vortices and their interaction has been well captured and resolved for both angles of attack.

Figures 17 and 18 show the pattern of surface streamtraces and surface c_f distribution. For both angles the pattern including the separation on the flap is in good agreement with the experimental results. The results of simulation after three adaptation steps for $\alpha = 13^\circ$ show the separation on upper surface of the flap at the Side Of the Body (SOB) (Fig. 17). The SOB separation exists in the experimental results however during simulation an adverse influence on lift and drag coefficients was observed. In order to present this effect the values in the table have been tagged as "prior to SOB" and "after SOB". Comparing with experimental values, both the lift and the drag coefficients are much better predicted prior to development of SOB separation.

Figure 19 presents velocity field for the slice at $y/b = 0.28$ obtained on adapted grid after 3 steps for $\alpha = 28^\circ$. It shows that the wakes behind the slat, the main body and the flap have been correctly resolved.

Figure 20 shows the evolution of the c_p distribution for sections $y/b = 0.41$ and $y/b = 0.98$ for both angles of attack. Especially, $y/b = 0.98$ shows that progressing adaptation improves the quality of the results.

Finally, we have compared the adaptive results of WUT with the results published in the HLPW-1 [16]. In order to pick most relevant results only solvers with Spalart-Allmaras turbulence model has been chosen for comparison. Additionally, only solution on finest grid submitted to the workshop (and for the geometry without brackets holding the flap) were taken into account. The details of the HLPW–1 simulations used for comparison are summarized in the following Table:

Entry	**Participant**	**Code name**	DOF (x10^6)	**Notes**
005.01	Cessna	FUN3D	32	
005.02	Cessna	NSU3D	11	thin-layer-like
008.01	DLR	TAU	31	
012.02	JAXA	UPACS	124	
015	NASA	USM3D	63	
017.03	NASA	CFL3D	161	thin-layer

The number of Degrees of Freedom (DOF) of almost all HLPW-1 computations exceeds substantially the adaptive simulations of WUT. Only the computation 005.02 is close to the finest grid of WUT used for $\alpha = 28^\circ$.

The comparison of the c_p distribution between HLPW-1 results and the obtained by the present adaptive method has been shown for the section at $y/b = 0.98$, as this section is closest to the wing tip. In this region good agreement with experiment is most difficult to obtain. The plot of c_p is shown in Figure 21 and the results of current adapted simulation are tagged with the name of the solver RED. As can be seen, despite existing differences between the simulation and the experiment, the results are better than the obtained in

HLPW-1. It is also good to note that this result was obtained on much smaller grids (e.g., for $\alpha = 13^\circ$ the finest grid from HLPW-1 has more around 20 times more DOFs than the finest grid of the presented adapted results).

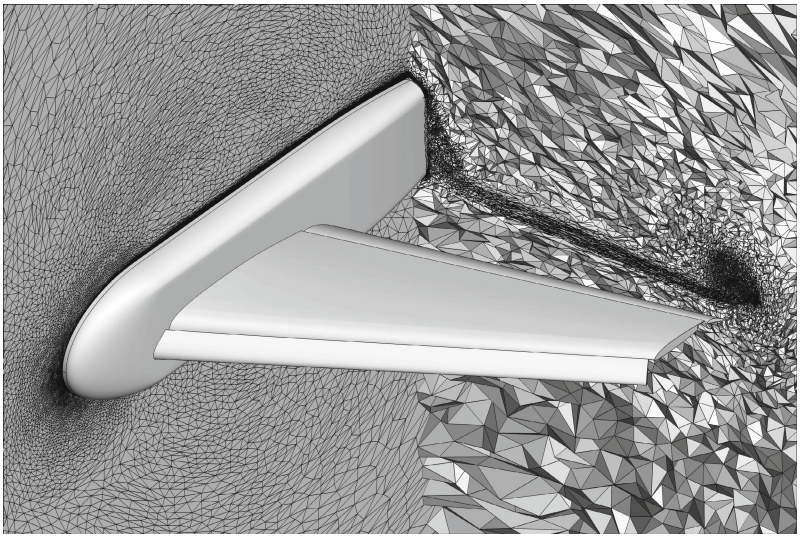


Fig. 9. The grid after 3^{rd} adaptation for $\alpha = 13^\circ$ - cut through the boundary layer and the volume grid

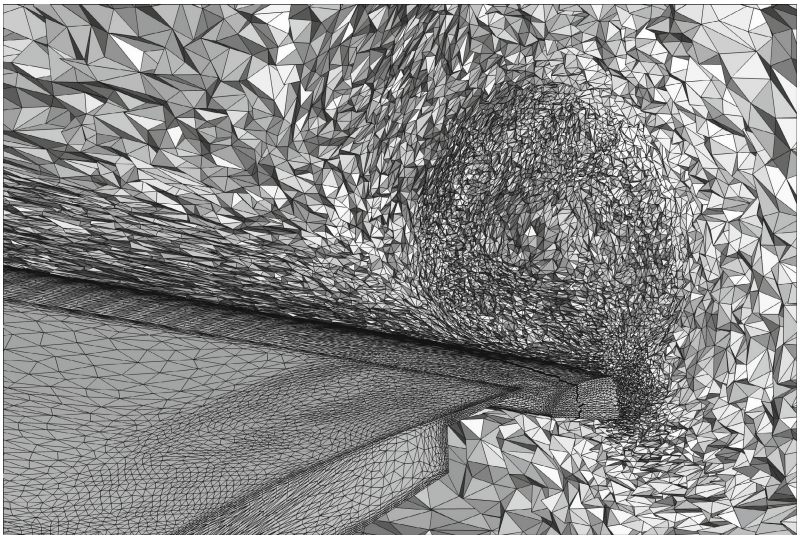


Fig. 10. The grid after 3^{rd} adaptation for $\alpha = 13^\circ$ - details of the volume grid in the area of the wing tip vortex

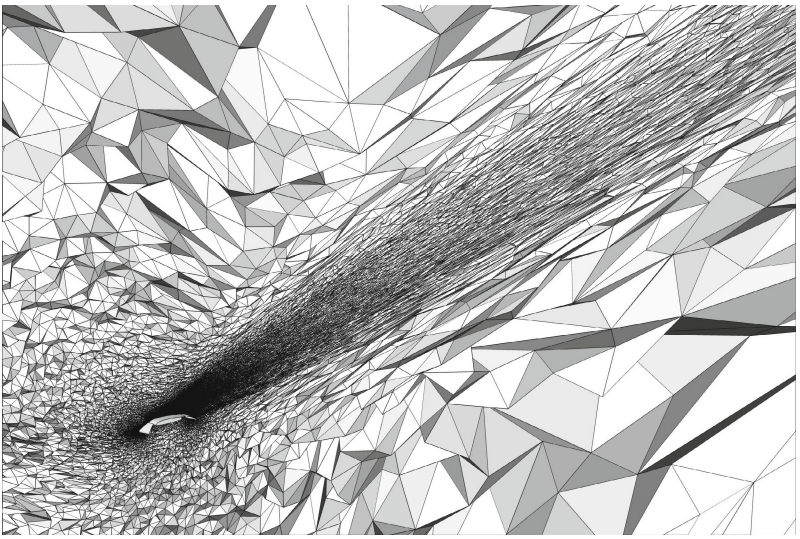


Fig. 11. The grid after 3^{rd} adaptation for $\alpha = 28^{\circ}$ - the wake

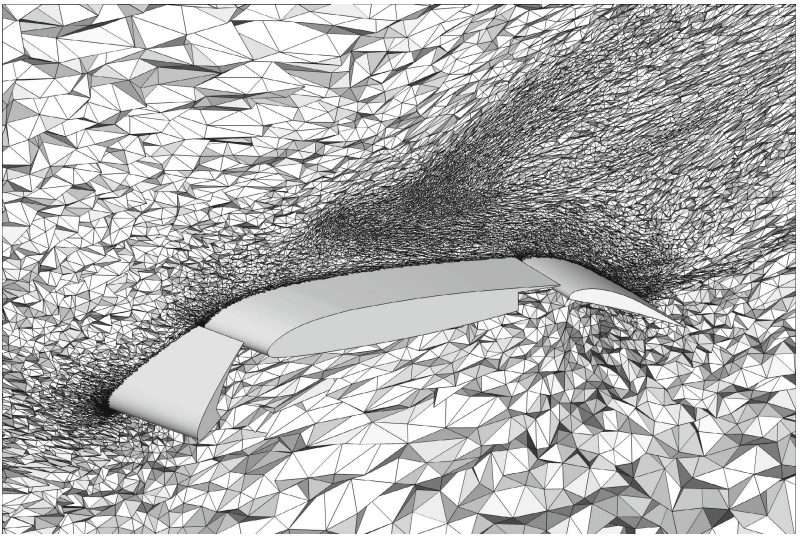


Fig. 12. The grid after 3^{rd} adaptation for $\alpha = 28^{\circ}$ - details of the volume grid in the area of the wing tip vortex

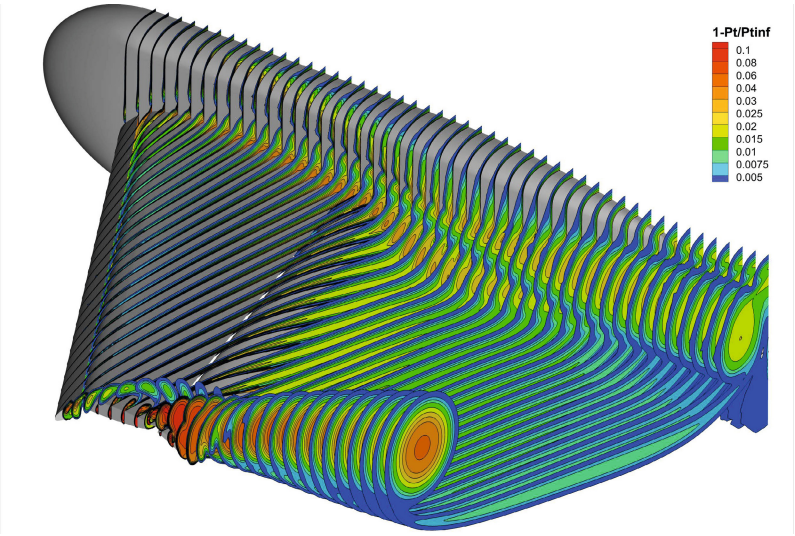


Fig. 13. Total pressure loss after 3^{rd} adaptation ($\alpha = 13^{\circ}$)

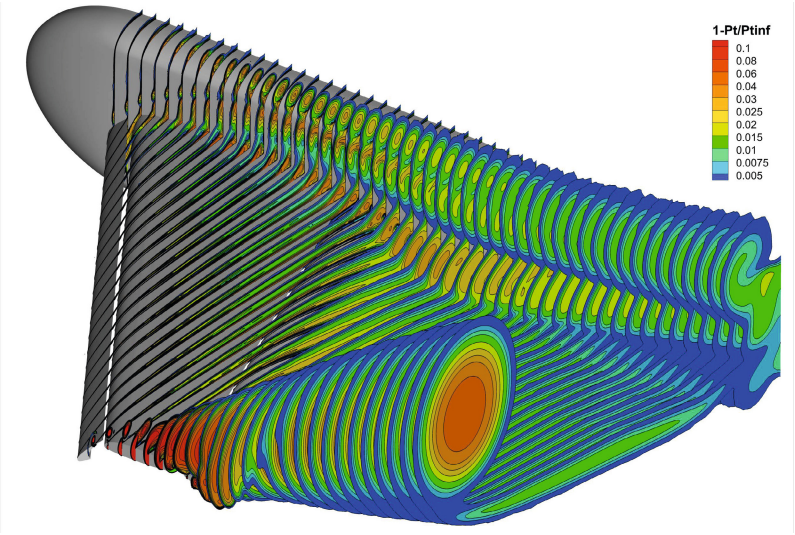


Fig. 14. Total pressure loss after 3^{rd} adaptation ($\alpha = 28^{\circ}$)

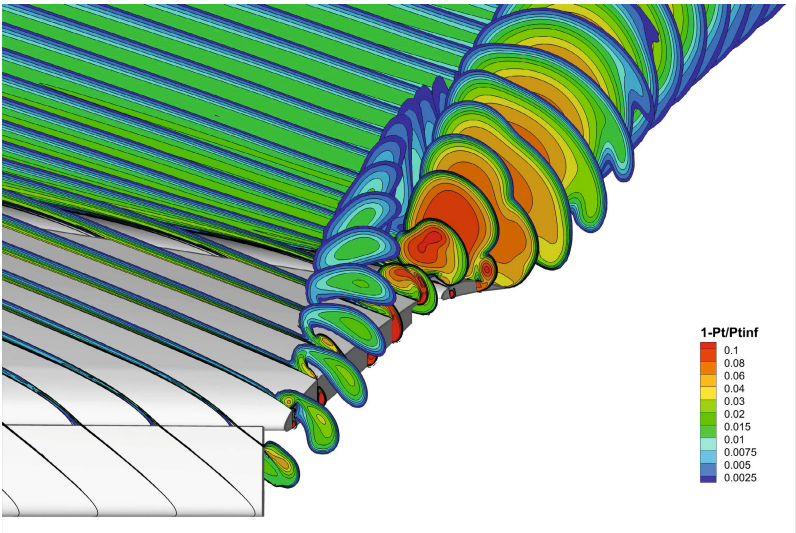


Fig. 15. Total pressure loss after 3^{rd} adaptation ($\alpha = 13°$) - details of the wing tip vortices

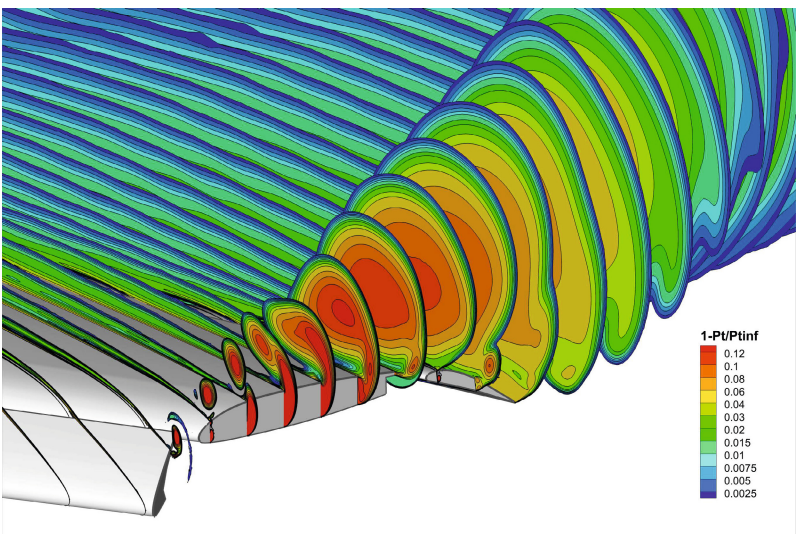


Fig. 16. Total pressure loss after 3^{rd} adaptation ($\alpha = 28°$) - details of the wing tip vortices

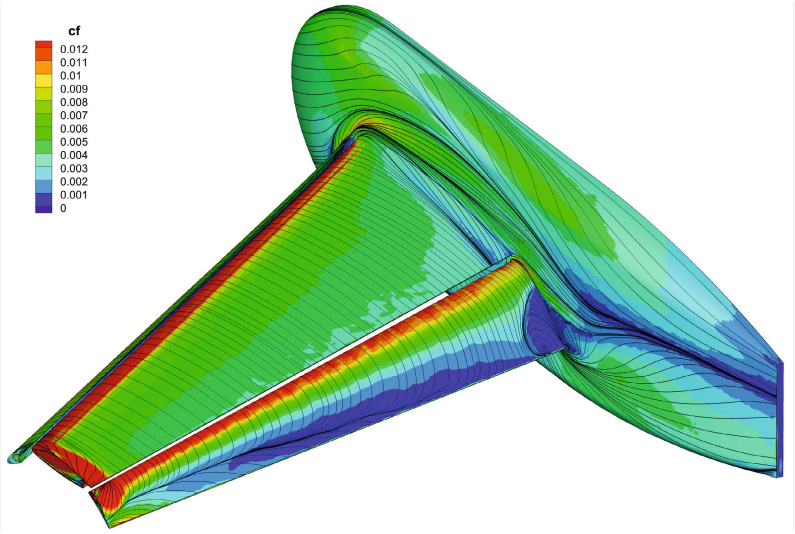


Fig. 17. Streamtraces and surface friction after 3^{rd} adaptation ($\alpha = 13^{\circ}$)

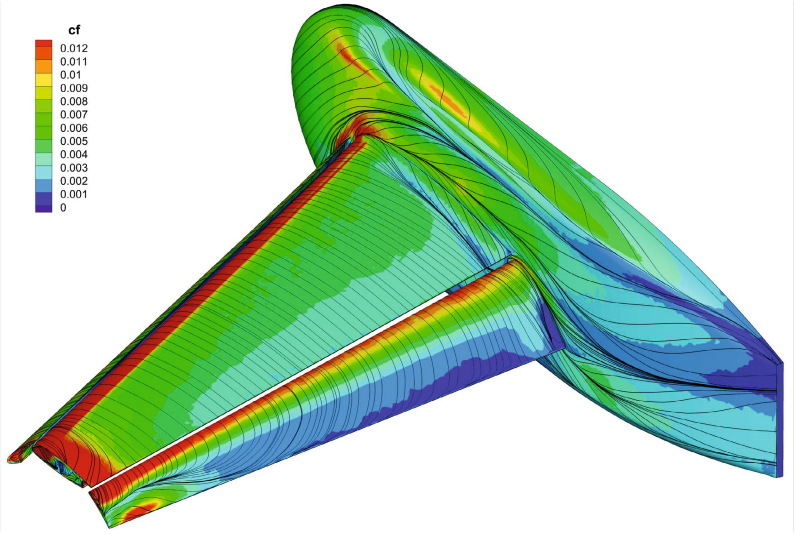


Fig. 18. Streamtraces and surface friction after 3^{rd} adaptation ($\alpha = 28^{\circ}$)

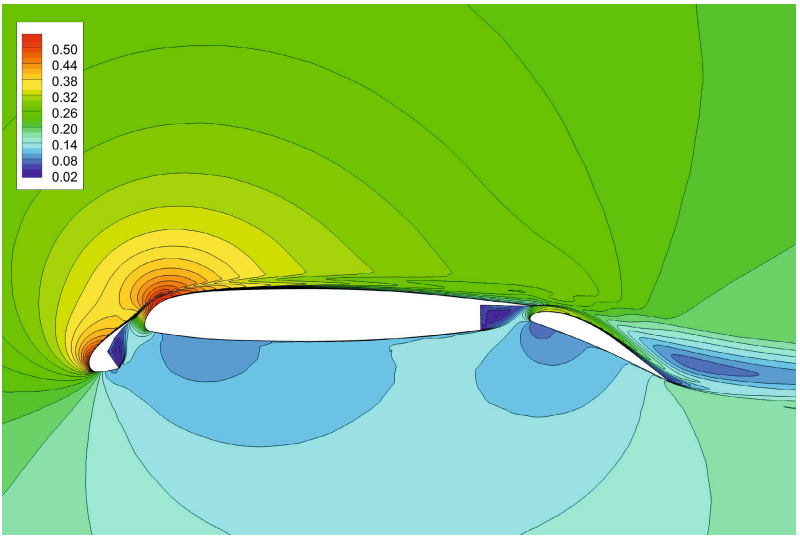


Fig. 19. The velocity field at $y/b = 0.28$ for $\alpha = 28°$ (3^{rd} adaptation)

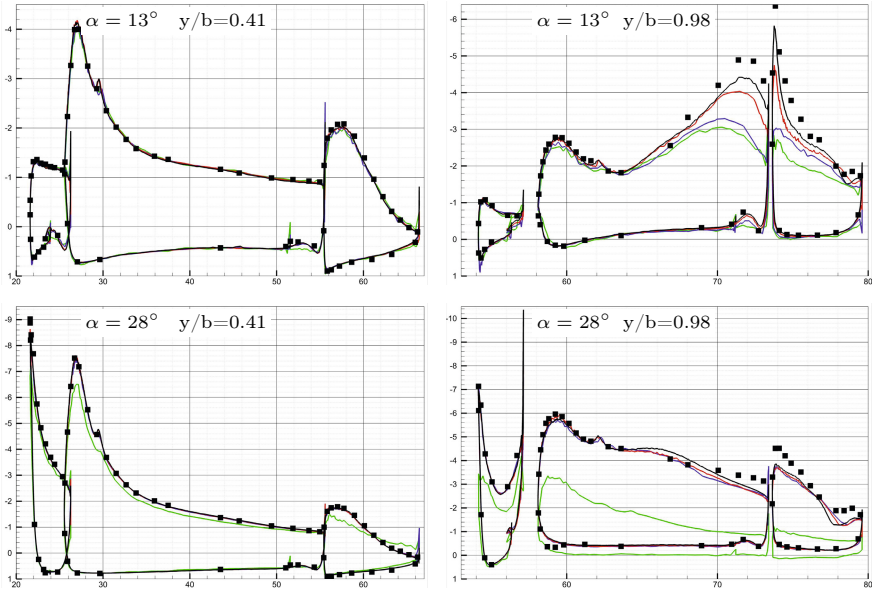


Fig. 20. c_p distribution for two chosen sections for A.3 (evolution during the adaptation process)
initial, 1^{st} adaptation, 2^{nd} adaptation, 3^{rd} adaptation, ■ - experiment

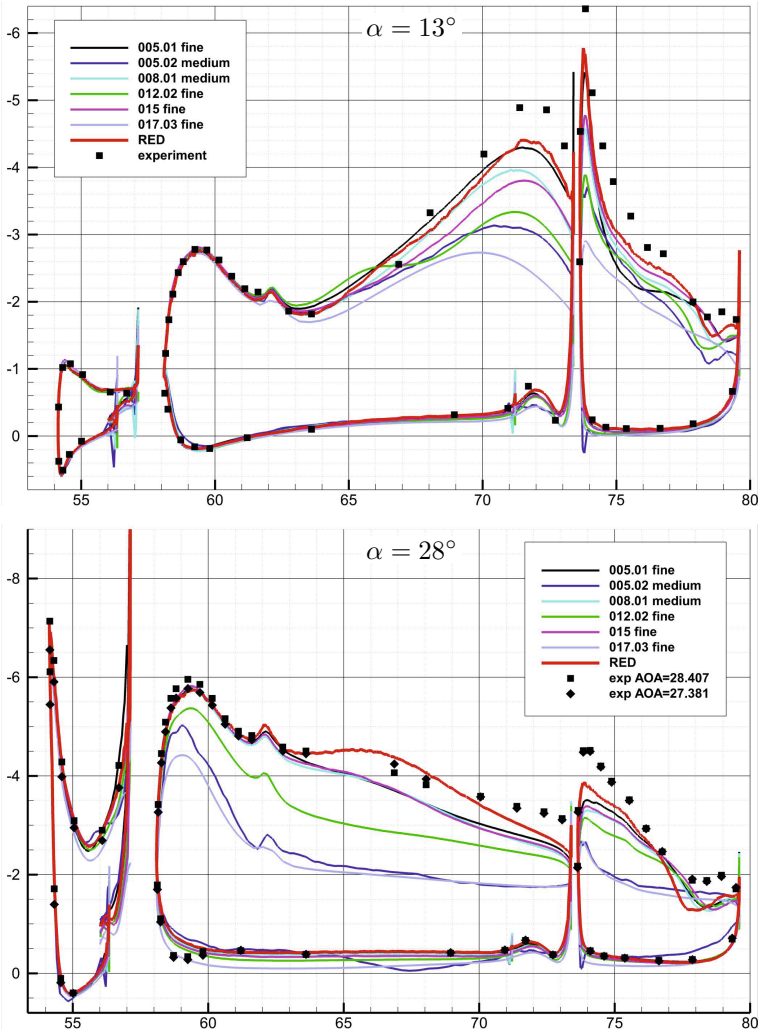


Fig. 21. Comparison of the c_p distribution with HLPW-1 results at $y/b = 0.98$ (RED tags the present adaptive simulations)

7 Conclusions and Future Work

The authors has successfully applied the anisotropic adaptation for simulations of turbulent high Reynolds number flows. The results of the simulations show clear advantage of the anisotropic adaptation – the results are comparable or even better than obtained on grids with much larger number of DOFs (e.g., see Section 6.2 for comparison with the HLPW–1 results). Different modifications added to the anisotropic error estimator improved capability of detection of flow features including viscous shear layers.

There are still few areas that can be addressed in order to further improve the adaptive simulation. They can be summarized in following points:

- The hybrid anisotropic grid generator developed by WUT is still insufficiently robust. The generated grid could be substantially improved by applying better post-optimization. This optimization must be aware of anisotropy of the grid cells which makes the algorithm quite challenging.
- The adaptation based on full remeshing is very demanding with respect to the grid generator. Furthermore, in most simulations it is not necessary to modify the whole grid while it is important to concentrate on relatively small areas which need more attention (especially during the last steps of the adaptation process). It seems therefore that some version of local adaptation might prove advantageous. By using the local operators, the grid could be adapted in anisotropic manner to the currently computed estimation of the required cell spacing. Then the cumbersome process of generation of the full new grid could be avoided while the adaptation could also become integrated into the solver.
- The error estimator should be improved especially by using goal-oriented anisotropic error estimator. Current implementation of the estimator works well when local flow features have to be extracted, however, when integral data (e.g., lift and drag forces) are needed, the results are not fully satisfactory.

Acknowledgments. The research presented here was done in the frame of the IDIHOM project (*Industrialisation of High-Order Methods - a Top-Down Approach*, FP7-AAT-2010-RTD-1, Grant agreement no.: 265780).

References

1. George, P.L., Borouchaki, H.: Delaunay Triangulation and Meshing - Application to Finite Element. Hermes (1998)
2. Castro-Díaz, M.J., Hecht, F., Mohammadi, B.: New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. Technical Report RR-2671, INRIA (1995)
3. Castro-Díaz, M.J., Hecht, F., Mohammadi, B., Pironneau, O.: Anisotropic unstructured mesh adaptation for flow simulation. International Journal for Numerical Methods in Fluids 25, 475–491 (1997)
4. Majewski, J.: Anisotropic adaptation for flow simulations in complex geometries. In: 36th Lecture Series on Computational Fluid Dynamics / ADIGMA Course on Hp-Adaptive and Hp-Multigrid Methods. Von Karman Institute for Fluid Dynamics (2009)
5. Majewski, J., Athanasiadis, A.: Anisotropic solution-adaptive technique applied to simulations of steady and unsteady compressible flows. In: Deconinck, H., Dick, E. (eds.) Computational Fluid Dynamics 2006 - Proceedings of the 4th International Conference on Computational Fluid Dynamics, ICCFD4, Ghent, Belgium, July 10-14, pp. 353–359. Springer (2006)

6. Venditti, D.A., Darmofal, D.L.: Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. J. Comput. Phys. 187, 22–46 (2003)
7. Loseille, A., Dervieux, A., Alauzet, F.: Fully anisotropic goal-oriented mesh adaptation for 3d steady euler equations. J. Comput. Physics. 229(8), 2866–2897 (2010)
8. Hartmann, R., Held, J., van der Ven, H., Dolejsi, V., Majewski, J., Tourrette, L.: Synthesis report on feature-based adaptation. Technical Report ADIGMA D5.1.3 (2009)
9. Majewski, J., Rokicki, J.: Anisotropic mesh adaptation in the presence of complex boundaries. In: Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.) ADIGMA. NNFM, vol. 113, pp. 441–453. Springer, Heidelberg (2010)
10. Najafiyazdi, M., Habashi, W., Fossati, M.: Improved transient-fixed-point mesh adaptation using orthogonality-preserving metric intersection. In: 20th AIAA CFD Conference (2011)
11. Alauzet, F.: Size gradation control of anisotropic meshes. Finite Elem. Anal. Des. 46(1-2), 181–202 (2010)
12. Alauzet, F.: Adaptation de maillage anisotrpe en trois dimensions. Application aux simulations instationnairesen Mécanique des Fluides. PhD thesis, de l'Université des Sciences et Techiques du Languedoc (2003)
13. Borouchaki, H., Hecht, F., Pascal, F.J.: Mesh gradation control. Int. J. Numer. Methods Eng., 131–141 (1998)
14. NPARC Alliance Verification and Validation Archive – Onera M6 wing, `http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html`
15. Bush, R., Power, G., Towne, C.: WIND: The production flow solver of the NPARC Alliance. In: 36th AIAA Aerospace Sciences Meeting and Exhibit. AIAA Paper 98-0935 (1998)
16. 1st AIAA CFD High Lift Prediction Workshop, `http://hiliftpw.larc.nasa.gov/index-workshop1.htm`

Deformation of Curvilinear Meshes for Aeroelastic Analysis

K. Kotecki, H. Hausa, M. Nowak, W. Stankiewicz,
R. Roszak, and M. Morzyński

University of Technologyät Poznań, Division of Virtual Engineering,
60-965, Poznań, Poland
krzysztof.kotecki@put.poznan.pl

Abstract. The article presents elastic analogy approach of deformation curvilinear meshes applied in aeroelastic simulations. The details of algorithm used in developed software with the new metrics designated for high-order mesh quality assessment are presented. The article ends the example of LANN wing deformed by featured tool. Presented software allows conducting the aeroelastic simulation based on CFD discontinuous Galerkin High Order solver.

Keywords: CFD, curvilinear grid, mesh deformation.

1 Introduction

Computational aeroelasticity examines the way a stream of fluid interacts with a deformable body immersed in that fluid [1]. If separate solvers are used for elasticity (CSM) and fluid flow (CFD), usually both domains are discretized with different, non-conforming meshes, which leads to the need of coupling of these codes. Coupling tools are responsible for the exchange of variables, like forces/-pressures and displacements/velocities, between the fluid and the structure [1].

Deformation is the last and one of the most significant parts in aeroelastic simulation. It relies on the movement of the nodes inside the domain, based on the displacement of the nodes on the surface. The main objectives state that quality of the deformed mesh is sufficient for CFD solver in the next step of aeroelastic loop. Considering the high order elements with curvilinear edges make the whole process much more complicated.

Within IDIHOM project the aeroelastic system developed in TAURUS project has been adapted for higher order methods. The most crucial stage was the replacement of the CFD solver from TAU Code to PADGE Code. This step resulted in the following changes, especially in interpolation and deformation tools. This article is focused on the second one.

2 Overview of Mesh Deformation Techniques

The deformation of the structure results in the need of the modification of CFD mesh such a way, that nodes on the CFD surface must follow the deformed object

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_7

being flown around. On the same time, the quality of the CFD mesh should be preserved.

The most widely used methods of mesh deformation are based on the spring analogy, where elements of CFD mesh (tetrahedra, pyramids, prisms and hexahedra) or their edges are replaced with spring elements. Among these methods, torsional spring analogy [2] [2], semi-torsional spring analogy [3] [5], ortho-semi-torsional (OST) spring analogy [6] and ball-vertex spring analogy [7] might be mentioned.

Another possibilities of mesh deformation include the use of beam elements [1] that combines benefits of spring and torsional orthotropic spring analogy, Kriging [9], Inverse Distance Weighting [10] or surface spline interpolation [11].

3 Deformation of Curvilinear Meshes

The tests, carried out on high order meshes using existing deformation tool from TAURUS AE System (based on spring analogy) [1], failed. The new, developed in IDIHOM Project, deformation tool base on elastic analogy with non-uniform material properties. In this method it is assumed that CFD domain is considered as the elastic body with clearly defined material of each cell. The nodal displacements on the "wet" boundary surface (structure-fluid interface) obtained from structural analysis are applied as kinematic loads. Static analysis of that prepared case returns the displacement of all nodes in the domain. The advantage of this method is that every cell in the mesh is considered as single continuous element, not as a set of separate edges as in case of spring analogy. The main challenges of this method are the preparation of metrics for the grid quality evaluation and an algorithm controlling material change in the grid.

The CFD grids for PADGE Code are based on Lagrange element types, while most of the CSM Solvers use Serendipity elements. Not many CSM Solvers support higher order elements as well. The proposed solution of this problem is the division of the high order Lagrange elements into first order ones. Therefore, to solve static structural analysis the PUT In-house parallel CSM Solver is used. The method of the element's division is shown in table 1:

Table 1. Division of second order volume elements

2^{nd} order element	**Equivalent 1^{st} order elements**
TETRA 10	8 x TETRA 4
PYRA 14	16 x TETRA 4
PRISM 18	8 x PRISM 6
HEXA 27	8 x HEXA 8

The deformation tool has been developed as iterative process. Assuming that the quality of every element is sufficient for CFD Solver before deformation, it is possible to deform grid obtaining quality not worse than original one. For

this purpose, the only elements that might be deformed are the ones for which there will be no deterioration in the quality. Controlling the material properties in elements, it is possible to prevent the element's degeneration, by applying sufficiently rigid material. Then, elements with more flexible material will carry the entire deformation. The main idea is to adjust material stiffness in every iteration in element according to its quality.

Every iteration consists of 3 stages:

- Preprocessing - preparation of the task for solver including application of boundary conditions, loads and materials and division of elements into first order ones
- Solution using static state analysis
- Postprocessing - deformation of the mesh and evaluation of its quality

The first iteration differs from the next ones, because there is initializing material process within it. This process is crucial for number of iterations. Predicting which elements are sensitive for decreasing quality after deformation is the most important in material initialization process. This group includes in particular the elements from boundary layer (prisms and hexahedra) that have high ratio between the longest and the shortest edge and elements with high degeneration ratio of skewness. Moreover the elements from a region near surface which has the highest deflection have to be inspected. The simplest method is to assign the same material for all elements. There are also available other methods depending on the kind of the mesh. For hybrid grids, it is easy to define boundary layer elements. The assessment of undeformed mesh provides information about other parameters like skewness, aspect ratio, etc., which ensure better assignment of materials. In structured grids the definition of the element's position in the grid is easier than in unstructured grids. The assessment of flat elements is more difficult because they mostly occur not only in boundary layer but propagate to the end of domain. Considering more metrics cause better material initialization.

Next step is the solution of the prepared case. In spite of larger number of first order elements resulting from the division of high order elements, the time of solving is relatively short. This is due to the use of MPI library and parallelization of whole process.

The last step is postprocessing. First, the grid is deformed and then evaluated. To ensure proper quality of the mesh for CFD Solver, all elements should have positive Jacobian.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial x}{\partial \xi_2} & \frac{\partial x}{\partial \xi_3} \\ \frac{\partial y}{\partial \xi_1} & \frac{\partial y}{\partial \xi_2} & \frac{\partial y}{\partial \xi_3} \\ \frac{\partial z}{\partial \xi_1} & \frac{\partial z}{\partial \xi_2} & \frac{\partial z}{\partial \xi_3} \end{bmatrix} \tag{1}$$

The equation (1) allows the calculation of the Jacobian's value in every node, but it is not sufficient for unambiguous assurance of correctness. A necessary requirement is that Jacobian's value is positive in whole element, not only in nodes.

The metrics based on Bezier functions, developed by Université Catholique de Louvain (UCL) [5] [6], determine the minimum value of Jacobian in the element.

$$B = TJ \tag{2}$$

where T is transformation matrix

This concept allowed to implement these metrics in the deformation tool. Checking the grid by this metric shows is the next iteration required. The pre-processor in next iterations modifies existing material assignment instead of the

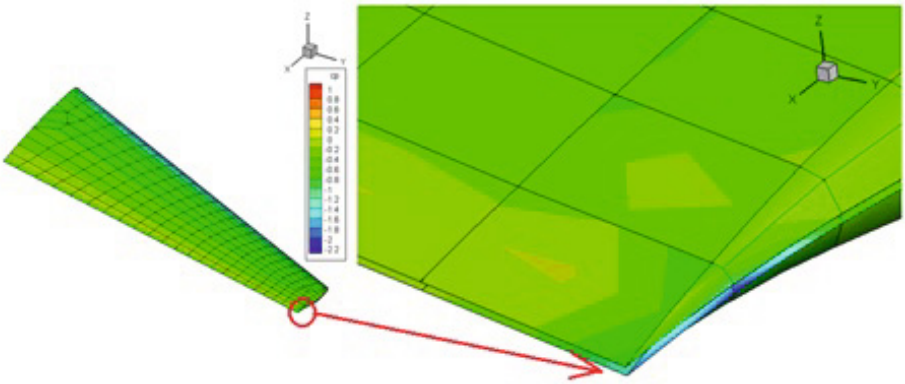


Fig. 1. Deformation of unstructured LANN wing grid

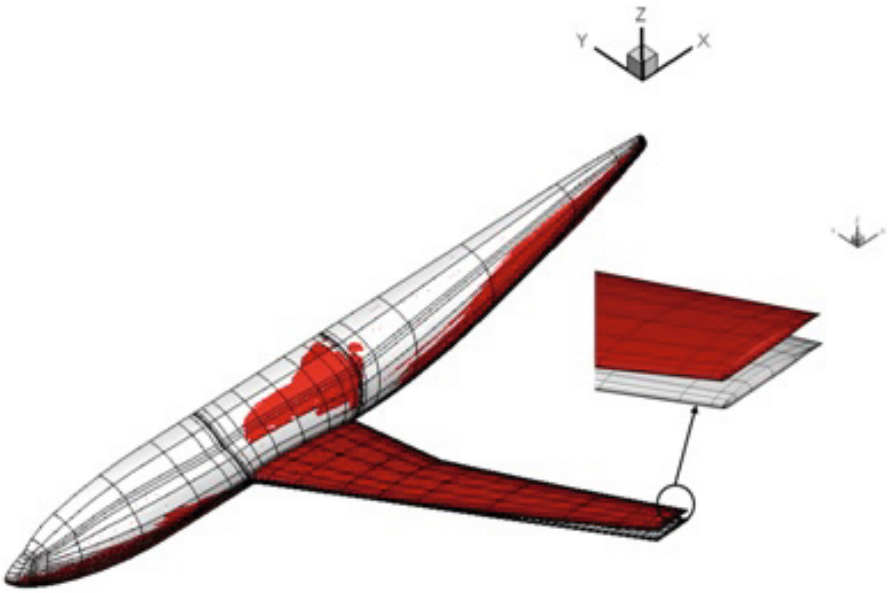
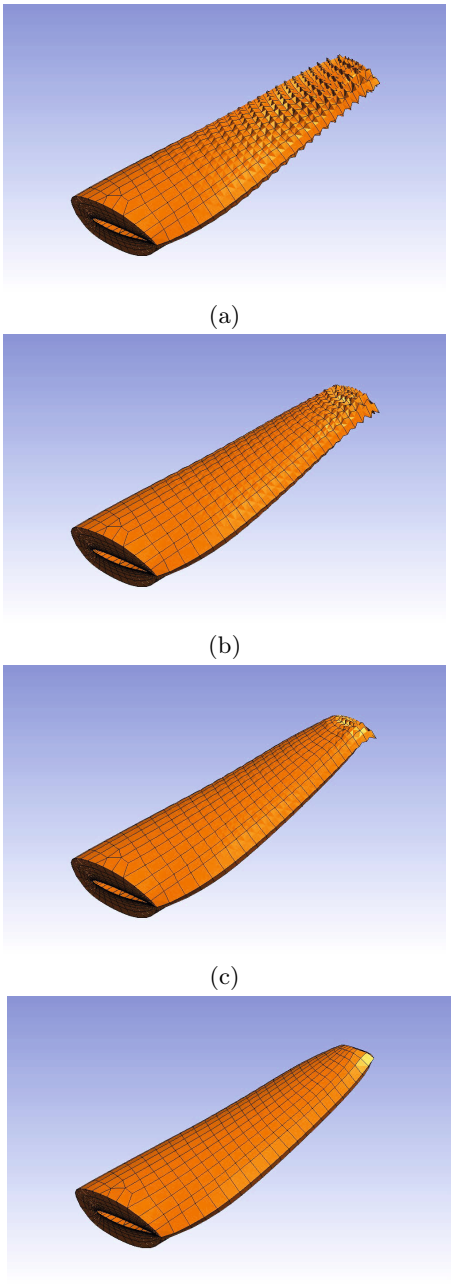


Fig. 2. Deformation of structured DLR-F6 wing-body configuration grid



(a)

(b)

(c)

(d)

Fig. 3. Deformation process of LANN wing in the a) 1st, b) 2nd, c) 3rd, d) 7th, iteration. Visualization of the boundary layer.

creation of new materials. The deformation process is repeated till obtaining correct mesh according to metrics based on Bezier functions.

Material properties changing in preprocessor part are chosen from material base. Tests of CSM Solver used in deformation process allowed for defining material base. The use of fixed material set prevents the application of two different materials with to high spreading of stiffness and resulting problems with solver convergence.

4 Testcases

The deformation tool is used in the aeroelastic (AE) system developed within IDIHOM project. It enables proper deformation of grids in the testcases (LANN wing [3] and DLR-F6 [4]) studied by PUT.

Further tests showed problems in the grid deformation at higher deflections of flow around body. Even applying the most and the less stiff material, some elements were still degenerated. In most cases these elements appear on the deformed surface (1st boundary layer). To prevent that occurrence, these elements are treated as non-deformable, with the ability of rigid movement and rotation. The deformations of CFD grid in aeroelastic testcases proposed in IDIHOM Porject are relatively small. They have amounted to 1-2% of the wing span. To test capabilities of the deformation tool, artificial data as the results from CSM solver was prepared. The maximum deflection had value equal 10% of wing span. The LANN wing mesh was deformed correctly (fig. 3).

More details on the aeroelastic analyses including the deformation described above might be found in chapters describing aeroelastc testcases.

5 Summary

The deformation tool for high order grids has been developed. It is a key enabler of aeroelastic analyses based on high-order methods and meshes. The testcases proposed in IDIHOM project, LANN wing and DLR-F6 wing-body configuration, have been successfully analyzed. Aeroelastic simulation required deformation of high-order CFD meshes, which was successfully achieved. The tests with artificial large deflection of the wing have shown promising direction in the deformation process, possible to be applied in industrial cases. Further development of algorithm could reduce the computational time of deformation process.

References

1. Kamakoti, R.: Computational Aeroelasticity Using a Pressure-Based Solver. PhD thesis University of Florida (2004)
2. Farhat, C., Degand, C., Koobus, B., Lesoinne, M.: Torsional springs for two-dimensional dynamic unstructured fluid mesh. Comput. Methods Appl. Mech. Engg. 163, 231–245 (1998)

3. Degand, C., Farhat, C.: A three-dimensional torsional spring analogy method for unstructured dynamic meshes. Comput. Struct. 80, 305 (2002)
4. Bloom, F.J.: Consideration on the spring analogy. Int. J. Numer. Methods Fluids 32, 647–668 (2000)
5. Zeng, D., Ethier, C.R.: A semi-torsional analogy model for updating unstructured meshes in 3D moving Domains. Finite Elem. Anal. Des. 41, 1118–1139 (2005)
6. Markou, G.A., Mouroutis, Z.S., Charmpis, D.C., Papadrakakis, M.: The ortho-semitorsional (OST) spring analogy method for 3D mesh moving boundary problems. Comput. Methods Appl. Mech. Engg. 196, 747–765 (2007)
7. Bottasano, C.L., Detomi, D., Serra, R.: The ball-vertex method: a new simple analogy method for unstructured dynamic meshes. Comput. Methods Appl. Mech. Engg. 194, 4244–4264 (2005)
8. Roszak, R., Posadzy, P., Stankiewicz, W., Morzyski, M.: Fluid structure interaction for large scale complex geometry and non-linear properties of structure. Archives of Mechanics 61(1), 3–27 (2009)
9. Stein, M.L.: Interpolation of spatial data: some theory for kriging. Springer (1999)
10. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM National Conference. ACM (1968)
11. Harder, R.L., Desmarais, R.N.: Interpolation using surface splines. Journal of Aircraft 9(2), 189–191 (1972)
12. Geuzaine, C.H., Remacle, J.-F.: Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)
13. Toulorge, T., Geuzaine, C.H., Remacle, J.-F., Lambrechts, J.: Robust untangling of curvilinear meshes. Journal of Computational Physics 254, 8–26 (2013)
14. Ruo, S.Y., Malone, J.B., Horsten, J.J., Houwink, R.: The LANN programan experimental and theoretical study of steady and unsteady transonic airloads on a supercritical wing. AIAA Paper 1983-1686
15. Brodersen, O., Strmer, A.: Drag Prediction of Engine Airframe Interference Effects using Unstructured Navier Stokes Calculations. In: 19th AIAA Applied Aerodynamics Conference, Anaheim, CA, USA, June 11-14. AIAA Paper 2001-2414 (2001)

Mesh Curving Techniques for High Order Discontinuous Galerkin Simulations

F. Hindenlang, T. Bolemann, and C.-D. Munz

Institute for Aerodynamics and Gasdynamics, Universität Stuttgart
Pfaffenwaldring 21, 70569 Stuttgart, Germany
{hindenlang,bolemann,munz}@iag.uni-stuttgart.de

Abstract. In the development of the next generation of numerical methods for CFD, Discontinuous Galerkin methods are promising a substantial increase in efficiency and accuracy. While the particular high order methods can be very distinct, they have in common that they must rely on a high-order approximation of curved geometries to maintain their high-order of accuracy. The generation of curved meshes is thus a topic whose importance cannot be overstated, if one truly wants to apply DG methods to problems with industrial relevance. Especially aerospace applications heavily rely on complex geometries and pose high requirements to the quality of geometry representation. In this work we present several techniques to produce high order meshes, relying on linear meshes, which can be generated by standard commercial mesh generation tools. We describe the generation of the curved surface meshes and curved volume meshes, which can be particularly difficult for curved boundary layers.

Keywords: mesh generation, curvilinear meshes, inner-cell stretching, grid deformation, elastic analogy, interpolation, normal vectors, unstructured grids.

1 Introduction

In industrial and especially aerospace applications, geometries are three-dimensional and typically comprise curved surfaces, curved borders and sharp edges. Here meshing itself becomes an issue and therefore, unstructured grids are needed. Most of the unstructured grid generators provide high quality grids consisting of hexahedra, prisms, tetrahedra and pyramids, whereas the element edges are in general straight lines, at most having an additional mid-point (quadratic elements).

When using high order methods, wall boundary conditions need a high order representation of the wall normal. Bassi and Rebay [2] showed, that in the case of curved boundaries a high order DG discretization with straight-sided 2D elements yields low order accurate and even physically wrong results. To overcome this issue, they propose to use at least elements with parabolic shaped sides on the boundary.

Approaches for high order grids should use CAD definitions to guarantee a correct approximation of the geometry. As high order mesh generation is still a topic

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_8

of ongoing research [11,12], CAD handling and modification as well as meshing of complex geometries with unstructured hybrid meshes have not reached the level of commercial grid generators. Therefore, our approach fully relies on commercial grid generators because of their ability to work on CAD definitions and produce high quality linear meshes, and use additional informations for the element curving.

Finding the curved element mapping is the key problem of the high order grid generation. In this work, we present several strategies to produce high order meshes, which are all implemented in a standalone preprocessing tool called the *preproctool*. It fills the gap between the linear grid generation and the simulation with the DG solver.

The paper is organized as follows: We introduce the high order element mapping in the section 2, followed by an an overview of the *preproctool* workflow in section 3. The first step to construct curved meshes consists of the curving of boundary faces of elements at curved surfaces, which is adressed in section 4. In the case of a boundary layer mesh, for example, only curving the boundary faces lead to invalid meshes. To overcome this problem, a mesh deformation strategy is presented in section 5.1 to propagate the boundary curvature into the mesh volume. Thereof independent is the strategy where block-structured volume meshes are directly transformed to fully curved hexahedral meshes, described in section 5.2.

2 Element Mapping

To be able to formulate the discontinuous Galerkin scheme on arbitrary elements, we have to describe the element geometry. The geometry of linear or curved elements is expressed by the mapping of an element between physical space $\boldsymbol{X}$ and reference space $\boldsymbol{\xi}$ by

$$\boldsymbol{X}(\boldsymbol{\xi}) \,, \tag{1}$$

with

$$\boldsymbol{X} = (X_1, X_2, X_3)^T = (X, Y, Z)^T \,, \quad \boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)^T = (\xi, \eta, \zeta)^T \,. \tag{2}$$

In practice, we represent the mapping by an interpolation polynomial

$$\boldsymbol{X}(\boldsymbol{\xi}) = \sum_{l=1}^{n_{\mathrm{IP}}} \hat{\boldsymbol{x}}_l \psi_l(\boldsymbol{\xi}) \,, \tag{3}$$

using a set of n_{IP} interpolation nodes, with physical coordinates $\{\hat{\boldsymbol{x}}_l\}_{l=1}^{n_{\mathrm{IP}}}$ and corresponding reference coordinates $\{\hat{\boldsymbol{\xi}}_k\}_{k=1}^{n_{\mathrm{IP}}}$, and the Lagrange basis functions $\psi_l(\boldsymbol{\xi})$ of polynomial degree N_g, having the cardinality property

$$\psi_l(\hat{\boldsymbol{\xi}}_k) = \delta_{kl} \,. \tag{4}$$

The choice of the reference node positions is free, however, it is common to include the boundaries of the element and to use an equidistant spacing.

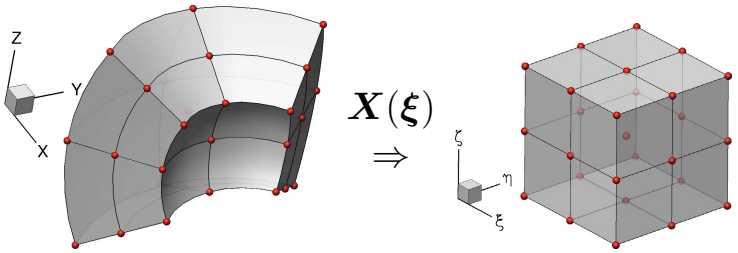


Fig. 1. Mapping of a hexahedral element from physical to reference space

This choices guarantee that the basic linear mapping ($N_g = 1$), using only the elements corner nodes, is included and also ensures that the transformation remains linear if physical nodes are equidistant. In addition, these node positions are very easy to implement for all element types.

As an example, the quadratic mapping ($N_g = 2$) of a curved hexahedra is shown in Figure 1. The element has $n_{\mathrm{IP}} = (N_g+1)^3$ interpolation nodes and the reference domain is $\boldsymbol{\xi} \in [-1, 1]^3$. The reference node positions are equidistant

$$\hat{\boldsymbol{\xi}}_{ijk} = (\hat{\xi}_i, \hat{\xi}_j, \hat{\xi}_k)^T \quad \hat{\xi}_i = 2\left(\frac{i}{N_g}\right) - 1, \quad i, j, k = 0, \ldots, N_g\,, \tag{5}$$

and the hexahedral element mapping reads as

$$\boldsymbol{X}(\boldsymbol{\xi})_{(\mathrm{Hex})} = \sum_{ijk=0}^{N_g} \hat{\boldsymbol{x}}_{ijk} \ell_i(\xi)\ell_j(\eta)\ell_k(\zeta)\,, \tag{6}$$

with a tensor-product polynomial basis consisting of 1D Lagrange basis functions $\ell_i(\xi)$.

3 Overview

In Figure 2 a flowchart of the curved mesh generation process is shown. The starting point is the linear unstructured mesh, created by the grid generator of choice. The CGNS standard mesh format is commonly used as input. The *point-normals* strategy uses normal vectors at the surface grid points of a given linear mesh. Either they are found by analytical functions, specified by the user, or for complex geometries directly from the CAD model via a CAD interface, providing a list of surface grid points and their associated normal vectors, see section 4.2 for details.

In this work, the commercial grid generators ANSA© and ICEM© were used to generate linear unstructured meshes. The software generates meshes using the original CAD geometry, and also provides CAD modifications, which are often necessary to clean up the geometry. The ANSA© mesh generator has a built-in

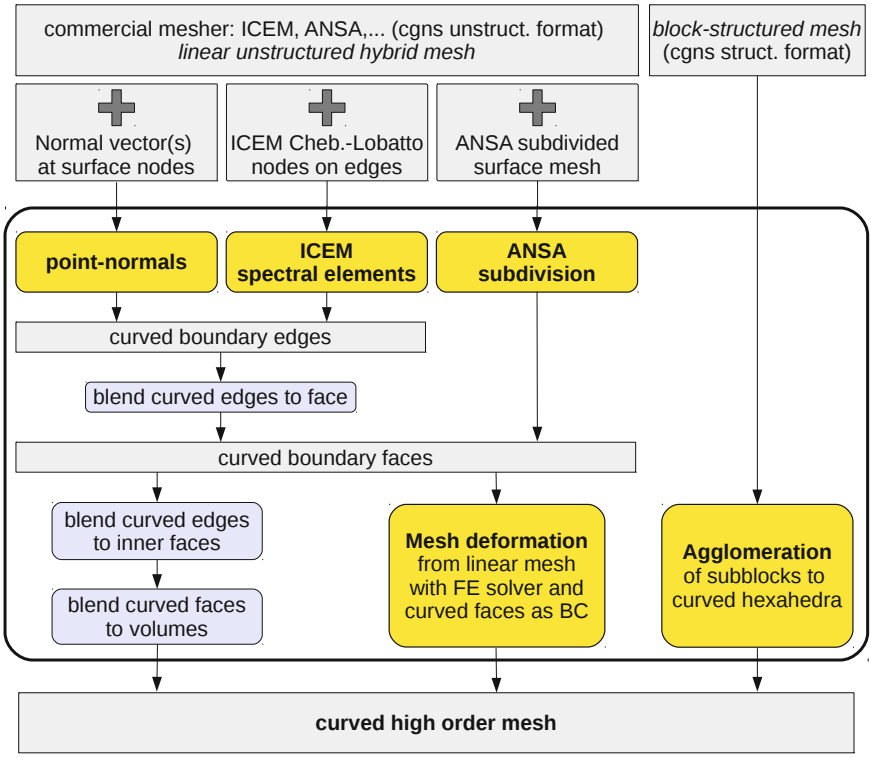


Fig. 2. Flowchart of the curved mesh generation process

feature (called *split*) to subdivide surface meshes, and ICEM© is able to extract Chebychev-Lobatto interpolation points along the element edges (called *spectral elements*). The points are again distributed along the original CAD surface, which makes it possible to use these points to construct high order boundary faces by simple interpolation, see section 4.4.

The ANSA© *subdivision* directly produces curved boundary faces, whereas the other two strategies only produce curved element edges, which are then blended to a curved boundary face. The same blending is used for the inner faces sharing edges with the curved boundary face, and finally the volume is again a blending of its faces. The blending techniques are detailed in section 4.3.

The surface curving only provides curved elements in the first layer at the boundary, and may lead to invalid element mappings, for example for boundary layer meshes. In this case, the mesh deformation strategy, first introduced in [17], is applied. Here, the mesh is modeled as a deformable solid, and the curved boundary is imposed as displacement from the linear mesh. A high order Finite Element Method solves for the deformed mesh, leading to a propagation of the curvature into the mesh volume. More details are found in section 5.1. Finally, a completely separated strategy is a simple agglomeration of block-structured

grids, directly leading to fully curved hexahedra, where intermediate structured points are used for interpolation of the element mapping, see section 5.2.

We introduce the surface mapping for the curved boundary faces

$$\boldsymbol{X}^S(\xi_1, \xi_2) = \sum_{j=1}^{n^s_{\mathrm{IP}}} \hat{\boldsymbol{x}}_j \psi_j(\xi_1, \xi_2)\,, \tag{7}$$

being a polynomial representation of the curved triangle or quadrangle. Here, (ξ_1, ξ_2) are the parameter directions, $\hat{\boldsymbol{x}}_j$ are the coordinates of the interpolation points on the curved surface and $\psi_j(\xi_1, \xi_2)$ are the corresponding Lagrange basis functions. The definition of multivariate Lagrange basis functions are found for the general case by using multi-variate modal basis functions, which are given for all element types in the book of Karniadakis and Sherwin [15].

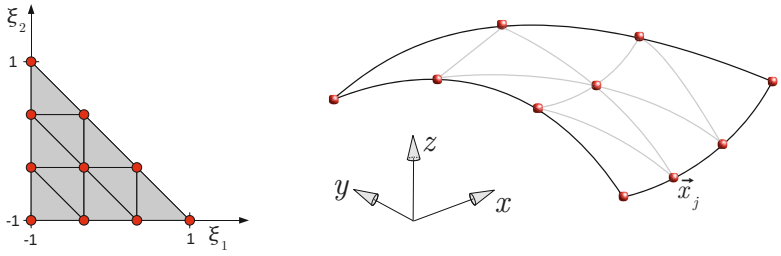


Fig. 3. Mapping from parameter space (ξ_1, ξ_2) to physical space using (7)

The number of interpolation points for triangles is $n^s_{\mathrm{IP}} = \frac{1}{2}(N_g + 1)(N_g + 2)$ and for quadrangles $n^s_{\mathrm{IP}} = (N_g + 1)^2$, where N_g is the polynomial degree of the mapping. In Figure 3 we show the mapping for a triangle and $N_g = 3$. We choose a regular spaced node distribution in reference space to represent the polynomial of the mapping. Regular spaced interpolation points in physical space are then directly mapped. When using other point distributions or basis functions, like a Bezier basis, polynomial coefficients are easily found by applying a Vandermonde matrix. We want to point out that this is only a question of polynomial representation and does not change the polynomial itself. The approximation error will only depend on the type of surface data.

4 Generation of the Surface Meshes

4.1 Continuity-Based Approach

The basic idea when reconstructing a curve by a set of points with spline interpolation is to enforce continuity at the point positions. The most popular is the cubic spline interpolation [9]. The curve is found by solving a tridiagonal linear

equation system with a size of the number of interpolation points. In two dimensions, it is a simple and robust technique to reconstruct the boundary curves. The only way to extend this technique to three-dimensional boundary surfaces is to use structured point distributions, which can be broken down into line-by-line interpolations. However, for unstructured meshes, the number of interpolation points increases significantly and in particular, a unique parametrization over element edges is not possible. Thus a completely local approach is attractive, using only the normal vectors at the element corner nodes. The surface elements are then G^1 continuous at the corner nodes. It is a so-called *point-normal* approach proposed in [9]. In Stiller [19], rational Bezier curves from point-normals are constructed, being able to represent spheres, cylinders and cones exactly. In contrast, we restrict ourselves to non-rational cubic polynomials.

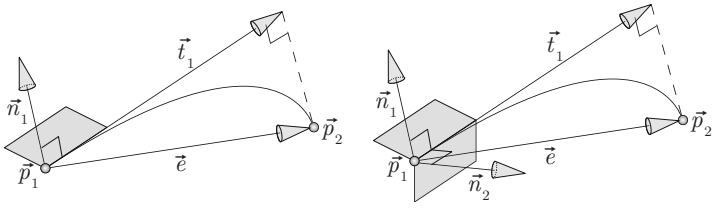
Clearly, the most difficult part is to provide the normal vector. In the *preproctool*, analytical expressions for simple geometric shapes are included, and it is easy to add other user-defined functions. We also need to consider sharp edges at the intersections of surfaces, where two point-normals are defined. To find these edges, the user marks each curved boundary patch with a unique index, as an additional information to the boundary conditions.

The point-normal assignment for complex geometries is realized via a CAD interface, see section 4.2 for the details.

A cubic polynomial curve is defined by its end points $(\boldsymbol{p}_1, \boldsymbol{p}_2)$ and two tangential vectors $(\boldsymbol{t}_1, \boldsymbol{t}_2)$, and reads in Bezier form as

$$\begin{gathered} \boldsymbol{X}_b(t) = \sum_{i=0}^{3} \boldsymbol{b}_i B_i^3(t)\,, \quad B_i^N(t) = \binom{N}{i}(1-t)^{N-i}t^i\,, \quad t \in [0;1] \\ \boldsymbol{b}_0 = \boldsymbol{p}_1\,, \quad \boldsymbol{b}_1 = (\boldsymbol{p}_1 + \frac{1}{3}\boldsymbol{t}_1)\,, \quad \boldsymbol{b}_2 = (\boldsymbol{p}_2 - \frac{1}{3}\boldsymbol{t}_2)\,, \quad \boldsymbol{b}_3 = \boldsymbol{p}_2\,. \end{gathered} \tag{8}$$

The Lagrange interpolation points with regular point spacing along the curve are simply found by interpolation of the Bezier curve.



(a) 1 point-normal (b) 2 points-normals

Fig. 4. Construction of tangential vectors (a) by projection and (a) by cross product

Assuming the point-normals are provided, the tangential vectors have to be constructed. As shown in see Figure 4, the point-normal defines a tangential plane, and the tangential vector is found by projection of the straight edge $\boldsymbol{e}$ connecting the two edge points. If two point-normals are given, the direction of

the tangential is found by a cross-product and its length again by projection onto the straight edge.

$$\boldsymbol{e} = \boldsymbol{p}_2 - \boldsymbol{p}_1\,, \quad \boldsymbol{t}_1 = \boldsymbol{e} - (\boldsymbol{n}\cdot\boldsymbol{e})\boldsymbol{n}\,, \quad \text{for 1 normal,} \tag{9}$$

$$\boldsymbol{t}_1 = ((\boldsymbol{n}_1 \times \boldsymbol{n}_2)\cdot\boldsymbol{e})\,(\boldsymbol{n}_1 \times \boldsymbol{n}_2)\,, \quad \text{for 2 normals.} \tag{10}$$

As depicted in Figure 5, the tangential vectors are constructed for all element edges and the interpolation points from the Bezier curve.

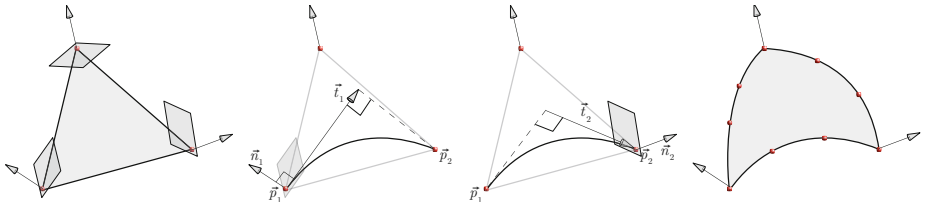


Fig. 5. Sequence of constructing curved element edges from surface normals

4.2 CAD Interface

For a complex three-dimensional geometry, we want to use the CAD definitions to find the exact normal vector at the grid points of the linear surface mesh. The CAD interface was developed by Thomas Bolemann in his diploma thesis [5] and is written in Visual Basic 8 and uses the Microsoft .Net 2.0 framework. It is connected to CATIA via a scripting interface (CAA-API), thus commands to load a model, do geometric operations and extract CAD definitions are made directly accessible for the CAD tool.

The work-flow to assign the point-normals obtained form the CAD model is shown in Figure 6. The CAD geometry is typically provided by a STEP file

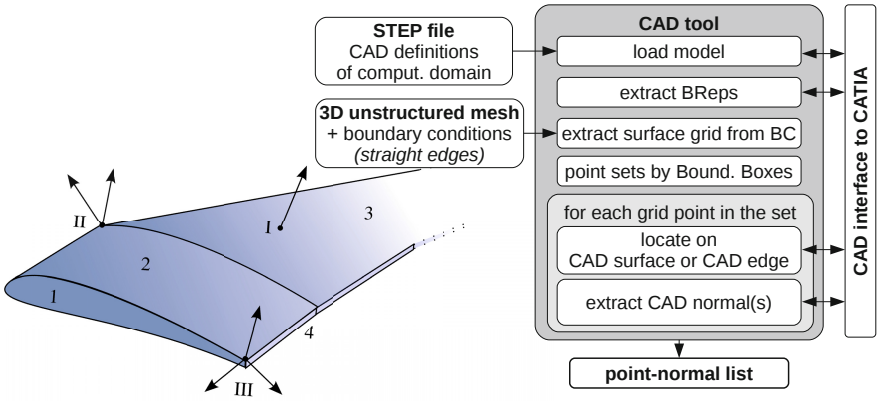


Fig. 6. Flowchart of the CAD tool to assign normal vectors from CAD

('.stp'), a widely used standard exchange format. In general, grid generators are able to import the STEP file and provide a mesh with straight-edged elements and boundary conditions (BCs).

In a preliminary step, the CAD tool reads the 3D grid, and extracts all grid points lying on boundaries, and even more important the connectivity of the surface grid. Then the STEP file is loaded and the topology is analyzed, and for each CAD surface, a bounding box is created. For each CAD surface, the distance between the grid points and the CAD surface and its edges is measured to decide whether or not the grid point lies on the surface or edge. To minimize the computational effort, only grid points inside the bounding box are considered. If the grid point lies on the surface (case I in Figure 6), one normal vector is evaluated at the grid point. Multiple normals are found for CAD edges (case II) or corners (case (III). A unique faceID is attached to every CAD surface, and each normal vector carries the faceID as a tag. The connectivity of the surface grid is important, because it helps to define local tolerances for the decision, wether the point lies on a surface, edge or corner, but also helps to find compound CAD faces, which were merged during mesh generation.

Finally, the output is a point-normal list, which includes the grid point index, the position, the number of normals at this point and the faceID for each normal. The data structure of the *preproctool*matches this format, where a flexible list of normal vectors can be allocated for each grid point. The faceIDs are very important, since they facilitate curved edge construction, especially at sharp edges and corners.

4.3 Volume Mapping from Blending Curved Edges

The volume mapping is defined, following Figure 7, by including the surface mapping of all element sides, and the surface mapping in turn is defined by curved element edges.

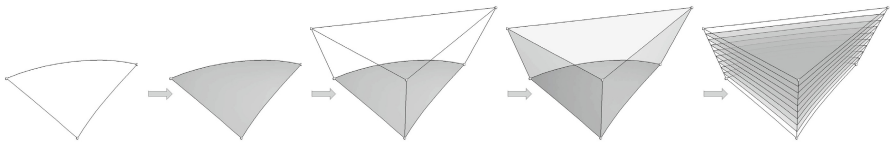


Fig. 7. Construction of mappings: from curved element edges to surface and volume mappings

1. All element edges lying on the curved boundary are *uniquely* defined by a polynomial curve. The uniqueness of the edges guarantees C^0 continuity of the high order grid. The polynomial curve is represented by equally spaced interpolation points and Lagrange basis functions. The inner edges remain straight lines.
2. Triangle and quadrangle mappings are also defined by Lagrange polynomials. The interpolation points on the edge remain the same, and inner points can

be expressed in terms of the edge points using blending functions. This is done for all element sides having at least one curved edge.

3. The volume mapping is a blending of all element sides. It is again a polynomial, resulting in the full volume mapping $\boldsymbol{X}(\boldsymbol{\xi})$. The blending functions for the standard volume elements can be constructured analogously by combinations of curved faces, edges and the corner nodes.

For element sides having at least one curved edge, we use a blending function to derive the curved element side. Blending three or four bounding curves yields to the so called *Coons* patches [9,10]. The blending technique is shown in Figure 8.

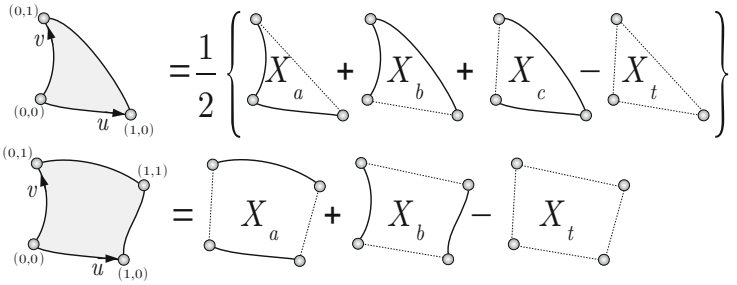


Fig. 8. Blending of curved element edges to a curved element side

We define the curves $C(u, v)$, representing the boundary curves of the element, evaluated at the parameter value $u, v \in [0; 1]$. The triangular Coons patch according to [9] is defined as

$$\begin{aligned} X(u,v) &= \frac{1}{2}\left(X_a + X_b + X_c - X_t\right), \quad \text{with} \\ X_a &= \frac{u}{u+v}C(u+v,0) + \frac{v}{u+v}C(0,u+v), \\ X_b &= \frac{1-u-v}{1-v}C(0,v) + \frac{u}{1-v}C(1-v,v), \\ X_c &= \frac{1-u-v}{1-u}C(u,0) + \frac{v}{1-u}C(u,1-u), \\ X_t &= (1-u-v)C(0,0) + uC(1,0) + vC(0,1), \end{aligned} \tag{11}$$

and the quadrangular Coons patch is a simple bilinear blending with

$$\begin{aligned} X(u,v) &= X_a + X_b - X_t, \quad \text{with} \\ X_a &= (1-v)C(u,0) + vC(u,1), \\ X_b &= (1-u)C(0,v) + uC(1,v), \\ X_t &= (1-u)(1-v)C(0,0) + u(1-v)C(1,0) \\ &\quad + (1-u)vC(0,1) + uvC(1,1). \end{aligned} \tag{12}$$

Finally, to describe the surface with the polynomial (7), the unknown inner interpolation points are found by evaluating the Coons surface $X(u, v)$ at the corresponding position in parameter space. Note that the formula of the triangular coons patch can be evaluated at inner surface points only.

Elements are curved only when elements they are linked – e.g. sharing at least one edge – with the curved wall boundary ('local curving'). If the elements are highly stretched or skewed, additional curving of inner elements would be required ('global curving') to avoid negative or very small Jacobians, which will be adressed in section 5.1. We found that using near-wall tetrahedra or pyramids generally leads to much smaller Jacobians than using prismatic extrusions of the surface grid, i.e. prisms and hexahedra.

4.4 Interpolation Approach

Assuming that we have access to a grid generator to create a mesh of a given CAD model, the mesh and the CAD are internally connected to each other. Once the mesh is exported, this information is typically lost and we are left with a mesh with linear edges. The idea is now to extract additional information using features of the grid generators to generate a refined surface grid and use the additional grid points for interpolation. In [1], a high order hexahedral grid with Chebychev-Lobatto (CL) point distribution is generated for a spectral element method using ICEM©. They started from a linear base mesh generated in ICEM© and were able to project CL point distributions of edges and surfaces onto the CAD faces via projection libraries of the ICEM© hexa module. Even though the projection libraries are not accessible anymore, a so-called *spectral elements* function is available to the curved boundary faces. One can export a file containing CL points for each edge of the surface mesh. The number of points can be chosen arbitrarily, two examples are shown in Figure 9. The file format provides the edges' end point node IDs and the additional point coordinates. We can therefore associate the linear mesh edges with the curved edges, but, like for the normal vector strategy, we need to blend the curved edges to find the curved element faces, see section 4.3.

Another feature is surface mesh refinement. It is naturally supported by the grid generator ANSA©. In each splitting step, surface elements are isotropically refined by a factor of two, and the new grid points are reprojected to the geometry, keeping the original aspect ratio of the surface element. In Figure 10, the original surface mesh and the refined mesh after two splitting steps are shown. Triangles and quadrangles are treated equally. Note that original mesh points remain unchanged. This property is important, since the *preproctool* applies a point matching routine between the boundary faces of the original mesh and the refined surface mesh.

Given the polynomial degree of the surface mapping ($N_g = 2, 4, 8$) corresponding to the number of refinement steps, an algorithm finds the connection between each boundary face and its refined surface elements by making use of the connectivity information of the refined surface mesh. We want to point out that any other grid generator which supports this type of isotropic surface mesh

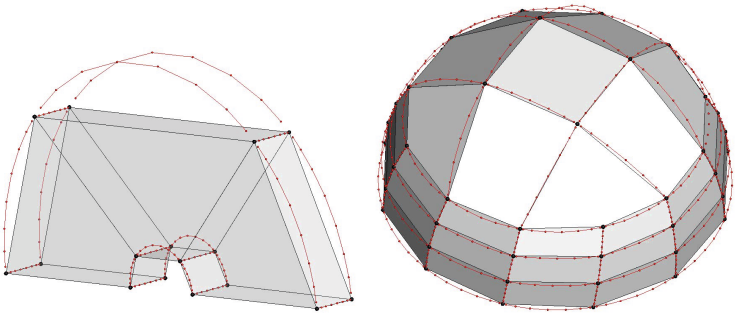


Fig. 9. Additional CL points on curved edges for a very coarse mesh of a cylinder and a sphere surface, generated with ICEM©

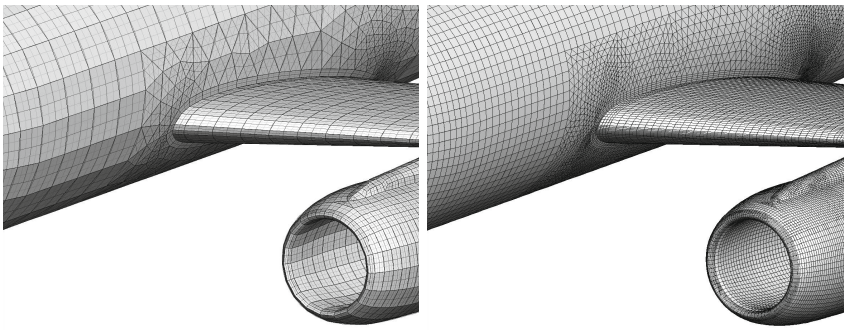


Fig. 10. Surface mesh of the DLR-F6 body-wing intersection, after two steps of isotropic refinement

refinement can be used for this strategy. For example, we note that the software CUBIT [4] also provides this feature.

5 Generation of Curved Volume Meshes

5.1 Mesh Deformation

Boundary layer meshes typically have very high aspect ratios near the wall, and if the wall is curved, a curved mesh generation based on linear meshes and surface curving may lead to inverted elements, see Figure 11. A curved boundary layer mesh is easy to construct with the agglomeration approach from section 5.2 and should always be considered if a structured mesh is available. Fairly complex geometries are meshed nowadays with block-structured meshes, however structured meshing remains a time-consuming task, and geometric restrictions often produce a large amount of additional elements. As the unstructured meshes available have only straight element edges, the surface curving must be propagated into the domain.

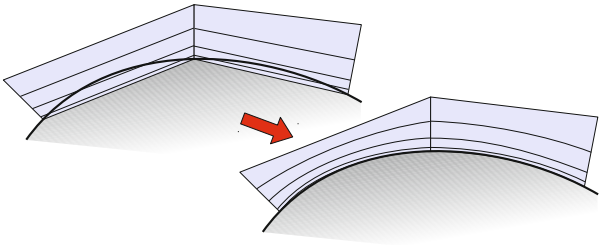


Fig. 11. The curved surface intersects the inner edges of a linear boundary layer mesh, and inner elements must be curved for a valid mesh

Provided that near-wall elements are simple prismatic extrusions of the surface mesh (prisms or hexahedra), a given stack of elements can be found from the mesh connectivity information and curved via a transfinite interpolation between the lower and upper surface. However, complex meshes may involve non-prismatic near-wall elements, thus a more general approach is needed. The mesh deformation issue is not new, it is found in applications involving fluid-structure interaction, where volumetric fluid meshes have to adapt to a deformation of the boundary surface to guarantee a valid mesh. For linear unstructured meshes, different approaches are found in literature. For example radial basis functions [7] or continuous approaches exploiting the internal mesh connectivity, from simple spring models [8] to Laplacian or biharmonic operators [3, 13] or solid body elasticity. The latter considers the mesh as a deformable solid and therefore prevents the elements to invert by construction. It was first applied successfully for curving of high order meshes by Persson and Peraire [17], who formulate a high order FE scheme with a non-linear material law to solve for the inner element deformations. We will adopt the same strategy and describe the details in the following.

We start with a second order system of equations

$$\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{P}(\boldsymbol{\nabla}_{\boldsymbol{x}} \boldsymbol{u}) = 0\,, \tag{13}$$

with the solution vector of the displacement $\boldsymbol{u} = (u_1, u_2, u_3)$ and the second order tensor $\boldsymbol{P} = (P_1, P_2, P_3)^T$ with the components of each vector $P_i = (p_{i1}, p_{i2}, p_{i3})$. We also introduce the undeformed mesh position $\boldsymbol{X}$, the displacement $\boldsymbol{u}$ and the deformed mesh position $\boldsymbol{x}$

$$\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u}\,. \tag{14}$$

Typically, the undeformed mesh position is the initial mesh with linear edges. We assume that we already applied the surface curving. Hence, we choose Dirichlet boundary conditions to impose the displacement as the difference between the linear and the curved surface

$$\boldsymbol{u}|_{\partial\Omega} = \boldsymbol{x}_{\text{curved}}|_{\partial\Omega} - \boldsymbol{X}|_{\partial\Omega}\,. \tag{15}$$

Other non-curved boundaries are clamped by setting $\boldsymbol{u}|_{\partial\Omega} = 0$. The choice of the second order tensor $\boldsymbol{P}$ will result in different models for the mesh deformation. Equation (13) yields the Laplace equation when choosing

$$p_{ij} = \frac{\partial u_j}{\partial X_i}, \quad \Rightarrow \quad \boldsymbol{\Delta}_{\boldsymbol{x}} U = 0. \tag{16}$$

It is the simplest model and the equation is linear.

In non-linear continuum mechanics, the tensor $\boldsymbol{P}$ refers to the first Piola-Kirchhoff stress tensor. The book by Bonet and Wood [6] describes several non-linear material laws, one of them is the compressible neo-Hookean material, which was used in [17] for the mesh deformation. The tensor writes as

$$\boldsymbol{P} = \left(\mu(\mathsf{F} - \mathsf{F}^{-T}) + \lambda J_{\mathsf{F}} \mathsf{F}^{-T}\right), \tag{17}$$

where F denotes the deformation tensor

$$\mathsf{F}_{ij} = \frac{\partial x_j}{\partial X_i} = \delta_{ij} + \frac{\partial u_j}{\partial X_i}, \tag{18}$$

and $J_{\mathsf{F}} = \det(\mathsf{F})$ is the determinant, μ is the shear modulus and λ Lamés first parameter. They are related to Youngs modulus E and the Poisson ratio $0. < \nu < 0.5$ by

$$\mu = E\frac{\nu}{(1+\nu)(1-2\nu)}, \quad \lambda = E\frac{1}{2(1+\nu)}. \tag{19}$$

In fact, the only parameter changing the way the boundary deformation is propagated is the Poisson ratio, since the Youngs modulus is simply a constant factor. Persson and Peraire [17] choose $\nu = 0.4$.

We solve the deformation equations using a high order Continuous Galerkin spectral element method (CG-SEM), see Kopriva [16] for details. The non-linear equation system can be solved via Newtons method, and in each Newton step, we solve the linear equation system with a diagonal preconditioned conjugate gradient method. The Laplace equation is linear and therefore the linear system can be solved directly without Newtons method. In case of a non-linear material, the deformation at the boundary cannot be applied at once but in load increments, so that elements remain valid after the initial boundary deformation. We also recommend a transfinite interpolation of the initial deformation into the first element layer to improve the convergence.

To show the applicability of the proposed method, we apply the mesh deformation to a sequence of boundary layer meshes of the NACA0012 airfoil, a coarse mesh with $n_e = 4$ elements along the chord and two refined levels by doubling the number of elements in each direction ($n_e = 8, 16$ elements). The total number of elements is $n_{e,\text{total}} = 288, 1152, 4608$, but we will refer to the meshes with the number of elements along the chord. The initial linear deformed meshes are depicted in Figure 12. The zoom into the boundary layer shows that the curved surfaces lies within 2-3 boundary layer elements for all mesh levels. The polynomial degree of the curved surface is $N_g = 4$ and it is convenient to

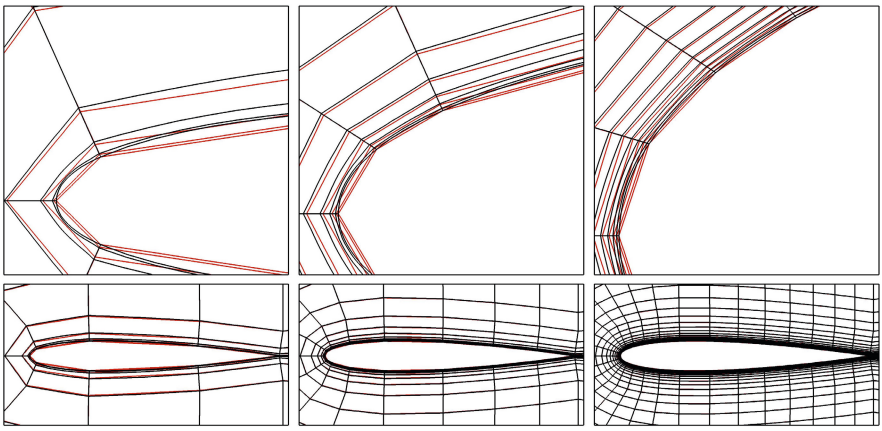


Fig. 12. Curved boundary layer meshes after deformation with Laplace equation and free corners. Initial linear mesh in red

choose the same polynomial degree for the solution of the mesh deformation. The Dirichlet boundary condition for the displacement is applied discretely at each interpolation node, as the vector between the linear and the curved nodes, see Eq. (15).

In Figure 12 , we applied the Laplace equation to deform the mesh, which is very fast, since we only need to solve the linear system once. The deformation is perfectly propagated into the mesh volume. Note that the deformation also pushes the element corners away from the boundary. This is typically not desired, since the resolution at the boundary should be given by the linear mesh. In our solver, we can easily clamp the element corners. There are two advantages, first the solver converges much faster, and the deformation propagates further into the domain yielding a better distribution of the curvature, see Figure 13.

In Figure 14, we show the effect of the material model. The non-linear neo-Hookean material (17) shows a different propagation of the deformation. We

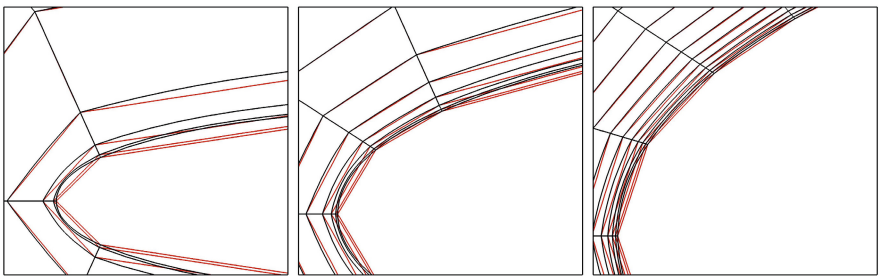


Fig. 13. Curved boundary layer meshes after deformation with Laplace equation and clamped corners. Linear mesh shown in red.

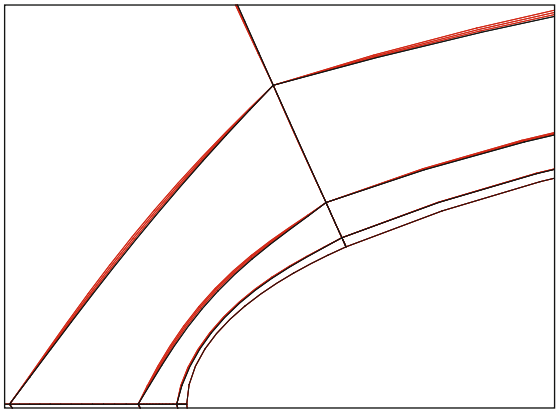


Fig. 14. Comparison of the material for the mesh deformation with clamped corners. The Laplace equation (black lines) and the neo-Hookean material law with three values of $\nu = 0.3, 0.4, 0.45$ (red lines).

can increase the depth of the deformation propagation by increasing the Poisson ratio $\nu \to 0.5$, and we recover the results of the Laplace equation for $\nu \to 0$.

A useful measure for quality and validity of a curved mesh is the normalized or scaled Jacobian, and depends on the variation of the Jacobian $J(\boldsymbol{\xi})$ inside an element Q

$$J_s = \frac{\min_{\boldsymbol{\xi} \in Q} J(\boldsymbol{\xi})}{\max_{\boldsymbol{\xi} \in Q} |J(\boldsymbol{\xi})|} \,. \tag{20}$$

For linear element mappings the Jacobian is constant the scaled Jacobian equals 1. The scaled Jacobians range from -1 to 1, but only elements with $J_{scaled} > 0$ are valid, quality increases with $J_{scaled} \to 1$. Note that the criterion applies to the

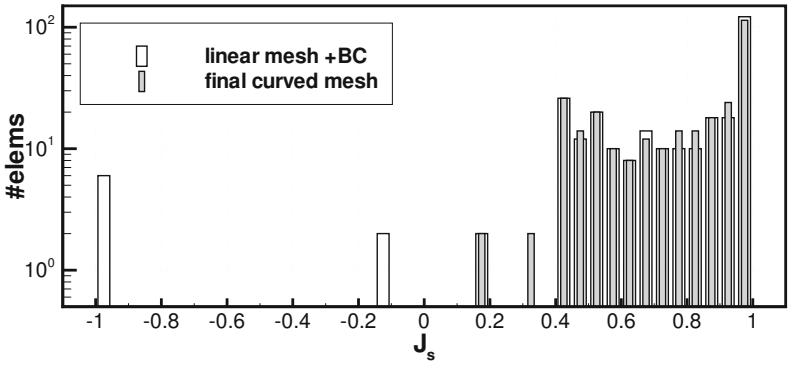


Fig. 15. Distribution of the scaled Jacobian, before and after the mesh deformation with clamped corners, using the Laplace equation

initial linear mesh, too, since Jacobians vary inside quadrangular or hexahedral elements with non-parallel edges.

We plot distribution of the number of elements belonging to a range of the scaled Jacobians (20) in Figure 15 for the coarse mesh $n_e = 4$ and clamped corners. There are 8 elements adjacent to the curved boundary, which are left inverted by the initial transfinite interpolation of the boundary deformation. The application of the mesh deformation pushes the elements back and yields positive Jacobians of the whole mesh. For all meshes and all models, the mesh deformation produced positive Jacobians. We summarized the results for the coarse mesh in Table 1 and look at the difference between the scaled Jacobian distribution of the final curved and the initial linear mesh. The linear elements at the boundary all have $J_s > 0.7$. After the application of the boundary deformation, negative Jacobians are found. The final deformed mesh pushes them back into the range $0.2 < J_s < 0.9$. The quality of the mesh increases for the neo-Hookean material law compared to the Laplace equation. Also a higher Poisson ratio has an overall positive effect and the mesh quality is higher if the element corners are clamped.

Table 1. Difference of the scaled Jacobian distributions between the final curved and the initial linear mesh, for the coarsest mesh $n_e = 4$ and different deformation models

$J_s >$	-1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$J_s \leq$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
linear mesh	0	0	2	0	0	38	30	22	22	28	146
only boundary	**+8**	0	0	0	0	0	0	0	**-2**	0	**-6**
free corners											
Laplace	0	0	0	**+2**	0	**+2**	0	**+2**	0	**+6**	**-12**
neo-H., $\nu = 0.3$	0	0	0	0	**+2**	**+2**	0	0	0	**+8**	**-12**
neo-H., $\nu = 0.4$	0	0	0	0	**+2**	0	**+2**	**-2**	**+2**	**+6**	**-10**
neo-H., $\nu = 0.45$	0	0	0	0	0	**+2**	**+2**	**-2**	-1	+5	**-6**
clamped corners											
Laplace	0	0	0	0	**+2**	**+2**	0	**-2**	**+2**	**+4**	**-8**
neo-H., $\nu = 0.3$	0	0	0	0	**+2**	**+2**	0	**-2**	0	**+4**	**-6**
neo-H., $\nu = 0.4$	0	0	0	0	0	**+2**	**+2**	**-2**	**-2**	**+6**	**-6**
neo-H., $\nu = 0.45$	0	0	0	0	0	0	**+4**	**-2**	**-2**	**+6**	**-6**

In Table 2, the total number of iterations to solve the linear system with the diagonal preconditioned conjugate gradient method is listed for the linear and the non-linear material. In the non-linear case, we need to apply the deformation in load increments and apply Newtons method. We choose 40 load increments for all cases and the Newton solver converges within 3 steps. Hence, the cost for the non-linear material compared to the Laplace equation is highly increased. In addition, the Poisson ratio influences the stiffness of the equation system, and the number of iterations increases for $\nu \to 0.5$. The clamping of the corner nodes

always reduces the number of iterations, and the number of iterations decreases slightly for the finer meshes, since deformations are smaller. The computational cost for each iteration however scales with the number of elements. The CPU time ranges between $1-8$ seconds for the linear and between $5-65$ seconds for the non-linear material. To reduce the cost for a full 3D configuration, only a certain amount of neighbors of the elements at the curved surfaces could be chosen for the computations, leaving the outer elements undeformed.

Table 2. Total number of iterations of the conjugate gradient solver, $||R|| < 1.0E-04$. Non-linear system solved with 40 load increments and ≤ 3 Newton steps per increment.

	$n_e = 4$		$n_e = 8$		$n_e = 16$	
corners:	free	clamped	free	clamped	free	clamped
Laplace	221	51	191	81	104	57
neo-H., $\nu = 0.3$	5015	2374	5287	3389	2740	1995
neo-H., $\nu = 0.4$	6468	2910	6737	4023	3304	2304
neo-H., $\nu = 0.45$	8583	3776	8501	5139	4276	2735

As long as the equation represents a material law, negative Jacobians are avoided by construction if the solution converges, and if the solver fails, the mesh remains invalid. In addition, positivity is not a quality measure, and small scaled Jacobians can lead to inferior solution approximation and smaller time steps. To remedy this issue, the GMSH research group of J.F. Remacle recently published in [18] a very promising multiple objective optimization strategy, where the positivity of the Jacobian and the timestep restriction are expressed as functionals of the curved node positions.

5.2 Agglomeration

A simple idea to produce fully curved elements is super-sampling. Like the surface refinement strategy for the surface curving, one could start from a refined volume mesh and agglomerate the elements to get the coarse DG mesh. However, the refinement process encounters the same problems as the curved mesh generation, since the meshes have to be adapted to curved geometry and inner volume points need to be adjusted. For a general unstructured volume mesh, it is a cumbersome task. In contrast, it is very easy when working with block-structured meshes, the number of points can be simply increased.

Assuming a 3D structured block of size $(I, J, K) = ([1; E_1], [1; E_2], [1; E_3])$ elements, we refine the mesh by the factor N_g. The curved hexahedral mapping of an element I, J, K, already introduced in Eq. (6) is found by interpolation

$$\boldsymbol{X}(\boldsymbol{\xi})^{IJK}_{(\mathrm{Hex})} = \sum_{ijk=0}^{N_g} \hat{\boldsymbol{x}}^{IJK}_{ijk} \ell_i(\xi)\ell_j(\eta)\ell_k(\zeta)\,, \tag{21}$$

where a regular point distribution in reference space is chosen (see Eq. (5)), to guarantee that the mapping remains linear, if the physical node position is regular, too. Given the grid points of the block, the mapping to the interpolation points is simply

$$\hat{\boldsymbol{x}}_{ijk}^{IJK} \mapsto \boldsymbol{x}_{mno}^{\text{block}}, \quad \begin{array}{l} m = (I-1)N_g + i\,,\, m = 0, \ldots, E_1 N_g \\ n = (J-1)N_g + i\,,\, n = 0, \ldots, E_2 N_g \\ o = (K-1)N_g + i\,,\, o = 0, \ldots, E_3 N_g \end{array} . \tag{22}$$

As an example, we show the curved mesh of the IDIHOM periodic hill test case in Figure 16, generated with agglomeration with a polynomial degree of $N_g = 4$, the initial structured mesh is shown in grey.

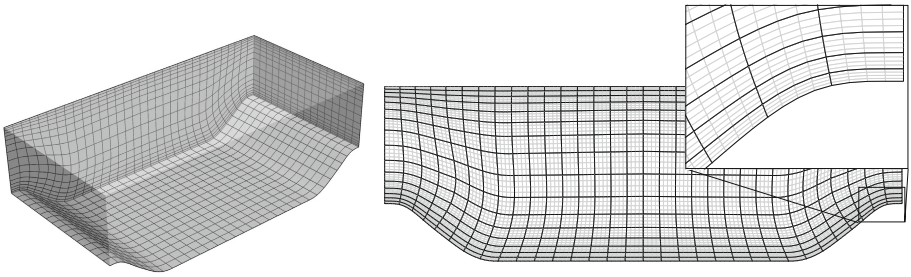


Fig. 16. Curved mesh of the periodic hill from an agglomerated structured grid ($N_g = 4$), 3D view (left) and front view (right) with initial grid lines shown in gray

Note if the mesh refinement is not possible, the structured point data still is a valuable information. From the points of each line of a structured block, we can compute a cubic spline interpolation curve to construct additional interpolation points. Sequentially applied in each structured direction to all lines, a fully curved mesh is generated. Two additional points inside each segment are sufficient to exactly represent the cubic spline segment, which would yield a polynomial degree of $N_g = 3$ for the element mapping.

In structured volume meshing, all blocks are connected, and therefore refinement or coarsening of the mesh is mainly done by stretching the mesh. Also the grid for a boundary layer is typically stretched. If we want to apply the agglomeration approach on a stretched grid, we have to be careful of the strength of the stretching, since element mapping may deteriorate by the interpolation. In [14], we conducted a simple accuracy analysis of the interpolation of a one-dimensional stretched grid with a stretching of a constant growth rate $f = \Delta x_{i+1}/\Delta x_i$ between adjacent elements. A very large growth rate would deteriorate the Jacobian of the interpolated mapping. We found that for polynomial degrees of $N = 2-11$, the maximum growth rate $f^{\max}$ of the fine grid should not exceed 2 for polynomial degrees ranging between $N_g = 3 - 11$. In Table 3 the maximum growth rates are listed for different approximation qualities of the stretched mapping.

Table 3. Maximum growth rate $f^{\max}$ of the fine grid for increasing mapping quality of the scaled Jacobian J_s, normalized by $J_s^{exact} = f^{-N_g}$

$\frac{J_s}{J_s^{\text{exact}}}$	$N_g = 2$	3	4	5	6	7	8	9	10	11
< 0	3.00	4.79	2.62	3.36	2.46	2.92	2.37	2.70	2.31	2.57
$\pm 10\%$	1.75	1.71	1.91	1.93	1.99	2.00	2.02	2.03	2.04	2.04
$\pm 1\%$	1.31	1.30	1.53	1.57	1.67	1.71	1.76	1.79	1.82	1.84

6 Conclusions

In this work, we present the framework which was developed to generate high order meshes. Due to the complexity of the mesh generation process, we rely on existing commercial grid generators which basically produce volume meshes with straight-edged elements. We developed a preprocessing tool filling the gap between the linear mesh and the high order simulation. We described different strategies to create additional information for the surface curving. The first strategy is to construct exact normal vectors at the surface grid points using the CAD model. In particular, sharp edges at the borders of CAD surfaces are automatically detected by our CAD interface. Here, the initial volume grid can be generated by nearly every mesh generator. The second strategy is an interpolation approach, which are however tailored to specific grid generators, using ANSA© for subdivision of the surface mesh or ICEM© for high order edge data. In some situations, like boundary layer meshes, the curving of element faces would result in inverted element mappings, and the surface curving must be propagated into the mesh domain to remove the problem. We confirm the findings of Persson and Peraire [17] that the inverted elements can be eliminated by a mesh deformation approach based on a high order solution of elliptic equations, where the linear mesh is deformed by the curved boundary. The comparison of the Laplace equation and the equation of non-linear solid mechanics reveals that both models produce valid curved meshes. The mesh quality increases for the non-linear material law, especially for high Poisson ratios, but also involves a more expensive non-linear solution procedure compared to the Laplace equation. We recommend to clamp the element corner nodes, since it improves the mesh quality and speeds up the solution process. Finally, the use of block-structured meshes enables the generation of fully three-dimensional curved boundary layer elements by agglomeration of a specific block of elements, leading to a very simple and robust curving technique.

Acknowledgments. This work was partially funded and supported by the IDIHOM project of the FP7 framework of the European Union and the cluster of excellence *Simulation Technology (SimTech), Universität Stuttgart.* We would also like to thank the *Bundesministerium für Bildung und Forschung (BMBF)* for the funding within the program *HPC-software for scalable parallel computers.*

References

1. Ait-Ali-Yahia, D., Robichaud, M.P., Stanescu, D., Habashi, W.G.: Spectral element grid generation and nonlinear computations for noise radiation from aircraft engines. In: 5th AIAA/CEAS Aeroacoustics Conference and Exhibit. AIAA, Bellevue (1999)
2. Bassi, F., Rebay, S.: High-order accurate discontinuous finite element solution of the 2D Euler equations. J. Comput. Phys. 138(2), 251–285 (1997)
3. Bathe, K.-J., Zhang, H.: Finite element developments for general fluid flows with structural interactions. International Journal for Numerical Methods in Engineering 60(1), 213–232 (2004)
4. Blacker, T., Bohnhoff, W., Edwards, T.: CUBIT mesh generation environment. Users manual. SAND94-1100 1 (May 1994)
5. Bolemann, T.: Flächenrekonstruktion mit CAD-basierten Normalenvektoren zur Erstellung von Gittern für Verfahren hoher Ordnung. Diplomarbeit, IAG, Universität Stuttgart (2010)
6. Bonet, J., Wood, R.D.: Nonlinear Continuum Mechanics for Finite Element Analysis. Cambridge University Press (1997)
7. de Boer, A., van der Schoot, M., Bijl, H.: Mesh deformation based on radial basis function interpolation. Computers & Structures 85(11–14), 784 – 795 (2007); Fourth MIT Conference on Computational Fluid and Solid Mechanics
8. Degand, C., Farhat, C.: A three-dimensional torsional spring analogy method for unstructured dynamic meshes. Computers & Structures 80(3–4), 305–316 (2002)
9. Farin, G.: Curves and surfaces for CAGD: a practical guide. San Francisco, CA (2002)
10. Farin, G., Hansford, D.: Discrete Coons patches. Comput. Aided Geom. Des. 16(7), 691–700 (1999)
11. Geuzaine, C., Remacle, J.-F.: Curvilinear mesh generation for CFD. Presentation from (2009)
12. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79, 1309–1331 (2009)
13. Helenbrook, B.T.: Mesh deformation using the biharmonic operator. International Journal for Numerical Methods in Engineering 56(7), 1007–1021 (2003)
14. Hindenlang, F., Gassner, G., Munz, C.-D.: Improving the accuracy of discontinuous galerkin schemes at boundary layers. International Journal for Numerical Methods in Fluids (2014)
15. Karniadakis, G.E., Sherwin, S.J.: Spectral element methods for CFD. Oxford University Press, New York (1999)
16. Kopriva, D.A.: Implementing Spectral Methods for Partial Differential Equations. Springer (2009)
17. Persson, P.-O., Peraire, J.: Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In: Proc. of the 47th AIAA Aerospace Sciences Meeting and Exhibit (2009)
18. Remacle, J.-F., Toulorge, T., Lambrechts, J.: Robust untangling of curvilinear meshes. In: Jiao, X., Weill, J.-C. (eds.) Proceedings of the 21st International Meshing Roundtable, pp. 71–83. Springer, Heidelberg (2013)
19. Stiller, J.: Point-normal interpolation schemes reproducing spheres, cylinders and cones. Computer Aided Geometric Design 4(5), 286–301 (2007)

Multigrid Solver Algorithms for DG Methods and Applications to Aerodynamic Flows

M. Wallraff, R. Hartmann, and T. Leicht

German Aerospace Center, Institute of Aerodynamics and Flow Technology,
Lilienthalplatz 7, 38108 Braunschweig, Germany
{Marcel.Wallraff,Ralf.Hartmann,Tobias.Leicht}@dlr.de

Abstract. In this chapter we collect results obtained within the IDIHOM project on the development of Discontinuous Galerkin (DG) methods and their application to aerodynamic flows. In particular, we present an application of multigrid algorithms to a higher order DG discretization of the Reynolds-averaged Navier–Stokes (RANS) equations in combination with the Spalart-Allmaras as well as the Wilcox-$k\omega$ turbulence model. Based on either lower order discretizations or agglomerated coarse meshes the resulting solver algorithms are characterized as p- or h-multigrid, respectively. Linear and nonlinear multigrid algorithms are applied to IDIHOM test cases, namely the L1T2 high lift configuration and the delta wing of the second Vortex Flow Experiment (VFE-2) with rounded leading edge. All presented algorithms are compared to a strongly implicit single grid solver in terms of number of nonlinear iterations and computing time. Furthermore, higher order DG methods are combined with adaptive mesh refinement, in particular, with residual-based and adjoint-based mesh refinement. These adaptive methods are applied to a subsonic and transonic flow around the VFE-2 delta wing.

Keywords: Discontinuous Galerkin methods, RANS equations, p-multigrid, h-multigrid, adaptivity.

1 Introduction

The Reynolds-averaged Navier–Stokes (RANS) equations in combination with suitable turbulence models such as the $k\omega$ model of Wilcox [30] or the Spalart–Allmaras model (SA) [23,25] can still be considered as state-of-the-art for many applications in exterior aerodynamics. It has been shown that Discontinuous Galerkin (DG) finite element methods are suitable for CFD application in the area of compressible aerodynamic flows [3,6,13,18]. One of the main drawbacks of DG methods is the relatively high computational cost per degree of freedom. High Reynolds numbers, the associated highly stretched meshes typically used for an optimal resolution of turbulent boundary layers, and source terms present in the turbulence models all contribute to an increased stiffness of the resulting algebraic system of equations that has to be solved. Implicit operators are generally accepted as key component for increased efficiency of iterative algorithms.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_9

Several authors suggested strongly implicit schemes that are close to Newton's method [3,13]. It is well-known that these implicit methods work best at the end of the iterative procedure in the regime of asymptotic convergence. Employing a mesh or order sequencing as start-up strategy helps to alleviate this problem by providing a good initial guess. In this work the application of nonlinear h- and p-multigrid algorithms are investigated as well as linear h- and p-multigrid algorithms used as a preconditioner.

Discontinuous Galerkin methods allow high-order flow solutions on unstructured or locally refined meshes by increasing the polynomial degree and using curvilinear instead of straight-sided mesh elements. Depending on the goal of the computation the mesh refinement is driven by either adjoint-based or residual-based indicators. While adjoint-based mesh refinement targets at efficiently and accurately approximating specific integral quantities like the aerodynamic drag or lift coefficients, the residual-based mesh refinement targets at resolving all flow features and is particular well suited for resolving vortical systems.

This chapter is structured as follows: After shortly introducing the RANS equations and the DG discretization in Section 2, Section 3 is devoted to the description of the solver algorithms. In particular, here we introduce the nonlinear h- and p-multigrid solver algorithms, the Backward-Euler iteration used as nonlinear smoother and the linear h- and p-multigrid algorithms used as a preconditioner within a linear Krylov solver. Then, after recalling the adjoint-based and residual-based mesh refinement in Section 4, the main Section 5 of this work is devoted to the application of these algorithms to aerodynamic test cases considered in the EU project IDIHOM. We note, that some of the results presented have already been published previously. In particular, this applies to some of the solver investigations on the VFE-2 delta wing in [27]. In addition, here we present results applying the proposed solver algorithms on third and fourth order RANS-SA solutions on an unstructured mesh around the L1T2 high lift configuration. Furthermore, in Section 5 we collect some results obtained by combining a fourth order DG discretization with residual-based mesh refinement and applying it to a subsonic and transonic flow around the VFE-2 delta wing with rounded leading edge (cf. [11] and [12], respectively). In addition, here we present results of a grid refinement study comparing third order DG solutions on residual-based and adjoint-based refined meshes against third order DG solutions on a hierarchy of nested block-structured meshes as well as against DG solutions of various polynomial degrees (p-refinement) on a purely tetrahedral mesh. Furthermore, the DG solutions are compared to Finite Volume solutions (computed with the DLR-TAU code [19]) on a hierarchy of unstructured mixed-element meshes in terms of accuracy and computational cost. In Section 4 we give some conclusions.

2 DG Discretization of the RANS Equations

In the following, we briefly introduce the DG discretization of the steady-state RANS-$k\omega$ [30,31] and RANS-SA [23,25] equations. These equations can be formulated in conservative form based on convective fluxes $\mathcal{F}^c$ and viscous fluxes

$\mathcal{F}^v$, corresponding to first and second order derivatives, respectively, and additional source terms $\mathcal{S}(\mathbf{u}, \nabla\mathbf{u})$, as follows

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})) = \mathcal{S}(\mathbf{u}, \nabla\mathbf{u}). \tag{1}$$

For the RANS-$k\omega$ equations the vector of conservative variables is given by $\mathbf{u} = (\rho, \rho v_1, \ldots, \rho v_d, \rho E, \rho k, \rho\omega)^T$, where ρ denotes the density, $\mathbf{v} = (v_1, \ldots, v_d)^T$ the velocity vector in spatial dimension d and E the total energy. In this model the eddy viscosity is given by $\mu_t = \frac{\rho k}{\omega}$, where the turbulent kinetic energy k and the specific turbulence dissipation rate ω are determined by two transport equations. As suggested by Bassi *et al.* [3] the transformation $\tilde{\omega} = \ln(\omega)$ is employed. Furthermore, a limitation for $\tilde{\omega}$ based on realizability constraints is applied. Note, that the limitation on $\tilde{\omega}$ avoids unphysical values and has been found in [3, 8, 13] to have a stabilizing effect on the numerical scheme.

For the RANS-SA equations the vector of conservative variables is given by $\mathbf{u} = (\rho, \rho v_1, \ldots, \rho v_d, \rho E, \rho\tilde{\nu})^T$, where the SA working variable $\tilde{\nu}$ is used to determine the eddy viscosity in this one equation turbulence model. We use a modification of the original Spalart–Allmaras turbulence model [25], in particular the negative Spalart–Allmaras turbulence model [23].

The DG discretization of (1) follows standard techniques. For the numerical representation of the viscous fluxes $\mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})$ we employ the second scheme of Bassi and Rebay (BR2) [3,10]. For the numerical representation of the convective flux $\mathcal{F}^c(\mathbf{u})$ the Roe flux [9] is used. Here, all equations are treated fully coupled, i. e., there is a single flux Jacobian used in the eigenvalue decomposition [29]. Furthermore, an entropy fix is used in order to ensure non-vanishing dissipation. The resulting discretization is given as follows: Find $\mathbf{u}_h \in \mathbf{V}_h^p$ such that

$$N_h(\mathbf{u}_h, \mathbf{v}_h) = 0 \qquad \forall \mathbf{v}_h \in \mathbf{V}_h^p, \tag{2}$$

where $\mathbf{V}_h^p$ is the discrete function space of vector-valued piecewise polynomial functions of degree p and $N_h(\cdot,\cdot) : \mathbf{V}_h^p \times \mathbf{V}_h^p \to \mathbb{R}$ is a semilinear form given in detail in [15]. Furthermore, $\mathbf{V}_h^p$ is based on an ortho-normal non-parametric formulation of the basis functions directly in the physical space. This is equivalent to the choice of a Taylor basis [20]. This way, the local basis can be defined independently from a specific reference element, which is necessary for the usage of agglomerated meshes in an h-multigrid. The choice of this space in connection with the formulation of a DG method on agglomerated meshes has been described in detail in [1, 14]. By choosing an ortho-normal basis for the DG discretization the mass-matrix is the identity matrix on each mesh element and globally over the whole mesh.

3 Solver Algorithms

In the following, we will present several strategies to exploit hierarchies of coarse level problems in solver algorithms, including both level sequencing and linear as well as nonlinear multigrid variations [26]. Given a hierarchy of discrete function

spaces $\mathbf{V}_{l_{\min}} \subset \ldots \subset \mathbf{V}_l \subset \ldots \subset \mathbf{V}_{l_{\max}} \equiv \mathbf{V}_h^p$, where $\mathbf{V}_l$ for $l < l_{\max}$ represents either an agglomerated coarse mesh or a lower order discretization the resulting algorithms can be characterized as h- or p-multigrid, respectively. The only difference between these multigrid algorithms is the use of different coarse level DG discretizations and, therefore, transfer operators. All other ingredients like smoothers, timestep control, usage of a Galerkin-transfer [26], start-up strategy, etc. are the same for both kinds of multigrid algorithms. The restriction transfer operator for both h- and p-multigrid algorithms of the nonlinear state vector is given by an orthogonal L^2-projection $\hat{I}_l^{l-1} : \mathbf{V}_l \to \mathbf{V}_{l-1}$ [4]. Furthermore, the prolongation transfer operator is obtained via a natural injection $I_{l-1}^l : \mathbf{V}_{l-1} \to \mathbf{V}_l$ and a restriction operator for the defect between the various approximation levels is defined by $I_l^{l-1} := (I_{l-1}^l)^T$. The Galerkin-transfer is defined as $\underline{\mathrm{R}}_{l-1} = I_l^{l-1} \underline{\mathrm{R}}_l I_{l-1}^l$ for a given matrix $\underline{\mathrm{R}}_l$ which is a linear map in $\mathbf{V}_l$. Moreover the resulting matrix $\underline{\mathrm{R}}_{l-1}$ is a linear map in $\mathbf{V}_{l-1}$. The actual form of the block matrices defining the transfer operators depend on the choise of basis functions. They take a particularly simple form if ortho-normal hierarchic basis functions are used on each level. For instance, for a p-multigrid the prolongation operator has the form of an identity matrix filled with additional zero rows at the bottom.

Algorithm 1. Nonlinear Multigrid Algorithm $NMG(\mathbf{f}_l, \mathbf{u}_l, m_1, m_2, l, l_{\min}, \tau)$

if $l = l_{\min}$ **then**
  Solve $\mathbf{L}_{l_{\min}}(\mathbf{u}_{l_{\min}}) = \mathbf{f}_{l_{\min}}$
  Set Solution to $\tilde{\mathbf{u}}_{l_{\min}}$.
end if
if $l > l_{\min}$ **then**
  for $i = 1$ to m_1 **do**
    $\mathbf{u}_l := GL_l^s(\mathbf{u}_l, \mathbf{f}_l)$ /* pre-smoothing */
  end for
  $\mathbf{u}_{l-1}^0 := \hat{I}_l^{l-1} \mathbf{u}_l$
  $\mathbf{f}_{l-1} := \mathbf{f}_{l-1} + I_l^{l-1} \left(\mathbf{f}_l - \mathbf{L}_l(\mathbf{u}_l)\right) - \left(\mathbf{f}_{l-1} - \mathbf{L}_{l-1}(\mathbf{u}_{l-1}^0)\right)$
  for $k = 1$ to τ **do**
    $\mathbf{u}_{l-1}^k := NMG(\mathbf{f}_{l-1}, \mathbf{u}_{l-1}^{k-1}, m_1, m_2, l-1, l_{\min}, \tau)$
  end for
  $\tilde{\mathbf{u}}_l := \mathbf{u}_l + I_{l-1}^l \left(\mathbf{u}_{l-1}^\tau - \mathbf{u}_{l-1}^0\right)$
  for $i = 1$ to m_2 **do**
    $\tilde{\mathbf{u}}_l := GL_l^s(\tilde{\mathbf{u}}_l, \mathbf{f}_l)$ /* post-smoothing */
  end for
end if
return $\tilde{\mathbf{u}}_l$

The nonlinear multigrid (Algorithm 1) is defined recursively. For $\tau = 1$ a V-cycle is performed, whereas $\tau = 2$ gives a W-cycle. The nonlinear problem on

every level is defined as $\mathbf{L}_l(\mathbf{u}_l) = \mathbf{f}_l$. For the nonlinear smoother $GL_l^s(\mathbf{u}_l, \mathbf{f}_l)$ we employ as underlying relaxation scheme a linearized Backward-Euler approach based on local pseudo-time steps that can be considered as a stabilized Newton's method and is also used predominantly as single-level solver [3, 13]. In Algorithm 2, $\underline{\mathrm{R}}_l := \frac{\partial \mathbf{L}_l(\mathbf{u}_l)}{\partial \mathbf{u}_l}$ is the fully implicit Jacobian matrix and $\underline{\mathrm{M}}$ is the mass

Algorithm 2. Backward–Euler $BWE(\mathbf{u}_{l,i-1})$

1: Solve $\left[(\alpha_i \Delta t)^{-1}\underline{\mathrm{M}} + \underline{\mathrm{R}}_l\right](\mathbf{u}_{l,i} - \mathbf{u}_{l,i-1}) = \left[\mathbf{f}_l - \mathbf{L}_l(\mathbf{u}_{l,i-1})\right]$,
2: **return** $\mathbf{u}_{l,i}$

matrix. In addition to that $\mathbf{u}_{l,j}$ is a given state vector, with $\mathbf{u}_{l,j} \in \mathbf{V}_l \; \forall \; j \in \mathbb{N}$. This method is controlled by a pseudo-time stepping scheme $(\alpha_i \Delta t)$, based on local time steps computed from the local state and a given CFL number. The pseudo-time step acts as a stabilizing mechanism and for CFL $\to \infty$ the Newton algorithm is recovered. Smaller time steps and accordingly CFL numbers are required in the initial phase of the solution process when the current iterate solution approximation is still too far from the converged solution. A *switched evolution relaxation* (SER) [22] technique is employed to modify the CFL number during the solution process, enabling to recover the optimal behavior of the Newton algorithm in the final phase of the solution process.

The resulting linear system is solved with a Krylov method, due to the non-symmetric Jacobian GMRes is used. This method is preconditioned either by a line-Jacobi [29] iterative scheme or a linear multigrid using the line-Jacobi scheme as a smoother. This line-Jacobi scheme (Algorithm 3) is stated as follows: let $\mathcal{L}_{l,k}(\mathfrak{u}_{l,k}) = \mathfrak{f}_{l,k}$ be the underlying linear problem on line k, where $\mathrm{R}_{l,k}^{-1}$ is the

Algorithm 3. Line–Jacobi scheme $LJ(\mathfrak{u}_{l,k,0}, n)$

1: **for** $i = 1$ to n **do**
2: $\quad \delta\mathfrak{u}_{l,k,i} := \mathfrak{u}_{l,k,i} - \mathfrak{u}_{l,k,i-1} = \mathrm{R}_{l,k}^{-1}(\mathfrak{f}_{l,k} - \mathcal{L}_{l,k}\mathfrak{u}_{l,k,i-1})$
3: $\quad \mathfrak{u}_{l,k,i} := \mathfrak{u}_{l,k,i-1} + \delta\mathfrak{u}_{l,k,i}$
4: **end for**
5: **return** $\mathfrak{u}_{l,k,n}$

inverse of the Jacobian matrix computed one line k in the mesh. We consider lines of mesh elements, i. e., disjoint subsets of elements. In order to find suitable lines a scalar advection-diffusion problem is defined and examined [10]. The inversion on a line based system matrix which is a block-tridiagonal system can be performed exactly.

The following choice of solver parameters is used in the numerical examples in Section 5. For a nonlinear multigrid V-cycles are used with one pre- and

post-smoothing step. On the lowest level one smoother step is done as well. The smoother is a Backward-Euler with SER timestep control [22]. A linear multigrid with four post-smoothing steps and no pre-smoothing steps is applied as preconditioner in the GMRes linear solver. In the linear multigrid a line-Jacobi scheme is used as a smoother. If no linear multigrid is used then four line-Jacobi iterations are performed as a preconditioner. These choices result from previous investigations [28]. For all computations a start-up strategy is applied. The basic idea is to start the computation on the lowest level and transfer the resulting solution to the next higher level to provide a suitable initial solution. On every sublevel the start-up strategy produces converged solutions according to a prescribed residual reduction tolerance. A p-start-up strategy generates solutions on the same computational mesh for all discretizations from $p_{\min}$ to $p_{\max}$. Here, $p_{\min} = 1$ is employed for the numerical examples. We do not consider $p = 0$ discretizations, as they do not represent a consistent first order discretization in the presence of viscous fluxes and source terms. The first order discretization could be modified, but this is not done here. Thus, second order ($p = 1$) is the lowest level considered in this work. For an h-start-up strategy the initial fine mesh is coarsened in a way such that the resulting coarse meshes build a hierarchy of nested meshes. The h-start-up strategy computes a solution on the coarsest mesh and uses this as an initial solution for the next mesh in the hierarchy until the algorithm gives a solution for the initial fine mesh. In every h-start-up strategy a p-start-up algorithm is performed on the coarsest mesh.

For the h-version of the considered multigrid method and start-up strategy an agglomeration of the underlying fine mesh is performed. A coarse level is constructed by connecting a certain number of fine mesh elements to form a single agglomerate. No simplification of the resulting geometry is performed, but faces between elements that are part of the same agglomerate can be ignored on coarse levels. Coarser levels are obtained by recursive application of the described procedure, i. e., the second agglomeration is performed based on the first level of agglomerates. Thus, a hierarchy of nested meshes is created which is appropriate for the transfer operators. For the numerical examples, the agglomeration is based on the `MGridGen` software, which employs graph-based algorithms to optimize the ratio of the agglomerate surface area to the agglomerate volume [21], i. e., it aims at isotropic coarse meshes. An example for the agglomeration can be seen in Fig. 1, which visualizes the mesh and agglomeration levels used in all h-multigrid DG computations in Section 5.1.

4 Adaptive Mesh Refinement Algorithms

Important quantities in aerodynamic flow simulations are the aerodynamic force coefficients like the pressure induced as well as the viscous stress induced drag, lift and moment coefficients. Local mesh refinement is a key ingredient for obtaining accurate results at reasonable cost in applications. The goal of the adaptive refinement is either to compute these quantities as accurate as possible or to resolve the overall flow field. Depending on this choice, either adjoint-based or

residual-based indicators are used for identifying the mesh elements to be refined. In the following, we present the associated mesh refinement algorithms.

Given a target quantity $J_h(\cdot)$ like an aerodynamic force coefficient and the discretization (2) of the governing equations (1), the discretization error in the computed target quantity $J_h(\mathbf{u}_h)$, i.e. the difference of the exact (and unknown) value, $J_h(\mathbf{u})$, and the computed one, $J_h(\mathbf{u}_h)$, can be approximated as follows

$$J(\mathbf{u}) - J_h(\mathbf{u}_h) \approx R_h(\mathbf{u}_h, \bar{\mathbf{z}}_h), \tag{3}$$

where $R_h(\mathbf{u}_h, \mathbf{z}) := -N_h(\mathbf{u}_h, \mathbf{z})$ includes the primal residuals multiplied by the discrete adjoint solution $\bar{\mathbf{z}}_h$ which is the solution to following discrete adjoint problem: Find $\bar{\mathbf{z}}_h \in \bar{\mathbf{V}}_h^p$ such that

$$N_h'[\mathbf{u}_h](\mathbf{w}_h, \bar{\mathbf{z}}_h) = J_h'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \bar{\mathbf{V}}_h^p. \tag{4}$$

Here, $N_h'[\mathbf{u}_h]$ and $J_h'[\mathbf{u}_h]$ are the derivatives of N_h and J_h with respect to $\mathbf{u}$ evaluated at the discrete solution $\mathbf{u}_h$. Note, that the discrete adjoint solution is computed in a richer space $\bar{\mathbf{V}}_h^p \equiv \mathbf{V}_h^{p+1}$ than the flow solution in order to avoid a vanishing error estimate (3). This estimate can be localized

$$J(\mathbf{u}) - J_h(\mathbf{u}_h) \approx R_h(\mathbf{u}_h, \bar{\mathbf{z}}_h) \equiv \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa, \tag{5}$$

where $|\bar{\eta}_\kappa|$ are the so-called adjoint-based indicators which include the primal local residuals weighted with the discrete adjoint solution. The associated adjoint-based mesh refinement algorithm (cf. Algorithm 4) is tailored to the accurate and efficient approximation of the quantity $J_h(\mathbf{u})$ of interest.

Algorithm 4. Adjoint-based mesh refinement algorithm

1: Construct an initial mesh $\mathcal{T}_h$.
2: Compute $\mathbf{u}_h \in \mathbf{V}_h^p$, see (2), on the current mesh $\mathcal{T}_h$.
3: Compute $\bar{\mathbf{z}}_h \in \bar{\mathbf{V}}_h^p = \mathbf{V}_h^{p+1}$, see (4), on the same mesh employed for $\mathbf{u}_h$.
4: Evaluate the approximate error representation $R_h(\mathbf{u}_h, \bar{\mathbf{z}}_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$.
5: **if** $|\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa| \leq$ `TOL` **then**
6: **stop**.
7: **else**
8: Refine and coarsen a fixed fraction of the total number of elements according to the size of $|\bar{\eta}_\kappa|$ and generate a new mesh $\mathcal{T}_h$; **go to** 2.
9: **end if**

Provided the adjoint solution related to an arbitrary target quantity is sufficiently smooth the corresponding error representation can be bounded from above by an error estimate as follows (cf. [15] for details)

$$|J(\mathbf{u}) - J_h(\mathbf{u}_h)| \leq C \left(\sum_{\kappa \in \mathcal{T}_h} \left(\eta_\kappa^{(\mathrm{res})} \right)^2 \right)^{1/2}, \tag{6}$$

where the so-called residual-based indicators $|\eta_\kappa^{(\mathrm{res})}|$ depend on the primal local residuals but are independent of the adjoint problem. Being independent of a specific target quantity the associated residual-based mesh refinement algorithm (cf. Algorithm 5) targets at resolving all flow features. Numerical experiments

Algorithm 5. Residual-based mesh refinement algorithm

1: Construct an initial mesh $\mathcal{T}_h$.
2: **for** $i = 1$ to #steps **do**
3: Compute $\mathbf{u}_h \in \mathbf{V}_h^p$, see (2), on the current mesh $\mathcal{T}_h$.
4: Evaluate the residual-based indicators $|\eta_\kappa^{(\mathrm{res})}|$ for all $\kappa \in \mathcal{T}_h$
5: Refine and coarsen a fixed fraction of the total number of elements according to the size of $|\bar{\eta}_\kappa|$ and generate a new mesh $\mathcal{T}_h$
6: **end for**

(cf. Section 5.2) show that it is particularly well suited for the resolution of vortical systems.

5 Numerical Results

In this section, we apply the solver algorithms and the mesh refinement algorithms described in Sections 3 and 4 to the IDIHOM test cases U.3 and U.1.

5.1 U.3: Subsonic Turbulent Flow around L1T2 High Lift Configuration

First we consider a subsonic fully turbulent flow around the L1T2 three element high lift configuration at an angle of attack of 20.18°, a Mach number of 0.197 and a Reynolds number of 3.52×10^6 [9]. Fig. 1 shows an unstructured mesh (courtesy of Jerzy Majewski and Piotr Szałtys, Warsaw University of Technology, WUT) with 25 757 curvilinear elements (quartic lines) together with two levels of agglomeration. This mesh is a coarse high order version of an original straight-sided mesh that has been created via anisotropic adaption to the solution of the RANS equations with the Spalart-Allmaras turbulence model using a second order Residual Distribution scheme. Therefore, in the following, we show Discontinuous Galerkin flow solutions for the same turbulence model. As an example, Fig. 2 shows the Mach number distribution of a third order DG solution on this mesh.

Within IDIHOM, reference solutions have been provided (courtesy of Axel Schwöppe, DLR) with the DLR-TAU code [19], a second order unstructured Finite Volume solver used in industry (e. g., by Airbus). Computations for the RANS-SA model have been performed on a hierarchy of four structured meshes from 4 268 elements up to 273 152 elements. Fig. 3 shows the second mesh in this

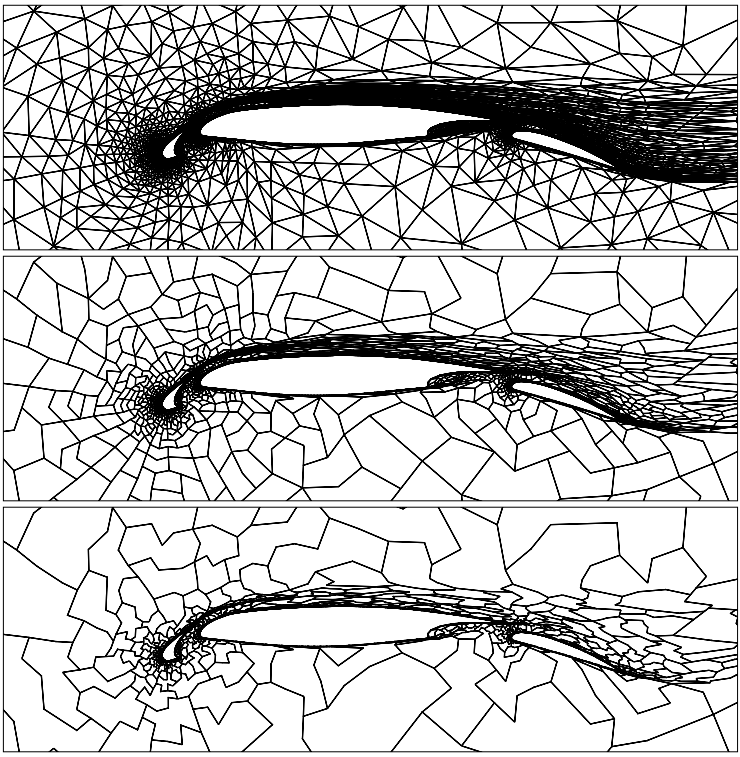


Fig. 1. Unstructured mesh from WUT with 25 757 curvilinear elements (quartic lines) around the L1T2 high lift configuration and two agglomeration levels

Fig. 2. L1T2 high lift configuration: Mach number distribution of a third order DG solution on the WUT mesh shown in Fig. 1

hierarchy with 17 072 elements. Extrapolation of the computed force coefficients according to the IDIHOM evaluation procedure yields asymptotic values of 4.051 for the lift and 0.0608 for the drag coefficient.

For the third order DG solution on the WUT mesh the lift and drag coefficients are 4.035 and 0.0612, respectively. These values are within the prescribed

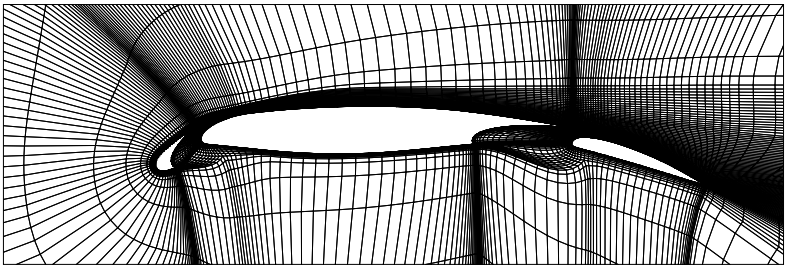


Fig. 3. L1T2 high lift configuration: second mesh in the multi-block structured mesh hierarchy with 17 072 elements

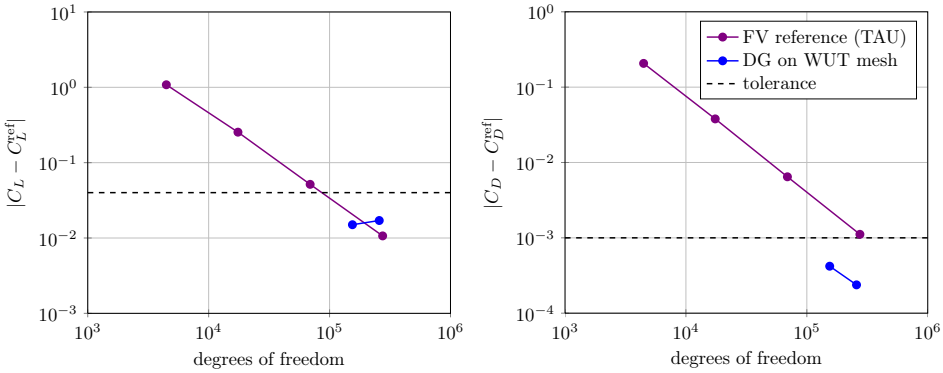


Fig. 4. L1T2 high lift configuration: Error in lift (left) and drag coefficients (right) for the reference computations on a hierarchy of structured meshes in comparison to DG results (of third and fourth order) on the unstructured mesh from WUT

tolerance band of the reference values, see Fig. 4. Here, the asymptotic values of the Finite Volume computations have been used for computing the errors in the DG case, as no mesh convergence study could be done on the adapted mesh. The actual asymptotic values might be slightly off due to subtly differences in the airfoil geometry approximation as well as the location and treatment of farfield boundaries. The two dots for the DG results in Fig. 4 represent third (p=2) and fourth (p=3) order solutions on the unstructured mesh from WUT.

Finally, Fig. 5 shows the computed pressure coefficient (c_p) and skin friction coefficient (c_f) distributions of the third order DG solution in comparison to that of the FV reference solution. We see a very good agreement. In fact, the only notable discrepancy is a pressure disturbance at the lower edge of the flap cove for the third order computation which might be due to the mesh topology in that region.

The DG results shown above can be obtained using various solver choices. For h-multigrid algorithms 3 levels are employed whereas for p-multigrid algorithms 2 or 3 levels are employed, depending on the desired polynomial degree $p_{\max} = 2$

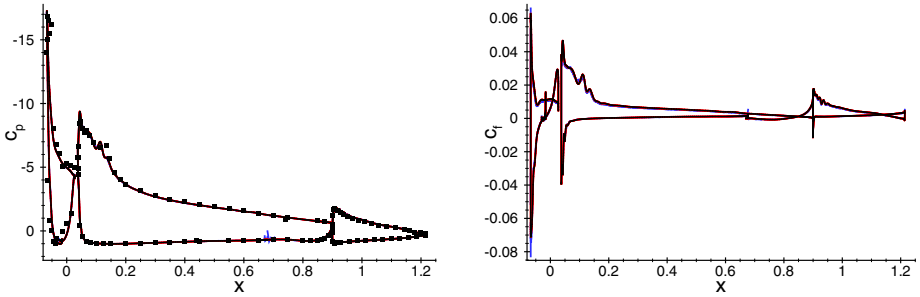


Fig. 5. L1T2 high lift configuration: Comparison of the FV reference solution (red) and the third order DG solution (blue) in terms of computed pressure coefficient (left) and skin friction coefficient (right). Symbols indicate experimental data.

or $p_{\max} = 3$. Found as best practice in [28], the Galerkin-transfer operator is employed in all multigrid computations. We note that the algorithmic development presented in this work is targeted at solving the RANS-$k\omega$ equations, which – according to our experience – is a more demanding task than for the RANS-SA case. Still, the immediate applicability to the RANS-SA case is a desirable property of the proposed algorithms. However, further experiments not reported here indicate that the use of a startup strategy is not as important for RANS-SA as in the RANS-$k\omega$ case. Thus, all multigrid variants use a single cycle only on coarser levels before proceeding to finer levels in the startup phase.

The convergence history of the nonlinear residual for the different solver choices is shown over the number of nonlinear iterations on the finest level and over the normalized CPU time in Fig. 6. All computations shown in Fig. 6 yield a $p = 2$ solution on the unstructured mesh from WUT with 25 757 elements.

The lines in Fig. 6 marked with a $+$ represent computations where a linear multigrid algorithm is applied. The lines marked with a $\times$ indicate the use of a nonlinear multigrid algorithm with a Backward-Euler as smoother and a line-Jacobi preconditioned GMRes method. Finally, the lines marked with a $\bullet$ represent the combination of nonlinear and linear multigrid algorithms. Note, that the cost per fine level iteration varies with the solver choice. The additional work associated to (linear or nonlinear) coarse level iterations is included in the CPU time in Fig. 6. Furthermore, for all computations in Fig. 6 the same initial CFL number and overall time step control was chosen for comparability reasons. Only for the single-level Backward–Euler solver (indicated with the solid unmarked line $-$) the initial CFL number had to be reduced in order to obtain a converged solution.

The dashed lines in Fig. 6 represent the p-multigrid algorithms. We see, that the linear p-multigrid algorithm (marked with $+$) and the nonlinear p-multigrid algorithm ($\times$) show good results in terms of number of nonlinear iterations and CPU time in comparison to a single-level Backward–Euler solver. In this case the combination of both p-multigrid approaches ($\bullet$) gives a further improvement in the number of iterations, but the nonlinear multigrid alone is faster in terms of

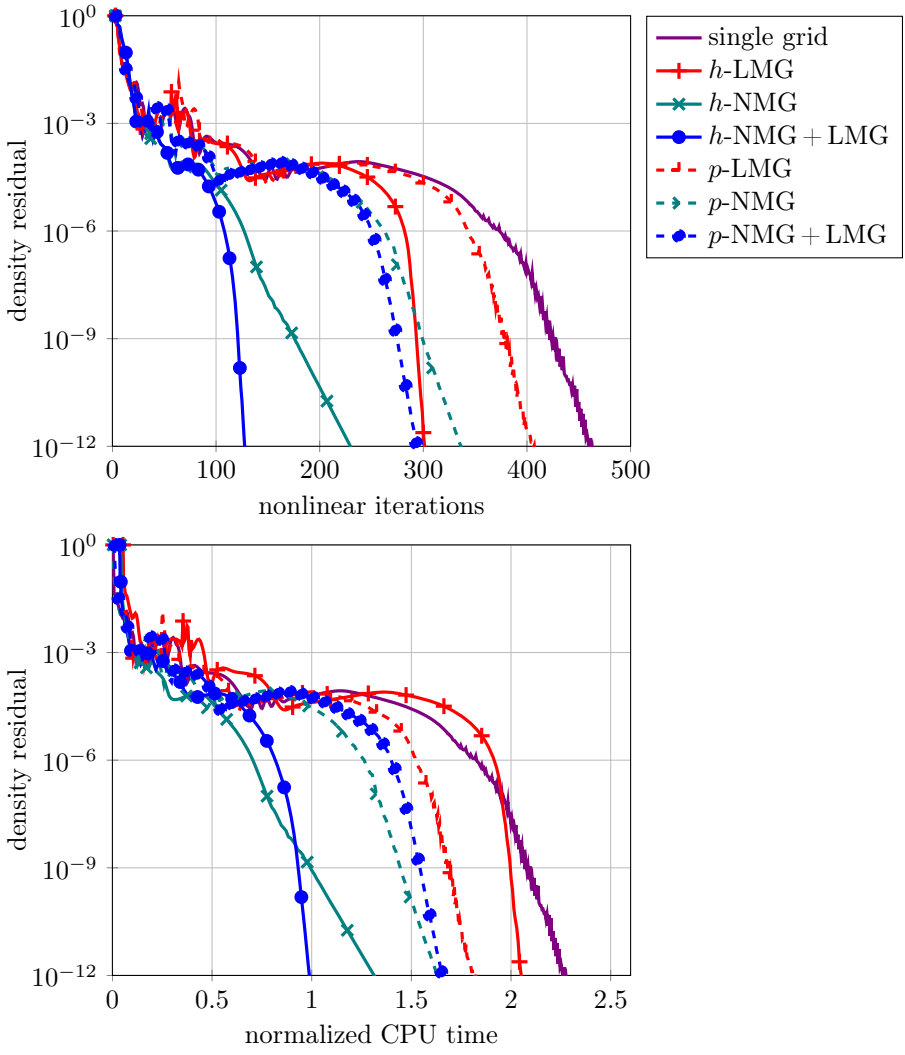


Fig. 6. L1T2 high lift configuration: Nonlinear iteration (top) and run time (bottom) comparison for the RANS-SA equations (p=2)

CPU time. This is in contrast to previous results in [27,28] where the combination of both multigrid algorithms always gave the best results in terms of nonlinear iterations *and* CPU time.

The solid marked lines in Fig. 6 represent the h-multigrid algorithms. First of all, we see that all h-multigrid variants improve the solution process in comparison to the single-grid Backward–Euler solver. Furthermore, the nonlinear h-multigrid computations (marked with $\times$), and the combination of nonlinear and linear h-multigrid algorithms ($\bullet$) are better in terms of nonlinear convergence and overall CPU time than their respective p counterpart. While the nonlinear multigrid algorithms are beneficial at the beginning of the computation on every level, the linear multigrid algorithms are superior at the end of the iterative procedure in the regime of asymptotic convergence, where they yield an improved rate of convergence. Therefore, the combination of a nonlinear and linear multigrid accomplishes the biggest reduction in nonlinear iterations for both the p- and h-multigrid computations. Finally, the combination of nonlinear and linear h-multigrid algorithms is superior in terms of normalized CPU time compared to all other proposed algorithms.

After this detailed analyis for $p_{\max} = 2$, Fig. 7 shows some of the proposed solver choices for a $p_{\max} = 3$ solution of this test case. Again, solver settings like the initial CFL number and the overall time step control are chosen identical for comparability reasons. In contrast to the $p_{\max} = 2$ computations, here the p-multigrid algorithms employ 3 levels just like the h-multigrid algorithms. The results indicate that the convergence rate in the regime of asymptotic convergence using the linear multigrid algorithm (lines marked with $\bullet$ and $+$) is better than without linear multigrid algorithm. In terms of computational time the linear h-multigrid alone is too costly in comparison to the single grid computation. Hence, the reduction in nonlinear iterations is not big enough to also achieve a reduction in CPU time.

The computations where a nonlinear multigrid algorithm is applied (lines marked with $\bullet$ and $\times$) seem to reduce the time after which the regime of asymptotic convergence is reached in comparison to the computations without nonlinear multigrid. Again, the combination of both multigrid variants can benefit from the improvements in both the initial and the asymptotic phase of convergence. This is particularly true for the h-multigrid case. Overall the solver behavior in the fourth order computations (cf. Fig. 7) is very similar to that in the third order computations (cf. Fig. 6).

This behavior of the proposed solver algorithms is typical for many test cases and will further be investigated in the next section. Similar results can be found in [27,28].

5.2 U.1: Turbulent Flow around the VFE-2 Delta Wing

In the following, we consider the 65° swept delta wing configuration with medium rounded leading edge of the second Vortex Flow Experiment [16]. The geometry of the VFE-2 configuration is shown in Fig. 8. We note, that the sting has been extended in the geometry model to about 1.5 chord lengths behind the wing and has a rounded trailing edge.

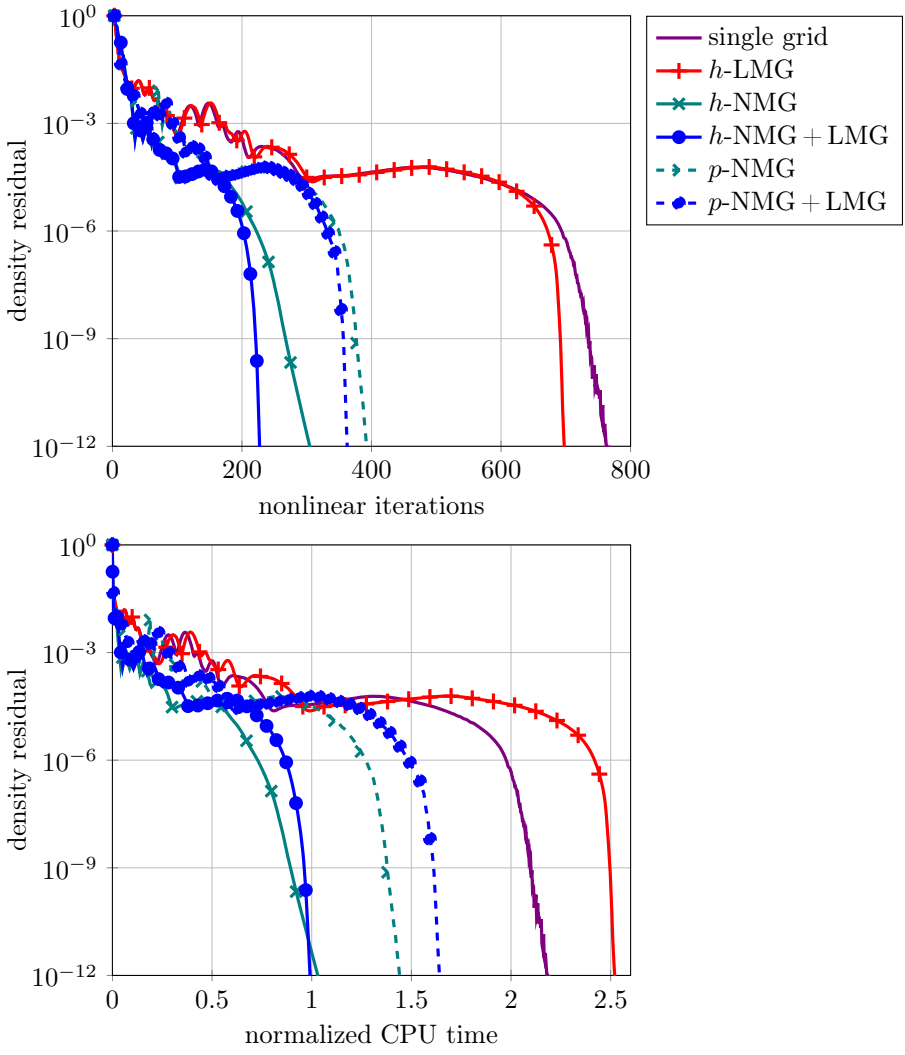


Fig. 7. L1T2 high lift configuration: Nonlinear iteration (top) and run time (bottom) comparison for the RANS-SA equations (p=3)

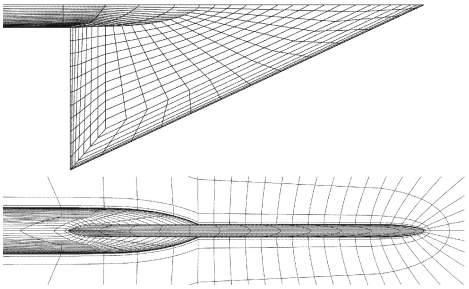
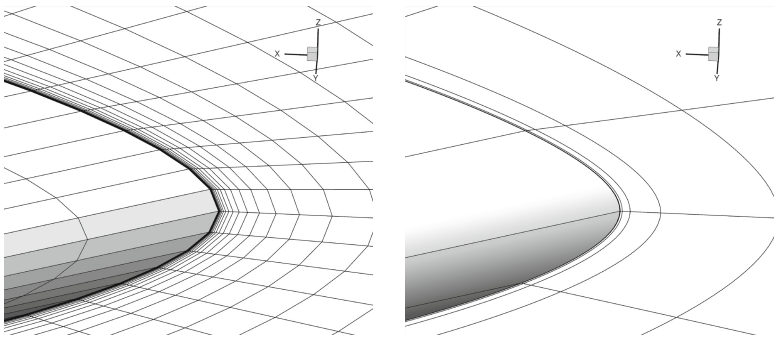


Fig. 8. VFE-2 delta wing: Top and side view of the 65° swept delta wing with medium rounded leading edge

An original multi-block structured mesh with 884 224 hexahedral elements (courtesy of Simone Crippa [7]) has been agglomerated twice resulting in a coarse mesh of 13 816 elements. The additional points of the original mesh have been used to define 13 816 curvilinear elements, where the curved lines are represented by polynomials of degree 4 and interpolate the original points. The surface mesh of the coarse mesh with curvilinear elements is depicted on the wing and the symmetry plane in Fig. 8. Figures 9(a) and (b) show the discrete representation of the rounded leading edge close to the symmetry plane for the original straight-sided mesh and the coarse curvilinear mesh, respectively. The original mesh shows clear kinks in the approximation of the leading edge. In contrast to that, the coarse mesh with curvilinear elements is smooth and approximates the curved boundary very well. Furthermore we note that in order to avoid an intersection of curved boundary faces with interior faces also interior faces in the coarse mesh are curved.



(a) (b)

Fig. 9. VFE-2 delta wing: Zoom of the rounded leading edge close to the symmetry plane for (a) the original straight-sided mesh, and (b) the coarse curvilinear mesh

U.1b: Subsonic Flow around the VFE-2 Configuration. In the following, we consider a subsonic fully turbulent flow at a Mach number $M = 0.4$, a

Reynolds number $Re = 3 \times 10^6$ and an angle of attack of $\alpha = 13.3°$ around the VFE-2 configuration. While typically only one primary vortex is encountered on flows around delta wings with sharp leading edges, the delta wing with a rounded leading edge considered here creates two primary vortices which makes the flow field particularly interesting and challenging for numerical simulation.

Solver algorithms The following results based on the RANS and $k\omega$-turbulence model equations represent the first application of the solver algorithms described in Section 3 in 3D and have previously been published in [27]. They do not constitute an exhaustive investigation. Instead, they are used to identify areas of future research for the extension of the algorithms from 2D to 3D.

Computations will be shown on the structured mesh with 13 816 curvilinear elements described above and on a once globally refined mesh with 110 528 curvilinear elements. In the following, a third order accurate solution ($p = 2$) is desired for all computations. Moreover, all multigrid algorithms use only two levels.

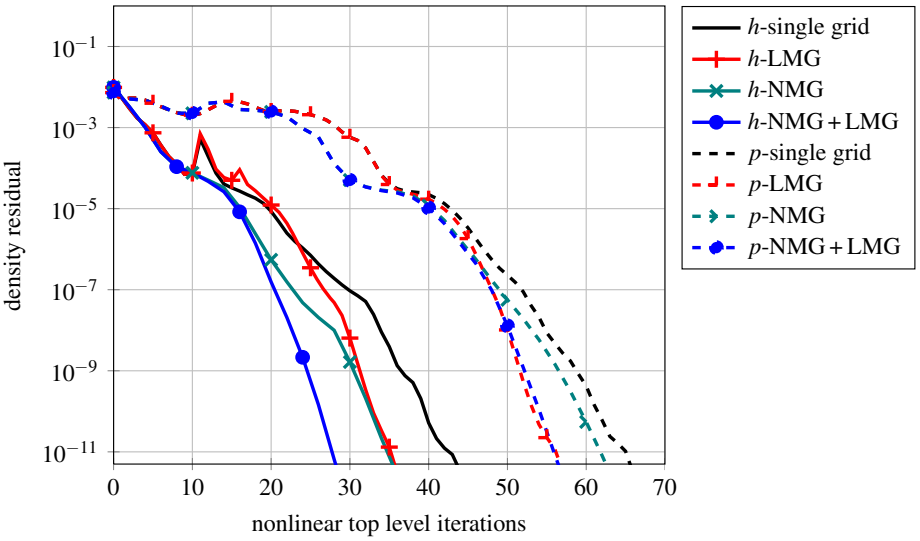


Fig. 10. VFE-2 delta wing: Computation of a $p = 2$ solution on the mesh with 13 816 elements using different multigrid algorithms [27]

Fig. 10 shows the nonlinear Backward-Euler iterations on the highest level of the start-up strategies. The dashed lines in Fig. 10 are initialized with a $p = 1$ solution on the same mesh to compute a $p = 2$ solution. All dashed line computations use a p-start-up strategy with the same lower level computation for $p = 1$. For all solid lines the h-start-up strategy is the same as well. Since we use two level multigrid algorithms the initial solution for the solid lines in Fig. 10 is indeed the same. This results from the fact that on the agglomerated grid a

p-start-up strategy is applied and after convergence of the $p = 2$ solution on the agglomerated grid it is used as an initial solution for the $p = 2$ computations on the mesh with 13 816 elements, as mentioned at the end of Section 3.

For comparability the timestep control for all computations shown in Fig. 10 is the same, including the same initial CFL number on the top level of the start-up strategies. For the solid and dashed unmarked lines a single-level Backward–Euler solver is applied. Thus, the only difference between the solid and dashed unmarked lines is the initial solution given on this level. In general our findings are that an h-start-up is more favorable than a p-start-up, like seen in this case for a single level solver.

Note, that the algorithms shown in Figures 10, 11 and 12 are indicated with the same markings as in Section 5.1. Moreover the results in Fig. 10 are in line with previous findings in Section 5.1. In addition to that the nonlinear p-multigrid alone gives a small improvement over the p-single grid computation, only. This supports our findings from [29].

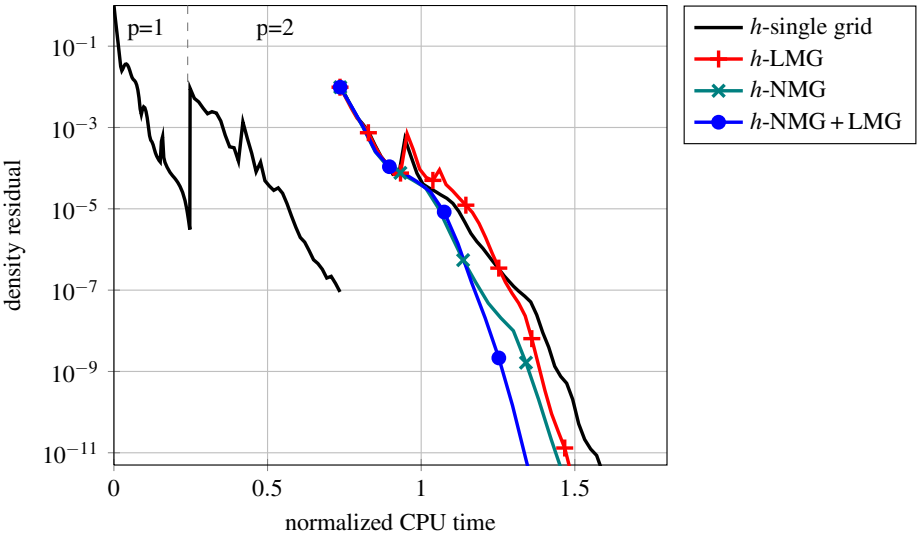


Fig. 11. VFE-2 delta wing: Computation of a $p = 2$ solution on the mesh with 13 816 elements with an h-start-up strategy [27]

Figures 11 and 12, which show the computing times of the computations, indicate that the better algorithmic convergence carries over to an improved runtime behavior.

In Fig. 11, most of the time is spent on the agglomerated mesh and thus on a coarse level. In contrast to that, in Fig. 12 most of the time is used for the $p = 2$ computation on the mesh with 13 816 elements and thus on the fine level. Even with the increased time spent on the coarser level, the overall h-multigrid $p = 2$ computations are slightly faster than their p counterparts with the same

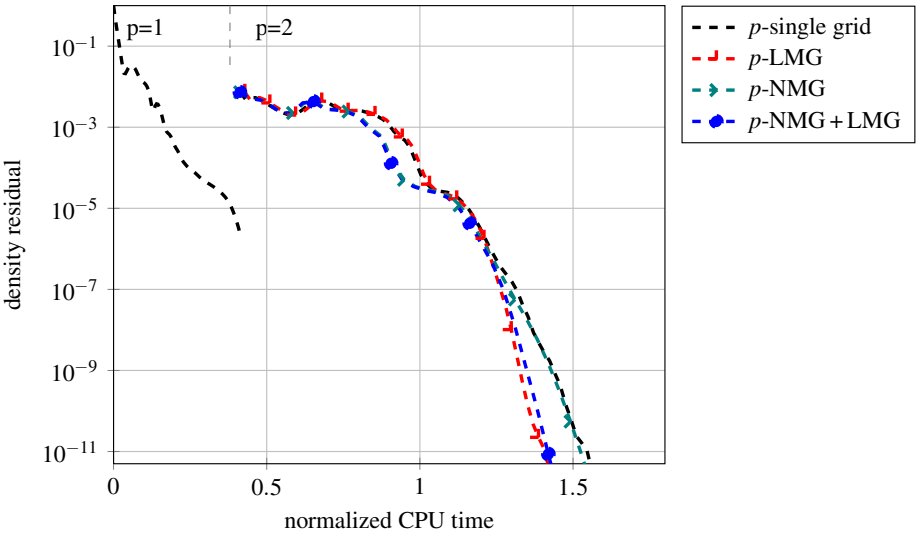


Fig. 12. VFE-2 delta wing: Computation of a $p = 2$ solution on the mesh with 13 816 elements with a p-start-up strategy [27]

solver settings. This results from the fact that fewer iterations are used on the highest level, as seen in Fig. 10. This again confirms the findings in 2D where an h-start-up strategy is more favorable than a p-start-up strategy in most cases.

Moreover, a beneficial run-time behavior of the multigrid algorithms can be seen in comparison with a single grid solver (unmarked lines in Figures 10, 11, 12), even though additional work is done between the top-level Backward-Euler smoother iterations in the multigrid algorithms.

Like in 2D test cases [28] also for the 3D test case considered here the combination of nonlinear and linear multigrid algorithms is more robust than the other algorithms. In particular, for the combination of nonlinear and linear h-multigrid algorithms a 3 times higher initial CFL number and for the p counterpart a five times higher initial CFL number can be used. The single grid algorithms from Fig. 10 do not converge for these increased initial CFL numbers. The combination of both multigrid algorithms is still the best algorithm in terms of CPU time and nonlinear top level iterations. This indicates that the findings in terms of robustness in 2D (cf. [28]) also hold in 3D. Moreover, with the reduced number of nonlinear top level iterations there is also an improvement in run-time. In particular, the number of nonlinear top level iterations in the h-multigrid is reduced from 30 to 24 which transfers to a reduction in normalized CPU time of only 0.12 for the whole computation. In contrast to that, in the p-multigrid case the nonlinear top level iterations are reduced from 58 to 32 which transfers to a reduction in normalized CPU time of 0.46 for the whole computation. Note, that the run time of this computation (not shown) including the start-up strategy is used for the normalization of the CPU times to unity in Figures 11 and 12.

The increased initial CFL number is beneficial at the beginning of the computation on each multigrid level. Due to the SER timestep control [22] which increases the CFL number at the end of the computation, with CFL$\to\infty$, the asymptotic convergence rate remains unchanged. These results align with the findings shown in Section 5.1 and other results published in [27, 28].

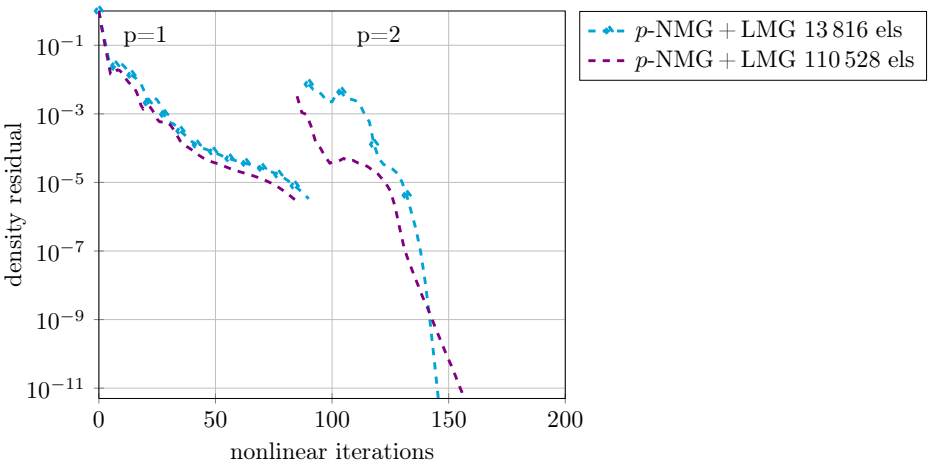
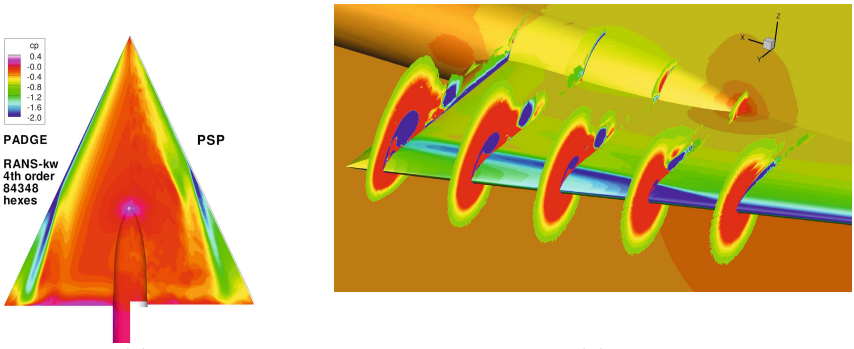


Fig. 13. VFE-2 delta wing: $p = 2$ computations on the mesh with 13 816 elements (lines with $\diamond$) and on a globally refined mesh with 110 528 elements (unmarked lines)

Fig. 13 shows a comparsion of the nonlinear and linear p-multilevel algorithm on two meshes, the coarse mesh with 13 816 element and the once globally refined mesh with 110 528 elements. Here, exactly the same parameters are used on the top level for both computations, the only difference is the underlying mesh. The overall convergence behavior on the globally refined mesh is similar to that on the coarse mesh, although some differences occur. In particular, the initial convergence on the fine level is improved for the refined mesh, whereas the asymptotic rate of convergence is inferior. This difference is due to the fact that with exactly the same linear solver settings the linear problems are not as accurately solved on the fine mesh as on the coarse one, since the size of the coarsest level is increased significantly for the refined mesh.

Adaptive mesh refinement algorithms In the following, we present numerical results obtained using the higher-order and adaptive discontinuous Galerkin flow solver PADGE [14]. We note, that in contrast to the previous results on the performance of the solver algorithms which are based on non-parametric basis functions and the Roe flux, the following results are based on parametric basis functions, the local Lax-Friedrichs flux and are solved with a fully implicit single grid solver. Furthermore, note that some of the following results have previously been published in [11].

Fig. 14 shows a fourth order flow solution on a residual-based adapted mesh with 84 348 curvilinear elements. The resulting c_p-distribution on the upper side of the wing is shown in the left part of Fig. 14(a) compared to pressure sensitive paint (PSP) measurements [17] in the right part of Fig. 14(a). In the measurements as well as in the fourth order flow solution on this rather coarse mesh we recognize the suction trace of the stronger outer vortex as well as the suction trace of the weaker inner vortex. Furthermore, Fig. 14(b) shows the c_p-distribution on the wing together with slices of the λ_2-criterion. The two primary vortices are clearly visible: the inner vortex that weakens and the outer vortex that strengthens while being advected downstream.



(a) (b)

Fig. 14. Subsonic flow around the VFE-2 delta wing: Fourth order DG solution on a residual-based adapted mesh with 84 348 curvilinear elements. (a) c_p-distribution of DG solution (left) in comparison to PSP measurements (right). (b) c_p-distribution and slices of the λ_2-criterion of the DG solution [11].

In the following we perform a grid refinement study for this subsonic flow around the VFE-2 delta wing. In particular, we compare the performance of the DG flow solver PADGE in predicting the drag and lift coefficients C_d and C_l for a third order DG solution

- under residual-based mesh refinement (cf. Algorithm 5),
- under adjoint-based mesh refinement (cf. Algorithm 4), and
- under global mesh refinement (h-refinement) of the hexahedral mesh,
- with global p-refinement on a tetrahedral mesh, and
- with the FV flow solver TAU on a sequence of mixed-element meshes.

Here, the hexahredral mesh refers to the curvilinear mesh (quartic lines) depicted in Figures 8 and 9(b). The tetrahedral mesh refers to a curvilinear mesh (courtesy of Oubay Hassan [32]) consisting of 1 145 797 tetrahedral elements with cubic lines shown in Fig. 16. Furthermore, Cassidian has generated a sequence

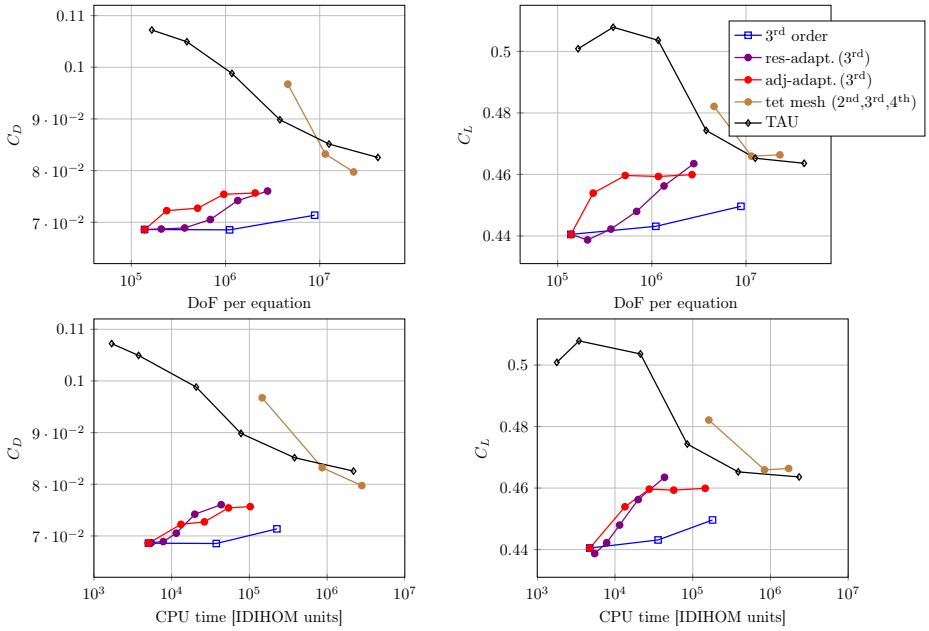


Fig. 15. Subsonic flow around the VFE-2 delta wing: Mesh convergence of drag (left) and lift (right) vs. degrees of freedom (top) and CPU time (bottom).

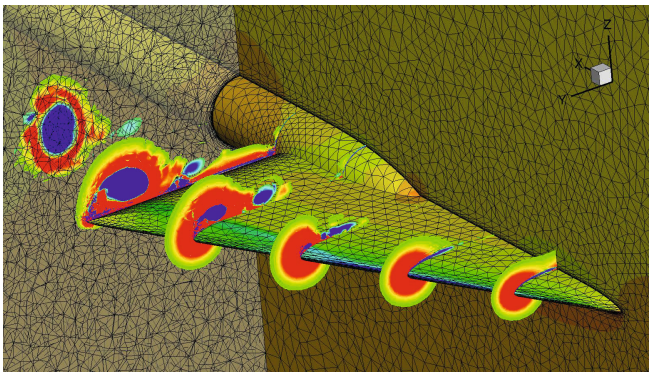


Fig. 16. Subsonic flow around the VFE-2 delta wing. Fourth order solution on an unstructured grid (courtesy of Oubay Hassan [32]) of about 1.15×10^6 tetrahedral elements with cubic lines: c_p-distribution and slices of the λ_2-criterion.

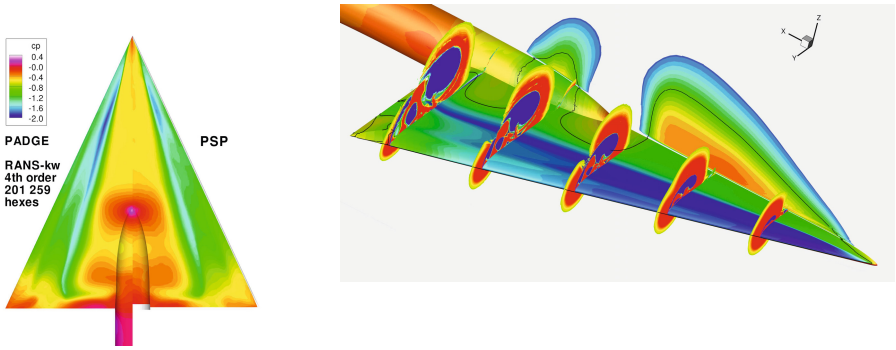
of unstructured mixed-element straight-sided meshes on which they applied the DLR-TAU code. In the top part of Fig. 15 we see that the accuracy of the second order DG solution on the tetrahedral mesh is lower than the second order Finite Volume computations on the sequence of mixed-element meshes at similar degrees of freedom. While this is mainly attributed to different distributions of points in the meshes compared, we see that under p-refinement, i. e., under increasing the order from two to three and four, the DG solution becomes more accurate (at least in C_d) on the tetrahedral mesh than the FV solution on finer mixed-element meshes. This demonstrates the high order DG solutions give accurate results even on purely tetrahedral meshes, while typically boundary layers are meshed with prisms or hexahedra to increase the quality of the solution, in particular for FV methods. Fig. 16 shows the c_p-distribution and slices of the λ_2-criterion of the fourth order DG solution on this tetrahedral mesh.

The remaining lines in the top part of Fig. 15 are based on the hexahedral mesh. Here, we see that global mesh refinement starting from the extremely coarse initial mesh of 13 816 curvilinear elements (cf. Figures 8 and 9(b)) converges very slowly and is thus not adequate for giving accurate results. The point distribution in this coarse mesh is not well suited for a third-order DG discretization. In contrast to that we see that the adjoint-based mesh refinement leads to "grid converged" force coefficients on significantly coarser meshes than the global mesh refinement as well as the other computations. Even though the timings in the bottom part of Fig. 15 of the adjoint-based mesh refinement includes the computational time of additional adjoint problems to be solved the advantage in the number of iterations of adjoint-based mesh refinement over the other computations seen in the top of Fig. 15 transfers to a similar (though slightly reduced) advantage in computing time in the bottom of Fig. 15.

5.3 U.1c: Transonic Flow around the VFE-2 Configuration

In the following, we consider a transonic fully turbulent flow at a Mach number $M = 0.8$, a Reynolds number $Re = 2 \times 10^6$ and an angle of attack of $\alpha = 20.5°$ around the VFE-2 configuration. This vortex dominated flow features a system of vortices and a shock. We note, that the following numerical results have previously been published in [12].

Fig. 17 shows a fourth order flow solution on a 4 times residual-based adapted mesh of 201 259 curvilinear elements. The resulting c_p-distribution on the upper side of the wing is shown in the left part of Fig. 17(a) compared to the pressure sensitive paint measurements [17] in the right part of Fig. 17(a). In the measurements as well as in the fourth order flow solution on this rather coarse mesh we recognize the suction trace of the strong primary vortex and the weaker secondary vortex. We note, however, that the suction trace of the computed primary vortex is slightly stronger than that in the measurements. Fig. 17(b) shows the c_p-distribution on the wing together with slices of the λ_2-criterion. The primary and secondary vortices are clearly visible. The resolution of the numerical solution is high enough such that some vorticity even at the position of a possible tertiary vortex can be recognized. At the symmetry plane the Mach number



(a) (b)

Fig. 17. Transonic flow around the VFE-2 delta wing: fourth order DG solution on a residual-based adapted mesh with 201 259 curvilinear elements. (a) c_p-distribution of DG solution (left) in comparison to PSP measurements (right). (b) c_p-distribution on the wing, Mach number distribution on the symmetry plane, and slices of the λ_2-criterion of the DG solution [12].

distribution is shown. Furthermore, on the symmetry plane and the wing the isoline of the critical c_p-value is shown. On the symmetry plane one clearly recognizes the region of supersonic flow over the wing which decelerates through a shock. As the flow passes the front of the sting it accelerates again to supersonic flow and then decelerates smoothly, i. e., without shock, to subsonic flow.

6 Conclusions

A flexible framework for h- and p-multigrid-based solver algorithms for high order DG discretizations has been developed and applied to RANS computations with both one and two-equation turbulence models. Results in 2D & 3D indicate that the best performance in algorithmic convergence and run-time behavior is achieved with a combination of a nonlinear multigrid with a Backward-Euler smoother on each level, in which the resulting linear systems are solved with a linear multigrid preconditioned GMRes method. The results also indicate that the nonlinear multigrid algorithms are beneficial at the beginning of the computation on every level and the linear multigrid algorithms are superior at the end of the iterative procedure in the regime of asymptotic convergence. Thus, the combination of a nonlinear and linear multigrid accomplishes the biggest reduction in nonlinear iterations for both the p- and h-start-up computations. The applicability of the algorithms to both unstructured mixed-element meshes and higher than third order has been demonstrated, at least in 2D.

Fourth order Discontinuous Galerkin solutions have been computed on an unstructured purely tetrahedral mesh as well as on locally refined purely hexahedral meshes with hanging nodes around the VFE-2 delta wing with rounded

leading edge at subsonic and transonic flow conditions. The results show that residual-based mesh refinement is particularly well suited for resolving the vortical systems. Furthermore, adjoint-based mesh refinement has been shown to be particularly well suited for accurately and efficiently approximating aerodynamic force coefficients. In particular, a grid refinement study performed at subsonic flow conditions revealed that a third order DG solution on an adjoint-based refined mesh gives converged force coefficients with a significantly smaller number of degrees of freedom and a by two orders of magnitude reduced computational effort compared to second order Finite Volume solutions on a hierarchy of hybrid meshes.

Future work will tackle the combination of a multigrid solver with mesh adaptation.

References

1. Bassi, F., Botti, L., Colombo, A., Pietro, D.D., Tesini, P.: On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. Journal of Computational Physics 231(1), 45–65 (2012)
2. Bassi, F., Botti, L., Colombo, A., Rebay, S.: Agglomeration based discontinuous Galerkin discretization of the Euler and Navier-Stokes equations. Computers and Fluids 61(0), 77–85 (2012)
3. Bassi, F., Crivellini, A., Rebay, S., Savini, M.: Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations. Computers & Fluids 34(4-5), 507–540 (2005)
4. Bassi, F., Ghidoni, A., Rebay, S., Tesini, P.: High-order accurate p-multigrid discontinuous Galerkin solution of the Euler equations. International Journal for Numerical Methods in Fluids 60, 847–865 (2009)
5. Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., Savini, M.: A high-order accurate discontinuous Finite Element method for inviscid and viscous turbomachinery flows. In: Decuypere, R., Dibelius, G. (eds.) 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, March 5-7, pp. 99–108. Technologisch Instituut (1997)
6. Burgess, N., Nastase, C., Mavriplis, D., Martinelli, L.: Efficient solution techniques for discontinuous Galerkin discretizations of the Navier-Stokes equations on hybrid anisotropic meshes. 48th AIAA Aerospace Sciences Meeting. AIAA 2010-1448 (2010)
7. Crippa, S.: Advances in Vortical Flow Prediction Methods for Design of Delta-Winged Aircraft. PhD thesis, KTH Engineering Sciences, Stockholm, Sweden (2008)
8. Ilinca, D.P.F.: Positivity preservation and adaptive solution for the k-w model of turbulence. AIAA, J. 36, 44–50 (1998)
9. Fejtek, I.: Summary of code validation results for a multiple element airfoil test case. In: 28th AIAA Fluid Dynamics Conference. AIAA Paper 97-1932 (1997)
10. Fidkowski, K.J., Oliver, T.A., Lu, J., Darmofal, D.L.: p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. Journal of Computational Physics 207(1), 92–113 (2005)

11. Hartmann, R.: Higher-order and adaptive discontinuous Galerkin methods applied to turbulent delta wing flow. In: Dillmann, A., Heller, G., Kreplin, H.-P., Nitsche, W., Peltzer, I. (eds.) New Results in Numer. & Exp. Fluid Mech. NNFM, vol. 121, pp. 497–506. Springer, Heidelberg (2013)
12. Hartmann, R.: Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. Int. J. Numer. Meth. Fluids 72(8), 883–894 (2013)
13. Hartmann, R., Held, J., Leicht, T.: Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations. Journal of Computational Physics 230(11), 4268–4284 (2011)
14. Hartmann, R., Held, J., Leicht, T., Prill, F.: Discontinuous Galerkin methods for computational aerodynamics – 3D adaptive flow simulation with the DLR PADGE code. Aerosp. Sci. Technol. 14, 512–519 (2010)
15. Hartmann, R., Leicht, T.: Higher order and adaptive DG methods for compressible flows. In: Deconinck, H. (ed.) VKI LS 2014-03: 37th Advanced VKI CFD Lecture Series: Recent Developments in Higher Order Methods and Industrial Application in Aeronautics, Belgium, December 9-12. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse (2014)
16. Hummel, D., Redeker, G.: A new vortex flow experiment for computer code validation. In: RTO-AVT Symposium on Vortex Fow and High Angle of Attack, Loen, Norway, 7.-11.05 (2001)
17. Konrath, R., Klein, C., Engler, R., Otter, D.: Analysis of PSP results obtained for the VFE-2 65° delta wing configuration at sub- and transonic speeds. In: 44th AIAA Aerospace Sciences Meeting and Exhibit. AIAA 2006-59-624 (2006)
18. Landmann, B., Kessler, M., Wagner, S., Krämer, E.: A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows. Computers & Fluids 37(4), 427–438 (2008)
19. Langer, S., Schwöppe, A., Kroll, N.: The DLR Flow Solver TAU - Status and Recent Algorithmic Developments. In: 52nd AIAA Aerospace Sciences Meeting. AIAA Paper 2014-0080 (2014)
20. Luo, H., Baum, J.D., Löhner, R.: A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. Journal of Computational Physics 227(20), 8875–8893 (2008)
21. Moulitsas, I., Karypis, G.: Multilevel algorithms for generating coarse grids for multigrid methods. In: Proceedings of the 2001 ACM/IEEE Conference on Supercomputing, p. 45. ACM (2001)
22. Mulder, W.A., Leer, B.V.: Experiments with implicit upwind methods for the Euler equations. Journal of Computational Physics 59(2), 232–246 (1985)
23. Allmaras, S.R., Johnson, F.T., Spalart, P.R.: Modifications and clarifications for the implementation of the spalart-allmaras turbulence model. In: ICCFD7, p. 1902 (2012)
24. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. Journal of Computational Physics 43(2), 357–372 (1981)
25. Spalart, P., Allmaras, S.: One-equation turbulence model for aerodynamic flows. Recherche Aerospatiale (1), 5–21 (1994)
26. Trottenberg, U., Oosterlee, C., Schüller, A.: Multigrid. Academic Press (2001)
27. Wallraff, M., Leicht, T.: 3D application of higher order multigrid algorithms for a RANS-$k\omega$ DG-solver. In: Abgrall, R., Beaugendre, H., Congedo, P.M., Dobrzynski, C., Perrier, V., Ricchiuto, M. (eds.) High Order Nonlinear Numerical Methods for Evolutionary PDEs. LNCSE, vol. 99, pp. 77–88. Springer, Heidelberg (2013)

28. Wallraff, M., Leicht, T.: Higher order multigrid algorithms for a discontinuous galerkin rans solver. In: 52nd AIAA Aerospace Sciences Meeting. AIAA Paper 2014-0936 (2014)
29. Wallraff, M., Leicht, T., Lange-Hegermann, M.: Numerical flux functions for Reynolds-averaged Navier-Stokes and $k\omega$ turbulence model computations with a line-preconditioned p-multigrid discontinuous Galerkin solver. International Journal for Numerical Methods in Fluids 71(8), 1055–1072 (2013)
30. Wilcox, D.C.: Reassessment of the scale-determining equation for advanced turbulence models. AIAA Journal 26(11), 1299–1310 (1988)
31. Wilcox, D.C.: Turbulence Modeling for CFD. DCW Industries, Inc., La Canada CA (1993)
32. Xie, Z.Q., Sevilla, R., Hassan, O., Morgan, K.: The generation of arbitrary order curved meshes for 3D finite element analysis. Computational Mechanics 51(3), 361–374 (2013)

Time Integration in the Discontinuous Galerkin Code MIGALE – Steady Problems

F. Bassi[1], L. Botti[1], A. Colombo[1], A. Crivellini[2], C. De Bartolo[4], N. Franchina[1], A. Ghidoni[3], and S. Rebay[3]

[1] Department of Industrial Engineering, University of Bergamo, viale Marconi, 5 - 24044 Dalmine (BG), Italy
{francesco.bassi,lorenzo.botti,alessandro.colombo, nicoletta.franchina}@unibg.it
[2] Department of Industrial Engineering and Mathematical Sciences, Polytechnic University of Marche,via Brecce Bianche, - 60100 Ancona, Italy
a.crivellini@univpm.it
[3] Department of Mechanical and Industrial Engineering, University of Brescia, Via Branze, 38 - 25123 Brescia, Italy
{antonio.ghidoni,stefano.rebay}@unibs.it
[4] Department of Mechanical, Energetic and Management, University of Calabria, via P. Bucci, 44/C - 87036 Rende (CS), Italy
carmine.debartolo@unical.it

Abstract. This chapter presents the high-order Discontinuous Galerkin (DG) solver named MIGALE for the steady solution of the RANS and k-ω turbulence model equations. During the IDIHOM project the MIGALE features have been enhanced both in terms of the prediction capability and solver efficiency, due to the implementation of an Explicit Algebraic Reynolds Stress Model (EARSM) and of the h- and p-multigrid (MG), respectively. algorithm. Several high-order DG results of 2D and 3D subsonic/transonic turbulent test cases, proposed within the IDIHOM EU project, demonstrated the capability of the method.

Keywords: Discontinuous Galerkin method, turbulence modeling, multigrid, RANS, EARSM.

1 Introduction

In the scientific and industrial communities there is a growing concern that high-order methods will become an essential tool in many technical areas, to achieve the CFD accuracy expected by the design offices. Among high-order methods, Discontinuous Galerkin (DG) methods turned out to be one of the most promising techniques for their great geometrical flexibility [1, 2], straightforward implementation of h/p adaptive techniques [3, 4], and compact stencil, useful for massively parallel computers platform. As demonstrated during the EU project ADIGMA [19], the higher accuracy comes at an increased computational cost with respect to standard finite volume (FV) methods, preventing a widespread

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_10

application in industry. In order to overcome this limitation, a considerable research effort has been recently devoted to devise more efficient computational strategies, both for the construction of the DG space operator [5,6] and for the time integration [7–9].

In this chapter we summarized the progress on such topics, achieved within the IDIHOM EU project and we assessed the capability of the high-order DG code MIGALE in computing complex turbulent steady flows for aeronautical and turbomachinery applications. The turbulent flow field was computed by means of the Reynolds Averaged Navier-Stokes (RANS) equations with the closure provided by the high-Reynolds number k-$\widetilde{\omega}$ turbulence model, following the original implementation proposed in [11], and by an Explicit Algebraic Reynolds Stress Model (EARSM).

The shock-capturing technique presented and assessed in [2] was adopted. We proposed a set of primitive variables based on the pressure and the temperature logarithms to ensure the positivity of the thermodynamic unknowns. Different time integration techniques were implemented: an implicit solver, a h- and p-multigrid algorithm [28,31]. Several high-order DG results for 2D and 3D subsonic/transonic turbulent test cases demonstrated the potential and capability of the method.

The paper is organized as follows. Section 2 describes the governing equations. In Section 3 the description of the DG discretization is given. Section 4 describes the time integration techniques, while Section 5 is dedicated to the discussion of the numerical results. Finally, Section 6 gives the concluding remarks.

2 Governing Equations

The complete set of RANS and k-$\widetilde{\omega}$ turbulence model equations can be written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \widehat{\tau}_{ji}}{\partial x_j}, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j H) = \frac{\partial}{\partial x_j}\left[u_i \widehat{\tau}_{ij} - \widehat{q}_j\right] - \tau_{ij}\frac{\partial u_i}{\partial x_j} + \beta^* \rho \overline{k} e^{\widetilde{\omega}_r}, \tag{3}$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \overline{\mu}_t)\frac{\partial k}{\partial x_j}\right] + \tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta^* \rho \overline{k} e^{\widetilde{\omega}_r}, \tag{4}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho \widetilde{\omega}) + \frac{\partial}{\partial x_j}(\rho u_j \widetilde{\omega}) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_j}\right] + \frac{\alpha}{\overline{k}}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta \rho e^{\widetilde{\omega}_r} + \\ + (\mu + \sigma \overline{\mu}_t)\,\frac{\partial \widetilde{\omega}}{\partial x_k}\frac{\partial \widetilde{\omega}}{\partial x_k},\end{aligned} \tag{5}$$

where the pressure, the turbulent and total stress tensors, the heat flux vector, the eddy viscosity and the limited value of turbulent kinetic energy are given by

$$p = (\gamma - 1)\rho\left(E - u_k u_k/2\right), \tag{6}$$

$$\tau_{ij} = 2\overline{\mu}_t\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] - \frac{2}{3}\rho\overline{k}\delta_{ij}, \tag{7}$$

$$\widehat{\tau}_{ij} = 2\mu\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] + \tau_{ij}, \tag{8}$$

$$\widehat{q}_j = -\left(\frac{\mu}{\mathrm{Pr}} + \frac{\overline{\mu}_t}{\mathrm{Pr}_t}\right)\frac{\partial h}{\partial x_j}, \tag{9}$$

$$\overline{\mu}_t = \alpha^*\rho\overline{k}e^{-\widetilde{\omega}_r}, \quad \overline{k} = \max\left(0, k\right). \tag{10}$$

Here γ is the ratio of gas specific heats, Pr and Pr_t are the molecular and turbulent Prandtl numbers and

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

is the mean strain-rate tensor. The closure parameters α, α^*, β, β^*, σ, σ^* are those of the high-Reynolds number k-ω model of Wilcox [58].

The turbulence k-$\widetilde{\omega}$ model given by Eqs. (11) and (12) is implemented as presented in [11], where $\widetilde{\omega} = \log(\omega)$ is used in place of ω to obtain a better behavior near solid wall and guarantees the positivity of this variable. Limited values $\overline{k}$ and $\overline{\mu}_t$ are used in Eqs. (10)–(12) to deal with possible negative value of the turbulent kinetic energy. The variable $\widetilde{\omega}$ in the source terms and in the eddy viscosity equation is replaced by $\widetilde{\omega}_r$, indicating it has to fulfill suitably defined "realizability" conditions, which set a lower bound on $\widetilde{\omega}$.

The "slightly-rough-wall" boundary condition is adopted to prescribe the value of $\widetilde{\omega}$ at the wall. In particular the approach proposed by Menter [46] is modified to introduce a dependence on the polynomial degree l of the solution. At the wall the $\widetilde{\omega}$ value is defined as the Taylor series expansion of the near wall solutions for $\widetilde{\omega}$ (with $\widetilde{\omega}_w \to \infty$) around h, truncated to l terms

$$\widetilde{\omega}_w^l = \log\left(\frac{6\nu_w}{\beta\left(\alpha^l h\right)^2}\right), \tag{11}$$

where

$$\alpha^l = e^{-\sum_{n=1}^{l}\frac{1}{n}}. \tag{12}$$

h is the distance from the wall of the element centroid next to the wall. Details of the implementation can be found in [7].

2.1 Explicit Algebraic Reynolds Stress Model

The EARSM proposed by Wallin and Johanson [24] was implemented in the MIGALE code. The constitutive relation for the turbulent stress tensor can be written as

$$\frac{\tau_{ij}}{\overline{\rho}\overline{k}} = -\frac{\overline{u_i u_j}}{\overline{k}} = -\alpha^* a_{ij} - \frac{2}{3}\delta_{ij} = \alpha^* \left(2C_\mu \tau S_{ij} - a_{ij}^{(\mathrm{ex})}\right) - \frac{2}{3}\delta_{ij}, \tag{13}$$

where, for implementation convenience, the anisotropy tensor a_{ij} is split in a linear part and a non-linear extra anisotropy contribution. The time scale τ and the variable coefficient C_μ are given by

$$\tau = \frac{1}{\beta^* e^{\tilde{\omega}}}, \qquad C_\mu = -\frac{1}{2}\left(\beta_1 + II_\Omega \beta_6\right). \tag{14}$$

The time scale τ does not include the near-wall lower bound, based on the Kolmogorov time scale, usually employed in k-ϵ implementations of EARSM since this limitation is provided by the finite value of ω set at the wall. The eddy viscosity $\overline{\mu}_t$ and the extra-anisotropy tensor $a_{ij}^{(\mathrm{ex})}$ are given by

$$\overline{\mu}_t = \alpha^* C_\mu \tau \overline{\rho} \overline{k}, \tag{15}$$

$$\begin{aligned} a_{ij}^{(\mathrm{ex})} &= \beta_3 \tau^2 \left(\Omega_{ik}\Omega_{kj} - \frac{1}{3} II_\Omega \delta_{ij}\right) + \beta_4 \tau^2 \left(S_{ik}\Omega_{kj} - \Omega_{ik}S_{kj}\right) \\ &+ \beta_6 \tau^3 \left(S_{ik}\Omega_{kl}\Omega_{lj} + \Omega_{ik}\Omega_{kl}S_{lj} - II_\Omega S_{ij} - \frac{2}{3} IV \delta_{ij}\right) \\ &+ \beta_9 \tau^4 \left(\Omega_{ik}S_{kl}\Omega_{lm}\Omega_{mj} + \Omega_{ik}\Omega_{kl}S_{lm}\Omega_{mj}\right), \end{aligned} \tag{16}$$

where the coefficients $\beta_{i\in\{1,3,4,6,9\}}$ are functions of the invariants II_S, II_Ω and IV

$$II_S = \mathrm{tr}\{\boldsymbol{S^2}\}, \qquad II_\Omega = \mathrm{tr}\{\boldsymbol{\Omega^2}\}, \qquad IV = \mathrm{tr}\{\boldsymbol{S\Omega^2}\}. \tag{17}$$

In the following sections, the notation EARSMx will indicate the EARSM model including anisotropy terms up to the x-th degree.

3 DG Approximation of the RANS and k-ω EARSM Equations

The system of governing equations can be written in compact form as

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}^{\mathrm{c}}(\mathbf{q}) + \nabla \cdot \mathbf{F}^{\mathrm{v}}(\mathbf{q}, \nabla\mathbf{q}) + \mathbf{s}(\mathbf{q}, \nabla\mathbf{q}) = \mathbf{0}, \tag{18}$$

where $\mathbf{q} \in \mathbb{R}^m$ is the unknown solution vector of the m variables, $\mathbf{F}^{\mathrm{c}} \in \mathbb{R}^m \otimes \mathbb{R}^d$ and $\mathbf{F}^{\mathrm{v}} \in \mathbb{R}^m \otimes \mathbb{R}^d$ are the inviscid and viscous flux functions, $\mathbf{s} \in \mathbb{R}^m$ the sum of turbulence source and volume forces vectors and d the number of dimensions.

A particular set of primitive variables is adopted to ensure the positivity of the thermodynamic unknowns, where p and T are replaced by their logarithms

$\widetilde{p} = \log(p)$ and $\widetilde{T} = \log(T)$. In fact, some flow configurations, *e.g.* abrupt expansions and strong shocks can produce numerical oscillations that can result in negative values of the pressure or the temperature.

Let $\mathbf{q} = [\rho, \rho u_1, \rho u_2, \rho u_3, \rho E, \rho k, \rho\widetilde{\omega}]^T$ and $\mathbf{w} = [\widetilde{p}, u_1, u_2, u_3, \widetilde{T}, k, \widetilde{\omega}]^T$ be the solution vector and the state vector in primitive variables based on logarithmic thermodynamic proprieties, and $\mathbf{P}(\mathbf{w}) = \frac{\partial \mathbf{q}}{\partial \mathbf{w}} \in \mathbb{R}^m \otimes \mathbb{R}^m$ the transformation matrix. The system of governing equations (18) for the new set of variables is rewritten as

$$\mathbf{P}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}^{\mathrm{c}}(\mathbf{w}) + \nabla \cdot \mathbf{F}^{\mathrm{v}}(\mathbf{w}, \nabla\mathbf{w}) + \mathbf{s}(\mathbf{w}, \nabla\mathbf{w}) = \mathbf{0}. \tag{19}$$

Notice that, while the equation for ω is explicitly rewritten for its logarithm, giving rise to an additional source term $\frac{\partial\widetilde{\omega}}{\partial x_k}\frac{\partial\widetilde{\omega}}{\partial x_k}$, the logarithms of p and T is introduced only in terms of the transformation matrix and their polynomial representation.

3.1 Discontinuous Galerkin Space Discretization

In order to construct a DG discretization of the governing equations, the weak formulation of Eq. 19 is obtained multiplying by an arbitrary smooth test function $\mathbf{v} = \{v_1, \cdots, v_m\}$, and integrating by parts

$$\begin{aligned}&\int_\Omega \mathbf{v} \cdot \left(\mathbf{P}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial t}\right) \mathrm{d}\Omega - \int_\Omega \nabla\mathbf{v} : (\mathbf{F}^{\mathrm{c}}(\mathbf{w}) + \mathbf{F}^{\mathrm{v}}(\mathbf{w}, \nabla\mathbf{w})) \ \mathrm{d}\Omega \\ &+ \int_{\partial\Omega} \mathbf{v} \otimes \boldsymbol{n} : (\mathbf{F}^{\mathrm{c}}(\mathbf{w}) + \mathbf{F}^{\mathrm{v}}(\mathbf{w}, \nabla\mathbf{w})) \ \mathrm{d}\sigma + \int_\Omega \mathbf{v} \cdot \mathbf{s}(\mathbf{w}, \nabla\mathbf{w}) \ \mathrm{d}\Omega = \mathbf{0},\end{aligned} \tag{20}$$

where $\boldsymbol{n}$ is the unit normal vector to the boundary. The physical domain is then approximated by a computational grid $\mathcal{T}_h = \{K\}$ consisting of a set of non-overlapping elements while a suitable discontinuous finite element space, spanned by polynomial functions of degree at most l continuous only inside each element K, is defined as follows

$$\mathbf{V}_h^l \stackrel{\mathrm{def}}{=} [\mathbb{P}_d^l(\mathcal{T}_h)]^m, \tag{21}$$

where

$$\mathbb{P}_d^l(\mathcal{T}_h) \stackrel{\mathrm{def}}{=} \left\{v_h \in L^2(\Omega_h) \ : \ v_h|_K \in \mathbb{P}_d^l(K), \forall K \in \mathcal{T}_h\right\}. \tag{22}$$

Considering the weak form of system (19) over $\mathcal{T}_h$ and using the finite space (21), the solution $\mathbf{w}$ and the test function $\mathbf{v}$ are replaced with a finite element approximation $\mathbf{w}_h$ and $\mathbf{v}_h$ respectively, belonging to the space $\mathbf{V}_h^l$. As a satisfactory basis for such space, orthogonal and hierarchical basis functions defined in the physical reference frame were employed, see [14]. Any function v_h can be regarded as a combination of the elements of the chosen basis. Let denote such elements as ϕ_i where $i \in \{1, \cdots, N_{\mathrm{DoF}}^K\}$ and N_{DoF}^K is the number of degrees of freedom of the polynomial space local to a given element K. Accounting for

this aspect, the DG formulation of the compressible RANS and k-ω equations consists in seeking the elements $w_{h,1}, \cdots, w_{h,m}$ of $\mathbf{w}_h \in \mathbf{V}_h^k$ such that

$$\sum_{K\in\mathcal{T}_h} \int_K \phi_i P_{j,k}\left(\mathbf{w}_h\right) \phi_l \frac{\mathrm{d}W_{k,l}}{\mathrm{d}t}\, \mathrm{d}\Omega - \sum_{K\in\mathcal{T}_h} \int_K \frac{\partial \phi_i}{\partial x_k} \left(F^{\mathrm{c}}_{j,k}\left(\mathbf{w}_h\right) + F^{\mathrm{v}}_{j,k}\left(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \mathbf{r}^g\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)\right) \mathrm{d}\Omega + \sum_{f\in\mathcal{F}} \int_f \llbracket \phi_i \rrbracket_k \left(\widehat{F}^{\mathrm{c}}_{j,k}\left(\mathbf{w}_h^{\pm}\right) + \widehat{F}^{\mathrm{v}}_{j,k}\left(\mathbf{w}_h^{\pm}, \left(\nabla_h \mathbf{w}_h + \eta_f \mathbf{r}^f\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)^{\pm}\right)\right) \mathrm{d}\sigma + \sum_{K\in\mathcal{T}_h} \int_K \phi_i s_j\left(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \mathbf{r}^g\left(\llbracket \mathbf{w}_h \rrbracket\right)\right) \mathrm{d}\Omega = 0, \quad \forall i \in \{1, \cdots, N^K_{\mathrm{DoF}}\}, \tag{23}$$

where $j = 1, \cdots, m$ and

$$w_{h,j} = \phi_l W_{j,l}, \tag{24}$$

denoting with $\mathbf{W}$ the global vector of unknown degrees of freedom (DoF).

In Eq. (42) $\mathcal{F}$ is the collection of all edges f and the functions $\mathbf{r}^g$ and $\mathbf{r}^f$: $[L^1(f)]^{m\times d} \rightarrow [\mathbb{P}^k_d(\mathcal{T}_h)]^{m\times d}$ are the global and local lifting operators used to obtain a consistent, stable and accurate DG discretization of the viscous part of the equations according to the BR2 scheme [15].

Being the functional approximation discontinuous, the flux is not uniquely defined over the elements boundary, and thus a numerical flux vector is suitably defined both for the inviscid $\widehat{\mathbf{F}^{\mathrm{c}}}$ and viscous $\widehat{\mathbf{F}^{\mathrm{v}}}$ part of the equations. In order to ensure conservation and correctly account for wave propagation the former is based on the Godunov flux computed with an exact Riemann solver [17] or alternatively on the van Leer vector split flux as modified by Hänel [18], while the latter is given by

$$\widehat{\mathbf{F}^{\mathrm{v}}}\left(\mathbf{w}_h^{\pm}, \left(\nabla_h \mathbf{w}_h + \eta_f \mathbf{r}^f\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)^{\pm}\right) \stackrel{\mathrm{def}}{=} \left\{\mathbf{F}^{\mathrm{v}}\left(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \eta_f \mathbf{r}^f\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)\right\}, \tag{25}$$

where the stability parameter η_f was defined according to [2].

The shock-capturing technique employed in this work is based on the scheme presented in [2]. The artificial diffusion is explicitly introduced within each element $K \in \mathcal{T}_h$ without using any shock sensor to detect the discontinuities location. The shock-capturing term is always and everywhere active but introduces numerical viscosity only where unphysical oscillations occur.

4 Time Integration: Implicit Solver

Assembling together all the elemental contributions, the discrete problem corresponding to Eq. (42) can be written as

$$\mathbf{M_P} \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} + \mathbf{R}\left(\mathbf{W}\right) = \mathbf{0}, \tag{26}$$

where $\mathbf{R}$ is the residuals vector and $\mathbf{M_P}$ is the global block diagonal matrix arising from the discretization of the first integral of Eq. (42). Matrix $\mathbf{M_P}$ couples the degrees of freedom of different variables within the element throug the variable transformation matrix $\mathbf{P}$.

Equation (44) defines a system of nonlinear ODEs which is solved by means of a linearized backward Euler (LBE) scheme

$$\left[\frac{\mathbf{M}_{\mathbf{P}}^{n}}{\Delta t}+\frac{\partial \mathbf{R}\left(\mathbf{W}^{n}\right)}{\partial \mathbf{W}}\right] \Delta \mathbf{W}=-\mathbf{R}\left(\mathbf{W}^{n}\right). \tag{27}$$

The linear system arising at each time step n from Eq. (44), is solved using the matrix-explicit or the matrix-free GMRES (Generalized Minimal RESidual) algorithm, see [8]. System preconditioning is required to make the convergence of the GMRES solver acceptable in problems of practical interest. The block Jacobi method with one block per process ILU(0) or the additive Schwartz method (ASM) are usually employed. The linear algebra and parallelization are handled through the PETSc [20] library (Portable Extensible Toolkit for Scientific Computations) and MPI, the standard for message-passing communication.

4.1 The Pseudo-Transient Continuation Strategy

The choice of the time step can significantly affect both the efficiency and the robustness of the method. For steady computations we relied on the pseudo-transient continuation strategy with the local time step given by

$$\Delta t_K = \mathrm{CFL}\frac{h_{K,\mathrm{CFL}}}{c_v + d_v},$$

where

$$c_v = |\mathbf{v}| + a, \qquad d_v = 2\frac{\mu_e + \lambda_e}{h_K}, \qquad h_{K,\mathrm{CFL}} = d\frac{\Omega_K}{S_K},$$

define convective and diffusive velocities and the reference dimension of the generic element K, respectively. The coefficients μ_e and λ_e are the effective dynamic viscosity and conductivity, while Ω_K and S_K denote the volume and the surface of K. All quantities depending on $\mathbf{w}_h$ in the above relations are computed from mean values of $\mathbf{w}_h$. Devising an effective and robust strategy to increase the CFL number as the residual decreases is not an easy task, especially for turbulent computations. The CFL law proposed in this work is based on the L^∞ and the L^2 norms of the residual and is defined as follows

$$\begin{cases} \mathrm{CFL} = \dfrac{\mathrm{CFL}_{min}}{x^\alpha} & \text{if } x \le 1 \\ \mathrm{CFL} = \mathrm{CFL}_{exp} + (\mathrm{CFL}_{min} - \mathrm{CFL}_{exp}) e^{\alpha \dfrac{\mathrm{CFL}_{min}}{\mathrm{CFL}_{min} - \mathrm{CFL}_{exp}}(1-x)} & \text{if } x > 1 \end{cases} \tag{28}$$

where, denoting by $x_{L^2} = \max\left(|R_i|_{L^2}/|R_{i0}|_{L^2}\right)$ and $x_{L^\infty} = \max\left(|R_i|_{L^\infty}/|R_{i0}|_{L^\infty}\right)$ for $i = 1, \ldots, m$

$$\begin{cases} x = \min\left(x_{L^2}, 1\right) & \text{if } x_{L^\infty} \le 1 \\ x = x_{L^\infty} & \text{if } x_{L^\infty} > 1, \end{cases}$$

and CFL_{min}, CFL_{exp} and α are the user-defined minimum CFL number, the maximum CFL number of explicit schemes and the exponent (usually ≤ 1) governing the growth rate of the CFL number, respectively.

5 Time Integration: Multigrid Solution Strategy

Standard iterative solvers show a slowdown of the convergence rate after few iterations, being unable to effectively reduce the error. In fact the error is a combination of "error modes", and the smoothers can damp only a part of these modes in the first iterations, thereby showing a stall in the convergence.

In order to remove all the error modes and increase the convergence rate, multigrid (MG) strategies can be adopted, both in the h and/or p variant. In h-MG methods, even if only a part of the modes is damped on each grid, the use of a hierarchical set of nested grids allows to reduce the solution error over the entire modes spectrum. In the p-MG context a sequence of progressively lower order approximations on a single grid is considered as coarse levels.

The relaxation scheme must provide, at an arbitrary level, an effective damping of all the error modes which can not be represented on "coarser" levels, where "coarse" represents an actual coarser grid in the h-multigrid case or a lower order approximation in the p-multigrid case. In both cases the modes that need to be damped at each level are called "oscillatory" modes, while the remaining modes are referred to as "smooth" modes.

The various levels can be visited following different paths. In this work, the commonly considered V-cycle was adopted. At each level, a number ν_1 of pre-smoothing iterations was performed prior to restricting the solution to the next coarser level, while, on the way back to "finer" levels, a number ν_2 of post-smoothing iterations was performed after prolongation. As a further improvement of the method, the full multigrid (FMG) algorithm was adopted, which exploits the coarser level solutions to obtain a good initial guess to initialize the computation of the finer levels. On each level before passing to the next finer level it is not necessary to compute the fully converged solution, since the discretization error on coarser levels can be relatively large. In the proposed algorithm this issue was addressed by prolongating the solution to the next finer level if a residual-based criterion was met, which is defined as:

$$\textbf{if} \quad (\max(||\mathbf{R}^l||_2) < 10^{-2}) \quad l \longrightarrow l+1, \tag{29}$$

where $||\mathbf{R}^l||_2$ is the L^2 norm of the residual vector.

The nonlinear MG algorithm is briefly summarized in the following with reference to a generic nonlinear problem $\mathbf{A}^l(\mathbf{W}^l) = \mathbf{b}$, where the superscript l indicates the level (the grid level h and the polynomial level p for the h- and p-MG algorithm, respectively). If $\widetilde{\mathbf{W}}^l$ denotes the approximate solution of the nonlinear problem during the iterative solution process and $\mathbf{R}^l(\widetilde{\mathbf{W}}^l) = \mathbf{b}^l - \mathbf{A}^l(\widetilde{\mathbf{W}}^l)$ the residual vector, the basic multigrid update of $\widetilde{\mathbf{W}}^l$ is performed as described in the Algorithms 1.

Algorithm 1. Multigrid algorithm

if (Pre-smoothing) **then**
  Compute $\mathbf{A}^l_{pre}, \mathbf{b}^l$
  $\widetilde{\mathbf{W}}^l = MGS(\mathbf{A}^l_{pre}, \mathbf{b}^l, \mathbf{s}^l_{pre}, \widetilde{\mathbf{W}}^l, \nu_1)$
  $\widetilde{\mathbf{W}}^{l-1}_0 = \widetilde{\mathbf{I}}^{l-1}_l \widetilde{\mathbf{W}}^l$
  $\mathbf{R}^{l-1} = \mathbf{I}^{l-1}_l \mathbf{R}^p(\widetilde{\mathbf{W}}^l)$
  $\mathbf{s}^{l-1}_{pre} = \mathbf{A}^{l-1}_{pre}(\widetilde{\mathbf{W}}^{l-1}_0) - \mathbf{R}^{l-1}$
else if (Post-smoothing) **then**
  $\mathbf{e}^{l-1} = \widetilde{\mathbf{W}}^{l-1} - \widetilde{\mathbf{W}}^{l-1}_0$
  $\widetilde{\mathbf{W}}^l = \widetilde{\mathbf{W}}^l + \widetilde{\mathbf{I}}^l_{l-1} \mathbf{e}^{l-1}$
  $\mathbf{A}^l_{pst} = \mathbf{A}^l_{pre}$
  $\mathbf{s}^l_{pst} = \mathbf{s}^l_{pre}$
  Compute $\mathbf{b}^l$
  $\widetilde{\mathbf{W}}^l = MGS(\mathbf{A}^l_{pst}, \mathbf{b}^l, \mathbf{s}^l_{pst}, \widetilde{\mathbf{W}}^l, \nu_2)$
end if

In the Algorithm 1 MGS is the multigrid smoother, ν_1 and ν_2 the number of pre and post iterations, $\mathbf{s}$ the forcing term, $\mathbf{e}$ the correction term, $\widetilde{\mathbf{I}}^{l-1}_l$ and $\mathbf{I}^{l-1}_l$ the solution and the residual restriction operators, $\widetilde{\mathbf{I}}^l_{l-1}$ the error prolongation operator. The matrix of the system $\mathbf{A}^l_{pre}$ can be computed (i) on each level or (ii) only on the finest level and projected on the other levels by means of the operator $\widehat{\mathbf{I}}^{l-1}_l$, the matrix restriction operator, depending on the adopted smoother. The transfer operators for the p-MG algorithm are reported in [28], while for the h-MG they are reported in [31, 32].

5.1 Smoothers

Three different smoothers were adopted:

1. a Line-Implicit RK (LIRK) scheme;
2. a Block-Implicit RK (BIRK) scheme;
3. an Implicit iterative smoother based on the linearized Backward Euler (LBE) scheme, as defined in Eq.27.

The RK schemes are described and assessed in [28]. The Algorithm 2 shows the LIRK scheme, where $\mathbf{T}(\mathbf{W}^0)$ is the block tridiagonal matrix associated to a line. The matrix $\left[\dfrac{\mathbf{M_P}}{\Delta t} + \alpha_m \mathbf{T}(\mathbf{W}^0)\right]$ is computed only for the first stage of the finest level at each MG cycle. The line smoother was employed only for the finest level because, as demonstrated in [28], on coarser levels it has no effect on convergence rate, while, on the other hand, it increases the computational cost.

The Algorithm 3 shows the BIRK scheme, where $\mathbf{D}(\mathbf{W}^0)$ is the block diagonal part of the full Jacobian matrix. The matrix $\left[\dfrac{\mathbf{M_P}}{\Delta t} + \alpha_m \mathbf{D}\left(\mathbf{W}^0\right)\right]$ is computed

Algorithm 2. The Line-Implicit Runge-Kutta (LIRK) scheme

1: $\mathbf{W}^0 = \mathbf{W}^n$
2: **for** $k = 1, m$ **do**
3: $\quad \left[\dfrac{\mathbf{M_P}}{\Delta t} + \alpha_m \mathbf{T}\left(\mathbf{W}^0\right)\right] \delta\mathbf{W}^k = -\alpha_k \mathbf{R}\left(\mathbf{W}^{k-1}\right)$
4: $\quad \mathbf{W}^k = \mathbf{W}^{k-1} + \delta\mathbf{W}^k$
5: **end for**
6: $\mathbf{u}^{n+1} = \mathbf{u}^m$

Algorithm 3. The Block-Implicit Runge-Kutta (BIRK) scheme

1: $\mathbf{W}^0 = \mathbf{W}^n$
2: **for** $k = 1, m$ **do**
3: $\quad \left[\dfrac{\mathbf{M_P}}{\Delta t} + \alpha_m \mathbf{D}\left(\mathbf{W}^0\right)\right] \delta\mathbf{W}^k = -\alpha_k \mathbf{R}\left(\mathbf{W}^{k-1}\right)$
4: $\quad \mathbf{W}^k = \mathbf{W}^{k-1} + \delta\mathbf{W}^k$
5: **end for**
6: $\mathbf{W}^{n+1} = \mathbf{W}^m$

only for the first stage if the smoother is used on the finest level, while, on coarser levels, it is just projected from the finest level.

The Jacobian of the LBE was recomputed only for the first smoothing iteration at each level and then was kept constant. Even if this choice decreased the effectiveness of the implicit smoother, it improved the robustness of the MG algorithms. The linear system was solved using the restarted GMRES algorithm preconditioned with the block Jacobi method (one block per process) as available in the PETSc library [20].

The time step for the LIRK/BIRK smoother was computed only on the finest level, because in the intermediate levels it was included in the matrix restricted from the finest level. The CFL law for the MG strategies is defined by the Eq.28.

6 Results

This section presents the results obtained in the computation of the compressible turbulent flows for different testcases considered in the IDIHOM EU project [40] with the solution strategies presented in the previous sections (implicit solver, h- and p-multigrid).

The robustness and reliability of the implicit solver was assessed in the computation of the subsonic/transonic turbulent flows around complex aeronautical configurations and through the NASA Rotor 37 with the standard k-$\widetilde{\omega}$ model. The influence of the different EARSM terms on the solution was also investigated. In particular the standard k-$\widetilde{\omega}$ model was compared with the EARSM in its linear and non-linear formulations for the computation of the subsonic turbulent flow around a delta wing.

The h- and p-multigrid algorithms were compared with the implicit solver, showing for both cases a speed-up in terms of CPU time needed to reach a converged solution. In particular the subsonic turbulent flow around a 2D multi-element airfoil was computed by using the h-MG, while the subsonic turbulent flow around a train-head and through a turbine cascade was computed by using the p-MG. For these testcases the standard k-$\widetilde{\omega}$ model was used.

The grids were obtained by means of an in-house agglomeration tool starting from fine linear grids.

The computing time is reported in the figures as a normalized value with respect to the TauBenchmark [33] value, $t_{TauBench}$, obtained on a full node of the cluster used for the CFD simulation[1]. The normalized computing time is measured in Work Units: $WU = t_{wall} * N_{cores}/t_{TauBench}$.

6.1 Full Aircraft FA5

This test case is about the flow at a high angle of attack around a 1:15 model of a delta-wing type generic fighter aircraft, dominated by strong vortices and large-scale turbulent flow phenomena. The farfield conditions of this test case were Mach number $\mathrm{M}_\infty = 0.85$, angle of attack $\alpha = 24°$ and Reynolds number based on the mean aerodynamic chord $\mathrm{Re}_{mac} = 46.5{\times}10^6$. At the engine intake a uniform pressure, $p/p_\infty = 1.2057$ and a mass flow ratio MFR $= 0.9$ were set, while at the engine exhaust free-stream conditions were imposed. The solution was computed up to $\mathbb{P}^2$ polynomial approximation on a grid consisting of 164636 hexahedral elements with quartic edges. The solver was not able to produce a fully converged solution, probably due to strongly nonlinear effects induced by the shock capturing approach implemented in the DG code MIGALE.

Figure 1 shows a detail of the surface grid and the surface pressure coefficient contours of the $\mathbb{P}^1$ solution. Figure 2 compares the C_p contours of the higher order $\mathbb{P}^2$ solution with the results of a second-order finite volume computation on a much finer grid. The improvement of the higher order solution is quite clear and supports the potential of high-order methods for such complex configurations. However, the comparison of $\mathbb{P}^1$, $\mathbb{P}^2$ and finite volume solutions seems to indicate that the DG solution most likely needs further h- or p-refinement.

6.2 Hovering Helicopter Rotor (HOTIS)

The hovering helicopter rotor was proposed and investigated experimentally within the HOTIS experiment in Braunschweig. The flow conditions were the Mach number at blade tip $M_{tip} = 0.633$, the Reynolds number at the blade tip $Re_{tip} = 1.5{\times}10^6$, and the air speed $v_a = 0.0$ m/s. The farfield boundary condition was perturbed with a jet and a potential sink at each iteration. In particular the velocity of the jet along the rotation axis is defined as:

$$u_j = -2|\mathbf{v}_{tip}|\sqrt{\frac{C_T}{2}},$$

[1] -n 250000 -s 10 define the reference TauBench workload for the hardware benchmark.

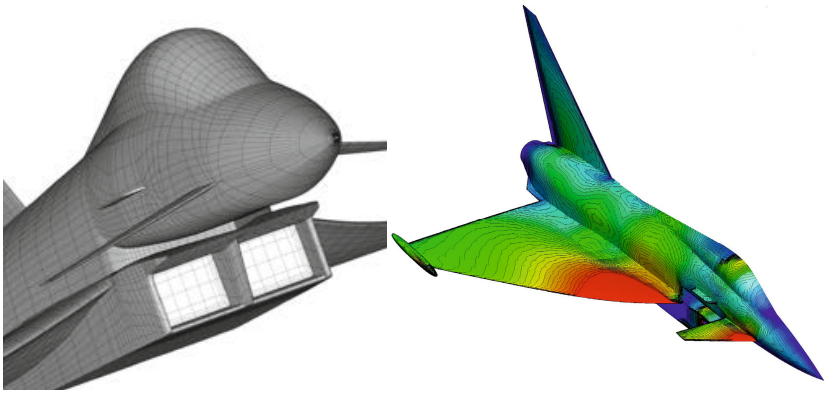


Fig. 1. FA5: grid consisting of 164636 hexahedral (quartic representation of the boundaries) elements (left) and surface pressure coefficient distribution for a $\mathbb{P}^1$ (right)

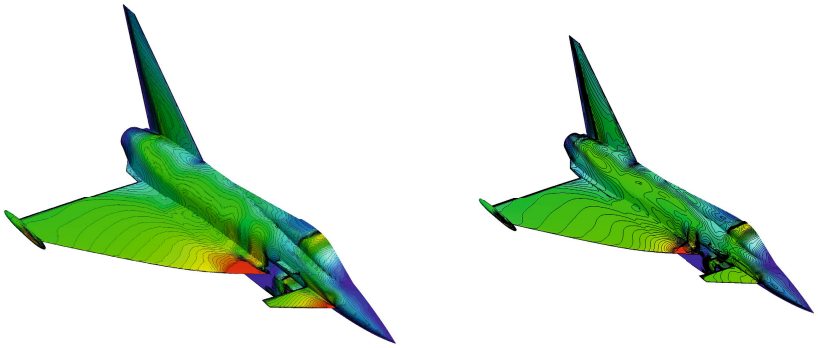


Fig. 2. FA5: comparison of the surface pressure coefficient distribution for a $\mathbb{P}^2$ solution (left) and for the industrial baseline computation conducted by Airbus D& S (right)

where $|\mathbf{v}_{tip}|$ is the velocity at the blade tip, and C_T the actual thrust coefficient. The velocity perturbation due to the potential sink is defined along the radial direction as:

$$u_r = -\frac{|\mathbf{v}_{tip}|}{4}\sqrt{\frac{C_T}{2}}\frac{R}{d}^2,$$

where R is the blade radius and d the distance of a generic farfield point from the rotational axis. Simulations were performed exploiting the relative reference frame formulation with a rotational speed $\omega_y = 1042$ r/min.

The solution was computed up to $\mathbb{P}^3$ on a grid consisting of 50176 hexahedral elements with quartic edges. Figure 3 shows a detail of the blade surface mesh (top) and the pressure contour on a blade for a $\mathbb{P}^3$ solution. Figure 4 shows a comparison between $\mathbb{P}^1$ and $\mathbb{P}^3$ solution in terms of iso contours of Q coloured with the vorticity magnitude. The computed thrust coefficient, $C_T(\mathbb{P}^3) = 2.71 \times 10^{-3}$, is in good agreement with the experimental value, $C_{T,exp} = 2.76 \times 10^{-3}$.

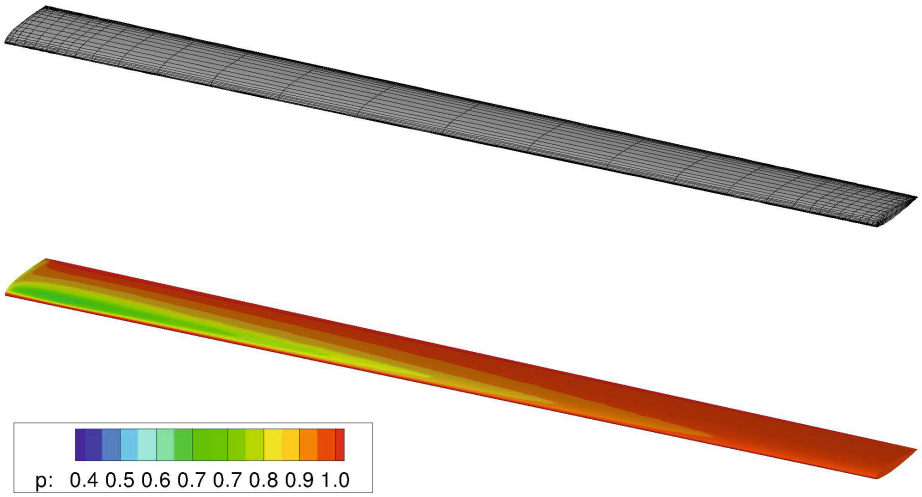


Fig. 3. HOTIS: surface grid of a blade (top) and pressure contour on a blade for a $\mathbb{P}^3$ solution (bottom)

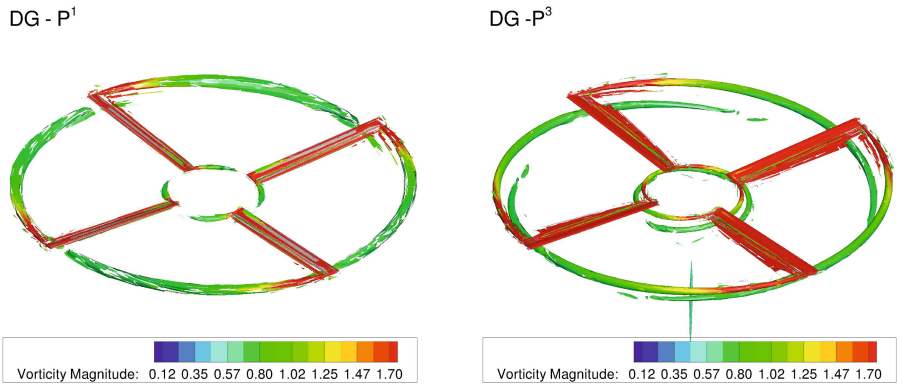


Fig. 4. HOTIS: iso contours of Q coloured with the vorticity magnitude for a $\mathbb{P}^1$ (left) and $\mathbb{P}^3$ (right) solution

6.3 NASA Rotor 37

The NASA Rotor 37 was thoroughly investigated both numerically and experimentally [2, 26].

Computations were performed up to $\mathbb{P}^3$ solution on a coarse grid composed of 20064 hexahedral elements with quartic polynomial functions for the representation of the boundaries. Figure 5 shows the surface grid (left) and the mesh details near the tip and the hub regions (right).

For this test case the governing equations have been written in a rotating coordinate system and include the acceleration contributions due to the rotating

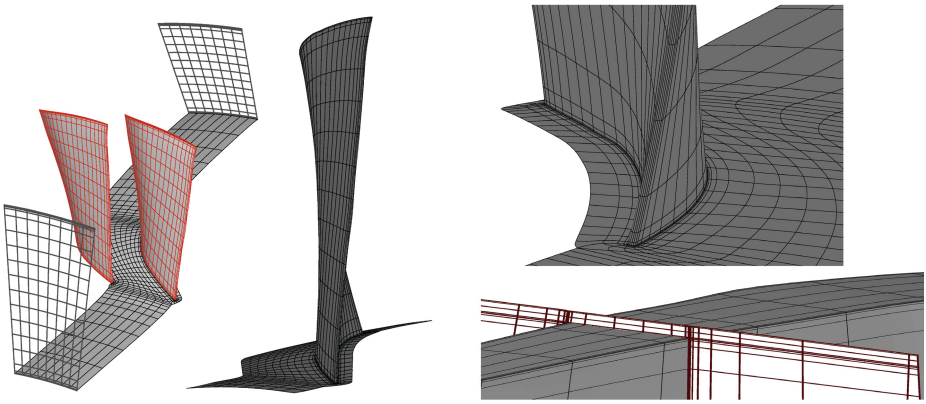


Fig. 5. Rotor 37: Surface mesh (left) and grid details of the hub and tip region (right)

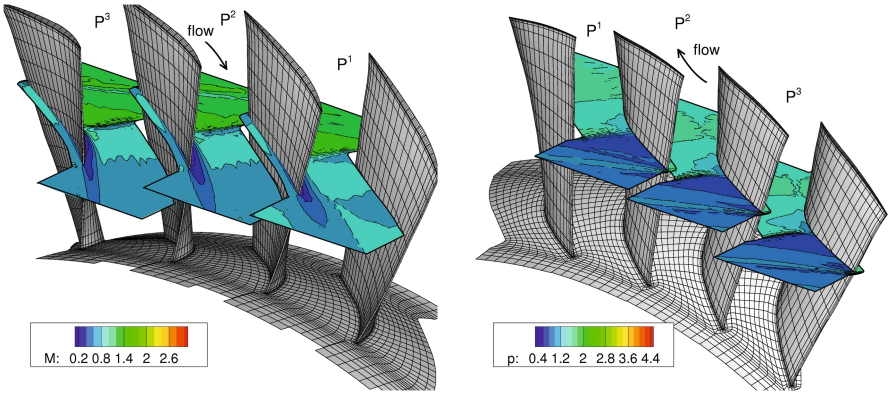
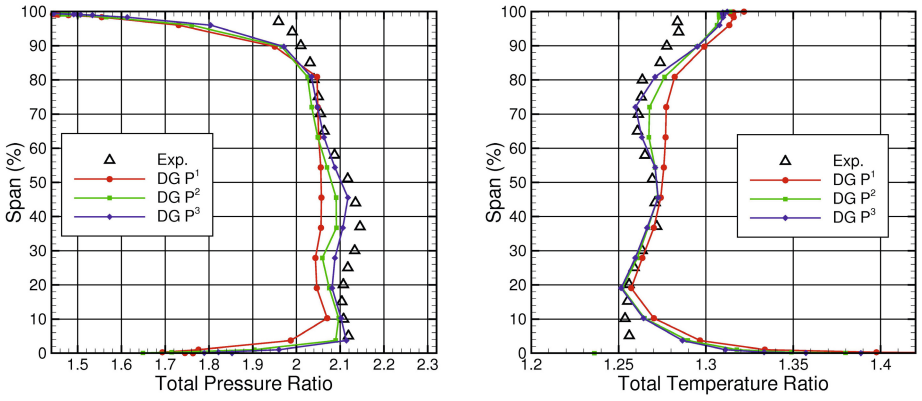


Fig. 6. Rotor 37: Relative Mach number (left) and pressure contours (right) at midspan, $\mathbb{P}^{1\to3}$ solutions

reference frame ($\omega_y = 1800$ rad/s). Blade walls and hub and tip surfaces were considered adiabatic. At inlet, the total pressure, total temperature, inflow angle $\alpha = 0°$ and turbulence intensity $\mathrm{Tu}_1 = 0.03$ were set. At outlet, the static pressure was prescribed at midspan and the spanwise pressure distribution was computed by means of a simplified radial equilibrium equation.

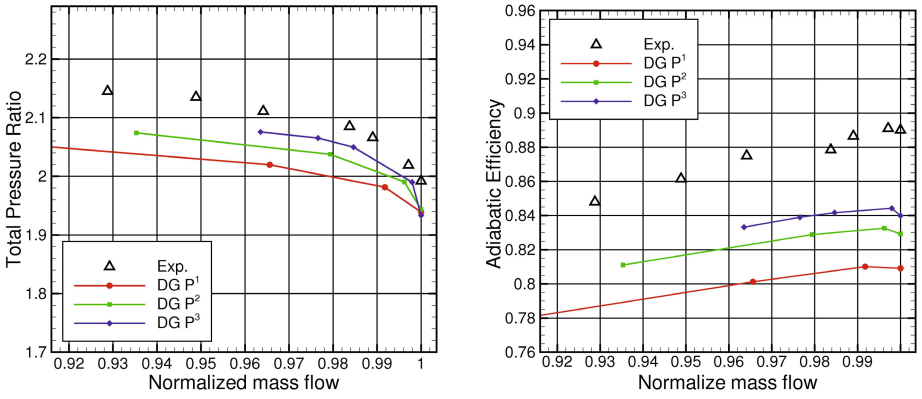
Figure 6 shows the main flow features at midspan. The shock wave originating at leading edge moves through the passage to the adjacent blade suction side, interacting with the boundary layer and entailing a flow separation which extends to the trailing edge. The radial distribution of the total pressure ratio, total temperature ratio and adiabatic efficiency at the outflow section are now considered to investigate the influence of the polynomial order on the solution. In Figure 7 the radial distribution of the pitchwise mass averaged $p_{0,2}/p_{0,1}$ and $T_{0,2}/T_{0,1}$ at 98% of the choked mass flow ($\mathbb{P}^{1\to3}$ solutions) are depicted and compared with experimental data. Good agreement of the $\mathbb{P}^3$ solution with experiments is

observed both for the pressure and temperature ratio profiles, ranging from 10% to about 90% of the span. Finally, the comparison of the overall performance maps for $\mathbb{P}^{1\to3}$ solutions with experimental data is shown in Figure 8. In particular, the pressure ratio (left) and the adiabatic efficiency (right) as functions of the normalized mass flow are depicted. The pressure ratio $p_{0,2}/p_{0,1}$ distributions indicate that for increasing polynomial orders, the computed results get closer to experiments. The $\mathbb{P}^3$ profile predicts fairly well the measured values, with a relative error of -2.8% at choke condition and of -1.7% for the nearest point to stall region, showing an overestimation of global losses. At choke condition the maximum error for η_{ad} is -5.5%, while in the stall region is about -4.7%.



(a) (b)

Fig. 7. Rotor 37: Pitch-wise total pressure ratio $p_{0,2}/p_{0,1}$ (left) and total temperature ratio $T_{0,2}/T_{0,1}$ (right), $\mathbb{P}^{1\to3}$ solution



(a) (b)

Fig. 8. Rotor 37: Total pressure ratio $p_{0,2}/p_{0,1}$ (left) and adiabatic efficiency η_{ad} (right) as a function of normalized mass flow, $\mathbb{P}^{1\to3}$ results and experiments

6.4 Delta Wing

The NASA 65° sweep delta wing was proposed and experimentally investigated within the second international Vortex Flow Experiment VFE-2, see [35]. The farfield conditions of this test case are $M_\infty = 0.4$, $\alpha = 13.3°$ and $Re_{mac} = 3\times10^6$. Solutions were computed using the standard k-$\widetilde{\omega}$ model and EARSM1→3 up to $\mathbb{P}^3$ polynomial approximation on a grid consisting of 13816 hexahedral elements with quadratic edges. The high-order grid was obtained by agglomerating a fine linear finite volume grid generated by [30].

Figure 9 compares the computed pressure coefficient contours on the wing surface with the experimental Pressure Sensitive Paint results reported in [29, 36]. The solution using the standard k-$\widetilde{\omega}$ model shows a delayed onset of the vortex developing along the wing with respect to the experimental data. Using EARSM,

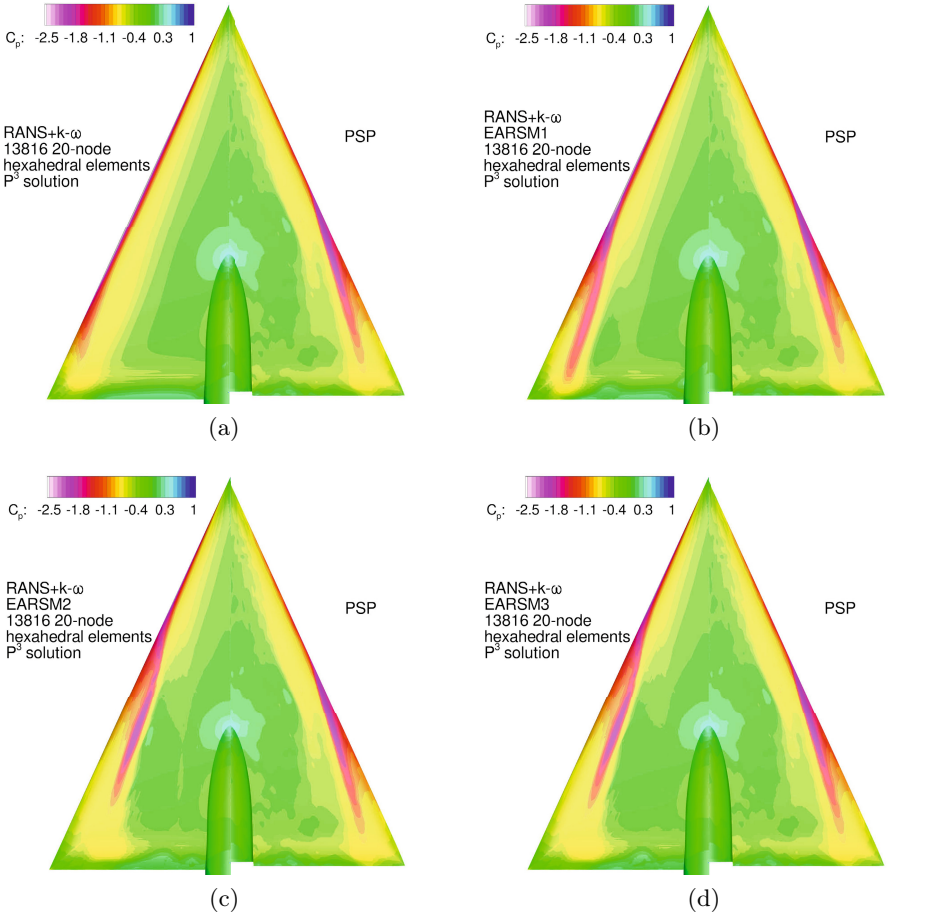
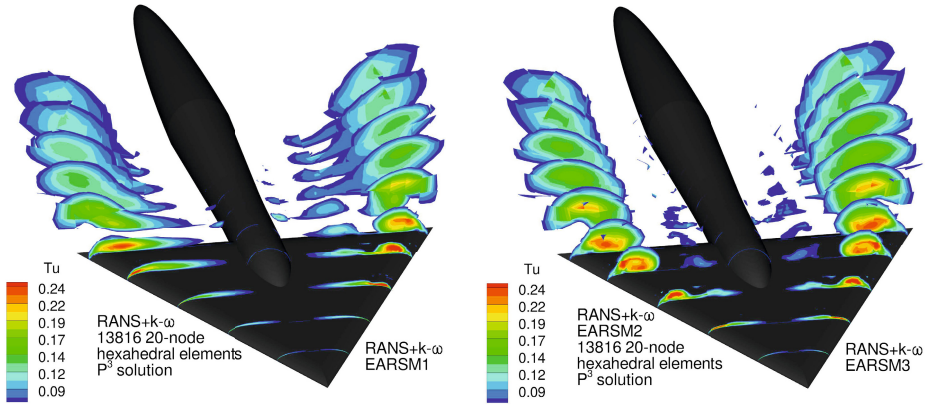


Fig. 9. VFE2: Pressure coefficient distribution, C_p. *Left side:* $\mathbb{P}^3$ results – *Right side:* Pressure Sensitive Paint, experimental data in [29]



(a) (b)

Fig. 10. VFE2: Turbulence intensity contours, $\mathbb{P}^3$ solutions

in its linear and non-linear versions, the vortex moves towards the wing apex showing also a sharper definition of turbulence intensity, as depicted in Figure 10. Overall, EARSM results appear to improve standard k-$\widetilde{\omega}$ predictions. In particular, the EARSM1 results are surprisingly good, while EARSM2-3 results are not as good. This behaviour needs further investigation, but we can already observe that the discrepancy with respect to experimental data is mainly due to an inaccurate prediction of the primary vortex origin. This issue requires closer numerical investigation of the flow behaviour around the nose of the wing.

6.5 2D Multi-element Airfoil (L1T2)

The flow around the three elements airfoil L1T2 was computed with the h-MG algorithm for a farfield Mach number $M_\infty = 0.197$, an angle of attack $\alpha = 20.18°$ and chord-based Reynolds number $Re_\infty = 3.52{\times}10^6$. This test case was computed on a hybrid mesh of 8969 elements with quadratic piecewise edges, generated with the high-order extension of an in-house mesh generator [34]. Three grid levels were adopted with an agglomeration ratio equal to 2 (8969→4696→2447), as depicted in Fig. 11.

The h-MG setting can be summarized as follows:

- p-sequencing on the coarsest mesh;
- LBE smoother on each level, GMRES method (15 iteration, no-restart) preconditioned with ILU(0) on each block;
- $l = h_{min}$, $\nu_c = 2$;
- $l > h_{min}$, $\nu_{1-2} = 1 - 1$.

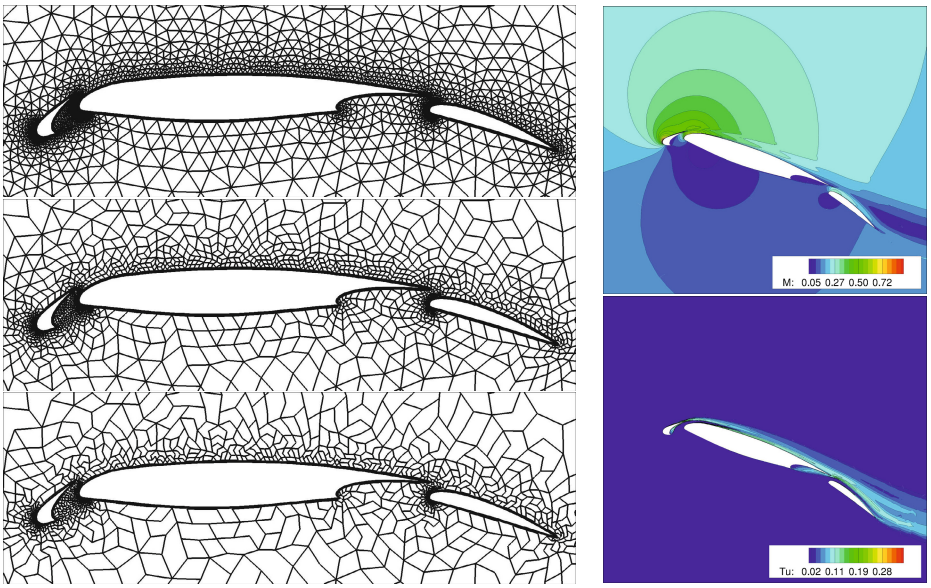


Fig. 11. L1T2: Three grid levels, agglomeration ratio equal 2, 8969 → 4696 → 2447 (left). Mach number and turbulence intensity contours for a $\mathbb{P}^2$ solution (right)

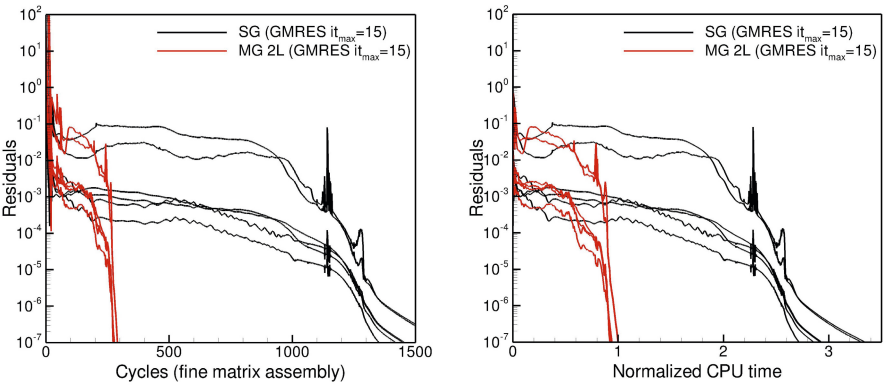


Fig. 12. L1T2: convergence history (L^2 norm of the residuals) in terms of iterations (left) and normalized CPU time (right) needed to reach a converged solution for the multigrid algorithm (MG 2L) and the implicit solver (SG), $\mathbb{P}^2$ soltuion

Figure 11 (right) shows the Mach number and turbulence intensity contours for a $\mathbb{P}^2$ solution. The comparison between the h-MG algorithm (MG 2L) and the implicit solver (SG) in terms of the iterations and the normalized CPU time (MG CPU time is the reference time) needed to reach a converged solution is depicted in Fig. 12. It is evident a great reduction of the CPU time due to the use of the h-MG strategy (around 66.6%) with respect to the implicit solver.

6.6 Train Head

The flow around a simplified train (single car) with the wind velocity $U_w = 70m/s$ and the Reynolds number based on the reference length $Re_l = 1.2 \times 10^6$ was considered. Computations were performed for a yaw angle $\beta = 10°$ on a grid with 7776 hexahedral elements (quartic representation of the boundaries). The linear structured grid used with the in-house agglomeration tool is generated using the meshing software from GridPro [37]. As the quadrature cost can be very expensive for a mesh with quartic edges, an error based adaptive procedure for the reduction of the degree of exactness of the quadrature formulae was used [31]. Figure 13 shows the surface mesh (left) and the streamline coloured with the turbulence intensity (right) for a $\mathbb{P}^4$ solution.

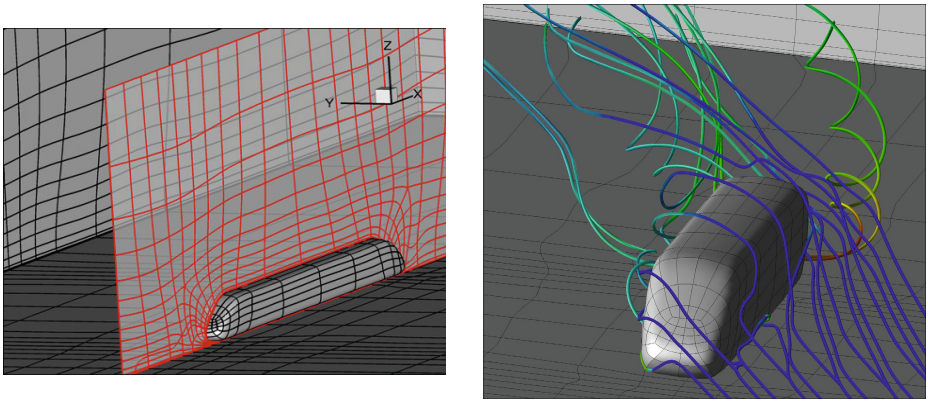


Fig. 13. Train: mesh (left) consisting of 7776 hexahedral elements with a quartic representation of the boundary and streamline coloured with the turbulence intensity for a $\mathbb{P}^4$ solution

The effect of different smoothers on the p-MG performance was investigated in this test case in terms of the memory usage (Mem) and the normalized CPU time (Time) needed to reach a converged solution. In particular two different sets of smoothers were considered: (i) the LIRK on the finest level, the LBE (GMRES method, 120 iteration, no-restart, preconditioned with ILU(0) on each block) on the coarsest level and the BIRK on the remaining levels, (ii) the LBE on every level (GMRES method, 120 iteration, no-restart, preconditioned with ILU(0) on each block).

The smoothing iterations adopted for the p-MG algorithm are as follows:

- $l = p_{min}$, $\nu_c = 5$;
- $l = p_{max}$, $\nu_{f,1-2} = 2 - 2$;
- $p_{min} < l < p_{max}$, $\nu_{1-2} = 3 - 3$.

Figure 14 shows the convergence history (L^2 norm of the density residual) for the p-MG with the two smoothers in terms of iterations (left) and non dimensional

CPU time needed to reach a converged solution (right). Table 1 summarizes performance for both strategies. The results show that the implicit smoother outperforms the line-implicit smoother in terms of CPU time, showing an average time reduction around 80%. However there is also a boost in the memory requirement, which increases with the polynomial order.

Table 1. Train: comparison of different smoothers for the p-MG algorithm. p-MG(LBE) is based on an implicit smoother (LBE) for each level. p-MG(LIRK) is based on the LIRK for the finest level, LBE for the coarsest level and the BIRK for the remaining levels. $\mathbb{P}^k$ is the solution approximation, Mem the memory usage, Δ_{RAM} the memory requirement reduction due to the use of different solution strategies, Time the normalized CPU time, Δ_{CPU} the computational time reduction due to the use of a different solution strategies (Memory usage and CPU time of the p-MG(LIRK) are taken as reference)

Scheme	$\mathbb{P}^k$	Mem [GB]	Δ_{RAM}	Time [IU]	Δ_{CPU}
p-MG(LIRK)	2	4.1	-	20449	-
p-MG(LBE)	2	12	+292%	5047	−75.3%
p-MG(LIRK)	3	12	-	194647	-
p-MG(LBE)	3	37.4	+311%	30427	−84.3%
p-MG(LIRK)	4	32	-	899902	-
p-MG(LBE)	4	122.4	+382%	191512	−78.7%

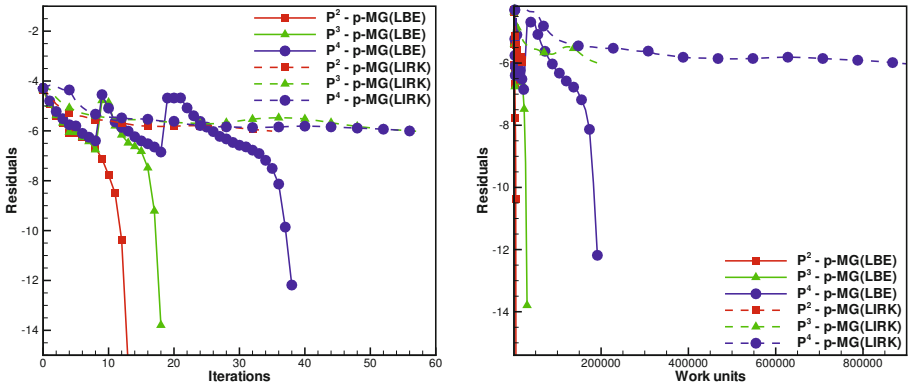


Fig. 14. Train: convergence history (L^2 norm of the density residual) as a function of the MG iterations (left) and of the normalized CPU time (right) for different polynomial solution approximations and smoothers

6.7 T106A Turbine Cascade

The T106A turbine cascade is a low-pressure turbine cascade designed by MTU Aero Engines, which has been extensively investigated in experimental and computational studies [21–23],

The computations were performed for a downstream isentropic Mach number $M_{2,is} = 0.59$, a Reynolds number based on the downstream isentropic conditions and on the blade chord $Re_{2,is} = 0.5 \times 10^6$, an inlet turbulence intensity $Tu_1 = 4.0\%$, and an inlet angles $\alpha_1 = 37.7°$. The grid used for the computations is shown in Figure 15 (left), consisting of 43200 hexahedral elements with quadratic edges.

The Mach number contours at midspan for different polynomial orders ($\mathbb{P}^{1\to3}$) are depicted in Figure 15 (right), while Figure 16 shows a comparison of the computed pressure coefficient distribution on the blade with experimental data for different polynomial approximations. On the pressure side all solutions are in good agreement with the experimental data. On the accelerating part of the suction side (right side of Figure 16) the experimental data are in good agreement with the C_p curve provided by the $\mathbb{P}^3$ solution, while the curve for the lower polynomial approximation is oscillating. In the rear part of the suction side there is a small laminar separation bubble, which is not captured by the turbulence model.

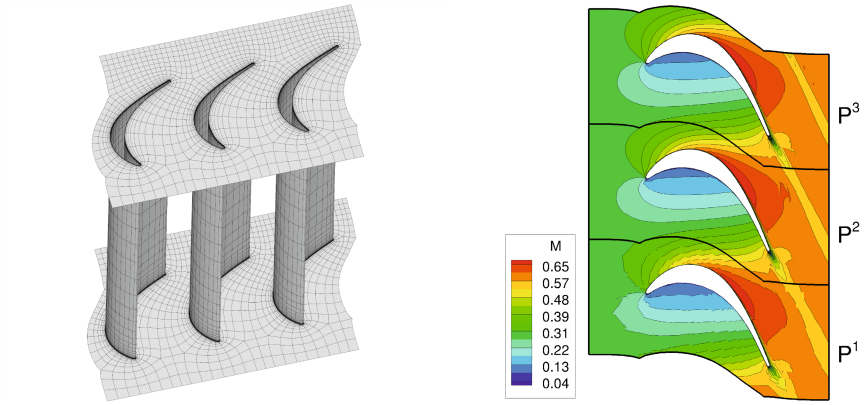


Fig. 15. T106A: mesh consisting of 43200 hexahedral elements with a quadratic representation of the boundary (left) and Mach number contours for different polynomial orders (right)

The comparison between the p-MG algorithm with the implicit smoother, p-MG(LBE), and the implicit solver, LBE, in terms of the memory usage (Mem) and the normalized CPU time (Time) needed to reach a converged solution is now described. The implicit smoother was chosen because as demonstrated for the previous testcase it guarantees the best performance in terms CPU time

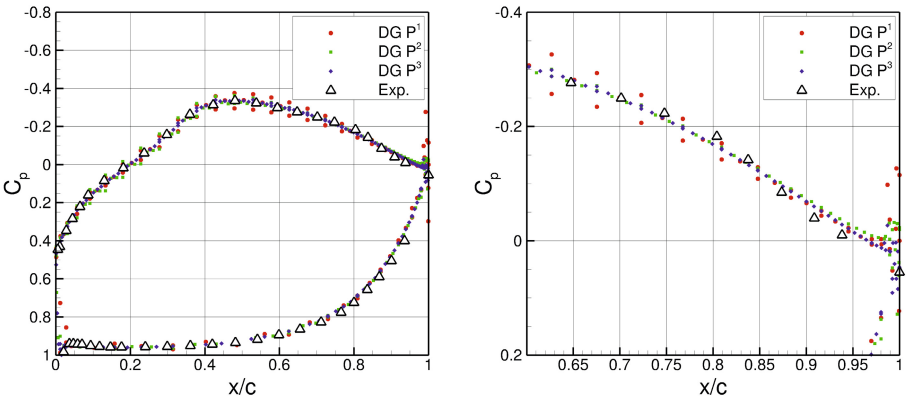


Fig. 16. T106A: comparison of the computed pressure coefficient distribution, C_p, on the blade for different solution approximations ($\mathbb{P}^{1\to 3}$) with experimental data (left). Detail of the C_p distribution in the rear part of the suction side

needed to reach a converged solution. The p-MG setting can be summarized as follows:

- LBE smoother on each level, GMRES method (120 iteration, no-restart) preconditioned with ILU(0) on each block;
- $l = p_{min}$, $\nu_c = 5$;
- $l = p_{max}$, $\nu_{f,1-2} = 2 - 2$;
- $p_{min} < l < p_{max}$, $\nu_{1-2} = 3 - 3$.

Figure 17 (left) shows a comparison of the convergence history (L^2 norm of the density residual) for the p-MG and the implicit solver as a function of the non dimensional CPU time needed to reach a converged solution. Table 2 summarizes performance for both strategies. The results show that the p-MG algorithm outperforms the implicit scheme for every solution approximation, both in terms of CPU time needed to reach a converged solution and memory requirement. However the different memory requirement is only due to a different setting of the GMRES parameters (the p-MG employs a small number of iteration and a higher tolerance). In particular the decrease in term of memory usage is around -35%, while the CPU time has been reduced by 79% and 82% for a $\mathbb{P}^2$ and a $\mathbb{P}^3$ solution, respectively. Also in this case to study the asymptotic behaviour of the density residual L^2 norm, the FMG solution with the LBE smoother has been computed for a $\mathbb{P}^2$ and $\mathbb{P}^3$ spatial discretization, as depicted in Figure 17 (right side).

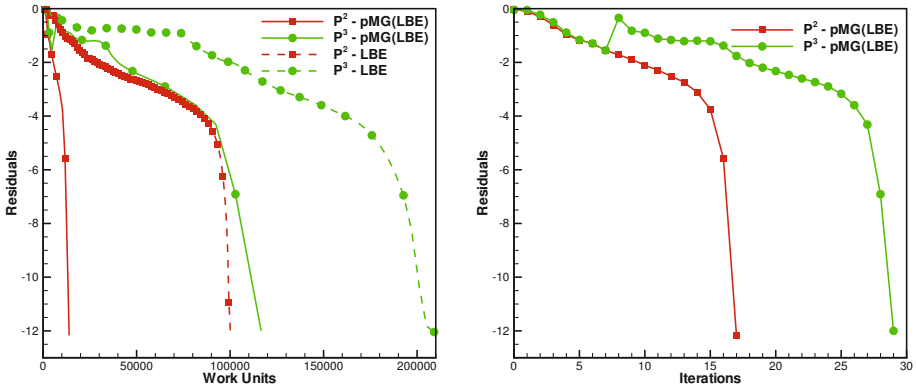


Fig. 17. T106A: convergence history (L^2 norm of the density residual) versus normalized CPU time for different polynomial solution approximations and strategy (left). Convergence history (L^2 norm of the density residual) versus MG cycles for a $\mathbb{P}^{2\to3}$ approximation (right)

Table 2. T106A: comparison of p-MG and implicit strategies performance. $\mathbb{P}^k$ is the solution approximation, Mem the memory usage, Δ_{RAM} the memory requirement reduction due to the use of different solution strategies, Time the normalized CPU time, Δ_{CPU} the computational time reduction due to the use of a different solution strategies (Memory usage and CPU time of the implicit scheme are taken as reference)

Scheme	$\mathbb{P}^k$	Mem [GB]	Δ_{RAM}	Time [IU]	Δ_{CPU}
LBE	2	63	-	29994	-
p-MG(LBE)	2	44	−30.1%	6218	−79.2%
LBE	3	244	-	223776	-
p-MG(LBE)	3	150	−38.5%	38915	−82.6%

7 Conclusions

In this chapter we demonstrated the capability of the DG code MIGALE to provide high-order solutions for complex subsonic/transonic turbulent flows. The potential of the high-order solver in the computation of three subsonic/transonic testcases (FA5, HOTIS and Rotor37) with very coarse meshes was shown, obtaining results in good agreement with reference solutions and experimental data. The progresses in terms of prediction capabilities and time integration techniques achieved within the IDIHOM project were also assessed. In particular the advantages of the EARSM in the prediction of the pressure coefficient distribution around a delta wing were demonstrated. Finally the efficiency of the h- and p-multigrid technique with respect to an implicit solver were shown in the computation of 2D and 3D turbulent subsonic testcases (L1T2, train-head, T106A).

Acknowledgements. This work was carried out within the EU FP7 IDIHOM project. The authors wish to thank Prof. Norbert Kroll for providing access to the HPC facilities at DLR.

References

1. Bassi, F., Botti, L., Colombo, A., Rebay, S.: Agglomeration based discontinuous Galerkin discretization of the Euler and Navier–Stokes equations. Computers & Fluids 61, 77–85 (2012)
2. Bassi, F., Botti, L., Colombo, A., Crivellini, A., Franchina, N., Ghidoni, A., Rebay, S.: Very high-order discontinuous Galerkin computations of transonic turbulent flows on aeronautical configurations. Notes on Fluid Mechanics 113, 25–38 (2010)
3. Hartmann, R., Held, J., Leicht, T., Prill, F.: Discontinuous Galerkin methods for computational aerodynamics – 3d adaptive flow simulation with the DLR PADGE code. Aerospace Science and Technology 14(7), 512–519 (2010)
4. Modisette, J.M., Darmofal, D.L.: An output–based adaptive and higher-order method for a rotor in hover. AIAA Paper 7343. AIAA Paper 7343, AIAA (2008), doi:10.2514/6.2008-7343
5. Cantwell, C., Sherwin, S., Kirby, R., Kelly, P.: From h to p efficiently: Strategy selection for operator evaluation on hexahedral and tetrahedral elements. Comput. & Fluids 43, 23–28 (2011)
6. Bassi, F., Franchina, N., Ghidoni, A., Rebay, S.: A numerical investigation of a spectral-type nodal collocation discontinuous Galerkin approximation of the Euler and Navier-Stokes equations. Int. J. Numer. Meth. Fluids 71, 1322–1339 (2013)
7. Bassi, F., Franchina, N., Ghidoni, A., Rebay, S.: Spectral p-multigrid discontinuous Galerkin solution of the Navier-Stokes equations. Int. J. Numer. Meth. Fluids 67, 1540–1558 (2011)
8. Crivellini, A., Bassi, F.: An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamics simulations. Comput. & Fluids 50(1), 81–93 (2011)
9. Wallraff, M., Leicht, T., Lange-Hegermann, M.: Numerical flux functions for Reynolds-averaged Navier-Stokes and k-ω turbulence model computations with a line-preconditioned p-multigrid discontinuous Galerkin solver. International Journal for Numerical Methods in Fluids 71(8), 1055–1072 (2013), doi:10.1002/fld.3702
10. Wilcox, D.C.: Turbulence Modelling for CFD. DCW industries Inc., La Canada (1993)
11. Bassi, F., Crivellini, A., Rebay, S., Savini, M.: Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations. Comput. & Fluids 34, 507–540 (2005)
12. Menter, F.R.: Two-equation eddy-viscosity turbulence model for engineering applications. AIAA Journal 32(8), 1598–1605 (1994)
13. Bassi, F., Botti, L., Colombo, A., Ghidoni, A., Rebay, S.: Discontinuous Galerkin for turbulent flows. In: Wang, Z.J. (ed.) Adaptive High-Order Methods in Computational Fluid Dynamics, ch. 3. Advances in Computational Fluid Dynamics, pp. 1–32. World Science Books (2011), `http://www.worldscibooks.com/mathematics/7792.html`
14. Bassi, F., Botti, L., Colombo, A., Di Pietro, D., Tesini, P.: On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. Journal of Computational Physics 231(1), 45–65 (2012), doi:http://dx.doi.org/10.1016/j.jcp.2011.08.018

15. Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., Savini, M.: A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere, R., Dibelius, G. (eds.) 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, pp. 99–108. Technologisch Instituut, Antwerpen (1997)
16. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. 39(5), 1749–1779 (2002)
17. Gottlieb, J., Groth, C.: Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. J. Comput. Phys. 78, 437–458 (1988)
18. Hänel, D., Schwane, R., Seider, G.: On the accuracy of upwind schemes for the solution of the Navier–Stokes equations. In: Proceedings of the AIAA 8th Computational Fluid Dynamics Conference, AIAA Paper, CP, AIAA, pp. 87–1105 (July 1987)
19. ADIGMA, Adaptive higher-order variational methods for aerodynamic applications in industry, Specific Targeted Research Project no. 30719 supported by European Commison, `http://www.dlr.de/as/en/Desktopdefault.aspx/tabid-2035/2979_read-4582/`
20. Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory (2013)
21. Hoheisel, H.: Entwicklung neuer entwurfskonzepte fur zwei turbinengitter, teil iii, ergebnisse T106, Technical report, Institut fur Entwurfsaerodynamik, Braunschweig (1981)
22. Ghidoni, A., Colombo, A., Rebay, S., Bassi, F.: Simulation of the transitional flow in a low pressure gas turbine cascade with a high-order discontinuous Galerkin method. Journal of Fluids Engineeing 135(7), doi:10.1115/1.4024107
23. Bassi, F., Botti, L., Colombo, A., Ghidoni, A., Rebay, S.: Investigation of near-wall grid spacing effect in High-Order Discontinuous Galerkin RANS computations of turbomachinery flows. In: Azaïez, M., EI Fekih, H., Hesthaven, J.S. (eds.) Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2012. LNCSE, vol. 95, pp. 125–134. Springer, Heidelberg (1980)
24. Wallin, S., Johansson, A.V.: An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. J. Fluid Mech. 403, 89–132 (2000)
25. Dunham, J.: CFD validation for propulsion system components, Tech. Rep. AGARD AR-355, AIAA (1994)
26. Chima, R.: Calculation of tip clearance effects in a transonic compressor rotor (96-GT-114)
27. IDIHOM, Industrialisation of High-Order Methods – a top-down approach, Specific Targeted Research Project supported by European Commison, `http://www.dlr.de/as/en/desktopdefault.aspx/tabid-7027/11654_read-27492/`
28. Ghidoni, A., Colombo, A., Bassi, F., Rebay, S.: Efficient p-multigrid discontinuous Galerkin solver for complex viscous flows on stretched grids. International Journal for Numerical Methods in Fluids 75(2), 134–154 (2014)
29. Konrath, R., Klein, C., Schrøder, A.: PSP and PIV investigations on the VFE-2 configuration in sub- and transonic flow. AIAA Paper 2008-379 (2008)
30. Crippa, S.: Advances in vortical flow prediction methods for design of delta-winged aircraft, Trita-AVE 2008:30. KTH, Aeronautical and Vehicle Engineering (2008)
31. Tesini, P.: An h-multigrid approach for high-order Discontinuous Galerkin methods. Università degli Studi di Bergamo (2008)

32. Bassi, F., Botti, L., Colombo, A.: Agglomeration based physical frame DG discretizations: an attempt to be mesh free. Mathematical Models and Methods in Applied Sciences (in Press, 2014), doi:10.1142/S0218202514400028
33. Integrated Performance Analysis of Computer Systems – Benchmarks for distributed computer systems (2005), http://www.ipacs-benchmark.org
34. Ghidoni, A., Pelizzari, E., Rebay, S., Selmin, V.: 3D anisotropic unstructured grid generation. Int. J. Numer. Meth. Fl. 51, 1097–1115 (2006)
35. Hummel, D.: The second international vortex flow experiment (VFE-2): Objectives and first results. J. Aerosp. Eng. 220(6), 559–568 (2006); 2nd International Symposium on Integrating CFD and Experiments in Aerodynamics
36. Konrath, R., Klein, C., Schrøder, A.: PSP and PIV investigations on the VFE-2 configuration in sub- and transonic flow. Aerosp. Sci. Technol. 24(1), 22–31 (2013)
37. GridPro 5.5.1, http://www.gridpro.com

Time Integration in the Discontinuous Galerkin Code MIGALE - Unsteady Problems

F. Bassi[1], L. Botti[1], A. Colombo[1], A. Crivellini[2], A. Ghidoni[3], A. Nigro[4], and S. Rebay[3]

[1] Universtità degli Studi di Bergamo, Engineering Department, 24044 Dalmine (BG), Italy
{francesco.bassi,lorenzo.botti,alessandro.colombo}@unibg.it
[2] Università Politecnica delle Marche, Department of Industrial Engineering and Mathematical Sciences, 60100 Ancona, Italy
a.crivellini@univpm.it
[3] Università degli Studi di Brescia, Department of Mechanical and Industrial Engineering, 25123 Brescia, Italy
{antonio.ghidoni,stefano.rebay}@unibs.it
[4] Università della Calabria, Department of Mechanical, Energy and Management Engineering, 87036 Rende (CS), Italy
alessandra.nigro@unical.it

Abstract. This chapter presents recent developments of a high-order Discontinuous Galerkin (DG) method to deal with unsteady simulation of turbulent flows by using high-order implicit time integration schemes. The approaches considered during the IDIHOM project were the Implicit Large Eddy Simulation (ILES), where no explicit subgrid-scale (SGS) model is included and the DG discretization itself acts like a SGS model, and two hybrid approaches between Reynolds-averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES) models, namely the Spalart-Allmaras Detached Eddy Simulation (SA-DES) and the eXtra-Large Eddy Simulation (X-LES). Accurate time integration is based on high-order linearly implicit Rosenbrock-type Runge-Kutta schemes, implemented in the DG code MIGALE up to sixth-order accuracy. Several high-order DG results of both incompressible and compressible 3D turbulent test cases proposed within the IDIHOM project demonstrate the capability of the method.

Keywords: Discontinuous Galerkin, unsteady flows, Rosenbrock-type Runge–Kutta schemes, ILES, hybrid RANS-LES, SA-DES, X-LES.

1 Introduction

In recent years, several approaches to the underresolved simulation of turbulent flows have been proposed to deal with high-fidelity simulations of flow conditions, like massively separated flows, that are poorly predicted by the Reynolds-averaged Navier-Stokes (RANS) equations with first-moment closures.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_11

For not-too-high Reynolds number flows, some recent papers have shown that the Implicit Large Eddy Simulation (ILES) approach seems well suited for a Discontinuous Galerkin (DG) discretization, due to the favourable dissipation and dispersion properties of this numerical method. Governing equations of ILES do not include any explicit sub-grid scale (SGS) model for the non-resolved turbulent scales. On the other hand, for those high Reynolds number flows where RANS models suffer from severe modelling limitations and Large Eddy Simulation (LES) is still too computationally expensive or even impossible, hybrid approaches between RANS and LES can play a significant role. In hybrid RANS-LES approaches the RANS model is used very close to solid walls, where LES would be prohibitively costly, while LES is performed in regions of separated flow where larger eddies are properly resolved by the numerical solution.

In this work we present the main features of the implementation in the DG code MIGALE, [7], of two hybrid RANS-LES models, namely the Spalart-Allmaras Detached Eddy Simulation (SA-DES) model of Spalart et al., [52], and the eXtra-Large Eddy Simulation (X-LES) model of Kok et al., [42]. In the SA-DES approach the simulation switches between RANS and LES modes by simply replacing the wall-distance parameter of the model with the minimum between this distance and a mesh size representative of the LES filter width. In the X-LES approach a composition of the two-equation TNT k-ω turbulence model, [41], and a k-equation SGS model is used. Among other hybrid approaches, X-LES has two attractive characteristics, *i.e.*, the use in LES mode of a clearly defined SGS model based on the k-equation and the independence of the wall distance, which is an ambiguous parameter for complex geometries.

A high-order implicit time integration approach seems well suited to be coupled with high-order DG space discretizations of hybrid RANS-LES models equations. In fact implicit methods can be very efficient in near wall RANS regions, characterized by highly-stretched grids, and, if accurate enough, they are also able to capture fine details of unsteady motions in LES regions farther from walls, even using large time steps. Here we focus on accurate time integration by means of high-order linearly implicit Rosenbrock-type Runge-Kutta schemes, implemented in the code MIGALE up to sixth-order accuracy. The main advantages of Rosenbrock schemes are that they have excellent stability properties, are self-starting and can use variable time steps, and the Jacobian matrix needs to be assembled and factored only once per time step. Other approaches to high-order implicit time integration, partially developed within the IDIHOM project, have been implemented in the DG code MIGALE and presented elsewhere, [31,48,49].

The equations of all models implemented in the DG code MIGALE are discretized to the same high-order accuracy, in the framework of a DG discretization for hybrid type elements, based on hierarchical and orthonormal polynomial basis functions, local to each element and defined in the physical space, [6]. All implementations employ analytically derived Jacobian matrices, which take full account of the dependence of the residuals on the unknowns and on their derivatives, including the implicit treatment of lifting operators and of boundary conditions.

This work summarizes the progress, achieved within the IDIHOM project, [40], on improved physical models and accurate time integration schemes implemented in the DG code MIGALE, and demonstrates the ability of the code to compute high-order unsteady flow solutions of several challenging test cases.

2 Governing Equations

In this section we review some details of the SA-DES and X-LES models implementation. Currently, the SA-DES model has been implemented in the incompressible DG code MIGALE, while the X-LES model has been implemented in the compressible version of the code. In the following, the two sets of governing equations are presented in turn.

2.1 SA-DES Equations

The incompressible flow equations of a hybrid SA-DES formulation can be written as

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right], \tag{2}$$

$$\frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial}{\partial x_j}(u_j \tilde{\nu}) = \frac{1}{\sigma}\frac{\partial}{\partial x_j}\left(\xi \frac{\partial \tilde{\nu}}{\partial x_j}\right) + s, \tag{3}$$

where the following definitions

$$\nu_t = f_{v1} \max(0, \tilde{\nu}), \tag{4}$$

$$\xi = \begin{cases} \nu(1+\chi) & \text{if } \chi \geq 0 \\ \nu & \text{if } \chi < 0 \end{cases}, \tag{5}$$

$$s = \begin{cases} c_{b1}\tilde{S}\tilde{\nu} - c_{w1} f_w \left(\frac{\tilde{\nu}}{\tilde{d}}\right)^2 + \frac{c_{b2}}{\sigma}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i} & \text{if } \chi \geq 0 \\ 0 & \text{if } \chi < 0 \end{cases}, \tag{6}$$

take account of possibly negative values of $\tilde{\nu}$ and of $\chi = \tilde{\nu}/\nu$. The set of closure functions is then completed by the following definitions

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 \tilde{d}^2} f_{v2}, \qquad S = \sqrt{2\Omega_{ij}\Omega_{ij}}, \qquad \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right),$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \qquad f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}},$$

$$g = r_r + c_{w2}\left(r_r^6 - r_r\right), \qquad r = \frac{\tilde{\nu}}{\tilde{S}(\kappa \tilde{d})^2}, \qquad r_r = \begin{cases} \min(r, r_{max}) & \text{if } r \geq 0 \\ r_{max} & \text{if } r < 0 \end{cases},$$

and model constants

$$c_{b1} = 0.1355, \qquad c_{b2} = 0.622, \qquad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma},$$
$$c_{v1} = 7.1, \qquad c_{w2} = 0.3, \qquad c_{w3} = 2,$$
$$\sigma = \frac{2}{3} \qquad \kappa = 0.41, \qquad r_{max} = 10.$$

In [52] the authors replaced the distance d from the nearest wall, introduced in the original formulation, [53], by a modified distance $\tilde{d}$, that involves the length scale $C_{des}\Delta$ and allows for DES computations, given by

$$\tilde{d} = \min\left(d, C_{des}\Delta\right). \tag{7}$$

Here C_{des} is a model constant equal to 0.65 and Δ is the largest spacing in the coordinate directions in each grid cell, given by $\Delta = \max\left(\Delta_x, \Delta_y, \Delta_z\right)$. With this modification the model reduces to the SA closure of the RANS equations when $d \ll \Delta$ and to a SGS model when $\Delta \ll d$.

The limitation on $\tilde{\nu}$ avoids numerical instabilities due to negative values of $\tilde{\nu}$ produced by higher-order solutions near the edge of boundary layers, where the $\tilde{\nu}$ profile displays a sharp transition to the freestream value of $\tilde{\nu}$, usually close to zero. Other forms of limitation have been proposed in the literature to address the issue of negative $\tilde{\nu}$ values, see, *e.g.*, [50, 51]. The approach here outlined is on the lines of an analogous modification for the k equation in the DG implementation of the k-ω model introduced by Bassi et al. in [10].

The modified function r, denoted by r_r, accounts for the "realizability" condition introduced by Crivellini et al. in [20]. According to the original SA formulation, the function r was defined as $r = l^2/(\kappa d)^2$, where l, a mixing length inspired by the algebraic models, should be strictly positive. However, writing the original r formulation as

$$r = \frac{\chi}{\frac{S(\kappa d)^2}{\nu} + \chi f_{v2}\left(\chi\right)},$$

it turns out that the function r can take negative values even if χ is positive. This happens when f_{v2} is negative, *i.e.*, when $1.003 \leq \chi \leq 18.4$ and $S(\kappa d)^2/\nu \leq 6.088$. The modified definition of r consists in limiting to a maximum value $r_{max} = 10$ both negative and large, positive, values of r. The reason for setting an upper bound on r can be understood considering that r becomes negative passing through a singular point where $r \to \pm\infty$. However, the upper bound on r practically does not affect the model behaviour since it was observed by Spalart et al. in [53] that $r > 1$ only occurs in adverse pressure gradient regions and then rarely it goes beyond $r = 1.1$.

2.2 X-LES Equations

The compressible flow equations of a hybrid X-LES formulation can be written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0, \tag{8}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \widehat{\tau}_{ji}}{\partial x_j}, \tag{9}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j H) = \frac{\partial}{\partial x_j}\left(u_i \widehat{\tau}_{ij} - \widehat{q}_j\right) - P_k + D_k, \tag{10}$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \overline{\mu}_t)\frac{\partial k}{\partial x_j}\right] + P_k - D_k, \tag{11}$$

$$\frac{\partial}{\partial t}(\rho \widetilde{\omega}) + \frac{\partial}{\partial x_j}(\rho u_j \widetilde{\omega}) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_j}\right] + (\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_k}\frac{\partial \widetilde{\omega}}{\partial x_k} + P_\omega - D_\omega + C_D, \tag{12}$$

where the pressure, the turbulent and total stress tensors, the heat flux vector and the limited values of eddy viscosity and of turbulent kinetic energy are given by

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2}u_k u_k\right), \tag{13}$$

$$\tau_{ij} = 2\overline{\mu}_t\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho \overline{k}\delta_{ij}, \tag{14}$$

$$\widehat{\tau}_{ij} = 2\mu\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) + \tau_{ij}, \tag{15}$$

$$\widehat{q}_j = -\left(\frac{\mu}{\mathrm{Pr}} + \frac{\overline{\mu}_t}{\mathrm{Pr}_t}\right)\frac{\partial h}{\partial x_j}, \tag{16}$$

$$\overline{\mu}_t = \alpha^* \frac{\rho \overline{k}}{\widehat{\omega}}, \quad \overline{k} = \max(0, k). \tag{17}$$

Here γ is the ratio of gas specific heats, Pr and Pr_t are the molecular and turbulent Prandtl numbers and

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

is the mean strain-rate tensor. The production, destruction and cross-diffusion terms in Eqs. (10), (11) and (12) are given by

$$P_k = \tau_{ij} \frac{\partial u_i}{\partial x_j}, \tag{18}$$

$$D_k = \beta^* \rho \overline{k} \hat{\omega}, \tag{19}$$

$$P_\omega = \alpha \left[\alpha^* \frac{\rho}{e^{\widetilde{\omega}_r}} \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho \delta_{ij} \right] \frac{\partial u_i}{\partial x_j} \tag{20}$$

$$D_\omega = \beta \rho \overline{k} e^{\widetilde{\omega}_r}, \tag{21}$$

$$C_D = \sigma_d \frac{\rho}{e^{\widetilde{\omega}_r}} \max \left(\frac{\partial k}{\partial x_k} \frac{\partial \widetilde{\omega}}{\partial x_k}, 0 \right), \tag{22}$$

where the closure parameters α^*, β^*, σ^*, α, β, σ, σ_d are those of Kok's TNT k-ω model, [41]. The hybrid X-LES formulation is based on a composite length scale, $\tilde{l}$, deriving from the $l_{k\text{-}\widetilde{\omega}}$ and l_{sgs} length scales of the k-$\widetilde{\omega}$ and SGS models, given by

$$l_{k\text{-}\widetilde{\omega}} = \frac{\sqrt{\overline{k}}}{e^{\widetilde{\omega}_r}}, \qquad l_{sgs} = C_1 \Delta, \tag{23}$$

where Δ is the SGS filter width and $C_1 = 0.05$. The composite length scale $\tilde{l}$ is then defined as

$$\tilde{l} = \min \left(l_{k\text{-}\widetilde{\omega}}, l_{sgs} \right) = \sqrt{\overline{k}} \min \left(\frac{1}{e^{\widetilde{\omega}_r}}, \frac{C_1 \Delta}{\sqrt{\overline{k}}} \right). \tag{24}$$

For implementation convenience, the composite length scale $\tilde{l}$ can be equivalently replaced by a composite specific dissipation rate $\hat{\omega}$ given by

$$\hat{\omega} = \frac{\sqrt{\overline{k}}}{\tilde{l}} = \max \left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1 \Delta} \right). \tag{25}$$

The implementation of Eqs. (11) and (12) follows the approach presented in [10], where $\widetilde{\omega} = \log(\omega)$ is used in place of ω to obtain a better behavior near solid walls and to guarantee the positivity of ω. The limited value $\overline{k}$ is used to deal with possibly negative values of the turbulent kinetic energy. Furthermore, the use of $\widetilde{\omega}_r$ instead of $\widetilde{\omega}$ indicates that $\widetilde{\omega}$ must fulfill suitably defined "realizability" conditions for the turbulent stresses, which set a lower bound on $\widetilde{\omega}$. According to the "slightly-rough-wall" boundary condition of Wilcox, [58], or to the popular formula proposed by Menter, [46], finite values of $\widetilde{\omega}$ are prescribed at solid walls. Inspired by these approaches and accounting for the polynomial degree k of the numerical solution, we set at the wall the value $\widetilde{\omega}_w$ defined by the first k terms of the Taylor series expansion of the analytical near-wall solution of $\widetilde{\omega}$ (with $\widetilde{\omega}_w \to \infty$) around a distance h, *i.e.*,

$$\widetilde{\omega}_w^k = \log \left(\frac{6\nu_w}{\beta \left(\alpha^k h \right)^2} \right), \tag{26}$$

where

$$\alpha^k = e^{-\sum_{n=1}^{k} \frac{1}{n}}. \tag{27}$$

In Eq. (26) h is set equal to the distance from the wall of the first cell centroid. More details on the implementation can be found in [7].

3 The DG Discrete Setting

Let $\mathcal{T}_h = \{K\}$ denote a mesh of the domain $\Omega \in \mathbb{R}^d, d \in \{2,3\}$ consisting of non-overlapping arbitrarily shaped elements K such that

$$\overline{\Omega}_h = \bigcup_{K \in \mathcal{T}_h} \overline{K}. \tag{28}$$

Following the idea to define discrete polynomial spaces in physical coordinates, see, *e.g.*, [8, 13–15, 24, 30], we consider DG approximations based on the space

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \left\{ v_h \in L^2(\Omega) \,|\, v_{h|K} \in \mathbb{P}_d^k(K),\, \forall K \in \mathcal{T}_h \right\}, \tag{29}$$

where k is a non-negative integer and $\mathbb{P}_d^k(K)$ denotes the restriction to K of the polynomial functions of d variables and total degree $\leq k$. To build a satisfactory basis for the space (29) we rely on the procedure presented in [55], see also [6,23], allowing to obtain orthonormal and hierarchical basis functions by means of the modified Gram-Schmidt (MGS) algorithm. The starting set of basis functions for the MGS algorithm are the monomials defined over each elementary space $\mathbb{P}_d^k(K)$, $K \in \mathcal{T}_h$, in a reference frame relocated in the element barycenter and aligned with the principal axes of inertia of K. For the sake of presenting the DG discretization, we introduce the set $\mathcal{F}_h$ of the mesh faces $\mathcal{F}_h \stackrel{\text{def}}{=} \mathcal{F}_h^i \cup \mathcal{F}_h^b$, where $\mathcal{F}_h^b$ collects the faces located on the boundary of Ω_h and for any $F \in \mathcal{F}_h^i$ there exist two elements $K^+, K^- \in \mathcal{T}_h$ such that $F \in \partial K^+ \cap \partial K^-$. Moreover, for all $F \in \mathcal{F}_h^b$, $\mathbf{n}_F$ denotes the unit outward normal to Ω_h, whereas, for all $F \in \mathcal{F}_h^i$, $\mathbf{n}_F^-$ and $\mathbf{n}_F^+$ are the unit outward normals to K^+ and K^-, respectively.

Since a function $v_h \in \mathbb{P}_d^k(\mathcal{T}_h)$ is double valued over an internal face $F \in \mathcal{F}_h^i$, we introduce the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ trace operators, that is

$$[\![v_h]\!] \stackrel{\text{def}}{=} v_{h|K^+}\mathbf{n}_F^+ + v_{h|K^-}\mathbf{n}_F^-, \qquad \{v_h\} \stackrel{\text{def}}{=} \frac{v_{h|K^+} + v_{h|K^-}}{2}, \tag{30}$$

and consider them to act componentwise when applied to vector functions. Finally, the DG discretization of second-order viscous terms employs the lifting operators $\mathbf{r}_F$ and $\mathbf{r}$. For all $F \in \mathcal{F}_h$, we define the local lifting operator $\mathbf{r}_F : \left[L^2(F)\right]^d \to [\mathbb{P}_d^k(\mathcal{T}_h)]^d$, such that, for all $\mathbf{v} \in \left[L^2(F)\right]^d$,

$$\int_\Omega \mathbf{r}_F(\mathbf{v}) \cdot \boldsymbol{\tau}_h d\mathbf{x} = -\int_F \{\boldsymbol{\tau}_h\} \cdot \mathbf{v} dF \quad \forall \boldsymbol{\tau}_h \in [\mathbb{P}_d^k(\mathcal{T}_h)]^d. \tag{31}$$

The global lifting operator $\mathbf{r}$ is then defined as

$$\mathbf{r}(\mathbf{v}) \stackrel{\text{def}}{=} \sum_{F \in \mathcal{F}_h} \mathbf{r}_F(\mathbf{v}). \tag{32}$$

3.1 DG Discretization of the RANS-LES Equations

The hybrid RANS-LES equations can be written in compact form as

$$\mathbf{P}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{w}) + \nabla \cdot \mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w}) + \mathbf{s}(\mathbf{w}, \nabla \mathbf{w}) = \mathbf{0}, \tag{33}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the unknown solution vector of the m variables, the tensors $\mathbf{F}_c \in \mathbb{R}^m \otimes \mathbb{R}^d$ and $\mathbf{F}_v \in \mathbb{R}^m \otimes \mathbb{R}^d$ are the inviscid and viscous flux functions and d the number of dimensions. A common choice for compressible flows is the set of $m = 4 + d$ conservative variables $\mathbf{w}_c = [\rho, \rho u_i, \rho E, \rho k, \rho\widetilde{\omega}]^T$ while for incompressible flows the set of $m = 2 + d$ primitive variables $\mathbf{w}_p = [p, u_i, \tilde{\nu}_t]^T$ is usually employed. With these choices the matrix $\mathbf{P}(\mathbf{w}) \in \mathbb{R}^m \otimes \mathbb{R}^m$ does not depend on $\mathbf{w}$ and reduces to the identity matrix $\mathbf{P} = \mathbf{I}$ in case of compressible flows and to the difference between the identity and a single-entry matrix $\mathbf{P} = \mathbf{I} - \mathbf{J}^{11}$ for incompressible flows.

Alternatives to the set of conservative variables for compressible flows were investigated by several authors in the past. In [37] the sets of entropy variables and primitive variables based on pressure were shown to be well defined in the incompressible limit of compressible flows, unlike the commonly used conservative variables or the primitive variables based on density. The primitive variables $(p, \mathbf{u}, T)$ are often preferred for low Mach number flows, [11, 17, 57], and are among the possible choices to design a numerical scheme suited for both compressible and incompressible flows, [36].

A proper choice of the dependent variables can also somewhat simplify the implicit implementation of a method. In particular, the contributions to the Jacobian matrix of viscous terms discretization, including the implicit treatment of boundary conditions, are easier to derive using the set of $(p, \mathbf{u}, T)$ variables.

As a variant to (p, T) variables, we propose to employ their logarithms $\widetilde{p} = \log(p)$ and $\widetilde{T} = \log(T)$ as working variables, thus ensuring, by design, the positivity of all thermodynamic variables at the discrete level. We remark that, unlike the equation for $\widetilde{\omega}$, we do not transform the governing equations, we only substitute the variables (p, T) with $(e^{\widetilde{p}}, e^{\widetilde{T}})$ and use a polynomial approximation for $\widetilde{p}$ and $\widetilde{T}$ in the DG discretization. We also remark that this approach certainly adds to the robustness of high-order simulations of transonic flows, but is not to be intended as a substitute for shock-capturing techniques. The transformation matrix $\mathbf{P}(\mathbf{w})$ then reads

$$\mathbf{w} = \left[\widetilde{p}, u_1, u_2, u_3, \widetilde{T}, k, \widetilde{\omega}\right]^T, \tag{34}$$

$$\mathbf{P}(\mathbf{w}) = \begin{bmatrix} \rho_{\widetilde{p}} & 0 & 0 & 0 & \rho_{\widetilde{T}} & 0 & 0 \\ \rho_{\widetilde{p}} u_1 & \rho & 0 & 0 & \rho_{\widetilde{T}} u_1 & 0 & 0 \\ \rho_{\widetilde{p}} u_2 & 0 & \rho & 0 & \rho_{\widetilde{T}} u_2 & 0 & 0 \\ \rho_{\widetilde{p}} u_3 & 0 & 0 & \rho & \rho_{\widetilde{T}} u_3 & 0 & 0 \\ \rho_{\widetilde{p}} H + \rho h_{\widetilde{p}} - e^{\widetilde{p}} & \rho u_1 & \rho u_2 & \rho u_3 & \rho_{\widetilde{T}} H + \rho h_{\widetilde{T}} & 0 & 0 \\ \rho_{\widetilde{p}} k & 0 & 0 & 0 & \rho_{\widetilde{T}} k & \rho & 0 \\ \rho_{\widetilde{p}} \widetilde{\omega} & 0 & 0 & 0 & \rho_{\widetilde{T}} \widetilde{\omega} & 0 & \rho \end{bmatrix}, \tag{35}$$

where

$$\rho = e^{(\widetilde{p}-\widetilde{T})}, \qquad \hat{e} = \frac{e^{\widetilde{T}}}{\gamma - 1}, \tag{36}$$

$$h_{\widetilde{p}} = \left.\frac{\partial h}{\partial \widetilde{p}}\right|_{\widetilde{T}} = \hat{e}_{\widetilde{p}} + \frac{e^{\widetilde{p}}}{\rho} - \frac{\rho_{\widetilde{p}}}{\rho^2} e^{\widetilde{p}}, \qquad h_{\widetilde{T}} = \left.\frac{\partial h}{\partial \widetilde{T}}\right|_{\widetilde{p}} = \hat{e}_{\widetilde{T}} - \frac{\rho_{\widetilde{T}}}{\rho^2} e^{\widetilde{p}}, \tag{37}$$

and assuming an ideal gas

$$\rho_{\widetilde{p}} = \left.\frac{\partial \rho}{\partial \widetilde{p}}\right|_{\widetilde{T}} = \rho, \qquad \rho_{\widetilde{T}} = \left.\frac{\partial \rho}{\partial \widetilde{T}}\right|_{\widetilde{p}} = -\rho, \tag{38}$$

$$\hat{e}_{\widetilde{p}} = \left.\frac{\partial \hat{e}}{\partial \widetilde{p}}\right|_{\widetilde{T}} = 0, \qquad \hat{e}_{\widetilde{T}} = \left.\frac{\partial \hat{e}}{\partial \widetilde{T}}\right|_{\widetilde{p}} = \hat{e}, \tag{39}$$

$$h_{\widetilde{p}} = 0, \qquad h_{\widetilde{T}} = \hat{e}_{\widetilde{T}} - \frac{e^{\widetilde{p}}}{\rho}. \tag{40}$$

By multiplying Eq. (33) by an arbitrary smooth test function $\mathbf{v} = \{v_1, \ldots, v_m\}$, and integrating by parts, we obtain the weak formulation

$$\int_\Omega \mathbf{v} \cdot \left(\mathbf{P}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial t} \right) d\mathbf{x} - \int_\Omega \nabla \mathbf{v} : \mathbf{F}(\mathbf{w}, \nabla \mathbf{w})\, d\mathbf{x} + \int_{\partial\Omega} \mathbf{v} \otimes \mathbf{n} : \mathbf{F}(\mathbf{w}, \nabla \mathbf{w})\, d\sigma + \int_\Omega \mathbf{v} \cdot \mathbf{s}(\mathbf{w}, \nabla \mathbf{w})\, d\mathbf{x} = \mathbf{0}, \tag{41}$$

where $\mathbf{F}$ is the sum of the inviscid and viscous flux functions and $\mathbf{n}$ is the unit vector normal to the boundary.

To discretize Eq. (41) we replace the solution $\mathbf{w}$ and the test function $\mathbf{v}$ with a finite element approximation $\mathbf{w}_h$ and a discrete test function $\mathbf{v}_h$, respectively, where $\mathbf{w}_h$ and $\mathbf{v}_h$ belong to the space $\mathbf{V}_h \stackrel{\text{def}}{=} [\mathbb{P}_d^k(\mathcal{T}_h)]^m$. For each of the m equations of system (41), and without loss of generality, we choose the set of test and shape functions in any element K coincident with the set $\{\phi\}$ of N_{dof}^K orthogonal and hierachical basis functions in that element. With this choice each component $w_{h,j}$, $j = 1, \ldots, m$, of $\mathbf{w}_h \in \mathbf{V}_h$ can be expressed, in terms of the elements of the global vector $\mathbf{W}$ of unknown degrees of freedom, as $w_{h,j} = \phi_l W_{j,l}$, $l = 1, \ldots, N_{dof}^K$, $\forall K \in \mathcal{T}_h$. Then, the DG discretization of the hybrid RANS-LES equations consists in seeking, for $j = 1, \ldots, m$, the elements of $\mathbf{W}$ such that

$$\sum_{K\in\mathcal{T}_h} \int_K \phi_i P_{j,k}(\mathbf{w}_h)\, \phi_l \frac{dW_{k,l}}{dt} d\mathbf{x} - \sum_{K\in\mathcal{T}_h} \int_K \frac{\partial \phi_i}{\partial x_n} F_{j,n}(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \mathbf{r}([\![\mathbf{w}_h]\!]))\, d\mathbf{x} + \sum_{F\in\mathcal{F}_h} \int_F [\![\phi_i]\!]_n \widehat{F}_{j,n}\left(\mathbf{w}_h^{\pm}, (\nabla_h \mathbf{w}_h + \eta_F \mathbf{r}_F([\![\mathbf{w}_h]\!]))^{\pm} \right) d\sigma + \sum_{K\in\mathcal{T}_h} \int_K \phi_i s_j(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \mathbf{r}([\![\mathbf{w}_h]\!]))\, d\mathbf{x} = 0, \tag{42}$$

for $i = 1, \ldots, N_{dof}^K$. In Eq. (42) repeated indices imply summation over the ranges $k = 1, \ldots, m$, $l = 1, \ldots, N_{dof}^K$, $n = 1, \ldots, d$.

The DG discretization of the viscous fluxes is based on the BR2 scheme, proposed in [14] and theoretically analyzed in [16] and [2]. According to this scheme, the viscous numerical flux is given by

$$\widehat{\mathbf{F}}_v \left(\mathbf{w}_h^{\pm}, \left(\nabla_h \mathbf{w}_h + \eta_F \mathbf{r}_F \left([\![\mathbf{w}_h]\!] \right) \right)^{\pm} \right) \stackrel{\text{def}}{=} \left\{ \mathbf{F}_v \left(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \eta_F \mathbf{r}_F \left([\![\mathbf{w}_h]\!] \right) \right) \right\} \quad (43)$$

where the stability parameter η_F is defined according to [2].

The inviscid numerical flux is computed from the solution of local Riemann problems in the normal direction at each integration point on elements faces. For compressible flows, we use either the exact Riemann solver of Gottlieb and Groth, [32], or, alternatively, the van Leer flux vector splitting method as modified by Hänel et al., [35]. For incompressible flows, we employ the approach proposed in [8], whereby the inviscid numerical flux is computed from the exact solution of local Riemann problems suitably modified by means of an artificial compressibility perturbation. According to the results of numerical experiments presented in [8] and [20], the value of the artificial compressibility parameter can be chosen in the range $[0.01, 100]$, without affecting the numerical accuracy. For the incompressible flow computations presented in this work it has been set equal to 1.

The DG discretization is best suited for a weak enforcement of boundary conditions, see, *e.g.*, [9, 14]. This can be easily achieved by properly defining boundary states which, together with the internal states, allow to compute the numerical fluxes and the lifting operators for all $F \in \mathcal{F}_h^b$. The boundary states and their derivatives must be defined according to the boundary types and, together with the internal states, enter in the Riemann solvers and ensure that the computed numerical fluxes are consistent with the physical ones.

3.2 Accurate Time Integration

Numerical integration of Eq. (42) by means of suitable Gauss quadrature rules leads to a system of nonlinear ODEs (or DAEs, for incompressible flows) that can be written as

$$\mathbf{M_P} \left(\mathbf{W} \right) \frac{d\mathbf{W}}{dt} + \mathbf{R} \left(\mathbf{W} \right) = \mathbf{0}, \quad (44)$$

where $\mathbf{R}(\mathbf{W})$ is the vector of residuals and $\mathbf{M_P}(\mathbf{W})$ is the global block diagonal matrix arising from the discretization of the first integral in Eq. (42). For the sets of variables $\mathbf{w}_c$ and $\mathbf{w}_p$, and using the set of orthogonal basis functions outlined above, $\mathbf{M_P}$ reduces to the identity matrix for compressible flows and to a modified identity matrix with zeros in the diagonal positions corresponding to the pressure degrees of freedom for incompressible flows. However, for sets of variables different than $\mathbf{w}_c$, the transformation matrix $\mathbf{P}$ couples the degrees of freedom of the variables $\mathbf{w}_h$ within each block of $\mathbf{M_P}$, so that this matrix can not be diagonal, even using a set of orthogonal basis functions.

Implicit and accurate time integration of Eq. (44) can be efficiently performed by means of linearly implicit Rosenbrock-type Runge-Kutta schemes that can be written as

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \sum_{j=1}^{s} b_j \mathbf{K}_j, \tag{45}$$

$$\left(\frac{\mathbf{I}}{\Delta t} + \gamma \widetilde{\mathbf{J}}\right)^n \mathbf{K}_i = -\widetilde{\mathbf{R}}\left(\mathbf{W}^n + \sum_{j=1}^{i-1} \alpha_{ij}\mathbf{K}_j\right) - \widetilde{\mathbf{J}}^n \sum_{j=1}^{i-1} \gamma_{ij}\mathbf{K}_j, \quad i = 1, \ldots, s, \tag{46}$$

where, omitting the dependence on $\mathbf{W}$ for notational convenience,

$$\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{W}}, \qquad \widetilde{\mathbf{R}} = \mathbf{M}_\mathbf{P}^{-1}\mathbf{R}, \qquad \widetilde{\mathbf{J}} = \frac{\partial \widetilde{\mathbf{R}}}{\partial \mathbf{W}} = \mathbf{M}_\mathbf{P}^{-1}\left(\mathbf{J} - \frac{\partial \mathbf{M}_\mathbf{P}}{\partial \mathbf{W}}\widetilde{\mathbf{R}}\right), \tag{47}$$

and b_i, α_{ij}, γ_{ij} are real coefficients. The Jacobian matrix $\mathbf{J}$ of the DG space discretization is computed analytically and fully accounts for the dependence of the residuals on the unknown vector and its derivatives including the implicit treatment of lifting operators and of boundary conditions. A direct implementation of Eq. (46) entails a matrix-vector product $\widetilde{\mathbf{J}}^n \sum_{j=1}^{i-1} \gamma_{ij}\mathbf{K}_j$ from the second stage on. In practice this can be avoided by noting that Eq. (46) can be reduced to the following equivalent formulation

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \sum_{j=1}^{s} m_j \mathbf{Y}_j, \tag{48}$$

$$\left(\frac{\mathbf{I}}{\gamma \Delta t} + \widetilde{\mathbf{J}}\right)^n \mathbf{Y}_i = -\widetilde{\mathbf{R}}\left(\mathbf{W}^n + \sum_{j=1}^{i-1} a_{ij}\mathbf{Y}_j\right) + \sum_{j=1}^{i-1} \frac{c_{ij}}{\Delta t}\mathbf{Y}_j, \quad i = 1, \ldots, s, \tag{49}$$

where, for $i = 1, \ldots, s$,

$$\mathbf{K}_i = \frac{1}{\gamma}\mathbf{Y}_i - \sum_{j=1}^{i-1} c_{ij}\mathbf{Y}_j. \tag{50}$$

The coefficients of the transformed scheme are given by

$$(m_1, \ldots, m_s) = (b_1, \ldots, b_s)\,\mathbf{\Gamma}^{-1}, \quad (a_{ij}) = (\alpha_{ij})\,\mathbf{\Gamma}^{-1}, \quad (c_{ij}) = \gamma^{-1}\mathbf{I}_s - \mathbf{\Gamma}^{-1},$$

where $\mathbf{\Gamma}^{-1} \overset{\text{def}}{=} (\gamma_{ij})^{-1}$ denotes the inverse of the matrix formed with the coefficients (γ_{ij}) of the schemes, see [34]. In this work we consider the second-order two-stage scheme of Iannelli and Baker, [39], the third-order three-stage ROS3P scheme of Lang and Verwer, [45], the fourth-order six-stage RODASP scheme of Steinebach, [54], and the fifth-order eight-stage RODAS5(4) scheme of

Di Marzo, [22]. Finally, in order to avoid the product $\mathbf{M_P^{-1}J}$ in Eq. (47), the actual implementation of Eq. (49) in the DG code MIGALE reads

$$\left(\frac{\mathbf{M_P}}{\gamma \Delta t} + \mathbf{J} - \frac{\partial \mathbf{M_P}}{\partial \mathbf{W}}\widetilde{\mathbf{R}}\right)^n \mathbf{Y}_i = - \mathbf{M_P^n}\left[\widetilde{\mathbf{R}}\left(\mathbf{W}^n + \sum_{j=1}^{i-1} a_{ij}\mathbf{Y}_j\right) - \sum_{j=1}^{i-1}\frac{c_{ij}}{\Delta t}\mathbf{Y}_j\right], \quad i = 1, \ldots, s. \quad (51)$$

A matrix-explicit or a less memory-demanding matrix-free, [19], GMRES algorithm can be used to actually solve Eq. (51) at each time step. In both cases system preconditioning is required to make the convergence of the GMRES solver acceptable in problems of practical interest. For this purpose, we usually employ the block Jacobi method with one block per process, each of which is solved with ILU(0), or the Additive Schwarz Method (ASM). In our codes we rely on PETSc, [5], for the linear solvers and to manage distributed arrays and the communication among them.

4 Numerical Results

The purpose of this section is to verify the design order of convergence of the Rosenbrock time integration schemes considered in Sec. 3.2 and to present the results of several unsteady turbulent flow computations obtained by using such schemes coupled to the high-order DG space-discretized ILES, SA-DES and X-LES models.

4.1 Convection of an Isentropic Vortex

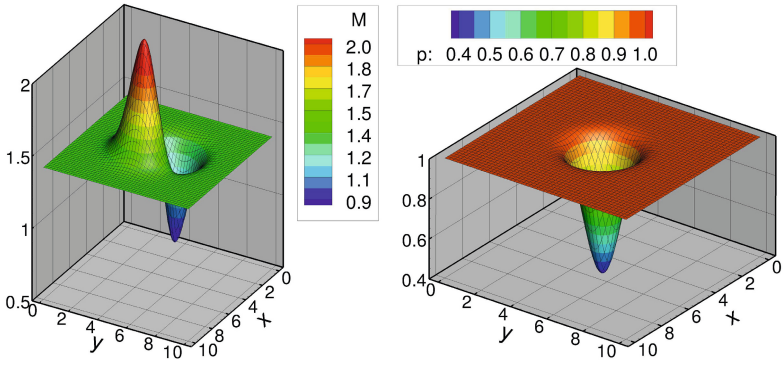
The accuracy of the Rosenbrock time integration schemes was assessed on the simple test case of an inviscid vortex transport by uniform flow, [38]. The isentropic vortex is defined by velocity and temperature perturbations of a uniform flow, with pressure, temperature and density equal to 1 and velocity components equal to $\sqrt{\gamma}$, given by

$$\delta u_1 = -\frac{\alpha}{2\pi}(y - y_0)\, e^{\phi\left(1-r^2\right)}, \quad (52)$$

$$\delta u_2 = \frac{\alpha}{2\pi}(x - x_0)\, e^{\phi\left(1-r^2\right)}, \quad (53)$$

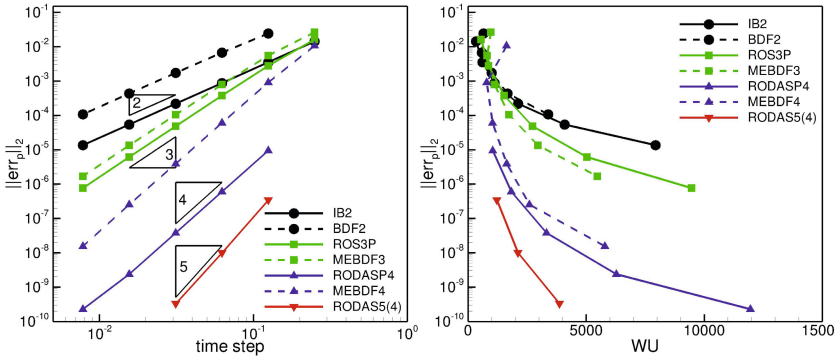
$$\delta T = -\frac{\alpha^2(\gamma - 1)}{16\phi\gamma\pi^2} e^{2\phi\left(1-r^2\right)}, \quad (54)$$

where $\gamma = 1.4$ is the ratio of specific heats, $\phi = \frac{1}{2}$ and $\alpha = 5$ are parameters that determine the vortex strength, r is the distance of point (x, y) from the vortex center (x_0, y_0), placed at $(5, 5)$ in a periodic domain $[0, 10] \times [0, 10]$ at time $t_0 = 0$. The exact solution, shown in Fig. 1 at $t_0 = 0$, can be easily computed



(a) Mach, $t_0 = 0$ (b) pressure, $t_0 = 0$

Fig. 1. Isentropic vortex - Initial flow field, Mach and pressure contours, $\mathbb{P}^6$ solution



(a) pressure error vs. time step (b) pressure error vs. WU

Fig. 2. Isentropic vortex - L^2 pressure error norm vs. time step and Work Units (WU), $\mathbb{P}^6$ solution (solid lines Rosenbrock schemes, dashed lines BDF and MEBDF schemes). Notations BDFx and MEBDFx denote schemes with x-th order of convergence.

by means of the perturbation functions of Eqs. (52–54) and assuming isentropic flow conditions throughout the domain.

The design order of convergence was verified by integrating in time, with smaller and smaller time steps, a very high-order DG space-discretized $\mathbb{P}^6$ solution defined on a uniform 50×50 quadrilateral grid. The results obtained by using several Rosenbrock schemes were compared with those of the second-order Backward Differentiation Formulae (BDF) and third- and fourth-order Modified Extended BDF (MEBDF) schemes, [31, 49].

The GMRES parameters used with the Rosenbrock schemes were $n_{rst} = 120$, $n_{max} = 240$, $tol_r = 10^{-14}$, where n_{rst} is the number of Krylov subspace vectors, n_{max} is the maximum number of iterations and tol_r is the convergence tolerance on the relative residual norm of the linear system. We did not attempt to optimize the GMRES parameters, but, according to our experience on this test case, we can say that tol_r could have been safely set to higher values without affecting the order of convergence, while significantly reducing the computational time. Instead, the iterative solution strategy of the BDF and MEBDF schemes was empirically optimized. For such schemes the convergence tolerance of the GMRES algorithm was set to a quite large value $tol_r = 10^{-2}$, while the convergence tolerance of the nonlinear systems, to be solved at each time step by means of the Newton method, was found to strongly affect the efficiency and accuracy of the schemes and needed to be carefully adjusted to the time step size and to the order of the schemes, [31, 49].

The L^2 pressure error norm of the solutions computed with the various time integration schemes is shown in Fig. 2a as a function of the time step and in Fig. 2b as a function of the computational time measured in terms of Work Units (WU), where WU $= t_{wall} N_{cores}/t_{TB}$, t_{wall} is the wall-clock time of a simulation run on N_{cores} cores and t_{TB} is the reference TauBenchmark time of the hardware. Comparing schemes of the same order, the errors of Rosenbrock schemes are lower, for a given time step size. Except for the third-order scheme, Rosenbrock schemes are also faster, for a given error level.

The results of this test case show that higher-order time integration schemes are much more efficient than lower-order ones if the required accuracy of time integration is high. This makes such schemes preferable to address complex and computationally demanding simulations of turbulent flows. Rosenbrock schemes have been here preferred to BDF and MEBDF schemes due to their greater flexibility, accuracy and efficiency. A thorough comparison of the schemes here considered and also of other high-order time integration schemes is the subject of ongoing work.

4.2 Transitional Flow around the SD7003 Airfoil

This test case is about the ILES of the transitional flow around the Selig-Donovan (SD) 7003 airfoil, one of the test cases proposed within the "International Workshop on High-Order CFD Methods", [56]. As ILES does not include any explicit SGS model, the flow model equations are simply the compressible Navier-Stokes equations, see [12].

The farfield conditions of this test case are $\mathrm{M}_\infty = 0.1$, $\alpha = 8^\circ$ and $\mathrm{Re}_c = 60000$. The unsteady flow is characterized by laminar separation, the formation of a transitional shear layer followed by turbulent reattachment on the suction side of the airfoil. The computational grid was obtained by agglomerating the hexahedral elements of a fine structured linear grid to form a mesh with 20064 50-node hexahedral elements and quartic edge representation and first cell height $y/c = 0.00029$, see Fig. 3a.

For this test case, the time integration scheme was the highly accurate fifth-order eight-stage RODAS5(4) scheme of Di Marzo, [22]. Solutions were advanced in time with a time step equal to a fraction $f = 5{\times}10^{-3}$ of the convective time, *i.e.*, $\Delta t = f\,(c/U_\infty)$. A p-sequence of lower-order, not fully statistically converged solutions was employed to initialize the $\mathbb{P}^3$ simulation. Presenting the results of this test case, by "mean" we mean "averaged both in time and spanwise direction".

Fig. 4 displays the contours of mean pressure coefficient C_p and of x-component of velocity of the $\mathbb{P}^4$ solution. Mean skin friction c_f and pressure coefficient distributions on the airfoil are shown in Fig. 5, while the mean drag, lift and moment coefficients of the airfoil, are reported in Table 1.

For the mean $\mathbb{P}^4$ solution, the laminar separation bubble on the suction side of the airfoil, along with the position of the separation and reattachment points, are shown in Fig. 3b. The mean velocity profiles of Fig. 6 highlight the quite important effect of increasing the accuracy of the DG space discretization and suggest that a $\mathbb{P}^5$ solution would be useful to corroborate the accuracy of the averaged $\mathbb{P}^4$ results.

For the $\mathbb{P}^4$ solution the instantaneous $Q = 500$ isosurface of Q-criterion and contours of the spanwise-averaged Reynolds stress $<u_1' u_1'>$ are displayed in Fig. 7. The results presented here are in reasonable agreement with those of Garmann and Visbal, [29], obtained using sixth-order compact finite-differencing and filtering.

Table 1. SD7003 - Mean aerodynamic coefficients

	C_D	C_L	C_m
$\mathbb{P}^3$	0.0423	0.9615	-0.0233
$\mathbb{P}^4$	0.0454	0.9534	-0.0224

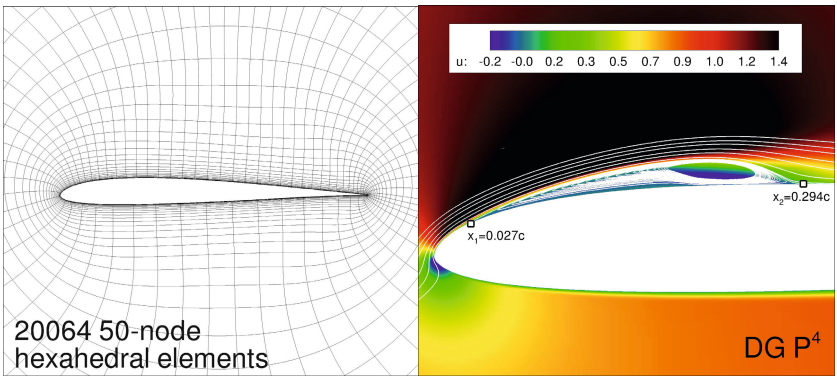


Fig. 3. SD7003 - Computational mesh with 20064 50-node hexahedral elements and detail of the laminar separation bubble on the suction side of the airfoil, $\mathbb{P}^4$ solution

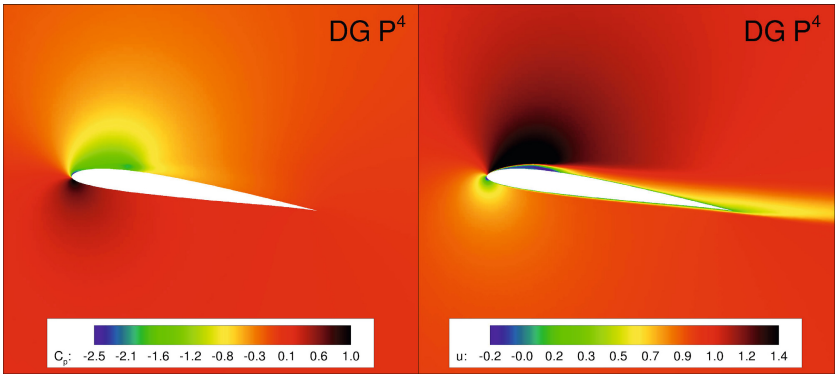


Fig. 4. SD7003 - Contours of mean pressure coefficient and x-component of velocity, $\mathbb{P}^4$ solution

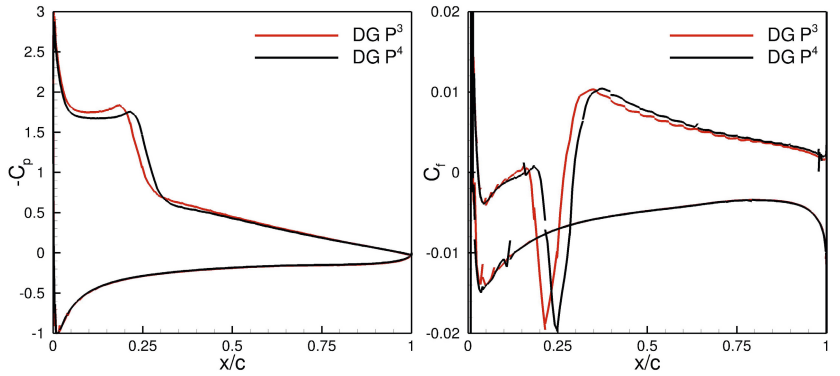


Fig. 5. SD7003 - Mean C_p and c_f distributions, $\mathbb{P}^3$ and $\mathbb{P}^4$ solutions

4.3 DESider Bump Water Tunnel Experiment (A.11a)

This test case is about the SA-DES of the water tunnel experiment known as DESider bump, proposed in the framework of the DESider European Project on hybrid RANS-LES modelling, [33]. The water tunnel experiment was carried out at ONERA in order to investigate flow separation, reattachment and post-reattachment recovery in a highly three-dimensional configuration, [4]. Detailed information about this configuration can be found in the DESider Project website [21].

The test case conditions are defined by the Reynolds number $\mathrm{Re}_L = 4{\times}10^6$, based on a reference length $L = 1\,\mathrm{m}$ and a reference velocity $U = 4\,\mathrm{m/s}$.

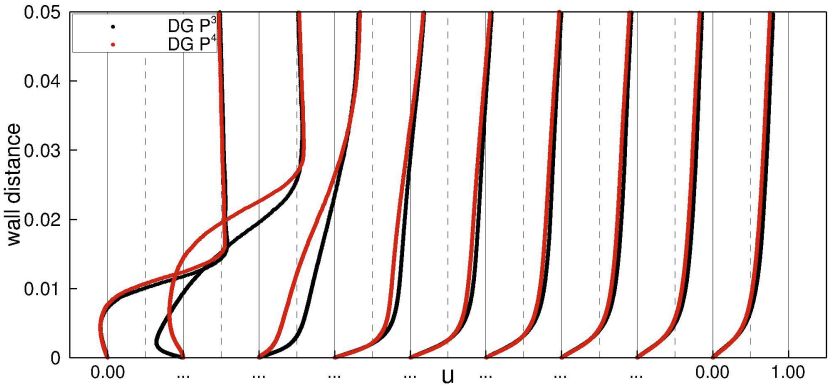


Fig. 6. SD7003 - Profiles of mean x-component of velocity at chordwise locations $x/c = \{0.1, \ldots, 0.9\}$, $\mathbb{P}^3$ and $\mathbb{P}^4$ solutions

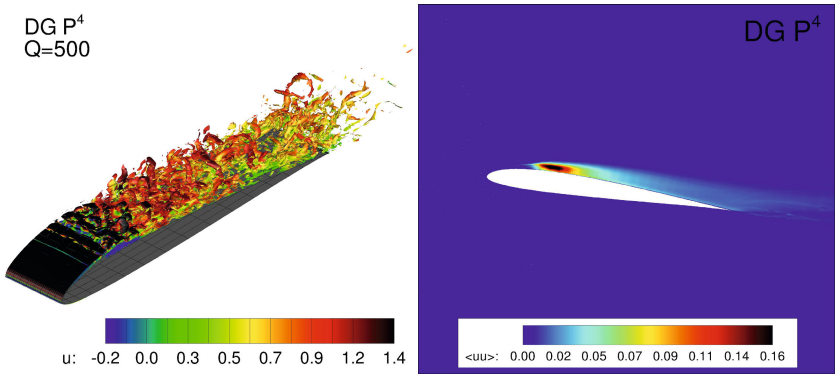


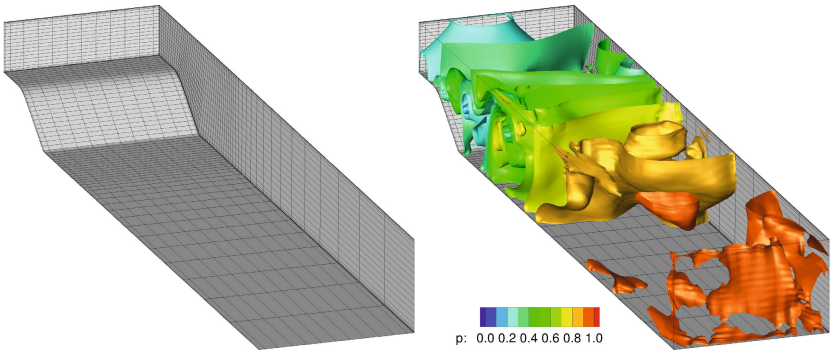
Fig. 7. SD7003 - Instantaneous $Q = 500$ isosurface of Q-criterion and contours of spanwise-averaged Reynolds stress $<u_1' u_1'>$, $\mathbb{P}^4$ solution

The experimental velocity profile was prescribed at inflow and the average static pressure at outlet. The high-order computational mesh, with 29700 20-node hexahedral elements and quadratic edge representation, was obtained by agglomerating a linear block-structured grid provided by ONERA. The surface grid is shown in Fig. 8a.

As a first step, the time integration scheme used here was the second-order, two-stage scheme of Iannelli and Baker, [39], with a time step $\Delta t = 10^{-3}$. This scheme was used to advance in time the SA-DES $\mathbb{P}^3$ solution, starting from a fully converged RANS-SA $\mathbb{P}^3$ steady solution. Fig. 9 shows the instantaneous and steady velocity contours of the SA-DES and RANS-SA $\mathbb{P}^3$ solutions, respectively.

Figs. 8b and 10a display the pressure isosurfaces and a λ_2-criterion isosurface coloured with pressure, respectively. The time-averaged velocity profiles shown in Fig. 10b indicate that the predicted separation region around $x = 0.35\,\mathrm{m}$ is significantly smaller than in the experiments. This result, however, is in agreement with the findings of the DESider project. Also a much more resolved X-LES simulation by Kok et al., [44], performed on a structured grid with 647520 elements using a fourth-order finite volume method, predicted similar velocity profiles even if the turbulent structures were, of course, much finer than those found here.

The main result of this test case was the demonstration of the hybrid SA-DES approach in the framework of our DG method for incompressible flows. More accurate, both in space and time, computations are clearly needed to improve the resolution of the preliminary solutions presented here. Nevertheless, the reasonably good agreement of the time-averaged velocity profiles with those predicted by more resolved simulations is very promising.



(a) surface grid (b) pressure isosurfaces

Fig. 8. DESider bump - Surface grid and instantaneous pressure isosurfaces, $\mathbb{P}^3$ solution

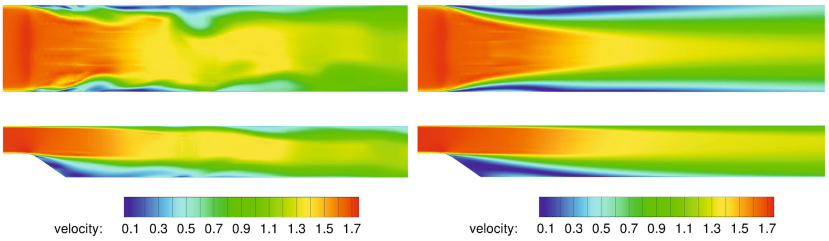
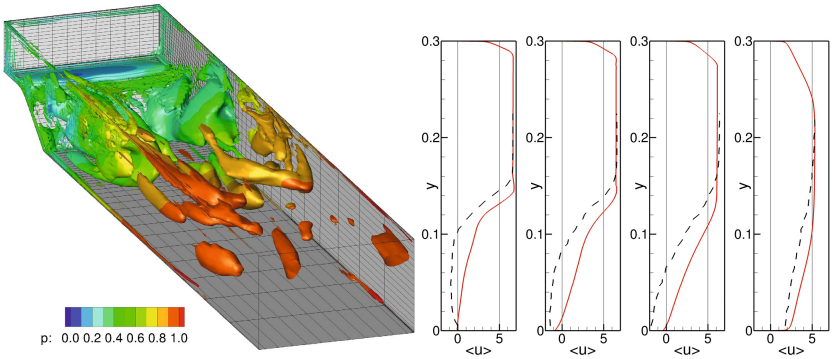


Fig. 9. DESider bump - Instantaneous and steady velocity contours, SA-DES and RANS-SA $\mathbb{P}^3$ solutions



(a) λ_2 isosurface (b) time-averaged streamwise velocity

Fig. 10. DESider bump - Instantaneous λ_2-criterion isosurface coloured with pressure and time-averaged streamwise velocity profiles on the spanwise midplane at positions $x = \{0, 0.15, 0.35, 0.925\}$ along the channel, $\mathbb{P}^3$ solution

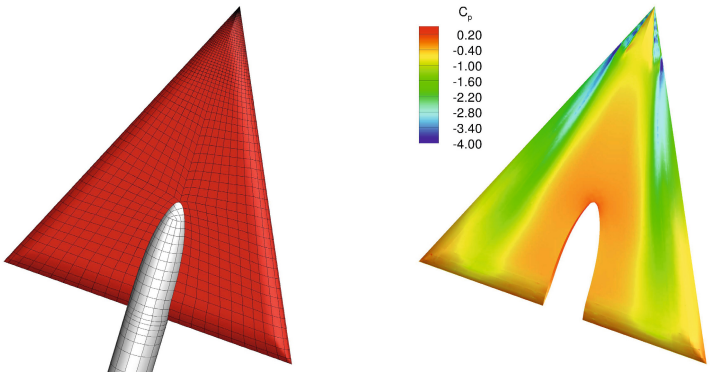
4.4 Vortical Flow over the VFE-2 Delta Wing (U.1a)

This test case deals with the X-LES of the flow around the VFE-2 (sharp edge) delta wing geometry of Chu and Luckring, [18], thoroughly investigated experimentally by Furman and Breitsamter, [25–28]. The farfield conditions of this test case are $\mathrm{M}_\infty = 0.07$, $\alpha = 23°$ and $\mathrm{Re}_{mac} = 1000000$, a flow regime for which vortex breakdown occurs.

The high-order computational mesh, with 98049 50-node hexahedral elements and quartic edge representation, was obtained by agglomerating a linear block-structured grid provided by J. C. Kok from NLR. A detail of the wall surface mesh is shown in Fig. 11a. The time integration scheme used here was the third-order, three-stage ROS3P scheme of Lang and Verwer [45]. The high-order $\mathbb{P}^3$ simulation was started from the last, not fully statistically converged, solution of a p-sequence of lower-order solutions. A constant filter width $\Delta = 5{\times}10^{-6}$ was used in this simulation. According to the experience of other IDIHOM partners, this is quite a small value, which was needed in order to ensure that most of the flow field on the upper side of the wing were computed in LES mode.

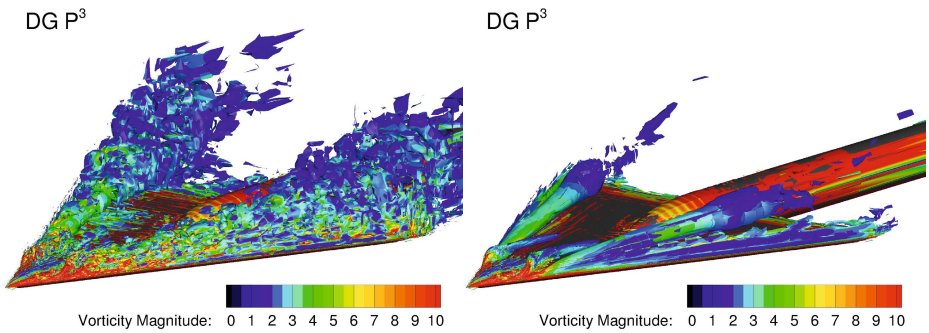
Compared to the results, not shown here, of Kok et al., [43,47], the picture of the downstream resolved turbulence looks reasonable and also the vortex breakdown seems to be captured by the simulation, see Figs. 12a and 13. However, the picture of the flow near the apex of the wing is not fully satisfactory, as the main vortex appears to start somewhat downstream of the apex along the wing leading edge, see Fig. 11b. Close to the symmetry axis, a small vortex, clearly visible in Fig. 12b, starts around $x/c_r = 0.2$ and is responsible for the large disagreement between computed and experimental C_p distributions shown in Fig. 14.

The time-averaged drag and lift coefficients (excluding the sting), $C_D = 0.3650$ and $C_L = 0.9169$, of our not fully statistically converged solution turned out roughly 6% higher than those obtained by other IDIHOM partners, *i.e.*, NLR and CA, [1, 47]. Such discrepancies with respect to reference solutions certainly need further investigation in order to understand the effect on the solution of X-LES parameters like the filter width. Nevertheless we can state that, from a numerical point of view, the proposed high-order implicit time integration schemes coupled with a high-order DG space discretization are well suited to the X-LES approach.



(a) surface mesh detail (b) time-averaged C_p

Fig. 11. VFE-2 - Wall surface mesh and time-averaged pressure coefficient contours, $\mathbb{P}^3$ solution



(a) instantaneous $Q = 2.5$ isosurface (b) time-averaged $Q = 2.5$ isosurface

Fig. 12. VFE-2 - Instantaneous and time-averaged $Q = 2.5$ isosurface of Q-criterion coloured with vorticity magnitude. Values made dimensionless with the freestream velocity and the mean aerodynamic chord, $\mathbb{P}^3$ solution.

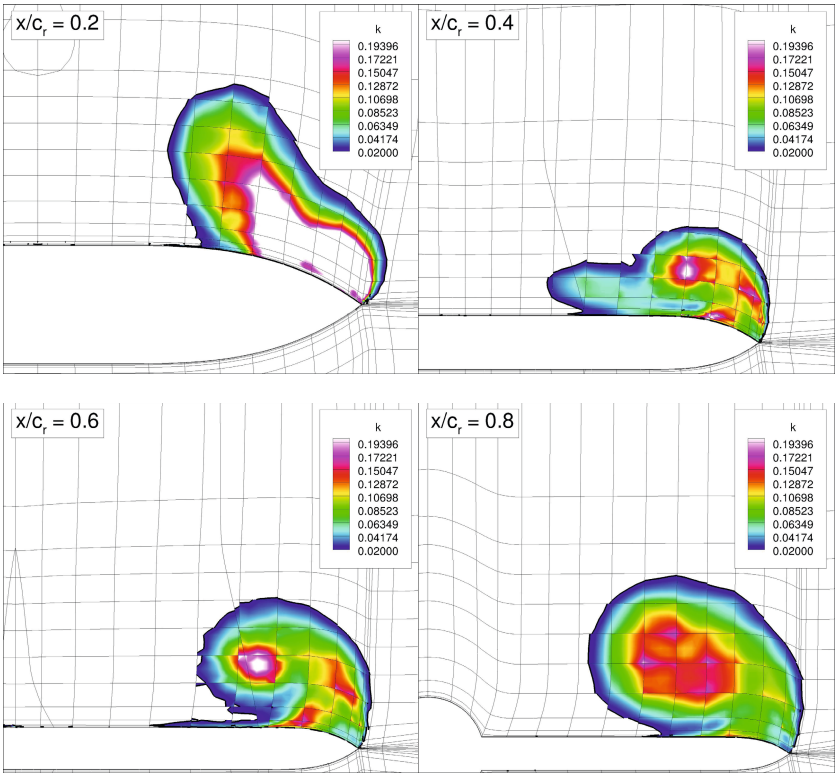


Fig. 13. VFE-2 - Resolved turbulent kinetic energy at different locations along the root chord c_r, $\mathbb{P}^3$ solution

4.5 Vortical Flow over the FA5 Model (A.2a)

This test case is about the flow around a 1:15 model of a delta-wing type generic fighter aircraft, formerly considered in the DESider, [21], and ATAAC, [3], European research projects. The farfield conditions of this test case are $\mathrm{M}_\infty = 0.125$, $\alpha = 15°$ and Reynolds number per meter $\mathrm{Re}_m = 2.78{\times}10^6$. The engine intake and exhaust boundaries where treated as outflow and inflow boundaries, respectively, with the prescribed pressure at outflow and the prescribed total pressure and total temperature at inflow set equal to the freestream values. The engine intake boundary condition corresponds to a flow-through condition with MFR = 1.0.

The X-LES solution was computed up to $\mathbb{P}^3$ polynomial approximation on a grid with 164636 50-node hexahedral elements and quartic edge representation. The time integration scheme used here was the third-order, three-stage ROS3P scheme of Lang and Verwer, [45]. A constant filter width $\Delta = 5{\times}10^{-6}$ was used in this simulation.

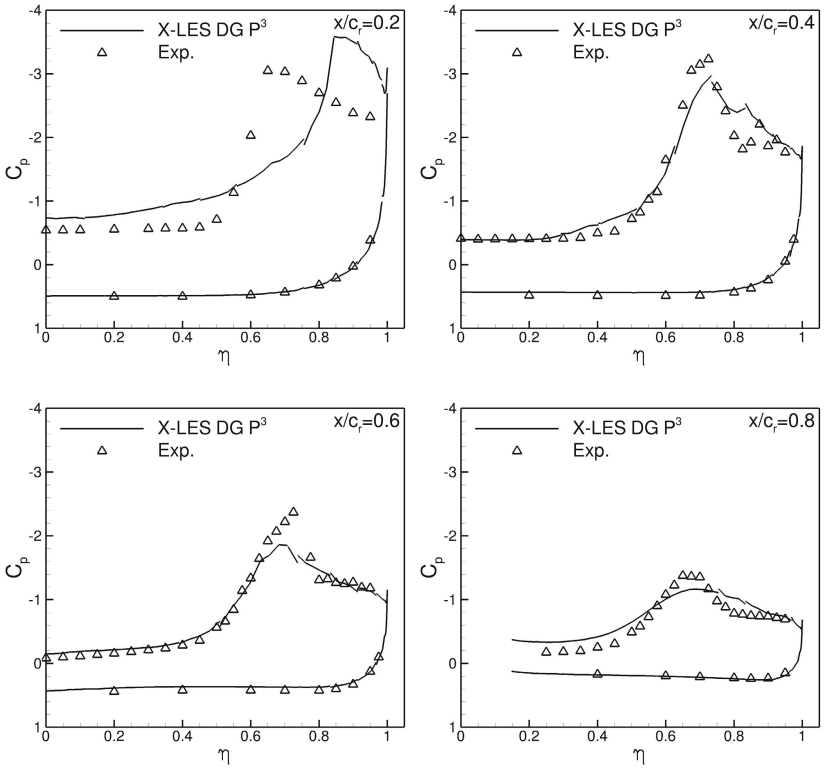


Fig. 14. VFE-2 - Time-averaged pressure coefficient distribution at different locations along the root chord c_r, $\mathbb{P}^3$ solution

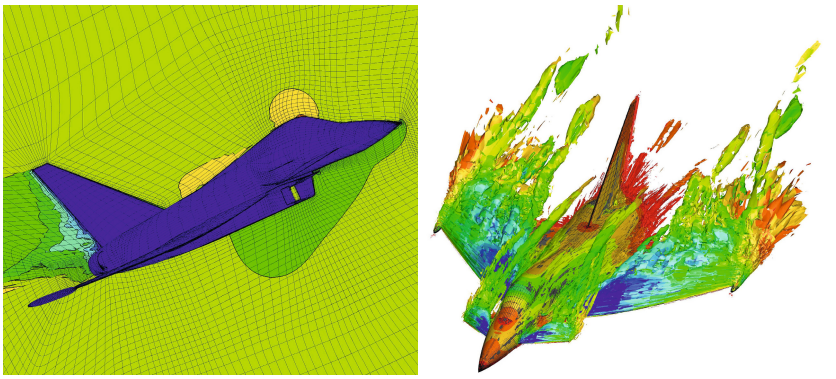


Fig. 15. FA5 - Time-averaged Mach contours and time-averaged Q-criterion isosurface coloured with C_p, $\mathbb{P}^3$ solution

Fig. 15, showing the time-averaged Mach contours and a time-averaged Q-criterion isosurface coloured with the pressure coefficient, appears to confirm the potential of the proposed approach to a high-order X-LES on very complex geometries. The solution could not be run until statistical convergence but, again, the numerical approach turned out to be both robust and accurate even advancing the solution in time with quite large time steps.

5 Conclusions

An implicit approach to high-order time integration, coupled with high-order DG space discretization, seems to be well suited for the numerical solution of those flow models that are currently receiving much attention by the CFD community for the simulation of underresolved turbulent flows.

In this work we focused on high-order linearly implicit Rosenbrock-type Runge-Kutta schemes, implemented in the code MIGALE up to sixth-order accuracy, applied to the time-accurate integration of the DG space-discretized ILES, SA-DES and X-LES models. The temporal order of accuracy of Rosenbrock schemes was assessed on a simple test case, with the result of an increasingly higher efficiency of higher order schemes for a given absolute error relative to the exact solution. When applied to ILES, SA-DES and X-LES models, Rosenbrock schemes proved to be numerically stable and accurate also for large time steps, providing solutions that appear in general good agreement with reference results.

Ongoing work is devoted to a thorough validation of our high-order space- and time-accurate DG method for ILES and RANS-LES computations of reference test cases available in the literature.

Acknowledgements. This work was carried out within the EU FP7 IDIHOM project. The authors thank Prof. Dr. Norbert Kroll for providing access to the HPC facilities at DLR during IDIHOM project and Dr. Johan C. Kok for valuable comments and suggestions concerning X-LES validation and for providing the VFE-2 computational grid.

References

1. Airbus D&S (CA). Test case report on U.1a. IDIHOM internal report (2013)
2. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. 39(5), 1749–1779 (2002)
3. ATAAC. ATAAC Web page (2014), `http://cfd.mace.manchester.ac.uk/twiki/bin/view/ATAAC/WebHome`
4. Aupoix, B., Barricau, P., Egorov, Y., Geiler, C., Gilliot, A., Menter, F.R., Monnier, J.-C., Pailhas, G., Perret, L., Stanislas, M., Touvet, Y.: The DESider bump experiment. In: In. Proc. 7th Int. ERCOFTAC Symp. Eng. Turb., Turkey, June 4–6 (2008)

5. Balay, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C.: PETSc Web page (2014), http://www.mcs.anl.gov/petsc
6. Bassi, F., Botti, L., Colombo, A., Di Pietro, D.A., Tesini, P.: On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. J. Comput. Phys. 231(1), 45–65 (2012)
7. Bassi, F., Botti, L., Colombo, A., Ghidoni, A., Rebay, S.: Discontinuous Galerkin for turbulent flows. In: Wang, Z.J. (ed.) Adaptive High-Order Methods in Computational Fluid Dynamics, ch. 3. Advances in Computational Fluid Dynamics, vol. 2, pp. 1–32. World Science Books (2011)
8. Bassi, F., Crivellini, A., Di Pietro, D.A., Rebay, S.: An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations. J. Comput. Phys. 218, 794–815 (2006)
9. Bassi, F., Crivellini, A., Di Pietro, D.A., Rebay, S.: An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. Comput. Fluids 36, 1529–1546 (2007)
10. Bassi, F., Crivellini, A., Rebay, S., Savini, M.: Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations. Comput. Fluids 34, 507–540 (2005)
11. Bassi, F., De Bartolo, C., Hartmann, R., Nigro, A.: A discontinuous Galerkin method for inviscid low Mach number flows. J. Comput. Phys. 228(11), 3996–4011 (2009)
12. Bassi, F., Rebay, S.: A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. J. Comput. Phys. 131, 267–279 (1997)
13. Bassi, F., Rebay, S.: A high order discontinuous Galerkin method for compressible turbulent flows. In: Cockburn, B., Karniadakis, G.E., Shu, C.-W. (eds.) Discontinuous Galerkin Methods. LNCSE, vol. 11, pp. 77–88. Springer, Heidelberg (2000)
14. Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., Savini, M.: A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere, R., Dibelius, G. (eds.) Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, March 5–7, pp. 99–108. Technologisch Instituut, Antwerpen (1997)
15. Botti, L.: Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces. J. Sci. Comput, 1–29 (2011)
16. Brezzi, F., Manzini, G., Marini, D., Pietra, P., Russo, A.: Discontinuous Galerkin approximations for elliptic problems. Numer. Meth. Part. D. E. 16, 365–378 (2000)
17. Choi, Y.-H., Merkle, C.L.: The application of preconditioning in viscous flows. J. Comput. Phys. 105(2), 207–223 (1993)
18. Chu, J., Luckring, J.M.: Experimental surface pressure data obtained on 65° delta wing across Reynolds number and Mach number ranges. TM 4645, NASA (1996)
19. Crivellini, A., Bassi, F.: An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. Comput. Fluids 50(1), 81–93 (2011)
20. Crivellini, A., D'Alessandro, V., Bassi, F.: A Spalart-Allmaras turbulence model implementation in a discontinuous Galerkin solver for incompressible flows. J. Comput. Phys. 241, 388–415 (2013)
21. DESider. DESider Web page (2014), http://cfd.mace.manchester.ac.uk/desider/

22. Di Marzo, G.: RODAS5(4) - Méthodes de Rosenbrock d'ordre 5(4) adaptées aux problemes différentiels-algébriques. MSc Mathematics Thesis; Faculty of Science, University of Geneva, Switzerland (1993)
23. Di Pietro, D.A., Ern, A.: Mathematical Aspects of Discontinuous Galerkin Methods. Mathématiques et Applications 69 (2012)
24. Dolejší, V.: Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. Commun. Comput. Phys. 4, 231–274 (2008)
25. Furman, A., Breitsamter, C.H.: Investigations of flow phenomena on generic delta wing. In: 25th Congress of the International Council of the Aeronautical Sciences, ICAS 2006-3.1.2, Hamburg, Germany, September 3–8 (2006)
26. Furman, A., Breitsamter, C.H.: Stereo-PIV and hot-wire investigations on delta wing with sharp and rounded leading edge. In: 1st CEAS European Air and Space Conference, CEAS-2007-430, Berlin, Germany, September 10–13 (2007)
27. Furman, A., Breitsamter, C.H.: Experimental investigations on the VFE-2 configuration at TU Munich, Germany. In: *Understanding and Modeling Vortical Flows to Improve the Technology Readiness Level for Military Aircraft*, Technical Report RTO-TR-AVT-113, chapter 21. NATO RTO, Neuilly-sur-Seine Cedex, France (October 2009)
28. Furman, A., Breitsamter, C.H.: Turbulent and unsteady flow characteristics of delta wing vortex systems. Aerosp. Sci. Technol. 24(1), 32–44 (2013)
29. Garmann, D.J., Visbal, M.R.: C3.3: Implicit large eddy-simulations of transitional flow over the SD7003 airfoil using compact finite-differencing and filtering. In: 2nd International Workshop on High-Order CFD Methods, Cologne, Germany, May 27–28 (2013)
30. Gassner, G.J., Lörcher, F., Munz, C.-D.: A discontinuous Galerkin scheme based on a space-time expansion II. Viscous flow equations in multi dimensions. J. Sci. Comput. 34, 260–286 (2008)
31. Ghidoni, A., Nigro, A., Rebay, S.: D3.2-09 – First results on performance of p-multigrid approach. IDIHOM internal report (2012)
32. Gottlieb, J.J., Groth, C.P.T.: Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. J. Comput. Phys. 78, 437–458 (1988)
33. Haase, W., Braza, M., Revell, A.: I The DESider Project. NNFM, vol. 103. Springer, Heidelberg (2009)
34. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, 2nd edn. Springer Series in Computational Mathematics (1996)
35. Hänel, D., Schwane, R., Seider, G.: On the accuracy of upwind schemes for the solution of the Navier–Stokes equations. AIAA Paper 87-1105 (1987)
36. Hauke, G., Hughes, T.J.R.: A unified approach to compressible and incompressible flows. Comput. Methods Appl. Mech. Engrg. 113(3–4), 389–395 (1994)
37. Hauke, G., Hughes, T.J.R.: A comparative study of different sets of variables for solving compressible and incompressible flows. Comput. Methods Appl. Mech. Engrg. 153(1–2), 1–44 (1998)
38. Hu, C., Shu, C.-W.: Weighted essentially non-oscillatory schemes on triangular meshes. J. Comput. Phys. 150(1), 97–127 (1999)
39. Iannelli, G.S., Baker, A.J.: A stiffly-stable implicit Runge-Kutta algorithm for CFD applications. AIAA Paper 88-0416 (1988)
40. IDIHOM. Industrialisation of High-Order Methods – A top-down approach, Specific Targeted Research Project supported by European Commission, `http://www.dlr.de/as/en/desktopdefault.aspx/tabid-7027/11654_read-27492/`

41. Kok, J.C.: Resolving the dependence on freestream values for the k-ω turbulence model. AIAA J. 38(7), 1292–1295 (2000)
42. Kok, J.C., Dol, H.S., Oskam, B., van der Ven, H.: Extra-large eddy simulation of massively separated flows. AIAA Paper 2004-264 (2004)
43. Kok, J.C., van der Ven, H.: Capturing free shear layers in hybrid RANS-LES simulations of separated flow. In: Third Symposium Simulation of Wing and Nacelle Stall, Braunschweig, Germany, June 21–22 (2012)
44. Kok, J.C., van der Ven, H.: Test case summary report - A.11a DESider test case: Bump in a square duct. IDIHOM internal report (2014)
45. Lang, J., Verwer, J.: ROS3P—An accurate third-order Rosenbrock solver designed for parabolic problems. BIT 41(4), 731–738 (2001)
46. Menter, F.R.: Two-equation eddy-viscosity turbulence model for engineering applications. AIAA J. 32(8), 1598–1605 (1994)
47. Nationaal Lucht- en Ruimtevaartlaboratorium (NLR). Test case report on U.1a. IDIHOM internal report (2013)
48. Nigro, A., De Bartolo, C., Bassi, F., Ghidoni, A.: Up to sixth-order accurate A-stable implicit schemes applied to the discontinuous Galerkin discretized Navier-Stokes equations. J. Comput. Phys. Under Review (2014)
49. Nigro, A., Ghidoni, A., Rebay, S., Bassi, F.: Modified Extended BDF scheme for the discontinuous Galerkin solution of unsteady compressible flows. Int. J. Numer. Methods Fluids Under Review (2014)
50. Oliver, T.A.: A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations. PhD thesis, MIT, Cambridge, MA, USA (2008)
51. Oliver, T.A., Darmofal, D.L.: Impact of turbulence model irregularity on high-order discretizations. AIAA Paper 2009-953 (2009)
52. Spalart, P., Jou, W., Strelets, M., Allmaras, S.: Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In: International Conference on DNS/LES, Ruston, Louisiana, August 4-8 (1997)
53. Spalart, P.R., Allmaras, S.R.: A one-equation turbulent model for aerodynamic flows. AIAA Paper 92-0439 (1992)
54. Steinebach, G.: Order-reduction of ROW-methods for DAEs and method of lines applications. Preprint-Nr. 1741, FB Mathematik, TH Darmstadt (1995)
55. Tesini, P.: An h-Multigrid Approach for High-Order Discontinuous Galerkin Methods. PhD thesis, University of Bergamo, Bergamo, Italy (January 2008)
56. Wang, Z.J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., Kroll, N., May, G., Persson, P.-O., van Leer, B., Visbal, M.: High-order CFD methods: Current status and perspective. Int. J. Numer. Methods Fluids 72(8), 811–845 (2013)
57. Weiss, J.M., Smith, W.A.: Preconditioning applied to variable and constant density flows. AIAA J. 33(11), 2050–2057 (1995)
58. Wilcox, D.C.: Turbulence Modelling for CFD. DCW industries Inc., La Cañada (1993)

High-Order, Linear and Non-linear Residual Distribution Schemes for Steady Compressible RANS Equations

R. Abgrall[1] and D. De Santis[2]

[1] Institut für Mathematik Universität Zürich Winterthurerstrasse 190
CH-8057 Zürich, Switzerland
remi.abgrall@math.uzh.ch
[2] INRIA Bordeaux Sud-Ouest, 200 avenue de la Vieille Tour,
33405 Talence Cedex, France
dante.de_santis@inria.fr

Abstract. Linear and non-linear Residual Distribution schemes for the discretization of the RANS equations are presented. Non-linear schemes are particularly suited for the discretization of transonic flows due to their capacity to give a monotone approximation of discontinuous solutions without the necessity to add artificial viscosity. A non-linear LU-SGS solver is considered to construct a robust implicit solver for the discretization of two and three-dimensional problems.

Keywords: Residual Distribution, turbulence modeling, implicit scheme.

1 Introduction

Residual Distribution (RD) schemes have been used for long time for the discretization of advective problems, like the Euler equations for example. RD methods, being based on a continuous approximation of the solution, introduce less degrees of freedom (DoFs) than high-order discontinuous methods. At the same time, for a compact stencil, they easily give an optimal order of accuracy for the advective terms. Furthermore, non-linear RD schemes are naturally able to give a monotone approximation of discontinuous solutions without the necessity to add artificial viscosity or using complex shock capturing approaches.

The extension of RD methods to advection-diffusion problems has always been less clear. Recently, a generally strategy for an accurate discretization of diffusive terms with the RD framework has been proposed [3, 7]. The approach relies on an accurate reconstruction of the gradient of the numerical solution in order to preserve the overall accuracy of the numerical scheme in advective and diffusive limits, and for the whole spectrum in between.

In this work linear and non-linear RD schemes are applied to the discretization of the steady RANS equations with the Spalart-Allmaras turbulence model at the second and third order of accuracy. Numerical results clearly show the benefit of using high-order approximation to get grid independent solutions with a smaller number of DoFs than that required by second-order approximation.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_12

2 Governing Equations

The compressible Reynolds Averaged Navier-Stokes equations coupled with the one equation Spalar-Allmaras turbulence model read

$$\frac{\partial \mathsf{u}}{\partial t} + \boldsymbol{\nabla}\cdot \mathbf{f}^a(\mathsf{u}) - \boldsymbol{\nabla}\cdot \mathbf{f}^v(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) = \mathsf{S}(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) \tag{1}$$

with the vectors of the conservative variables u, the advective and the diffusive fluxes defined as follows

$$\mathsf{u} = \begin{pmatrix} \rho \\ \boldsymbol{m} \\ E^t \\ \mu_t^\star \end{pmatrix}, \qquad \mathbf{f}^a(\mathsf{u}) = \begin{pmatrix} \boldsymbol{m} \\ \dfrac{\boldsymbol{m}\otimes\boldsymbol{m}}{\rho} + P\mathbb{I} \\ \big(E^t + P\big)\dfrac{\boldsymbol{m}}{\rho} \\ \mu_t^\star \dfrac{\boldsymbol{m}}{\rho} \end{pmatrix}, \qquad \mathbf{f}^v(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) = \begin{pmatrix} 0 \\ \mathbb{S} \\ \mathbb{S}\cdot\dfrac{\boldsymbol{m}}{\rho} - \boldsymbol{q} \\ \dfrac{\eta}{\sigma_a}\boldsymbol{\nabla}\left(\dfrac{\mu_t^\star}{\rho}\right) \end{pmatrix},$$

where ρ, $\boldsymbol{m}$, E^t and $\mu_t^\star$ denote the density, momentum, total energy per unit of volume and the turbulent working variable of the Spalar-Allmaras equation, respectively. The pressure P and the temperature T are given by the equations of state for the ideal gas. The stress tensor $\mathbb{S}$ is defined as follows

$$\mathbb{S} = -\frac{2}{3}(\mu+\mu_t)(\boldsymbol{\nabla}\cdot\boldsymbol{v})\mathbb{I} + (\mu+\mu_t)(\boldsymbol{\nabla}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{\nabla}\boldsymbol{v}),$$

where $\boldsymbol{v}$ is the velocity vector, $\mathbb{I}$ is the identity matrix, μ and μ_t are the fluid dynamic viscosity and the eddy viscosity, respectively. The laminar viscosity is computed via the Sutherland law, while the eddy viscosity is computed according to the Spalart-Allmaras model as $\mu_t = \mu_t^\star f_{v1}$ where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \text{with} \quad \chi = \frac{\mu_t^\star}{\mu_t}.$$

In the diffusive flux, the following definition has been used $\eta = \mu + \mu_t^\star$.

The heat flux $\boldsymbol{q}$ is computed by the means of the Fourier law

$$\boldsymbol{q} = -(\kappa + \kappa_t)\boldsymbol{\nabla}T$$

where κ, κ_t are, respectively, the laminar and the eddy thermal conductivity, which are computed using the Prandtl law, namely

$$\kappa = \frac{c_p\mu}{\mathrm{Pr}} \quad \text{and} \quad \kappa_t = \frac{c_p\mu_t}{\mathrm{Pr}_t},$$

where c_p is the specific heat at constant pressure, Pr and Pr_t are the laminar and the turbulent Prandtl numbers. They are set to be $\mathrm{Pr} = 0.72$ and $\mathrm{Pr}_t = 0.9$, respectively.

The source term of the RANS equations is defined as $\mathsf{S}(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) = (0, \mathbf{0}, 0, S_{\mathrm{SA}})^{\mathrm{T}}$, with S_{SA} the source term of the Spalart-Allmaras equation. In the actual implementation of the Spalart-Allmaras equation, two modifications have been adopted which have shown to be very effective in improving the robustness of the numerical scheme. The first modification consists in replacing the variable χ with the following function [8]

$$\psi_{\mu_t^\star} = \begin{cases} 0.05 \log\left(1 + e^{20\chi}\right), & \chi \leq 10, \\ \chi, & \chi > 10, \end{cases}$$

it has the role to deactivate the production and the destruction terms of the turbulence model equation when $\mu_t^\star$ becomes negative. The second modification re-defines the source term as follows

$$S_{\mathrm{SA}} = \begin{cases} c_{b1}\hat{\omega}\mu\psi_{\mu_t^\star} - \rho c_{w1} f_w \left(\dfrac{\mu\psi_{\mu_t^\star}}{\rho\, d_{\min}}\right)^2 + \dfrac{1}{\sigma_{SA}}\rho c_{b2} \left\| \boldsymbol{\nabla}\left(\dfrac{\mu_t^\star}{\rho}\right)\right\|^2, & \chi > 0, \\ c_{b1}\hat{\omega}\mu\psi_{\mu_t^\star} - \rho c_{w1} f_w \left(\dfrac{\mu\psi_{\mu_t^\star}}{\rho\, d_{\min}}\right)^2 & \chi \leq 0, \end{cases} \tag{2}$$

and the different functions involved in the Spalart-Allmaras equation read

$$\hat{\omega} = \begin{cases} \|\boldsymbol{\omega}\| + \bar{\omega}, & \bar{\omega} > -c_{v2}\|\boldsymbol{\omega}\|, \\ \|\boldsymbol{\omega}\| + \dfrac{\|\boldsymbol{\omega}\|\left(c_{v2}^2\|\boldsymbol{\omega}\| + c_{v3}\bar{\omega}\right)}{(c_{v3} - 2c_{v2})\|\boldsymbol{\omega}\| - \bar{\omega}}, & \bar{\omega} \leq -c_{v2}\|\boldsymbol{\omega}\|, \end{cases}$$

$$\eta = \mu(1 + \psi_{\mu_t^\star}),\ \bar{\omega} = \frac{\mu\psi_{\mu_t^\star} f_{v2}}{\rho\, k^2\, d_{\min}^2},\ f_{v1} = \frac{\psi_{\mu_t^\star}^3}{\psi_{\mu_t^\star}^3 + c_{v1}^3},\ f_{v2} = 1 - \frac{\psi_{\mu_t^\star}}{1 + \psi_{\mu_t^\star} f_{v1}},$$

$$f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}},\quad g = r + c_{w2}\left(r^6 - r\right),\quad r = \frac{\mu\psi_{\mu_t^\star}}{\rho\,\hat{\omega}\, k^2\, d_{\min}^2},$$

where $\|\boldsymbol{\omega}\|$ is the magnitude of the vorticity vector, $d_{\min}$ is the distance to the nearest wall, $c_{b1} = 0.1355$, $\sigma_{SA} = 2/3$, $c_{b2} = 0.622$, $k = 0.41$, $c_{w1} = \dfrac{c_{b1}}{k^2} + \dfrac{(1 + c_{b2})}{\sigma_{SA}}$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$.

Using the fact that the viscous flux is homogeneous with the respect of the gradient of the conservative variables, it is useful to re-write the viscous flux as follows

$$\mathbf{f}^v(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) = \mathbb{K}(\mathsf{u})\boldsymbol{\nabla}\mathsf{u}.$$

3 RD Space Discretization

The computational domain Ω is discretized with N_e non-overlapping elements with characteristic length h, the set of all the elements is denoted by $\mathcal{E}_h$, the

list of the DoFs is denoted by $\mathcal{N}_h$, and the total number of DoFs is N_{dof}. The solution is approximated on each element by k-th order polynomials which are assumed to be continuous within the elements and on the faces of the elements. If Lagrangian shape functions are used, the approximated solution u_h can be written as

$$\mathsf{u}_h(\boldsymbol{x}) = \sum_{i \in \mathcal{N}_h} \psi_i(\boldsymbol{x})\, \mathsf{u}_i, \quad \boldsymbol{x} \in \Omega,$$

with u_i the numerical solution at the generic DoF i. The approximated solution, will give rise to a residual on each element e, namely

$$\Phi^e\big(\mathsf{u}_h, \boldsymbol{\nabla}\mathsf{u}_h\big) = \int_{\Omega_e} \Big(\boldsymbol{\nabla}\cdot \big(\mathbf{f}^a\big(\mathsf{u}_h\big) - \mathbf{f}^v(\mathsf{u}_h, \boldsymbol{\nabla}\mathsf{u}_h)\big) - \mathsf{S}(\mathsf{u}_h, \boldsymbol{\nabla}\mathsf{u}_h)\Big)\, \mathrm{d}\Omega \tag{3}$$

The integral quantity $\Phi^e\big(\mathsf{u}_h, \boldsymbol{\nabla}\mathsf{u}_h\big)$ is called total residual of the element e. In practice, the total residual is never computed as reported in Eq. (3), the volume integral is transformed instead into a surface integral by the means of the divergence theorem. This procedure is legitimate for the divergence of the advective flux, since the numerical solution by definition is continuous across the face of two neighbor elements, however, in handling the divergence of the viscous flux one has to cope with the fact that the normal component of gradient of the numerical solution is discontinuous across the face of two adjacent elements. The continuity of the normal flux across the element boundary is one of the key arguments in [6] to show the consistency of the numerical scheme with respect to the original PDE. One way to overcome this difficulty would be to introduce numerical flux for the viscous contribution, in the spirit of [2], leading to the introduction of additional parameters that might be difficult to understand.

Suppose, now, that a unique value of the gradient of the numerical solution is available at each DoF, the gradient can be interpolated with the same shape functions used for the solution, and the total residual on the element can be written as follows

$$\begin{aligned} \Phi^e &= \oint_{\partial\Omega_e} \Big(\mathbf{f}^a\big(\mathsf{u}_h\big) - \mathbf{f}^v\big(\mathsf{u}_h, \widetilde{\boldsymbol{\nabla}\mathsf{u}_h}\big)\Big)\cdot \boldsymbol{n}\, \mathrm{d}\partial\Omega - \int_{\Omega_e} \mathsf{S}(\mathsf{u}_h, \boldsymbol{\nabla}\mathsf{u}_h)\, \mathrm{d}\Omega \\ &= \oint_{\partial\Omega_e} \Big(\mathbf{f}^a\big(\mathsf{u}_h\big) - \mathbb{K}\big(\mathsf{u}_h\big)\widetilde{\boldsymbol{\nabla}\mathsf{u}_h}\Big)\cdot \boldsymbol{n}\, \mathrm{d}\partial\Omega - \int_{\Omega_e} \mathsf{S}(\mathsf{u}_h, \boldsymbol{\nabla}\mathsf{u}_h)\, \mathrm{d}\Omega, \end{aligned} \tag{4}$$

where $\widetilde{\boldsymbol{\nabla}\mathsf{u}_h}$ is the interpolated gradient of the numerical solution, which is now continuous on the faces of the elements. Recovering the gradients at each DoF, with the same order of accuracy of the solution, is extremely important in order to preserve the overall accuracy of the numerical scheme, as will be showed later.

In order to handle only nodal values, the total residual is first distributed, in some way, to the DOFs of the element as follows

$$\Phi_i^e = \beta_i^e\big(\mathsf{u}_h\big)\, \Phi^e\big(\mathsf{u}_h, \widetilde{\boldsymbol{\nabla}\mathsf{u}_h}\big), \quad \forall i \in \mathcal{N}_h^e, \tag{5}$$

where $\mathcal{N}_h^e$ is the list of the DoFs of the element e and β_i^e are the distribution coefficients, which can be in general function of u_h. To obtain an equation for

each nodal value of the numerical solution, the following relations are written for each DoF

$$\sum_{e \in \mathcal{E}_{h,i}} \Phi_i^e = 0, \quad \forall i \in \mathcal{N}_h, \tag{6}$$

where $\mathcal{E}_{h,i}$ is the set of the elements which share the DoF i. The previous relations define a set of non-linear equations that must be solved for the nodal values of the solution $[\mathsf{u}_i]_{i=1,\ldots,N_{\text{dof}}}$.

3.1 Linear and Non-linear Schemes

The linear scheme proposed in this work is the extension to the integral formulation of the classical Ni's Lax-Wendroff scheme [9]. The distributed residual, for a generic DoF i of the element e, can be written as follows

$$\Phi_i^{e,\text{LW}} = \frac{\Phi^e}{N_{\text{dof}}^e} + \int_{\Omega_e} \boldsymbol{A} \cdot \boldsymbol{\nabla} \psi_i \, \Xi \, \boldsymbol{A} \cdot \boldsymbol{\nabla} \mathsf{u}_h \, \mathrm{d}\Omega, \tag{7}$$

where Ξ is a scaling matrix which is taken as

$$\Xi = \frac{1}{2} |\Omega_e| \left(\sum_{i \in \mathcal{N}_i^e} \mathsf{R}_{n_i}(\bar{\mathsf{u}}) \, \Lambda_{n_i}^+(\bar{\mathsf{u}}) \, \mathsf{L}_{n_i}(\bar{\mathsf{u}}) \right)^{-1}$$

where $\bar{\mathsf{u}}$ is the arithmetic average of the conservative variables on the element, R_n, L_n are respectively the matrices of the right and left eigenvectors along the direction of the generic vector $\boldsymbol{n}$, and $\Lambda_n = \text{diag}(\lambda_n)$ is the corresponding diagonal matrix of the eigenvalues. The operator $(\cdot)^+$ selects only the positive values and sets the negative ones to zero. The vector $\boldsymbol{n}_i$ is taken as follows

$$\boldsymbol{n}_i = \frac{1}{N_{\text{dim}}} \int_{\Omega_e} \boldsymbol{\nabla} \psi_i \, \mathrm{d}\Omega,$$

such that it has the dimensions of a length (surface) in two (three) spatial dimensions.

Non-linear schemes are needed to combine the non-oscillatory behavior with the higher-order discretization. The basic idea to construct a non-linear scheme is to start with a first-order, monotone scheme, and to map its distributed residuals onto a set of positive and non-linear residuals [1]. To see in practice how to construct a non-linear scheme, consider the first-order accurate and positive Rusanov' scheme (also know as Lax-Friedrichs scheme) defined as

$$\Phi_i^{e,\text{Rv}} = \frac{1}{N_{\text{dof}}^e} \Phi^e + \frac{1}{N_{\text{dof}}^e} \alpha^e \sum_{\substack{j \in \mathcal{N}_h^e \\ j \neq i}} (\mathsf{u}_i - \mathsf{u}_j),$$

with α^e taken as

$$\alpha^e = \max_{j \in \mathcal{N}_h^e} |\lambda_{n_j}| > 0.$$

In the next step, the distribution coefficients of the low-order scheme are mapped into non-linear bounded distribution coefficients by the means of a non-linear map. As described in [1], the mapping for a system of equations is constructed in the characteristic space, consequently the distributed and the total residuals are first rewritten as

$$\Phi_i^{e,\star} = \mathsf{L}_n \Phi_i^{e,\mathrm{Rv}} \qquad \text{and} \qquad \Phi^{e,\star} = \sum_{i \in \mathcal{N}_h^e} \Phi_i^{e,\star},$$

where the mean fluid velocity vector on the element is used as direction vector for the computation of the eigenvectors. The distributed high-order residuals are obtained by applying a non-linear mapping to the original unbounded distribution coefficients, $\beta_i^{e,\star} = \Phi_i^{e,\star} / \Phi^{e,\star}$. The map is constructed as follows

$$\hat{\beta}_i^{e,\star} = \frac{\left(\dfrac{\Phi_i^{e,\star}}{\Phi^{e,\star}} \right)^+}{\displaystyle\sum_{j \in \mathcal{N}_h^e} \left(\frac{\Phi_j^{e,\star}}{\Phi^{e,\star}} \right)^+},$$

which corresponds to the scalar PSI limiter. The limited residual in the characteristic space is firstly computed as follows

$$\hat{\Phi}_i^{e,\star} = \hat{\beta}_i^{e,\star}\, \Phi^{e,\star},$$

and finally, the high-order distributed residuals are projected back onto the physical space: $\hat{\Phi}_i^e = \mathsf{R}_n \hat{\Phi}_i^{e,\star}$.

The use of a central scheme, like the Rusanov' scheme, in combination with the limiting technique, produces undamped spurious modes and a poor iterative convergence to the steady state solution [1], due to the fact that no upwind mechanism is included. The cure to this problem consists in adding a filtering term by means of a streamline dissipation term

$$\hat{\Phi}_i^{e,\mathrm{Rv}} = \hat{\Phi}_i^e + \varepsilon_h^e(\mathsf{u}_h) \int_{\Omega_e} \boldsymbol{A} \cdot \boldsymbol{\nabla} \psi_i \, \Xi \, \boldsymbol{A} \cdot \boldsymbol{\nabla} \mathsf{u}_h \, \mathrm{d}\Omega, \tag{8}$$

where

$$\Xi = \frac{1}{2} |\Omega_e| \left(\sum_{i \in \mathcal{N}_h^e} \mathsf{R}_{n_i}(\bar{\mathsf{u}})\, \Lambda_{n_i}^+(\bar{\mathsf{u}})\, \mathsf{L}_{n_i}(\bar{\mathsf{u}}) \right)^{-1},$$

and $\varepsilon_h^e(\mathsf{u}_h)$ is a smoothness sensor which assures that the filtering term is added only in the smooth regions of the solution, namely $\varepsilon_h^e(\mathsf{u}_h) \sim 1$ in smooth regions and $\varepsilon_h^e(\mathsf{u}_h) \sim 0$ near discontinuities [5].

As described in [4], an improved version of the previous scheme takes into account the difference between the reconstructed and the internal gradient on the element, namely

$$\Phi_i^{e,\mathrm{LW}} = \frac{\Phi^e}{N^e_{\mathrm{dof}}} + \int_{\Omega_e} \boldsymbol{A}\cdot\boldsymbol{\nabla}\psi_i\, \Xi\left(\boldsymbol{A}\cdot\boldsymbol{\nabla}\mathsf{u}_h - \boldsymbol{\nabla}\cdot\left(\mathbb{K}\widetilde{\boldsymbol{\nabla}\mathsf{u}}\right)\right)\mathrm{d}\Omega$$

$$+ \int_{\Omega_e} \mathbb{K}\boldsymbol{\nabla}\psi_i\cdot\left(\boldsymbol{\nabla}\mathsf{u}_h - \widetilde{\boldsymbol{\nabla}\mathsf{u}_h}\right)\mathrm{d}\Omega$$

$$\hat{\Phi}_i^{e,\mathrm{Rv}} = \hat{\Phi}_i^e + \varepsilon_h^e(\mathsf{u}_h)\int_{\Omega_e}\left(\boldsymbol{A}\cdot\boldsymbol{\nabla}\psi_i - \mathbb{K}\boldsymbol{\nabla}\psi_i\right)\Xi\left(\boldsymbol{A}\cdot\boldsymbol{\nabla}\mathsf{u}_h - \boldsymbol{\nabla}\cdot\left(\mathbb{K}\widetilde{\boldsymbol{\nabla}\mathsf{u}_h}\right)\right)\mathrm{d}\Omega$$

$$+ \int_{\Omega_e}\mathbb{K}\boldsymbol{\nabla}\psi_i\cdot\left(\boldsymbol{\nabla}\mathsf{u}_h - \widetilde{\boldsymbol{\nabla}\mathsf{u}_h}\right)\mathrm{d}\Omega$$

3.2 Gradient Recovery Strategies

As explained in the previous Section, one has to assume that a continuous value of the gradient of the numerical solution is available on the faces of the elements. The strategy adopted here to achieve this goal consists in recovering the gradients at every DoF of the element, then the nodal values of gradients are interpolated with the Lagrangian shape functions. It is evident that the gradient reconstruction represents a key point to get a higher-order of accuracy; on the other hand the computational effort in the gradient reconstruction should be as limited as possible.

In reference [4] different reconstruction strategies were considered, and it was found that the so-called Super-convergent Patch Recovery method proposed by Zienkiewicz and Zhu [14, 15] (SPR-ZZ) offers the possibility to reconstruct the gradients with the same order of accuracy of the solution, without spoiling the compactness of the numerical scheme. For sake of clarity this method is briefly recalled here.

Assume that u^h is the piecewise continuous polynomial interpolation of the solution of degree k. The aim is to reconstruct the gradient of the solution, $\widetilde{\boldsymbol{\nabla}\mathsf{u}^h}$, at the DoFs, with $(k+1)$-th order of accuracy. The components of the reconstructed gradient, for each grid vertex, are written in a polynomial form as follows

$$\widetilde{\frac{\partial \mathsf{u}_h}{\partial x}} = \boldsymbol{p}^{\mathrm{T}}\boldsymbol{a}_x, \quad \text{and} \quad \widetilde{\frac{\partial \mathsf{u}_h}{\partial y}} = \boldsymbol{p}^{\mathrm{T}}\boldsymbol{a}_y,$$

with $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x, y, x^2, \ldots, x^k, x^{k-1}y, \ldots, y^k)$, $\boldsymbol{a}_x = (a_{x_1}, a_{x_2}, \ldots, a_{x_m})$ and $\boldsymbol{a}_y = (a_{y_1}, a_{y_2}, \ldots, a_{y_m})$. Assuming that N_{s_i} sampling points, $\boldsymbol{x}_{i,\ell}, \ell = 1 \ldots N_{s_i}$, are available for each vertex i, the values of the coefficients $\boldsymbol{a}_x$ and $\boldsymbol{a}_y$ are obtained by minimizing, with respect to the polynomial coefficients, the functions

$$F_{x_i} = \sum_{k=1}^{N_{s_i}}\left(\frac{\partial \mathsf{u}_h}{\partial x}(\boldsymbol{x}_{i,k}) - \boldsymbol{p}_{i,k}^{\mathrm{T}}\boldsymbol{a}_{x_i}\right)^2 \quad \text{and} \quad F_{y_i} = \sum_{k=1}^{N_{s_i}}\left(\frac{\partial \mathsf{u}_h}{\partial y}(\boldsymbol{x}_{i,k}) - \boldsymbol{p}_{i,k}^{\mathrm{T}}\boldsymbol{a}_{y_i}\right)^2,$$

with $\boldsymbol{p}_{i,k} = \boldsymbol{p}(\boldsymbol{x}_{i,k})$. The minimization problems require, for each vertex i of the grid, the solution of the following linear systems: $A_i \boldsymbol{a}_{x_i} = \boldsymbol{b}^h_{x_i}$, and $A_i \boldsymbol{a}_{y_i} = \boldsymbol{b}^h_{y_i}$, in a least-square sense, where

$$\boldsymbol{b}^h_{x_i} = \begin{pmatrix} \dfrac{\partial \mathsf{u}_h}{\partial x}(\boldsymbol{x}_{i,1}) \\ \dfrac{\partial \mathsf{u}_h}{\partial x}(\boldsymbol{x}_{i,2}) \\ \vdots \\ \dfrac{\partial \mathsf{u}_h}{\partial x}(\boldsymbol{x}_{i,N_{s_i}}) \end{pmatrix}, \quad \boldsymbol{b}^h_{y_i} = \begin{pmatrix} \dfrac{\partial \mathsf{u}_h}{\partial y}(\boldsymbol{x}_{i,1}) \\ \dfrac{\partial \mathsf{u}_h}{\partial y}(\boldsymbol{x}_{i,2}) \\ \vdots \\ \dfrac{\partial \mathsf{u}_h}{\partial y}(\boldsymbol{x}_{i,N_{s_i}}) \end{pmatrix}$$

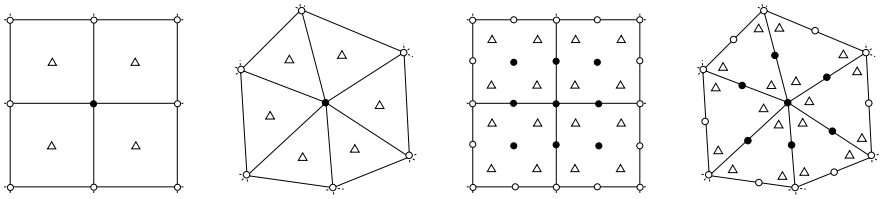
$$A_i = \begin{pmatrix} 1 & x_{i,1} & y_{i,1} & \dots & y^k_{i,1} \\ 1 & x_{i,2} & y_{i,2} & \dots & y^k_{i,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i,N_{s_i}} & y_{i,N_{s_i}} & \dots & y^k_{i,N_{s_i}} \end{pmatrix}.$$

Once the vectors $\boldsymbol{a}_{x_i}$ and $\boldsymbol{a}_{y_i}$ are known, the gradient $\widetilde{\boldsymbol{\nabla} \mathsf{u}_h}$ can be evaluated at every point within the patch of vertex i. Since the matrix A_i depends only on the geometry, it is computed and factorized only once.

The least-square problem is solved for each grid vertex, but not for the extra nodes introduced by a higher-approximation of the solution. For these nodes, the gradient is reconstructed by simply evaluating on the patch, at the coordinates of the nodes, the polynomial function constructed for the nearest grid vertex. Note that these node may belong to more than one patch, each of them is equally valid to evaluate the gradient. An arithmetic average is used to uniquely define the reconstructed gradient at these nodes.

The dimensions of the matrix A_i, for each vertex i, are determined by the number of sampling points N_{s_i} and by the degree of the polynomials used to express the reconstructed gradient, that is $A_i \in \mathbb{R}^{N_{s_i} \times m}$, where m is the number of the coefficients in the vectors $\boldsymbol{a}_{x_i}$ and $\boldsymbol{a}_{y_i}$. The problem admits a unique solution if Rank $A_i = m$, which is always satisfied in the case in which $N_{s_i} \geq m$.

Generally, the number of elements contained in the patch is such that the condition $N_{s_i} \geq m$ is always satisfied, this means that the gradient reconstruction is compact because it involves only the elements contained within the support of a grid node. For the nodes belonging to the boundary, the condition $N_{s_i} \geq m$ might not be satisfied without enlarging the stencil, otherwise the problem is ill conditioned. To avoid the use of larger stencils for boundary nodes, it is possible to obtain the value of the reconstructed gradient at the boundary by evaluating, at the boundary node, the polynomial expansion already computed for the nearest domain vertex, *i.e.*, the interior patch is used to evaluate the gradient at the boundary.



(a) (b) (c) (d)

Fig. 1. Interior super-convergent patches for quadrilateral and triangular elements: (a-b) linear elements, (c-d) quadratic elements. The symbols (∘) indicate the patch assembly points, the symbols (•) indicate the points where the gradient is reconstructed and the symbols (△) indicate the super-convergent sampling points.

4 Implicit Time Integration

Since high-order methods are generally less robust than standard low-order methods, the construction of a reliable numerical scheme becomes crucial. In fact, it has been observed in numerical simulations of turbulent flows at very high Reynolds number, typical of aerodynamic applications, that the iterative scheme is very prone to fail with a sudden breakdown of the solver or it stagnates at high values of the residual.

For these reasons, instead of using a Newton Krylov method, a new strategy has been adopted for the solution of the non-linear system obtained from the discretization of the RANS equations. The new approach is based on the manipulation of the Gauss-Seidel method in order to avoid the calculation of the off-diagonal elements in the Jacobian matrix.

Consider the use of the backward Euler method for the time integration solved with a Newton's approach, the resulting scheme can be written as

$$\left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}\right) \Delta \mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n).$$

The previous equation can be recasted in the following form

$$\left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i}\right) \Delta \mathsf{u}_i^n = -\mathsf{R}_i(\mathsf{u}^n) - \sum_{\substack{j \in \mathcal{N}_h^i \\ j \neq i}} \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_j} \Delta \mathsf{u}_j^n, \quad \forall\, i \in \mathcal{N}_h, \tag{9}$$

with $\mathcal{N}_h^i$ the set of the DoFs which belong to the stencil of the DoF i. By applying a symmetric variation of the Gauss-Seidel method, in which the forward and backward sweeps are applied alternatively, one obtains

$$\left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i}\right) \Delta \mathsf{u}_i^{(k+1)} = -\mathsf{R}_i(\mathsf{u}^n) - \sum_{\substack{j \in \mathcal{N}_h^i \\ j \neq i}} \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_j} \Delta \mathsf{u}_j^{(*)}, \quad \forall\, i \in \mathcal{N}_h, \tag{10}$$

where k is an inner iteration index, the superscript $^{(*)}$ indicates the most recently updated solution, and $\Delta \mathsf{u}^{(0)} = 0$. At the end, $\mathsf{u}^{n+1} = \mathsf{u}^n + \Delta \mathsf{u}^{(k+1)}$.

The right-hand side of the previous equation can be further manipulated as follows. Denoting the latest available solution as $\mathsf{u}^{(*)} = \mathsf{u}^n + \Delta \mathsf{u}^{(*)}$, it is possible to linearize $\mathsf{R}_i\big(\mathsf{u}^{(*)}\big)$ as

$$\begin{aligned}\mathsf{R}_i\big(\mathsf{u}^{(*)}\big) &\approx \mathsf{R}_i\big(\mathsf{u}^n\big) + \sum_{j \in \mathcal{N}_h^i} \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_j} \Delta \mathsf{u}_j^{(*)} \\ &= \mathsf{R}_i\big(\mathsf{u}^n\big) + \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i} \Delta \mathsf{u}_i^{(*)} + \sum_{\substack{j \in \mathcal{N}_h^i \\ j \neq i}} \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_j} \Delta \mathsf{u}_j^{(*)},\end{aligned}$$

and the following relation is obtained

$$\mathsf{R}_i\big(\mathsf{u}^n\big) + \sum_{\substack{j \in \mathcal{N}_h^i \\ j \neq i}} \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_j} \Delta \mathsf{u}_j^{(*)} = \mathsf{R}_i\big(\mathsf{u}^{(*)}\big) - \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i} \Delta \mathsf{u}_i^{(*)}.$$

Substituting the previous relation in Eq. (10), one obtains the following scheme

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i}\right] \Delta \mathsf{u}_i^{(k+1)} - \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i} \Delta \mathsf{u}_i^{(*)} = -\mathsf{R}_i\big(\mathsf{u}^{(*)}\big),$$

which at last can be recasted as

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i}\right] \left(\Delta \mathsf{u}_i^{(k+1)} - \Delta \mathsf{u}_i^{(*)}\right) = -\mathsf{R}_i\big(\mathsf{u}^{(*)}\big) + \frac{\Delta \mathsf{u}_i^{(*)}}{\Delta t^n}. \tag{11}$$

The Eq. (11) is solved with the forward and backward sweeps, and at the beginning of each step the small diagonal blocks of the Jacobian matrix in the left-hand side can be inverted using LU decomposition. Note that the right-hand side of the previous equation is nothing but the residual evaluated with the latest available solutions. The last term in the right hand side of the (11) is usually dropped for steady simulations, in order to accelerate the convergence rate. When the symmetric Gauss-Seidel converges, one is actually solving the original equation

$$\frac{\mathsf{u}_h^{n+1} - u_h^n}{\Delta^n} = -R(\mathsf{u}_h^{n+1})$$

instead of the linearized version (9), for this reason the algorithm is called nonlinear LU-SGS [13].

The expression (11) still requires the diagonal blocks of the Jacobian matrix, in some works [10,13] the diagonal blocks of the Jacobian are approximated with a finite difference method, however this strategy has been found unsatisfactory for RANS simulations, thus diagonal blocks of the Jacobian matrix computed analytically are used instead.

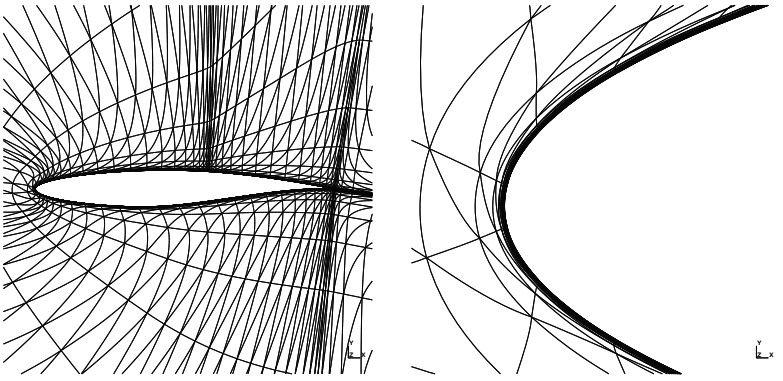
5 Numerical-Results

5.1 RAE-2822 Airfoil

The first test case considered is the subsonic flow over a RAE2822 airfoil at angle of incidence $\alpha = 2.79°$, the Mach number based on the free-stream conditions is $M = 0.4$ and the Reynolds number based on the free-stream conditions and the airfoil chord is $Re = 6.5 \times 10^6$.

The problem is solved, with the linear scheme and the SPR-ZZ method, on a sequence of unstructured grids of triangles obtained with subsequent uniform refinements of the coarsest grid, shown in Figure 2, which is made of 4 048 elements. As practice in this work, an iso-parametric representation is used for the solution and the geometry. Due to the strong stretching of the boundary layer elements, some of the quadratic elements have curved boundaries non only on the airfoil wall but also at the interior. The solution is converged using the non-linear LU-SGS method; solutions are considered to be at convergence when the L^2 norm of the residual of all the variables (mean flow and turbulence equation) is reduced by ten orders of magnitude.

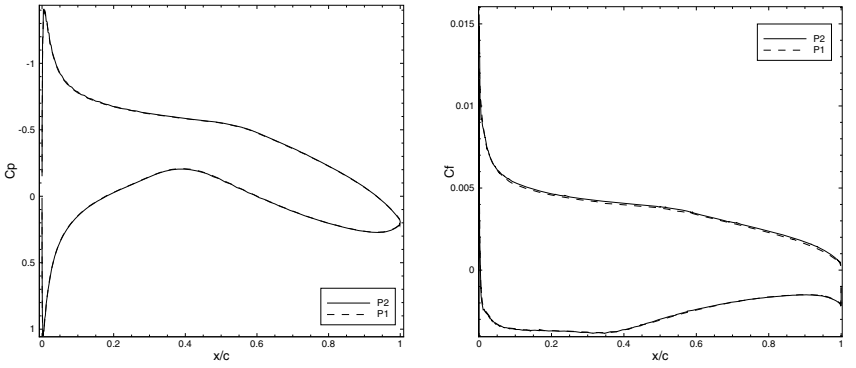
In Figure 3 are shown the pressure and friction coefficient along the airfoil computed with linear and quadratic elements on two different grids, such that the number of degrees of freedom of the two simulations is the same ($N_{\text{dof}} = 32\,784$); the respective iterative convergence histories are reported in Figure 4.



(a) (b)

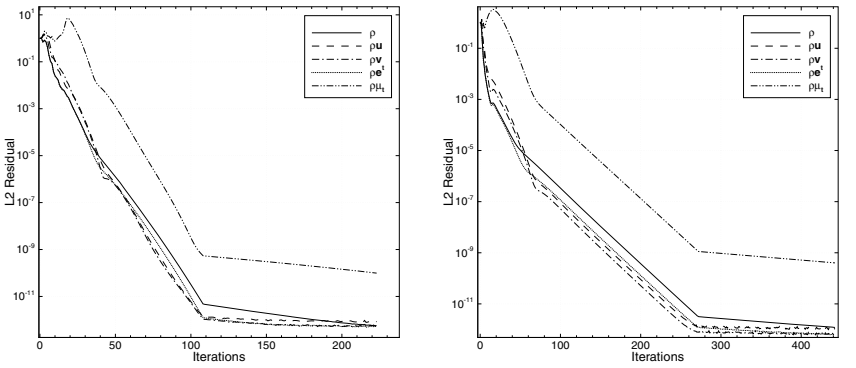
Fig. 2. An example of coarse grid used for the numerical simulation of a turbulent flow over the RAE2822 airfoil and a zoom of the mesh near the leading edge of the airfoil to show the curvilinear elements

For completeness, in Figure 5 are reported the values of the lift and drag coefficients computed on different grids with linear and quadratic elements. Note how the use of quadratic elements allows a faster convergence with the mesh refinement of the lift and drag coefficients to their respective asymptotic values. In addition, Figure 6 reports the errors for the force coefficients; the error for the



(a) (b)

Fig. 3. Pressure (a) and friction (b) coefficient along the RAE2822 airfoil with linear and quadratic elements on two different grids, such that $N_{\text{dof}} = 32\,784$

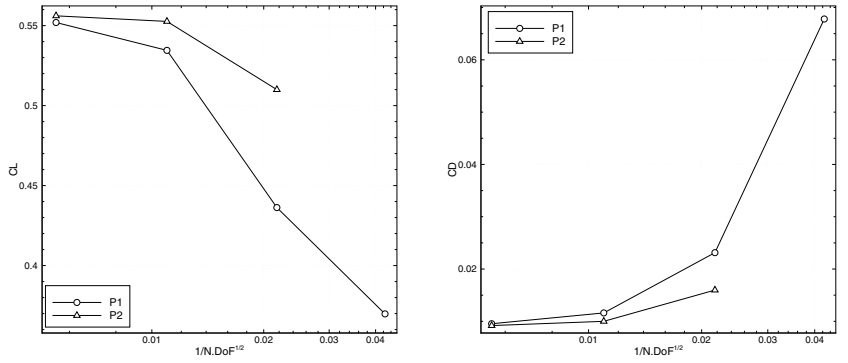


(a) (b)

Fig. 4. Iterative convergence for the simulation of the subsonic flow over the RAE2822 airfoil with linear (a) and quadratic (b) elements on two different grids, such that $N_{\text{dof}} = 32\,784$

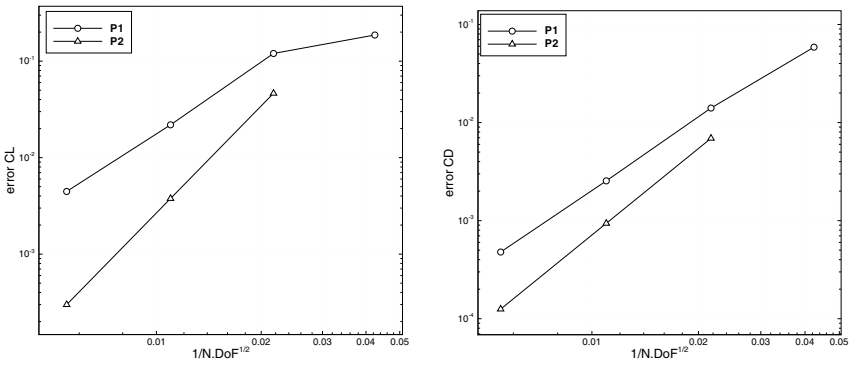
lift coefficient is computed as $err_{\text{CL}} = |c_{L_h} - c_{L_{\text{ex}}}|$, where c_{L_h} is the value of the lift coefficient computed numerically and $c_{L_{\text{ex}}}$ is a reference value obtained with a Richardson extrapolation [12] of the lift coefficients computed with quadratic elements. The same procedure is also used to compute the error on the drag coefficient. For the lift coefficient, it is clear the advantage of using a high-order approximation over standard second-order schemes even for the simulations of RANS equations. For the drag coefficient, the difference between second and third-order schemes reduces but nevertheless the quadratic approximation of the solutions gives a smaller error, for the same number of degrees of freedom.

In the next test case a transonic, turbulent flow over a RAE2822 airfoil is considered. The free-stream Mach number is M = 0.734, the angle of incidence



(a) (b)

Fig. 5. Values of the lift (a) and drag (b) coefficients as function of the number of the degrees of freedom, for the subsonic turbulent flow over the RAE2822 airfoil, with linear and quadratic elements



(a) (b)

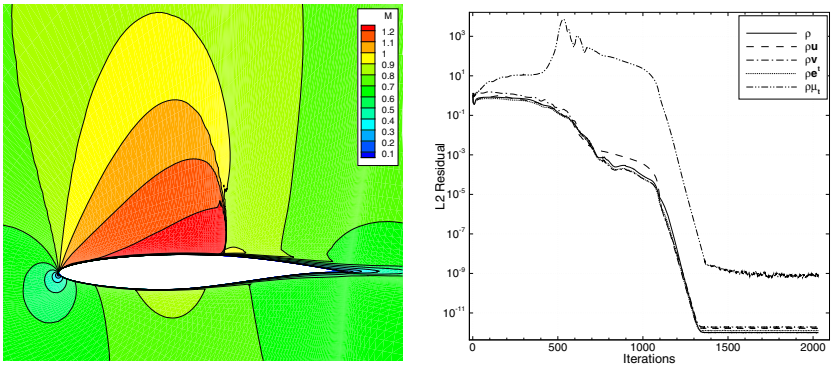
Fig. 6. Errors with linear and quadratic elements on the lift (a) and drag (b) coefficients for the subsonic turbulent flow over the RAE222 airfoil

and the Reynolds number are the same of the previous test case, namely $\alpha = 2.79°$ and $\mathrm{Re} = 6.5 \times 10^6$. In these conditions, the upper surface of the airfoil is characterized by the presence of a shock wave which impinges the boundary layer, generating a small recirculation bubble behind the shock.

Due to the presence of the shock, the problem is discretized with the non-linear scheme, and the SPR-ZZ method is used for the gradient reconstruction. Furthermore, the solution is converged using the non-linear LU-SGS method; the L^2 norm of the residual of the mean flow and of the turbulent equation is reduced at least by nine orders of magnitudes.

The same type of grids used in the previous test case is adopted and three levels of refinement are considered; simulations have been performed in parallel

on 8, 16 and 32 processors, respectively. In Figure 7 are reported the Mach number contours for the solution on the fine grid (64 768 elements) with quadratic elements, together with the iterative convergence history. Note that the shock is sharply captured, and the number of iterations to reach the steady state is much higher compared to the subsonic test case, due to the presence of the shock and also due to the use of the non-linear scheme.

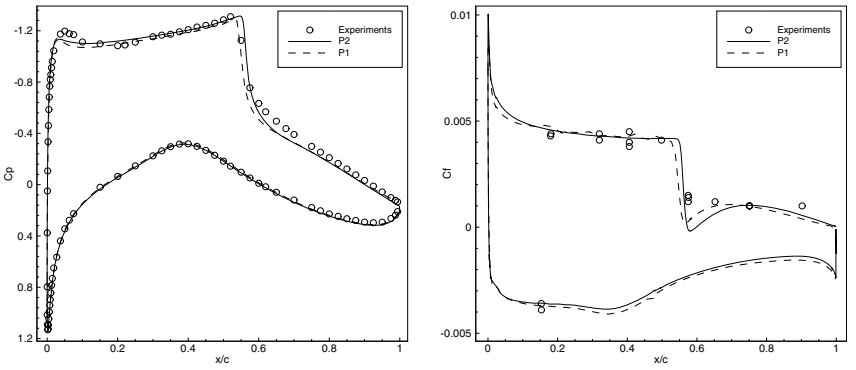


(a) (b)

Fig. 7. Mach number contours (a) and iterative convergence history (b) for the transonic turbulent flow over a RAE2822 airfoil computed with quadratic elements

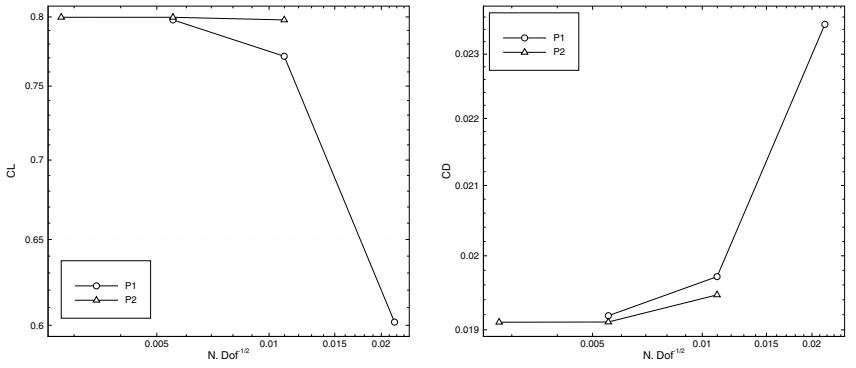
In Figure 8 are shown the pressure and the friction coefficients along the airfoil with linear elements on a finer grid and quadratic elements on a coarse grid, such that the number of degrees of freedom in the two cases is the same: 32 784. The values of the computed pressure coefficient agree very well with the experimental data, and with the quadratic discretization of the solution the agreement is even improved. The shock is sharply captured and its position correctly predicted. The leading edge suction peak of the pressure is slightly under-predicted by the numerical simulations, due to the fact that the simulations have been performed in a fully turbulent regime. Even the friction coefficient agrees well with the experimental data. On the upper part of the airfoil, note the negative values of the friction coefficient just behind the shock, obtained with the quadratic approximation of the solution, that identify the separation bubble induced by the shock.

Finally, in Figure 9 are reported the values of lift and drag coefficients for linear and quadratic approximation of the solution with three levels of grid refinement; the quadratic solution reaches much faster than the linear one the reference value, for the same number of degrees of freedom.



(a) (b)

Fig. 8. Pressure (a) and friction (b) coefficients over the RAE2822 airfoil, for the transonic turbulent flow computed with linear and quadratic elements on two grids, such that $N_{\text{dof}} = 32\,784$

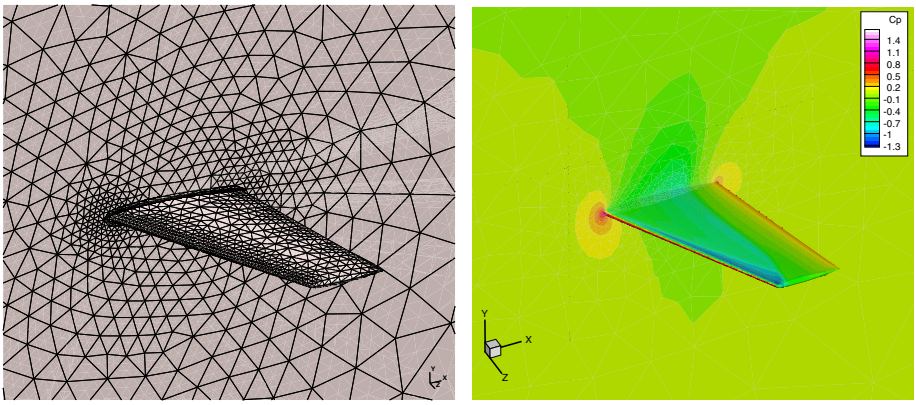


(a) (b)

Fig. 9. Lift (a) and drag (b) coefficients on different grids, for the transonic turbulent flow over the RAE2822 airfoil, with linear and quadratic elements

5.2 ONERA M6 Wing

The simulation of the transonic turbulent flow over the ONERA M6 wing is performed. The free-stream Mach number is $\mathrm{M} = 0.8395$, the angle of incidence is $\alpha = 3.06°$ and the Reynolds number based of the free-stream values and the mean aerodynamic chord is $\mathrm{Re} = 11.72 \times 10^6$. The grid is composed by 123 444 tetrahedra, and the mean height of elements on the wing in dimensionless units is approximately $y_1^+ = 5$. Far-field boundary conditions are applied on the outer part of the domain, symmetry boundary conditions are applied on the vertical plane that intersects the wing root and the wing surface is modeled as a non-slip adiabatic wall. In Figure 10 are reported the grid used for the simulation, and the pressure coefficient contours for the third-order simulation.



(a) (b)

Fig. 10. Computational mesh used for the simulation of the turbulent flow over the M6 wing (a) and pressure coefficient contours (b) for the third-order simulation

The non-linear solver has been used to discretize the governing equations with the SPR-ZZ gradient reconstruction method, and the non-linear LU-SGS method has been used to make the scheme converge to the steady state solution. Convergence is considered achieved when the L^2 residual of all the equations is dropped by eight orders of magnitude. Simulations have been performed in parallel on 64 processors.

In figure Figure 11, the values of the pressure coefficient for the third-order simulation at different stations along the span-wise direction of the wing are compared against the experimental data. Despite the quite coarse mesh, a good agreement with the experimental data is obtained and even the lambda shock is well represented. However, a finer grid is required to better capture the shock structure at wing tip and also to better resolve the complex shock-boundary layer interaction occurring here. Finally, in Table 1 are reported the values of lift and drag coefficients for the second and third-order simulations.

5.3 NASA Delta Wing

The simulation of the subsonic turbulent flow over the NASA 65° sweep delta wing with a medium radius leading edge is now considered. The free-stream Mach number is $M = 0.4$, the angle of attach is $\alpha = 13.3°$ and the Reynolds number based on the free-stream conditions and the mean aerodynamic chord is $\mathrm{Re} = 3 \times 10^6$. This test case has been computed with the linear scheme using linear and quadratic approximation of the solution on a grid of 1 145 797 tetrahedra (Figure 12-a). Simulations has been performed in parallel using 96 cores. With linear elements the residual was reduced by 8 orders of magnitude, while with quadratic elements the residual could not be reduced by more than 5 orders of magnitude.

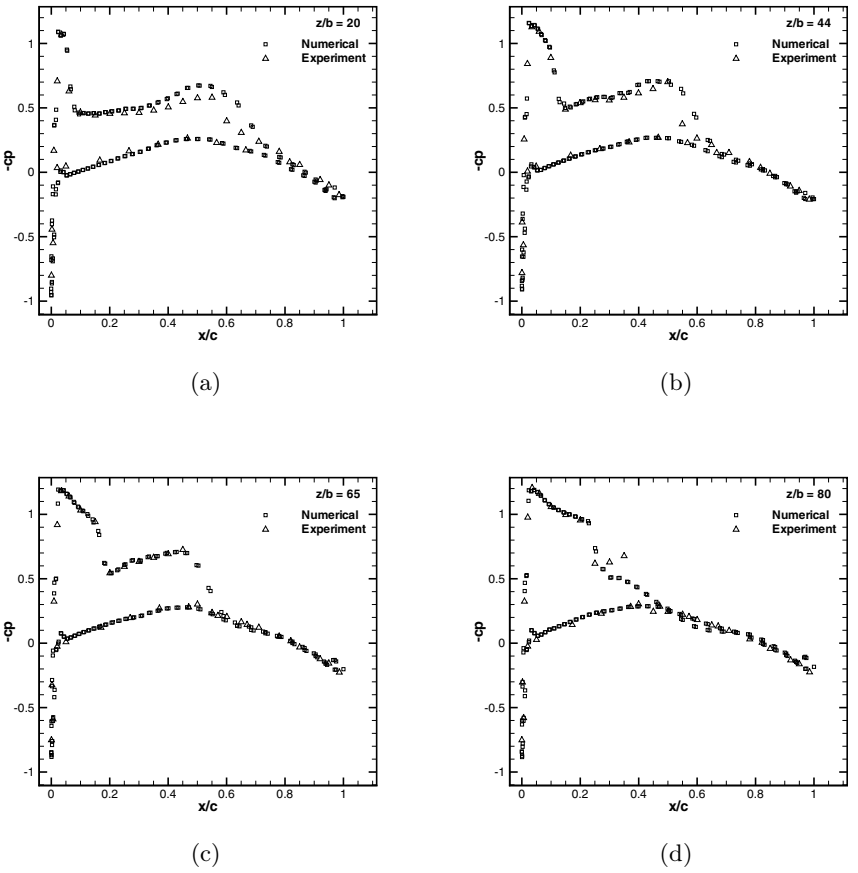
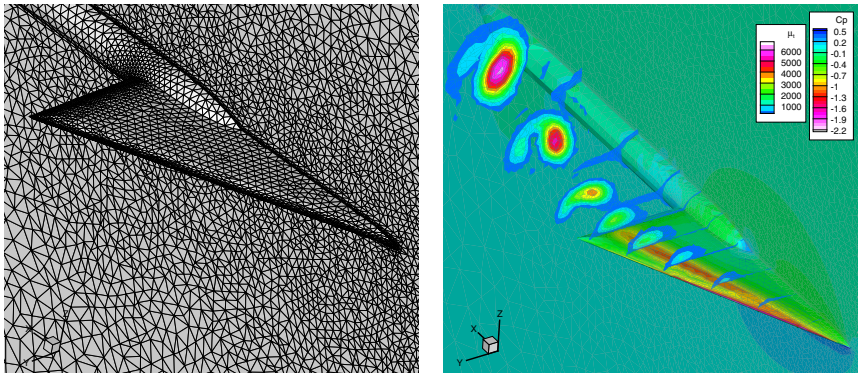


Fig. 11. Pressure coefficient distribution at different span-wise locations over the M6 wing, for the third-order simulation

Table 1. Lift and drag coefficients for the second and third-order turbulent simulations over the M6 wing

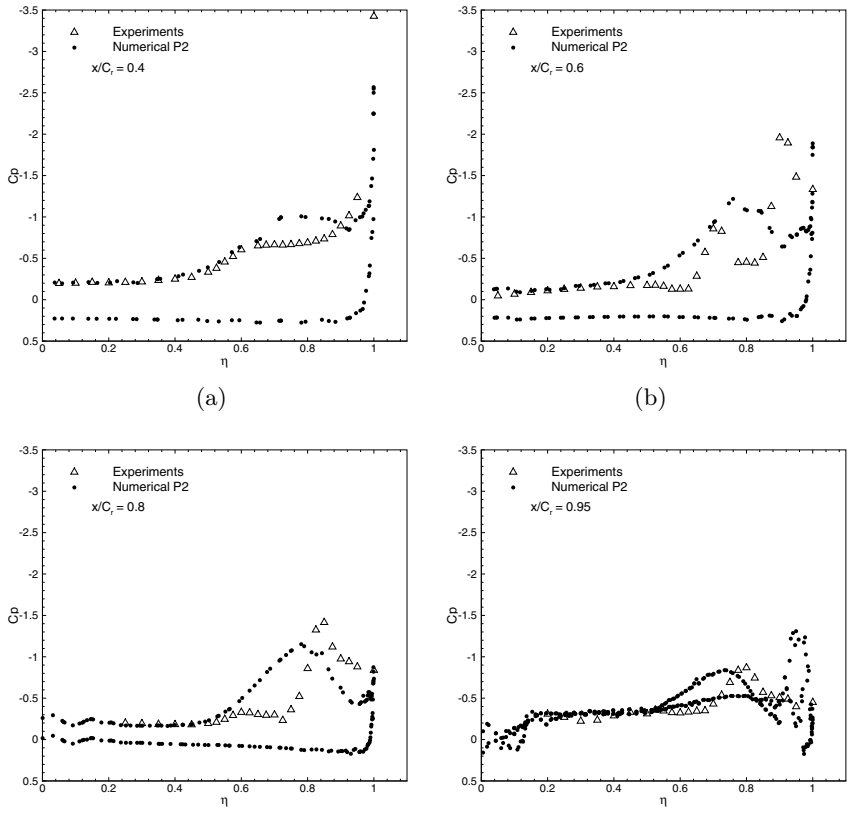
	N. DoF	CL	CD
$\mathbb{P}^1$	22 276	0.268231	0.019002
$\mathbb{P}^2$	170 751	0.270758	0.018554

In Figure 12-b are shown, for the solution with P2 elements, the contours of the pressure coefficients and slices of the turbulent working variable at different stations along and behind the wing. The comparison with the experimental results [11] is shown in Figure 13. The agreement between the numerical results and the experimental data is mildly good considering the quite coarse mesh and the simple turbulent model used.



(a) (b)

Fig. 12. 3D turbulent subsonic flow over a delta wing. Left: the mesh used, right pressure coefficient contours and slices of the turbulent working variable, with P2 elements.



(a) (b)

(c) (d)

Fig. 13. 3D turbulent subsonic flow over a delta wing. Pressure coefficients distribution at different wing sections. Comparison between the quadratic solution and the experimental data.

6 Conclusions

Flow computations with linear and non-linear RD method for compressible turbulent flows on unstructured grids have been reported in this work. The solution is approximated with linear and quadratic Lagrangian polynomials, and the one equation Spalart-Allmaras turbulence model has been used. The non-linear LU-SGS method has been used to construct a robust and efficient implicit scheme.

The benefit of using high-order approximation has been established with smooth and shocked flows, in two and three spatial dimensions. Numerical results confirm the benefit of using high-order approximation. In fact, it is shown that with high-order methods grid independent quantities, *e.g.* force coefficients, can be obtained with a reduced number of DoFs compared to lower-order approximation.

References

1. Abgrall, R.: Essentially non-oscillatory residual distribution schemes for hyperbolic problems. Journal of Computational Physics 214, 773–808 (2006)
2. Abgrall, R.: A residual distribution method using discontinuous elements for the computation of possibly non smooth flows. Advances in Applied Mathematics and Mechanics 2, 32–44 (2010)
3. Abgrall, R., De Santis, D.: Linear and non-linear high order accurate residual distribution schemes for the discretization of the steady compressible navier-stokes equations. Journal of Computational Physics (2013) (submitted)
4. Abgrall, R., De Santis, D.: High order preserving residual distribution schemes for advection-diffusion scalar problems on arbitrary grids. SIAM Journal on Scientific Computing (2014) (accepted)
5. Abgrall, R., Larat, A., Ricchiuto, M.: Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. Journal of Computational Physics 230, 4103–4136 (2011)
6. Abgrall, R., Roe, P.L.: High-order fluctuation schemes on triangular meshes. Journal of Scientific Computing 19(3), 3–36 (2003)
7. De Santis, D.: Development of a high order residual distribution method for navier Stokes and RANS equations. PhD thesis, University of Bordeaux (2013)
8. Moro, D., Nguyen, N.C., Peraire, J.: Navier-Stokes solution using hybridizable discontinuous Galerkin methods. In: 20th AIAA Computational Fluid Dynamics Conference. AIAA Paper 2011-3407 (2011)
9. Ni, R.H.: A multiple grid scheme for solving the Euler equations. In: 5th Computational Fluid Dynamics Conference, pp. 257–264 (1981)
10. Parsani, M., Ghorbaniasl, G., Lacor, C., Turkel, E.: An implicit high-order spectral difference approach for large eddy simulation. Journal of Computational Physics 229(14), 5373–5393 (2010)
11. Schröder, A., Konrath, R., Klein, C.: PSP and PIV investigations on the VFE-2 configuration in sub- and transonic flow. AIAA-Paper 2008-379 (2008)
12. Roache, P.J.: Verification and Validation in Computational Science and Engineering. Hermosa Publishers (1998)
13. Sun, Y., Wang, Z.J., Lun, Y.: Efficient implicit non-linear lu-sgs approach for compressible flow computation using high-order spectral difference method. Communications in Computational Physics 5(2-4), 760–778 (2009)

14. Zienkiewicz, O.C., Zhu, J.Z.: A simple error estimator and adaptive procedure for practical engineerng analysis. International Journal for Numerical Methods in Engineering 24(2), 337–357 (1987)
15. Zienkiewicz, O.C., Zhu, J.Z.: The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity. International Journal for Numerical Methods in Engineering 33(7), 1365–1382 (1992)

Development and Validation of a Massively Parallel High-Order Solver for DNS and LES of Industrial Flows

C.C. de Wiart and K. Hillewaert

Cenaero, Rue des frères Wright 29, B6041 Gosselies, Belgium
{corentin.carton,koen.hillewaert}@cenaero.be

Abstract. This work is part of the development of a new generation CFD solver, Argo, based on the discontinuous Galerkin Method (DGM), specifically targeted towards accurate, adaptive, reliable and fast DNS and LES of industrial aerodynamic flows. Several aspects were investigated in IDIHOM. A first activity was the optimisation of the parallelisation strategy, resulting in highly efficient scaling, demonstrated on some of the largest computers in Europe. A second activity concerned the assessment and validation on several academic benchmark problems of the capability of DGM to perform direct numerical simulation (DNS) and (implicit) Large Eddy Simulation (iLES). Two moderately complex flows are treated, namely the ILES of the transitional flow in the low pressure turbine cascade T106C and the isothermal jet issueing from the JEAN nozzle.

Keywords: Discontinuous Galerkin Method (DGM), High Performance Computing (HPC), Direct Numerical Simulation (DNS), (implicit) Large Eddy Simulation ((I)LES), Transitional flow, Industrial CFD.

1 Introduction

Computational Fluid Dynamics (CFD) has become an indispensable tool for both academic and industrial aerodynamical studies. The increase of computational power combined to the industrialisation of the solvers opened the way to the use of CFD in many sectors, such that it is now daily used for the design of aircraft, cars, energy production units, production processes, buildings, ... Nevertheless, a large gap still remains between the accuracy of the solvers used in industry and those developed by the academic community, in particular as far as the impact of turbulence is concerned. Indeed, the academic community is primarily focused on fundamental aspects of turbulence or the development of physical models, and therefore considers relatively simple setups on which very accurate methods and very low turbulence modeling contributions can be used. On the other hand, the industrial applications target a large range of flow regimes in complex configurations. Therefore, the solvers used by the industry often sacrifice accuracy for higher flexibility, robustness and a low computational cost. As a consequence, less accurate numerical schemes together with significant

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_13

contributions of empirical turbulence models, mainly *Reynolds averaged Navier Stokes (RANS)*, are the current industrial standard. Indeed in most cases the complexity of the geometry is such that only the averaged flow can be computed.

Due to the continuously increasing economical and environment constraints, the current standard methods are no longer sufficient to answer the design requirements of the industry, in particular when off-design performance, noise or combustion needs to be predicted. A need for better accuracy, together with smaller empirical turbulence modelling contributions, is required. In particular, industry would like to use *scale-resolving modeling* approaches such as *Direct Numerical Simulation (DNS)* or *Large Eddy Simulation (LES)*, which respectively represent all or the largest turbulent scales. This type of computation requires very high resolution and accuracy, such that in academia high-accuracy schemes tend to be used. In an industrial context it is commonly accepted that computational codes need to be revised. Indeed, as scale-resolving simulations require a nearly flawless representation of the turbulent scales, current industrial solvers require huge computational resources to provide sufficient accuracy, and hence, most computations appear to be under-resolved so far (see Tucker [35,36] for a recent review in turbomachinery). In a first step, discretisation techniques probably need to be changed to obtain higher order of accuracy. Furthermore, codes need to be optimised to provide high efficiency and strong scalability, in order to make computation time compatible with design lead times.

In the course of the last decade, CFD has seen the advent of new unstructured, finite element-like high order methods, such as the *discontinuous Galerkin method (DGM)* [20], *spectral element method (SEM)* [17,27], flux reconstruction (FR) [25] or *spectral difference method (SDM)* [33]. These methods appear to provide a compromise between the flexibility of industrial *finite volume methods (FVM)* [34] and the accuracy of academic solvers, such as high order *finite difference methods (FDM)* [37] or *pseudo-spectral methods (PSM)* [11]. Due to their computational compacity, most of these methods - in particular those with discontinuous interpolation - also provide an excellent serial and parallel computational efficiency. In view of these advantages, the industrial need for scale-resolving simulations and appropriate CFD codes, it is precisely in this domain that new high-resolution methods offer the best perspectives.

One of those novel methods, the discontinuous Galerkin method with symmetric interior penalty (DGM/SIP) has been implemented in Argo, the multi-physics platform of Cenaero, during the thesis of Koen Hillewaert [22], in close collaboration with Prof. Remacle of the *Université catholique de Louvain (UCL)*. The solver has been subsequently applied on the scale-resolving simulations of moderate Reynolds flows during the thesis of Corentin Carton de Wiart (Cenaero/UCL) [16].

Next to this introduction and the conclusions, this chapter comprises four technical sections. Section 2 discusses briefly the main characteristics of the DGM discretisation, whereas the second section 3 details the parallellisation strategy and demonstrates the scalability of the solver. In section 4 the method is assessed for DNS and LES on academic benchmarks. Finally, in section 5 the

method is applied on three IDIHOM test cases: the benchmark computation of the 2D periodic hill, and the semi-industrial configurations of the T106C turbine blade and the JEAN nozzle.

2 Numerical Method

For compacity reasons, the compressible Navier-Stokes equations are written as a set of generic convection-diffusion equations for the state vector $\tilde{w}$ containing the conservative variables $\tilde{w} = [\rho\ \rho\mathbf{u}\ \rho E]^T$ defined on a domain Ω:

$$\frac{\partial \tilde{w}_m}{\partial t} + \frac{\partial}{\partial x^k}\ \mathcal{F}_m^k(\tilde{w}) + \frac{\partial}{\partial x^k}\ \mathcal{D}_m^k(\tilde{w}, \nabla\tilde{w}) = 0\ ,\ \forall m. \tag{1}$$

Here m is the index running on the different variables in the state vector, whereas $\mathcal{F}$ and $\mathcal{D}$ respectively denote the convective and diffusive fluxes. We furthermore define the Jacobian of the diffusive fluxes with respect to the solution gradients as the fourth order tensor D_{mn}^{kl}, such that we have:

$$\mathcal{D}_m^k \approx D_{mn}^{kl}(\tilde{w})\frac{\partial \tilde{w}_n}{\partial x^l}\,. \tag{2}$$

Appropriate Dirichlet, Neumann and Robin boundary conditions are defined on the boundary $\partial\Omega$.

The discontinuous Galerkin method [6,15] is a Galerkin finite element method based on an interpolation space Φ, composed of functions v that are polynomials of order p on each of the elements e in the mesh $\mathcal{E}$, but not required to be continuous across the interfaces f between the elements. Such an interpolant is illustrated in Fig. 1. After choosing an appropriate set of basis functions ϕ_i for the test space Φ, such that

$$\Phi = span\{\phi_0, \ldots, \phi_N\},$$

DGM approximates the different components $\tilde{w}_m$ of $\tilde{w}$ by w_m

$$w_m = \sum_i W_{im}\phi_i \tag{3}$$

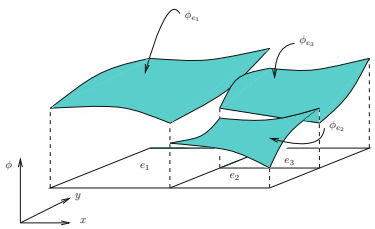


Fig. 1. Artists impression of the DGM interpolation space

As for any Galerkin method, the expansion weights W_{im} are found by requiring that the residual of the model equations, evaluated with w, is orthogonal to any function $v \in \Phi$; as the ϕ_i form a basis for Φ, it is sufficient to check this orthogonality for each of the ϕ_i and each of the components m. A naive application of the Galerkin principle not only leads to the typical volume contributions, but additional interface terms. To stabilise the discretisation, these interface contributions are replaced by consistent Dirichlet-like terms that enforce solution continuity across the elements. For the convective contributions, this is done using an *approximate Riemann solver (ARS)* (here the Roe ARS is used), whilst the diffusive terms are discretised according to the *Symmetric Interior Penalty (SIP)* method [6], which can be derived from an application of penalty Dirichlet contributions:

$$\begin{aligned}\forall \phi_i \in \Phi \ , \ \forall m : & \underbrace{\sum_e \int_e \phi_i \frac{\partial w_m}{\partial t} dV}_{GI} - \underbrace{\sum_e \int_e \frac{\partial \phi_i}{\partial x^k} (\mathcal{F}^k_m + \mathcal{D}^k_m) \ dV}_{GV} \\ & + \underbrace{\sum_f \int_f [[\phi_i]]^k \ n^k \ \mathcal{H}_m \left(w^-, w^+, \mathbf{n}\right) dS}_{CI} + \underbrace{\sum_f \int_f [[\phi_i]]^k \left\{ D^{kl}_{mn} \ \frac{\partial w_n}{\partial x^l} \right\} dS}_{DI} \\ & + \underbrace{\sum_f \int_f [[w_n]]^k \left\{ D^{kl}_{nm} \cdot \frac{\partial \phi_i}{\partial x^l} \right\} dS}_{DS} + \underbrace{\sum_f \sigma \int_f [[w_m]]^k \ [[\phi_i]]^k \ dS}_{DP} = 0\end{aligned} \tag{4}$$

Using $+$ and $-$ to indicate lower and upper values with respect to the face normal $\mathbf{n}$, we define $\mathbf{n}^- = \mathbf{n}$ and $\mathbf{n}^+ = -\mathbf{n}$. The interface average $\{.\}$ and jump $[[.]]$ operators are then defined as

$$\{a\} = \frac{(a^+ + a^-)}{2} \ , \ \ [[a]] = a^- \mathbf{n}^- + a^+ \mathbf{n}^+ \ . \tag{5}$$

For the convective terms, the Riemann solver $\mathcal{H}$ assures the correct flux of characteristics sent to, and received from, neighbouring cells, thus assuring correctly posed elementwise problems and global energy stability. For the diffusive term, the method generalises a boundary penalty method to enforce weak coupling at the interfaces. The penalty parameter σ must be chosen large enough to guarantee stability, yet not too large. Sharp bounds for the value of σ have been elaborated for simplices [32], and recently for hybrid meshes [22].

All Dirichlet boundary conditions, both for the convective and diffusive terms, are imposed weakly by providing a fictitious exterior state for the convective interface flux terms **CI**, the consistent diffusive flux **DI**, the symmetrising term **DS** and the penalty term **DP**. Neumann boundary conditions for the diffusive terms are enforced by modifying the consistent term **DI** and symmetrising term **DS** on the concerned faces.

By choosing base functions ϕ_i that are supported on a single element, one clearly identifies that this method boils down to a straightforward Galerkin

discretisation of the problem, consisting of typical volume inertial **GI** and flux conservation contributions **GV**, complemented with weak Dirichlet conditions for the convective **CI** and diffusive terms **DI**, **DS** and **DP** on the element boundary.

Both explicit and implicit temporal discretisations are available in Argo. The explicit time-stepping can be achieved using several methods such as standard Runge-Kutta schemes. The implicit method is performed using a second-order accurate three-point backward difference scheme (BDF2) [7] which, for a constant time step, is written as:

$$\frac{df}{dt} \widehat{=} \frac{3f(t) - 4f(t - \Delta t) + f(t - 2\Delta t)}{2\Delta t}. \tag{6}$$

At each time step, the non-linear problem resulting from the implicit integration is solved through a Newton/GMRES method, preconditioned with elementwise block-Jacobi, as described in [22]. This strategy allows for a very efficient parallellisation, as described in the following section.

3 Large Scale Parallellisation

3.1 Parallellisation Strategy

Fig. 2 shows the elements at the interface (blue) between two partitions and the quadrature points that contribute to the volume terms **GI** and **GV** (red), as well as those used for the interface terms **CI**, **DI**, **DT** and **DP** (green). As a consequence of the elementwise independent interpolation, one sees that the parallelism of the volume terms **GI** and **GV** in Eq. (4) is straightforward as it does not require any communication between elements. Indeed, all the quadrature points involved in the computation of the volume term are located within the element, and therefore in the same partition. On the other hand, the interface

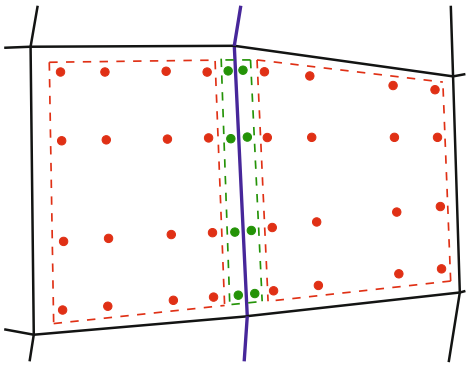


Fig. 2. Sketch of the volume term (red) and the interface term (green) with their interaction with the quadrature points. The interface between the two partitions is drawn in blue.

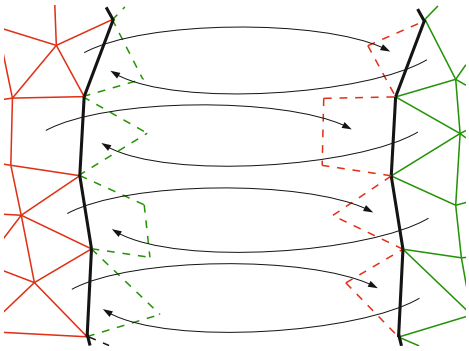


Fig. 3. Point to point communication in the ghost cell approach. For each cell next to the partition boundary (solid black line), a virtual cell (dashed) is provided which copies solution from the cell on the other partition.

contributions need the solution of the neighbouring cells, and therefore require communication between the MPI processes. Practically, this is implemented using ghost cells as illustrated shown in Fig. 3 Virtual neighbour cells are created for interfaces on the partition boundary, in which the solution is copied from the real cell on the other process using peer-to-peer communications. In this way the interface terms at the partition boundary can be computed in exactly the same way as at the interfaces internal to the domain. As the interface term involves the solution and its spatial derivatives, all the interpolation points of the neighbouring element are communicated to the ghost cell. Indeed, all the interpolation points of the element are required to collocate the solution and its gradients in the quadrature points of the interface.

Since the volume as well as the boundary term do not require the ghost cells, non-blocking communications can be used to hide the point-to-point communication behind their computation, as proposed by Altmann *et al.* [4]. Thereto send and reception operations for the ghost cells are initiated ahead of the evaluation of volume and boundary terms, and finalised only just ahead the interface term evaluation. This processus is illustrated in Fig. 4.

3.2 Scaling Tests

The weak and strong scalability of the code has been assessed on some of the largest HPC infrastructures in Europe available at the time, namely the BlueGene P machine JUGENE and its successor, the BlueGene/Q machine JUQUEEN, at the Jülich Supercomputing Centre (JSC). *Weak scalability* characterises the capacity of the code to tackle problems of growing size, typically proportional to the number of processors. *Strong scalability* concerns the capability of a code of performing the same computation faster by distributing it on growing numbers of processors.

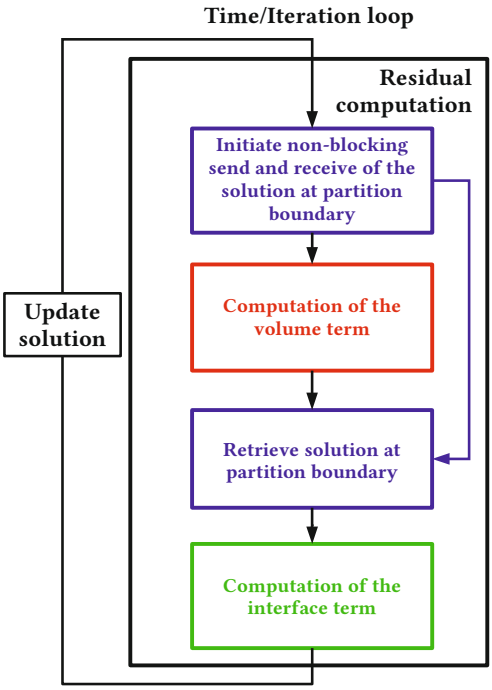


Fig. 4. Schematic of the computation of the residual, including non-blocking communications

All the scaling tests have been performed using the fourth order version of the solver with the time-implicit discretisation, as this is the standard setup of Argo for LES of wall-bounded flow. The time-implicit method is a second order backward differencing scheme, solved with a Newton method. The linear system is solved using preconditioned GMRES. As GMRES requires a large number of global communications, this iterative strategy is harder to parallellise than an explicit strategy, where only point-to-point communications are required. A second issue is that the domain decomposition method may degrade the iterative convergence, due to its impact on the preconditioner efficiency. Therefore a block-Jacobi preconditioner is used [22]. Even if the block-Jacobi features low convergence rates than block-ILU for instance, it is deemed the best candidate for massive parallel computations. Indeed, since it does not take into account inter-element dependencies, its iterative convergence is not impacted by parallellisation.

Weak Scaling. The weak scaling was assessed on JUGENE, the former Blue Gene/P machine at the Jülich Supercomputing Centre (JSC), during the PRACE

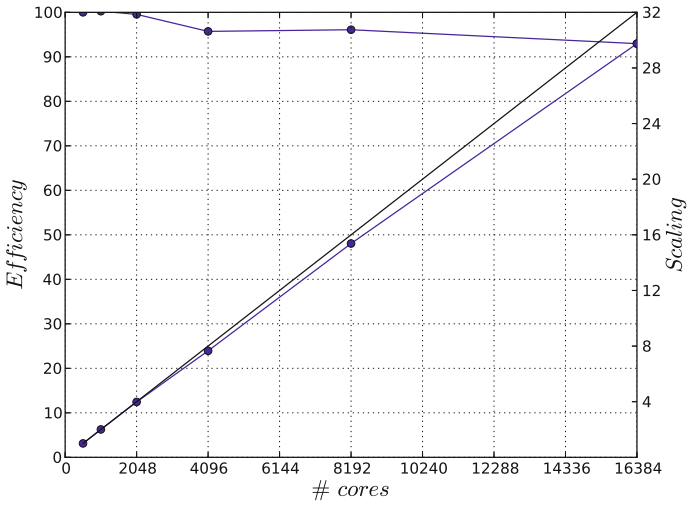


Fig. 5. Weak MPI scaling obtained on JUGENE. Efficiency and scaling when using from 512 to 16384 cores (*i.e.* 128 to 4096 nodes).

project noFUDGE. The BG/P architecture is specifically designed for very large scale scientific computing computations using tens of thousands of cores. It consists of a very large number of low memory, relatively slow processors in combination with a very efficient 3D toroidal interconnect, with an auxiliary network for global communications. This configuration was designed to give a favorable ratio between computational speed and communication efficiency. JUGENE was composed of 73728 nodes, each node containing one PowerPC 450 CPU@850MHz with 4-way SMP (294912 cores in total).

Fig. 5 shows the results of the weak scaling test performed on JUGENE. In this test, the size of the problem is increased simultaneoously with the number of processors. It gives a measure on how efficiently HPC infrastructure is used when computing larger and larger problems. The test ranges from 128 to 4096 nodes or alternatively from 512 to 16384 cores, using the full amount of the 2Gb of memory available per nodes. A hybrid parallellisation strategy is used, in which the work on a single node is distributed, using the OpenMP paradigm, over the 4 cores which access the same shared memory. In between nodes, the MPI paradigm is used to provide communication between the nodes. The efficiency remains above 92% even when using more than $15k$ cores.

Strong Scaling. Later on, using a further optimised version of the code, strong scaling tests were undertaken on the BlueGene/Q machine JUQUEEN at JSC. Again a hybrid parallellisation strategy is used. JUQUEEN is composed of 28672 nodes, each nodes containing a 16-cores IBM PowerPC A2@1.6GHz (458752

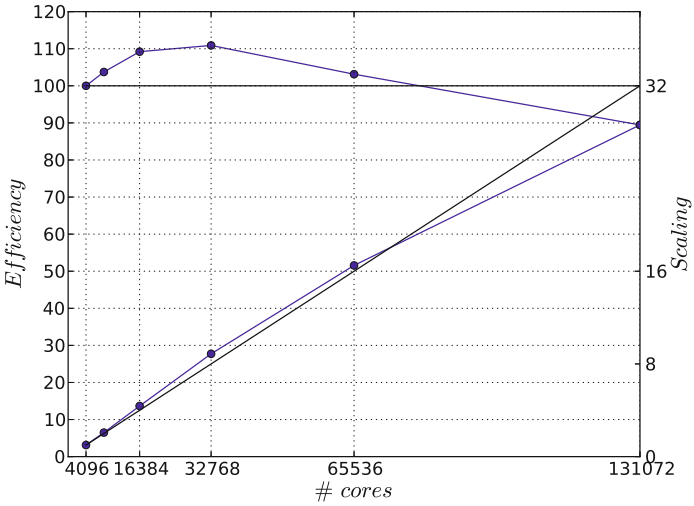


Fig. 6. Strong MPI scaling obtained on JUQUEEN when using the best configuration (*i.e.* 2 MPI processes ans 16 OpenMP threads per node). Efficiency and scaling when using from 4096 to 131072 cores (256 to 8192 nodes).

cores in total). Each core is capable of supporting 4 virtual threads, such that in theory up to 64 threads can be run efficiently on a single node. The communication channels follow a 5D toroidal configuration, with again a dedicated network for small size global communications.

On this machine, the choice of the hybrid parallellisation paradigm was not immediately obvious, and therefore the efficiency was compared between different configurations. Finally 2 MPI processes, each combining 16 shared memory threads, are used per node. This configuration, although it does not use all of the threads available, was found to provide the best computational efficiency. The scaling tests were performed during the PRACE project PADDLES. The parallel efficiency is presented in Fig. 6. An efficiency of slightly less than 90% is obtained when using more than $130k$ cores (with the smallest number of partitions using the full amount of the 16Gb of memory available per node). For the largest number of partitions, each partition contains only about hundred elements. A peak of efficiency (around 110%) is obtained around $32k$ cores, probably due to cache effects resulting from the continuous decrease in memory requirements. At larger core counts, this effect is compensated by the growing cost of communication with respect to computation. These results shows that the restitution time of the solver can be decreased by a factor of 32 on this architecture, while keeping a high efficiency.

4 Validation on Academic Benchmarks

The main goal of this section is to assess a *Computational Fluid Dynamics (CFD)* code Argo, based on the discontinuous Galerkin method with symmetric interior penalty (DGM/SIP), for scale-resolving simulations of turbulent flows. Academic test cases are considered, allowing to compare the accuracy of the method with respect to state-of-the-art academic solvers.

The first test case concerns the DNS of the Taylor-Green vortex, representative for free stream transition and turbulence. Its simplicity allows to get rid of uncertainties on the setup, such as boundary conditions or initial condition, and to focus the study on the effects of the numerical schemes only. The Taylor-Green vortex has been investigated at the first and the second International Workshops on High-Order CFD Methods [2,3,39], allowing to many participants to compare their results and assess the convergence rate of their solvers. Results by Cenaero are further detailed in [14].

A particular aspect is the use of an *implicit LES (ILES)* strategy, where the inherent dissipation of the method is used to provide the destruction of the smallest scale structures, which would give rise to the creation of sub resolution structures. The approach is simple, since no tuning is required, and provides by definition a smooth transition between DNS and LES resolution. The results obtained on the Taylor-Green vortex provide an indication for the viability of the ILES approach, since in underresolved computations, the numerical dissipation is observed to compensate the deficit in resolved dissipation.

A second section looks into a more rigourous assessment of the ILES approach on wall-bounded flows. It concerns the DNS and implicit LES of the well known plane channel flow benchmark at different Reynolds numbers. The mesh for the lowest Reynolds number case is designed with DNS criterion while for the higher Reynolds number, LES criterion are chosen to assess the *Implicit LES (ILES)* capabilities of the method.

4.1 Taylor-Green Vortex

This section presents the results obtained on the DNS of the Taylor-Green vortex at $Re = 1600$. A detailed study is available in Carton de Wiart *et al.* [14].

The Taylor-Green vortex is a three-dimensional periodic and transitional canonical flow defined by an analytic initial condition, given by

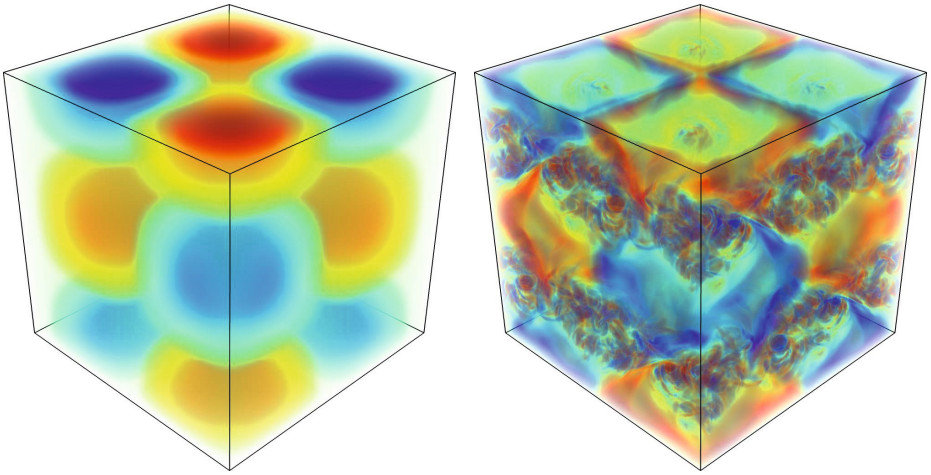
$$u = \quad V_0 \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right) \ , \tag{7}$$

$$v = -V_0 \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right) \ , \tag{8}$$

$$w = 0 \ , \tag{9}$$

$$p = p_0 + \frac{\rho_0 V_0^2}{16}\left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right)\right)\left(\cos\left(\frac{2z}{L}\right) + 2\right) \ . \tag{10}$$

The characteristic convective time is defined as $t_c = \frac{L}{V_0}$. This flow features transition to anisotropic small-scale turbulence and subsequent decay, as il-



(a) $t/t_c = 0$ (b) $t/t_c = 10$

Fig. 7. Taylor-Green vortex: volume rendering of the z-component of the vorticity.

lustrated in Fig. 7. The flow is computed within a periodic square box defined as $-\pi L \leq x, y, z \leq \pi L$. The Reynolds number of the flow is defined as $Re = \frac{\rho_0 V_0 L}{\mu}$ and is equal to 1600. The physical duration of the computation is set to $t_{final} = 10\, t_c$.

In the simulations using the compressible DGM code, the fluid is considered to be a perfect gas with $\gamma = c_p/c_v = 1.4$ and Prandtl number $Pr = \frac{\mu\, c_p}{\lambda} = 0.71$, where c_p and c_v are the heat capacities at constant pressure and volume respectively, μ is the dynamic shear viscosity and λ is the heat conductivity. It is furthermore assumed that the gas has zero bulk viscosity ($\mu_v = 0$). The Mach number is set to $M_0 = \frac{V_0}{c_0} = 0.10$ (with c_0 the speed of sound corresponding to the temperature $T_0 = \frac{p_0}{R\,\rho_0}$), which is indeed small enough that the solutions obtained for the velocity and pressure fields are very close to those obtained assuming an incompressible flow. The initial temperature field is taken uniform, $T = T_0$, and hence the initial density field is given by $\rho = \frac{p}{R\,T_0}$.

A grid convergence study is performed on four grids, and using third order polynomial interpolants ($p = 3$), resulting in a fourth order of accuracy. The number of cells of each computation, the number of numerical *degrees of freedom (dof)*, as well as the number of cell subdivisions as defined by the high order control nodes (here denoted as *effective resolution*), are given in Tab. 1. In order to avoid confusion when comparing to other discretisations, the effective resolution will be used throughout the study instead of the number of dof or the number of elements. Indeed, the smallest wavelength that can be captured depends directly on the number of potential zero crossings in an element. This number is equal to the polynomial order p, such that this wavelength is $2h/p$, whereby h is the mesh size. Therefore the latter measure of *effective resolution* is more appropriate than the number of *dof*. The time integration is based on a

Table 1. Taylor-Green vortex: specifications for grid convergence study at $p = 3$

Elements	dof	Effective resolution
48^3	192^3	144^3
64^3	256^3	192^3
96^3	384^3	288^3
128^3	512^3	384^3

standard explicit 4-stage Runge-Kutta integrator, whereas the time step corresponds to at most a unitary acoustic CFL number. In terms of convective speed, this corresponds to a maximum CFL number of 0.1, which results in negligible temporal error.

The evolution of the global kinetic energy is presented in Fig. 8. As indicated before, it is difficult to distinguish the computations on this criterion as differences are minimal. This first result shows that, despite the presence of dissipation in the numerical scheme, DGM preserves the kinetic energy rather well, even at the coarsest resolution.

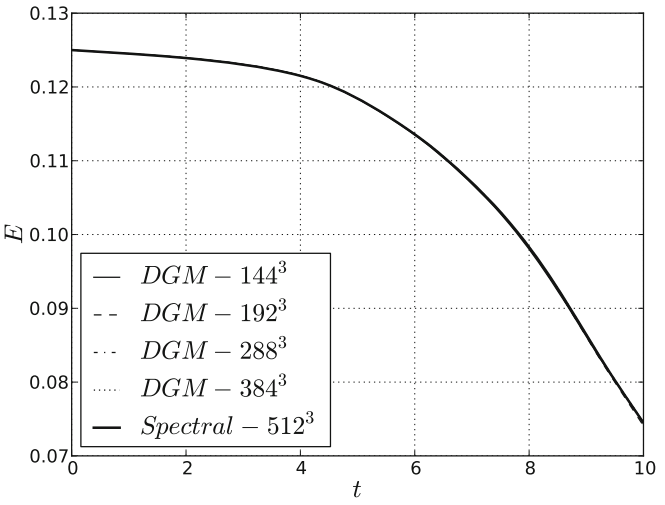
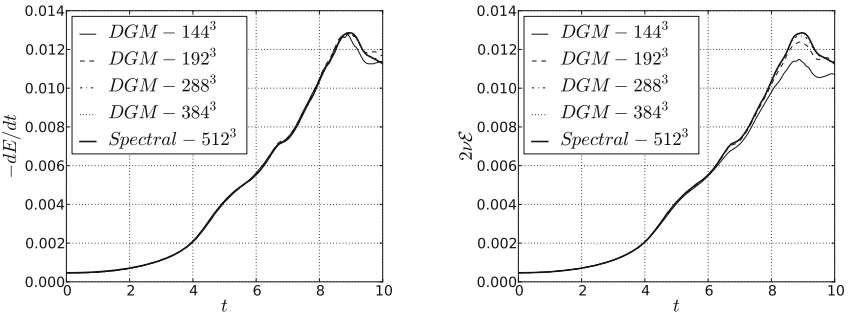


Fig. 8. Taylor-Green vortex: convergence of the temporal evolution of the kinetic energy



(a) Measured (b) Resolved

Fig. 9. Taylor-Green vortex: convergence of the temporal evolution of the dissipation rate computed using the time derivative of the energy (measured) and the integral of the enstrophy (resolved). See also zoom in Fig. 10.

The differences are much more pronounced when the dissipation rate is considered. For incompressible flow the theoretical dissipation rate is proportional to the enstrophy $\mathcal{E}$

$$-\frac{dE_k}{dt} = 2\nu\mathcal{E} \tag{11}$$

with the enstrophy the energy associated to the vorticity ω

$$\mathcal{E} = \frac{1}{2}\int_V \omega \cdot \omega dV \tag{12}$$

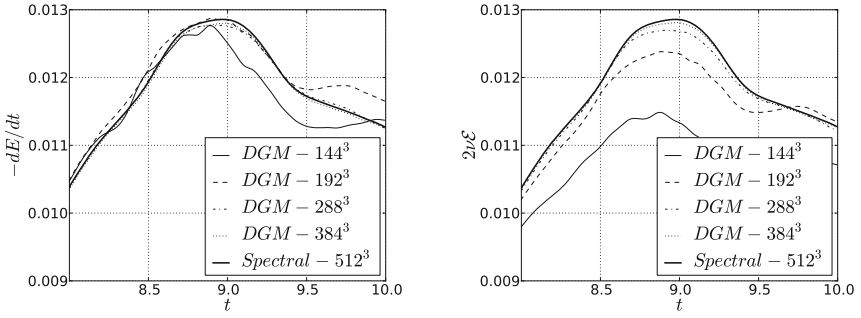
Figs. 9 and 10 (zoom) show that the *measured* ($-\frac{dE_k}{dt}$) and *resolved* or *theoretical* dissipation rate based on enstrophy are not numerically equivalent. Typically, the measured dissipation, which include numerical dissipation, is consistently larger than the resolved dissipation. Although DGM is dissipative, this is not obvious since dispersion errors will impact the *temporal evolution* of the dissipation. It is clear that the measured dissipation rate, comprising both resolved and numerical dissipation, converges faster to the reference results. Therefore, for this case and at resolutions near DNS, the numerical dissipation could be considered to act as a relatively accurate LES subgrid scale model, thereby indicating a potential applicability of DGM for *implicit LES*.

The numerical dissipation of DGM is triggered by the jumps of the solution across element interfaces when the resolution is insufficient to capture all of the features of the flow. Its precise value depends obviously on the Riemann solver in the convective interface flux CI and the penalty terms DT and DP. However this is only for a fixed value of the jump, such that their precise impact is not easy to predict. Further studies turbulence have indicated that neither may have a significant impact for isotropic turbulence or channel flow [13].

Fig. 11 shows the resulting spectrum of each computation. In these figures, the vertical lines show the respective cutoff wave number associated with the element

size and the effective resolution. Each DGM computation captures correctly the spectrum below the cut-off wave number corresponding to the element size and progressively dissipates the energy when going towards the cut-off wave number corresponding to the high-order nodes.

Both prismatic and tetrahedral meshes are also considered. They have the same characteristic mesh size as the hexahedral mesh with 96^3 elements. The prismatic mesh is created by extrusion of an unstructured triangular mesh on one of the faces. The tetrahedral mesh is created directly from the 96^3 mesh by a regular splitting of the hexahedral elements; this approach is indeed often



(a) Measured (b) Resolved

Fig. 10. Taylor-Green vortex: convergence of the dissipation rate computed using the time derivative of the energy (measured) and the integral of the enstrophy (resolved); zoom at the peak.

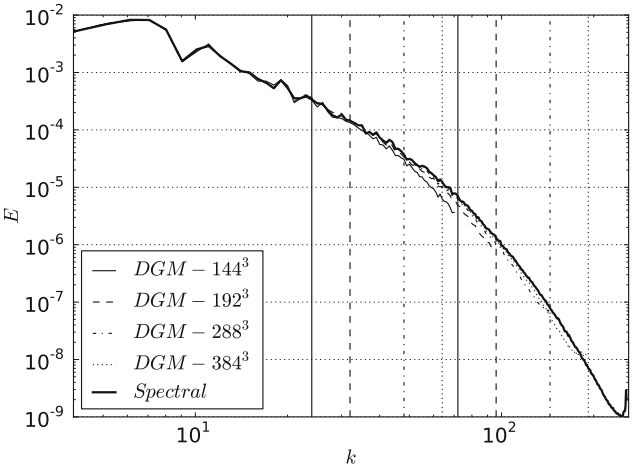


Fig. 11. Taylor-Green vortex: convergence of the energy spectrum at $t = 9$. The vertical lines indicate the cut-off wave numbers corresponding to the *mesh* (h) and the *effective* resolution (h/p) respectively.

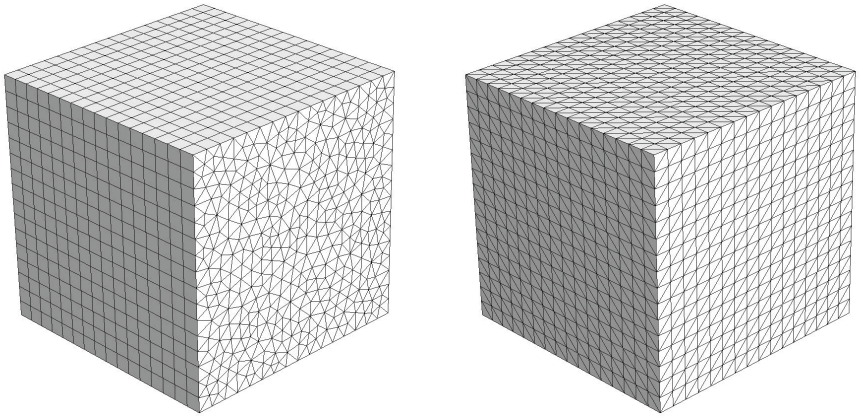


Fig. 12. Taylor-Green vortex: prismatic mesh (left) and tetrahedral mesh (right). To enable visualisation, a 6^3 times coarser resolution is shown.

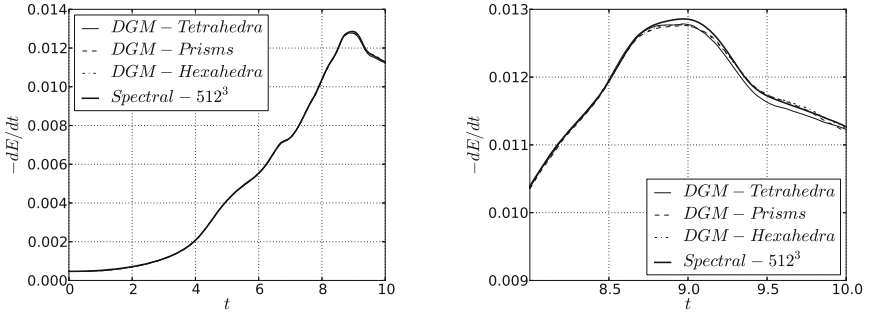
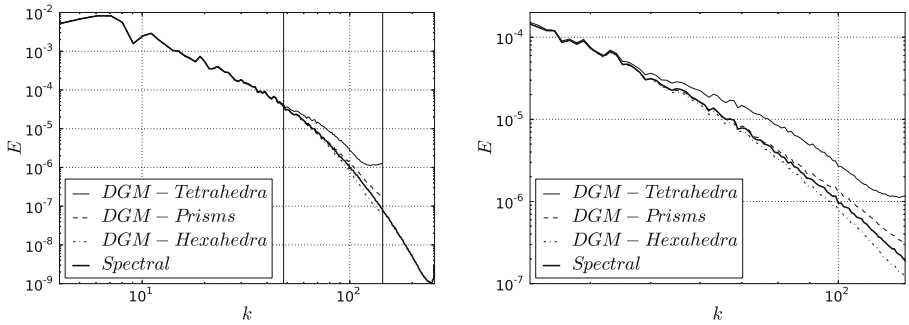


Fig. 13. Taylor-Green vortex: temporal evolution of the measured dissipation rate. Comparison of DGM using tetrahedral, prismatic and hexahedral meshes to the pseudo-spectral reference.

used by many large scale industrial meshers for regions that are far away from boundaries. A fully unstructured tetrahedral mesh has not been considered due to limitation on the stable time step, resulting from the presence of a small number of very small elements. Fig. 12 shows the grid configurations. The prismatic mesh is composed of $2.34M$ prisms and the tetrahedral mesh of $5.31M$ tetrahedra (corresponding to six tetrahedra per hexahedron). Fig. 13 compare the computed dissipation rate to the reference results. All the computations are very close to the pseudo-spectral reference: both prismatic and tetrahedral meshes gives globally the same results as the hexahedral grid. The tetrahedral mesh appears to be slightly less accurate than the hexahedral grid, whilst the prismatic element mesh provides similar accuracy.



(a) Full spectrum (b) Zoom near cut-off wave numbers

Fig. 14. Taylor-Green vortex: energy spectrum at $t = 9$. Comparison between tetrahedral, prismatic and hexahedral meshes. The vertical lines indicate the cut-off wave numbers corresponding to the *mesh* (h) and the *effective* resolution (h/p) respectively.

The kinetic energy spectrum is shown in Fig. 14. For wave numbers up to the cut-off wave number corresponding to the grid size, the spectrum is perfectly captured by all computations. However, kinetic energy appears to be accumulated at higher wave numbers in the case of the tetrahedral mesh. The spectrum obtained with prismatic elements is between that obtained with hexahedra and tetrahedra. The dissipation of the method in the large wavenumbers is thus highly dependent of the element type. This is also observed on more theoretical studies, see for instance the analysis of DGM for wave propagation performed by Hu *et al.* [24]. Hu shows that meshes composed of unstructured triangles can engender less anisotropy compared to regular mesh composed of quadrangles. The same analysis should be extended to three-dimensional meshes to measure the true level of dissipation of each element type for complex turbulent flows without predefined direction.

4.2 Periodic Channel Flow

This section summarises the results presented in Carton de Wiart *et al.* [13]. The wall-resolved LES of the fully developed turbulent flow between two parallel walls separated by a distance 2δ in the y-direction is investigated. The flow is assumed periodic in the streamwise (x) and the spanwise (z) direction. Four Reynolds numbers are considered: $Re_\tau = u_\tau \delta/\nu = 180$, 395, 590 and 950, with u_τ the friction velocity based on the wall shear stress $u_\tau = \sqrt{\tau_w/\rho}$. The friction Reynolds number is imposed through a constant forcing in the x-momentum equation. This forcing is given by the pressure gradient $\frac{dp}{dx} = -\frac{\tau_w}{\delta}$. Here again, a Mach number of 0.1 (based on the estimated bulk velocity) is chosen to approximate the incompressible conditions of the reference DNS of Moser *et al.* [30] ($Re_\tau = 180,\ 395,\ 590$) and Hoyas *et al.* [23] ($Re_\tau = 950$). The fourth order version of the method is used ($p = 3$).

Table 2. LES of the channel flow. Parameters of the simulations.

Re_τ	Grid size $nx \times ny \times nz$	Effective resolution $nx \times ny \times nz$	$\Delta x^+ \times \Delta y^+ \times \Delta z^+$	Δt^+
180	$42 \times 42 \times 42$	$126 \times 126 \times 126$	$17.95 \times 0.17 \times 8.96$	1×10^{-3}
395	$21 \times 16 \times 16$	$63 \times 48 \times 48$	$38.78 \times 2.45 \times 25.85$	2×10^{-3}
590	$32 \times 20 \times 32$	$96 \times 60 \times 96$	$38.61 \times 2.36 \times 19.3$	1×10^{-3}
950	$64 \times 32 \times 64$	$192 \times 96 \times 192$	$31.01 \times 1.52 \times 15.5$	5×10^{-4}

The grid and time step parameters are summarised in Table 2, with $\Delta t^+ = u_\tau \Delta t/\delta$ the dimensionless time step. On the $Re_\tau = 180$ case, a very high resolution is chosen, such that DNS can be assumed. At this Reynolds number, the level of turbulence is assumed too low to perform a meaningful LES. The computational domain is the same for the LES (from $Re_\tau = 395$): $2\pi\delta \times 2\delta \times \pi\delta$. According to Moser *et al.* [30], a larger domain is chosen for the DNS ($4\pi\delta \times 2\delta \times 4/3\pi\delta$) to capture the low frequency phenomena. The grid is stretched in the wall normal direction using the following cosine law transformation (for the semi-channel):

$$\frac{y(\xi)}{\delta} = \left(1 - \cos\left(\frac{\pi\xi}{2\delta}\right)\right), \quad \xi \in [0, 1].$$

The grid points are taken at regular interval $\Delta\xi$ in the transformed coordinates.

Figs. 15 and 16 present instantaneous volume renderings of the obtained velocity fields for all cases, showing the complex and intense turbulent structures near the wall and the effects of the Reynolds number. Fig. 16 highlights the elongated structures at the near wall, typical of the channel flow.

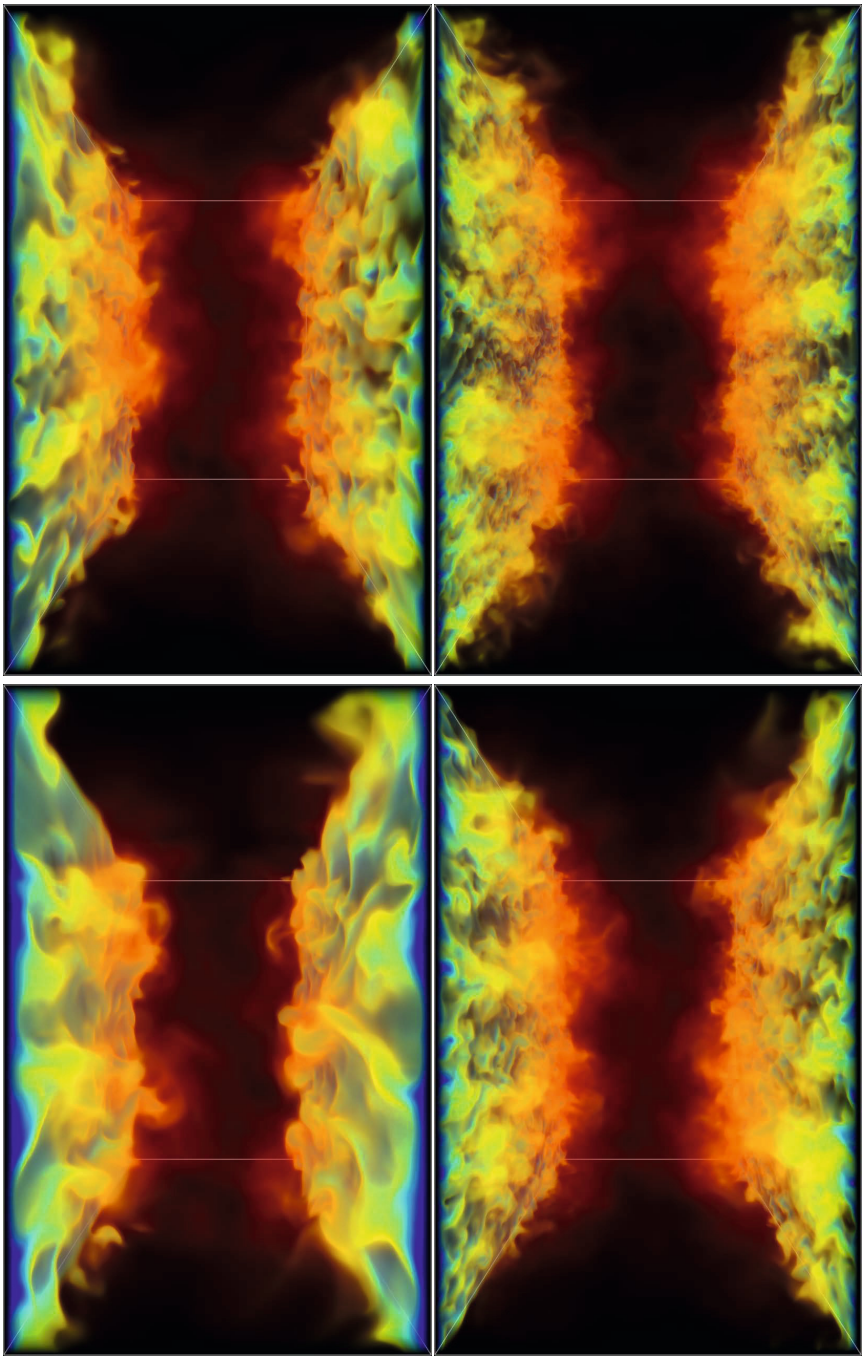


Fig. 15. Channel flow at $Re_\tau = 180$ (DNS), 395, 590 and 950 (LES). Volume rendering of the velocity field. Values from 75% of the velocity at the center of the channel are transparent. Only a portion of the domain is shown for the $Re_\tau = 180$ case.

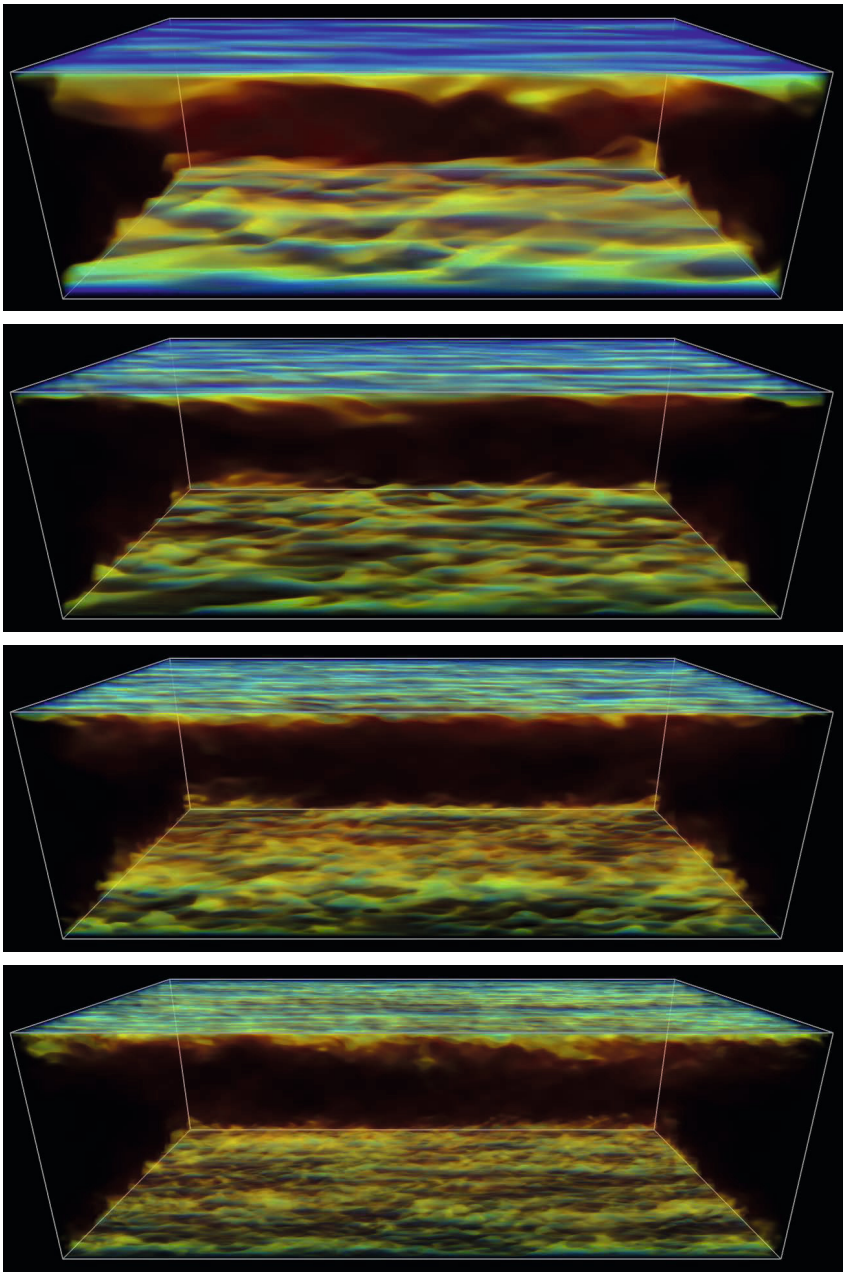


Fig. 16. Channel flow at $Re_\tau = 180$ (DNS), 395, 590 and 950 (LES). Volume rendering of the velocity field. Values from 75% of the velocity at the center of the channel are transparent. Only a portion of the domain is shown for the $Re_\tau = 180$ case.

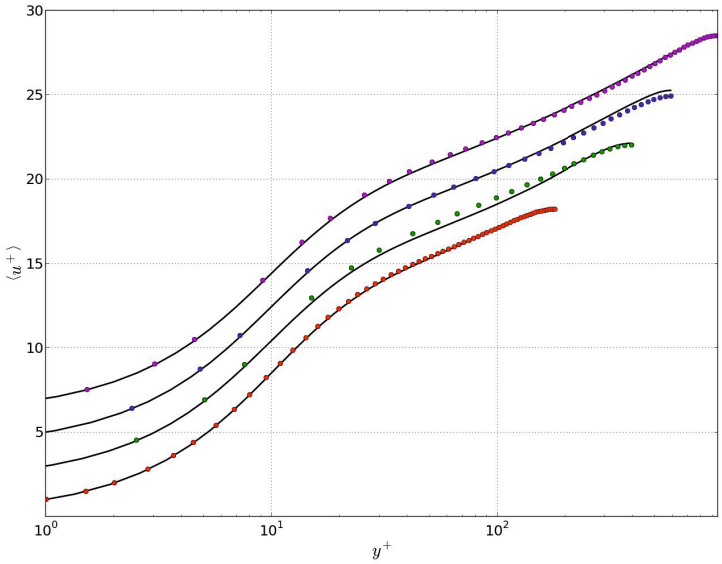


Fig. 17. LES of the turbulent channel flow at $Re_\tau = 180$, 395, 590 and 950. Mean velocity profiles $\langle u^+ \rangle$. DGM/ILES (circle) compared to DNS (solid line) of Moser *et al.* [30] ($Re_\tau = 180$, 395 and 590) and Hoyas *et al.* [23] ($Re_\tau = 950$). The curves were shifted vertically by $\langle \Delta u^+ \rangle = 2$ for clarity.

Fig. 17 shows the resulting time and spatially averaged profiles of the through-flow velocity $\langle u^+ \rangle = \langle u \rangle / u_\tau$. Each point corresponds to a high-order interpolation point, whereas the solution is averaged at the points located on element interfaces. The notation $\langle \cdot \rangle$ indicates values that have been averaged in time and in both the x and z homogeneous directions. The curve of the DNS performed at $Re_\tau = 180$ is perfectly superimposed with the DNS of Moser. Globally, the LES results are also seen to be in very good agreement with the DNS of Moser and Hoyas, especially for $Re_\tau = 950$. This might be due to the better resolution of this computation, relative to that required for the DNS of Moser and Hoyas, compared to those used for the lower Reynolds numbers. It is interesting to stress that the LES results are obtained on a relatively coarse mesh in the near wall region. Indeed, in these DGM computations, the first interpolation point (*i.e.* at a distance h/p away from the wall is located at a wall distance $y^+ > 1.5$ (and even 2.5 for $Re_\tau = 395$ and 590); whereas, for most other discretisations (FDM, FVM, PSM, etc.), one needs to stay below $y^+ = 1$ to obtain satisfactory results.

Fig. 18 shows the components of the averaged turbulence intensity profiles. According to [40], a fair comparison of the turbulent intensities between a DNS

and LES computations can only be based on the traceless values, computed as:

$$\left\langle u_i'^{+} \right\rangle_{rms}^{*} = \frac{\sqrt{\langle u_i' u_i' \rangle - \frac{1}{3} \sum_{k=1}^{3} \langle u_k' u_k' \rangle}}{u_\tau}.$$

The DNS at $Re_\tau = 180$ captures almost perfectly the fluctuations. A fair agreement is obtained for the LES, even if the fluctuations at the center of the channel are slightly underpredicted, in particular at higher Re_τ. The peak for $\langle u'^{+} \rangle_{rms}$ is also overpredicted for all Reynolds numbers. This behaviour is typical of LES of channel flow and is observed for state-of-the-art LES codes and models (see, for instance, Bricteux *et al.* [9]). This effect appears to be further reduced as the Reynolds number increases. This might be due to the slightly better spanwise resolution of the high Reynolds cases. Small oscillations can also be seen on the profiles. Those oscillations, also observed in [28], follow the mesh pattern, thereby reflecting the discontinuity of the DGM solution and the difference in the numerical treatment at the interfaces. Nevertheless, this phenomenon does not affect significantly the values.

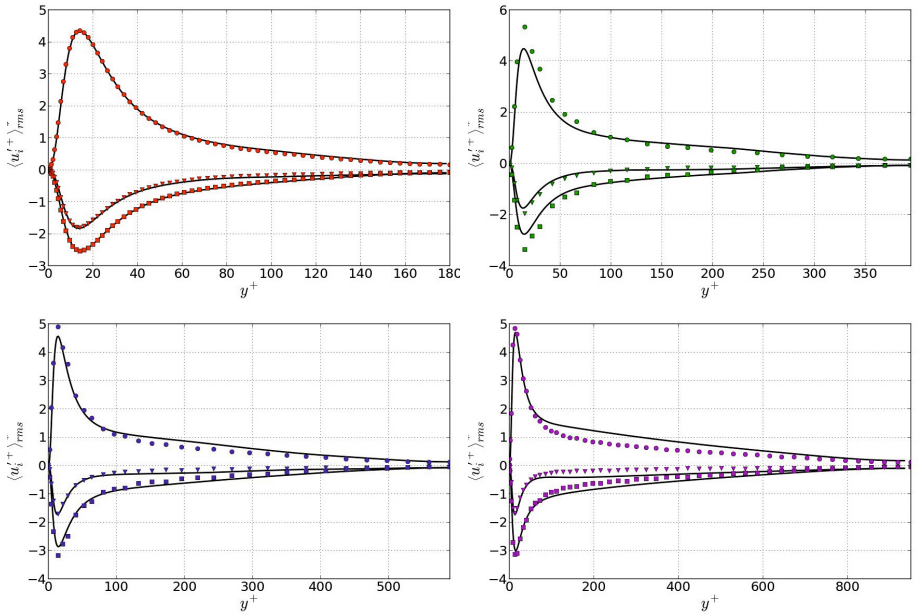


Fig. 18. LES of the turbulent channel flow at $Re_\tau = 180$ (upper left), 395 (upper right), 590 (lower left) and 950 (lower right). RMS turbulent velocities: $\left\langle u'^{+} \right\rangle_{rms}^{*}$ (circle), $\left\langle v'^{+} \right\rangle_{rms}^{*}$ (square) and $\left\langle w'^{+} \right\rangle_{rms}^{*}$ (triangle) of DGM/ILES compared to DNS (solid line) of Moser *et al.* ($Re_\tau = 180$, 395 and 590) and Hoyas *et al.* ($Re_\tau = 950$).

Finally, the shear component $-\langle u'v' \rangle / u_\tau^2$ of the Reynolds stress is presented in Fig. 19. Here also, the DNS at $Re_\tau = 180$ is in very good agreement with the

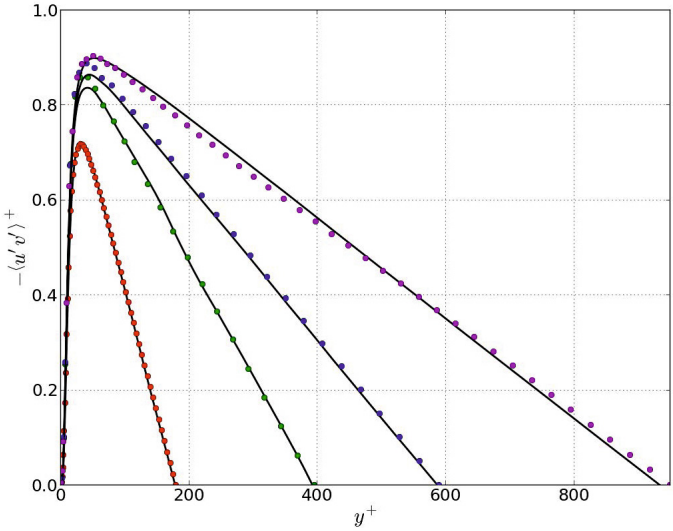


Fig. 19. LES of the turbulent channel flow at $Re_\tau = 395$, 590 and 950. Mean turbulent shear stress profiles $-\langle u'v' \rangle^+$: DGM/ILES (circle) compared to DNS (solid line) of Moser *et al.* [30] ($Re_\tau = 395$ and 590) and Hoyas *et al.* [23] ($Re_\tau = 950$).

reference. The LES results are also very close to those of the reference. Small overpredictions can be observed at $Re_\tau = 395$ and 590 around the maximal value. It can also be seen that the results of Hoyas *et al.* [23] have been obtained at a slightly lower Reynolds number of 935 rather than the 950 announced. Indeed, the curve intersects the zero value around $y^+ = 935$. The DGM/ILES result is correctly located at $y^+ = 950$. The authors were not aware of this small discrepancy at the time, and hence simulations was run at $Re_\tau = 950$ and not 935.

5 IDIHOM Test Cases

Finally, the method is applied to three IDIHOM test cases. The first is the periodic flow over a 2D hill from the ERCOFTAC QNET CFD data base [1]. Subsequently, two semi-industrial cases are tackled, namely the LES of the transitional flow in the T106C turbine cascade and the LES of the turbulent jet flow from the JEAN nozzle [26].

5.1 DNS and ILES of the Periodic Flow over a 2D Hill

The 2D periodic hill flow is a generic case of a flow separating from a smoothly curved surface. The flow over periodically arranged hills features separation at

the hill crest and reattachment downstream, creating a large recirculation bubble. The flow conditions are determined by the Reynolds number $Re_b = \frac{u_b h}{\nu}$ based upon the bulk velocity, evaluated at the crest

$$u_b = \frac{1}{2.035h} \int_h^{3.035h} u(y)dy \tag{13}$$

and the hill height h. The case has been studied numerically and experimentally, from $Re_b = 100$ to 37000, by numerous research groups. The major contributions have been performed experimentally in a water channel by Rapp *et al.* [31] and targeted Reynolds numbers from $Re_b = 5600$ to 37000. Numerical studies, including DNS and LES, have been performed by Breuer *et al.* [8], Manhart and Friedrich [29], Fröhlich *et al.* [18], amongst others. A detailed description together with numerical and experimental reference results are available on the ERCOFTAC QNET-CFD wiki [1]. It was also test case C3.6 of the second International Workshop on high-order CFD methods [12].

The computational domain is periodic in both the stream- and spanwise direction. The mass flow rate is ensured by the addition of a body force. This body force is calibrated continuously in order to provide the correct mass flow rate, using a procedure proposed by Benocci and Pinelli [10, 38], as mentioned in the test case description on the ERCOFTAC QNET-CFD wiki [1].

Two flow conditions are considered, corresponding to $Re_b = 2800$ and $Re_b = 10595$. The low Reynolds number case is computed using DNS mesh criteria and the high Reynolds number with LES mesh criteria. The Mach number is set to 0.1 to approximate incompressible conditions. The size of the domain is $9h \times 3.035h \times 4.5h$. Both Reynolds numbers are computed on a second order curvilinear mesh, composed of $128 \times 64 \times 64$ elements (see Fig. 20). Using fourth order accurate interpolants ($p = 3$), this leads to more than $33M$ *dof* per equation. For the $Re_b = 10595$ case, a coarser mesh is also considered to perform a more challenging ILES. It is composed of $64 \times 32 \times 32$ elements, leading to slightly more than $4M$ (effective resolution of $1.4M$).

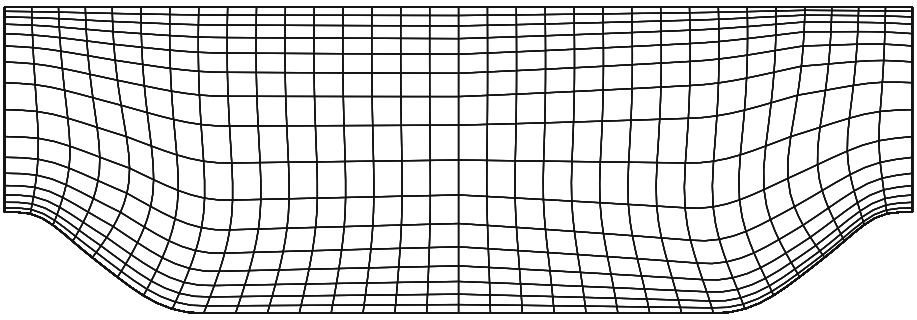


Fig. 20. Second order curved mesh for the 2D periodic hill. For clarity, the mesh shown is coarsened four times in each direction.

Fig. 21 shows an instantaneous volume rendering of the vorticity field for the $Re_b = 2800$ case at $t = 300t_c$, with t_c the convective time scale, here defined as $t_c = h/u_b$. It can be seen that the turbulence is fully established. Due to the recycling of the flow, the shear layer emanating from the hill crest is turbulent immediately, without a transitional region. Intense elongated structures are created in the shear layer emanating from the hill crest. Due to the presence of low frequency phenomena, the flow is averaged over a relatively long period (*i.e.* $150t_c$) and in the spanwise direction, such that fully converged statistics are obtained. The averaged results are compared to those obtained by Breuer *et al.* on a $48M$ *dof* DNS using the MGLET and the LESOCC finite volume solvers. The mean velocity profiles $\overline{u}/u_b$ and the turbulent intensity $\overline{u'v'}/u_b^2$ are presented in Fig. 22. Some very small discrepancies can be observed on the vertical velocity in the recirculation bubble but, globally, excellent agreement is found with the reference results. The length of the recirculation bubble can be approximately measured on the horizontal velocity profile, and is located just after $x/h = 5$. The differences are slightly more pronounced when velocity fluctuations are considered, in particular in the shear layer. Nevertheless, these remain very small and globally an excellent agreement is achieved, especially with MGLET.

Fig. 23 shows a snapshot of the volume rendering of the Q criterion for the $Re_b = 10595$ case. As expected, much smaller vortical structures are obtained, even if the flow is computed on the same mesh as the lower Reynolds number. Fig. 24 shows the mean velocity profiles $\overline{u}/u_b$ and the turbulent intensity $\overline{u'v'}/u_b^2$ obtained on the baseline and the coarse mesh. At this Reynolds number experimental results, obtained by Rapp *et al.*, are available. The results are also compared to those obtained by Breuer *et al.* using the LESOCC solver on a 13M *dof* grid. Globally, a good agreement is obtained between the computations and the experiment for the velocity profiles. The results obtained using Argo are even slightly better than those obtained using the LESOCC solver. Unfortunately, fully converged velocity fluctuations were not obtained for the baseline mesh at the end of the study. Indeed, the flow has only been averaged on $20t_c$. This short averaging period is responsible for the under-prediction of the fluctuations. On the other hand, the solution on the coarse mesh is averaged on more than $200t_c$. The results show a good agreement with the reference, considering this coarseness of the resolution.

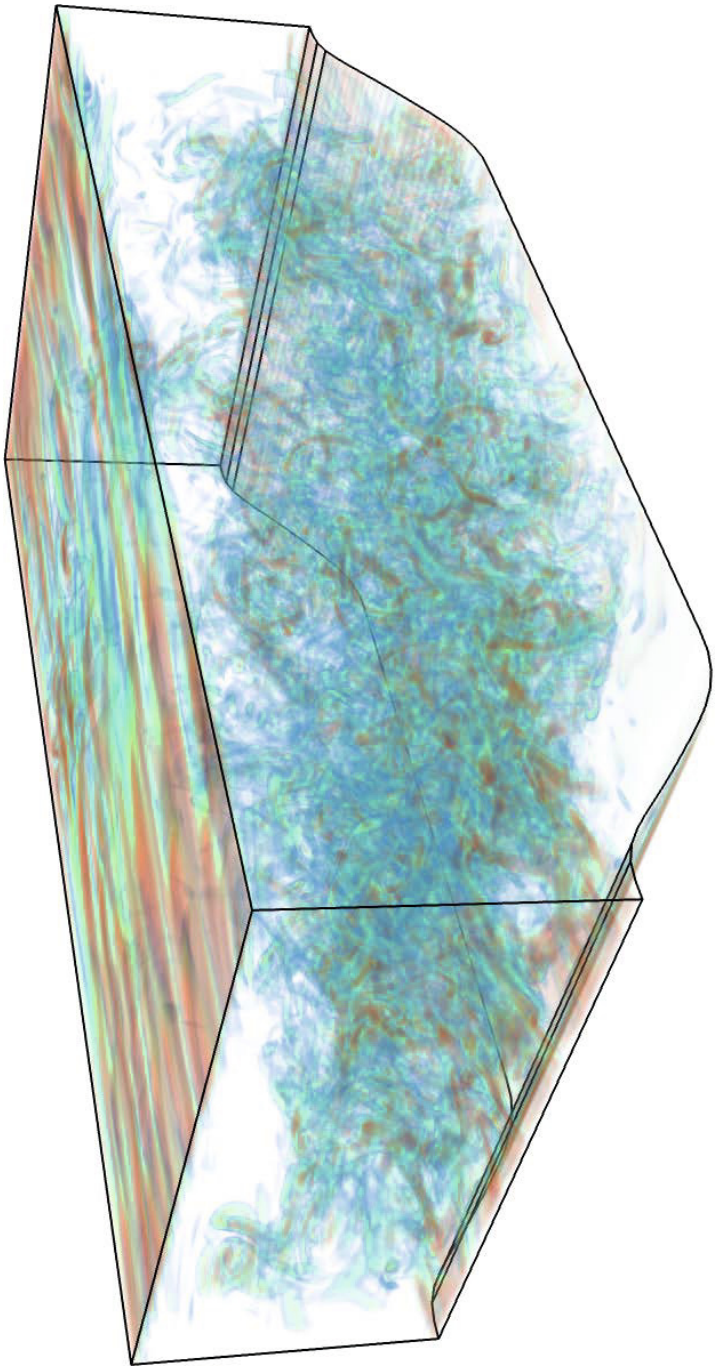


Fig. 21. DNS of the 2D periodic flow at $Re_b = 2800$. Volume rendering of an instantaneous vorticity field. Remark that due to the limitations of the 3D visualisation chain, minor spurious 2D effects appear on the curved walls.

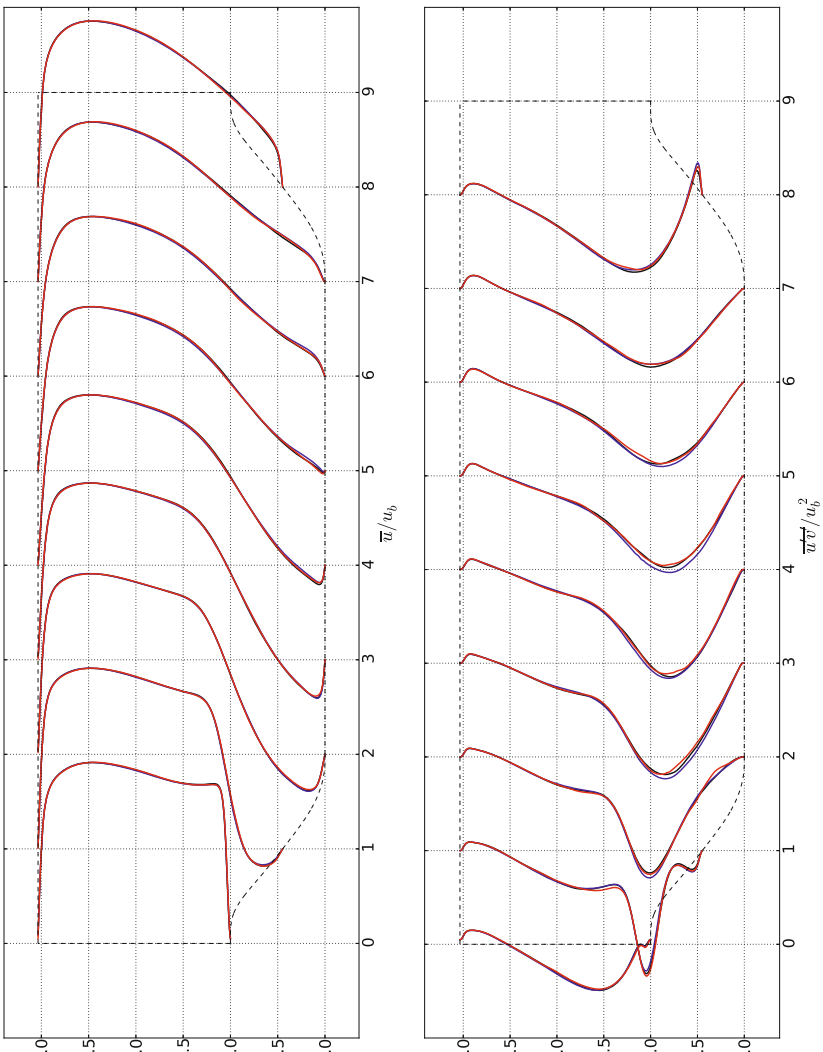


Fig. 22. DNS of the 2D periodic hill flow at $Re_b = 2800$. Averaged velocity $\overline{u}/u_b$ (top) and turbulent intensity $\overline{u'v'}/u_b^2$ (bottom). Red line: Argo (DGM). Black line: MGLETT (FVM). Blue line: LESOCC (FVM).

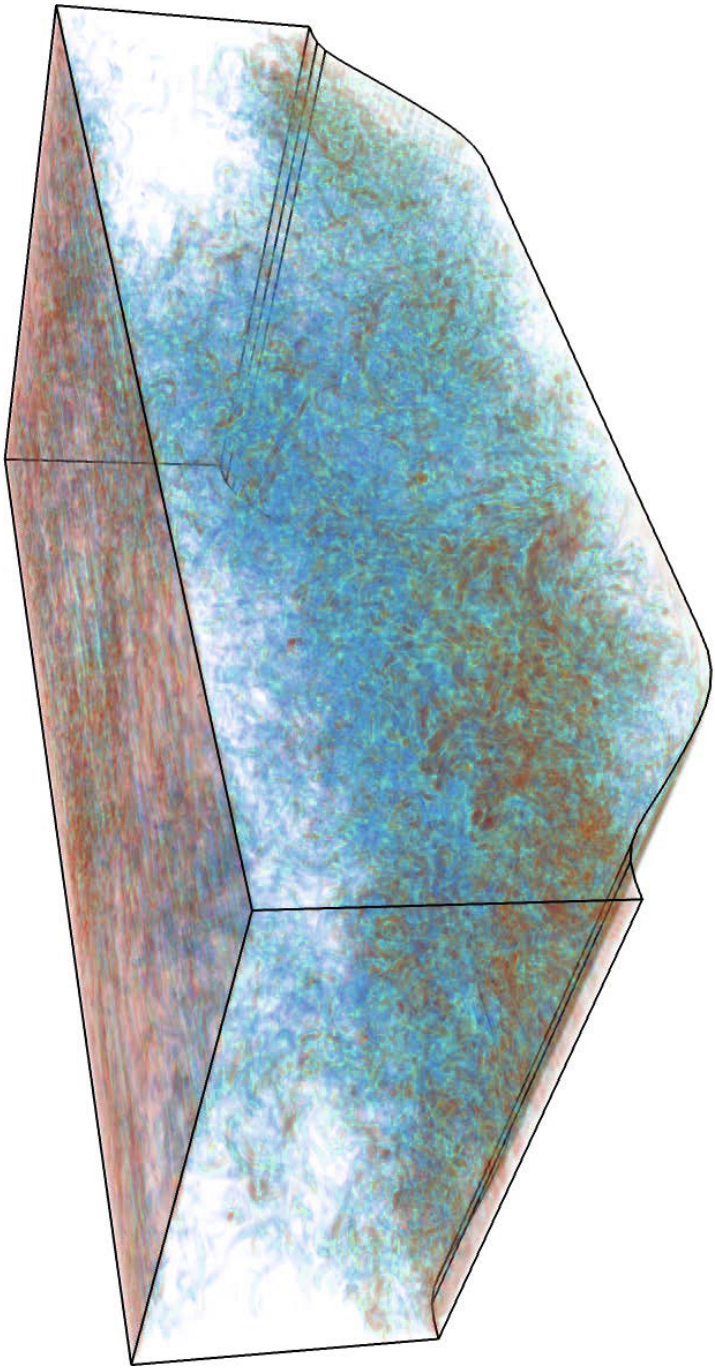


Fig. 23. ILES of the 2D periodic hill flow at $Re_b = 10595$. Volume rendering of the instantaneous Q criterion field. Remark that due to limitations of the visualisation chain, minor spurious 2D effects appear on the curved walls.

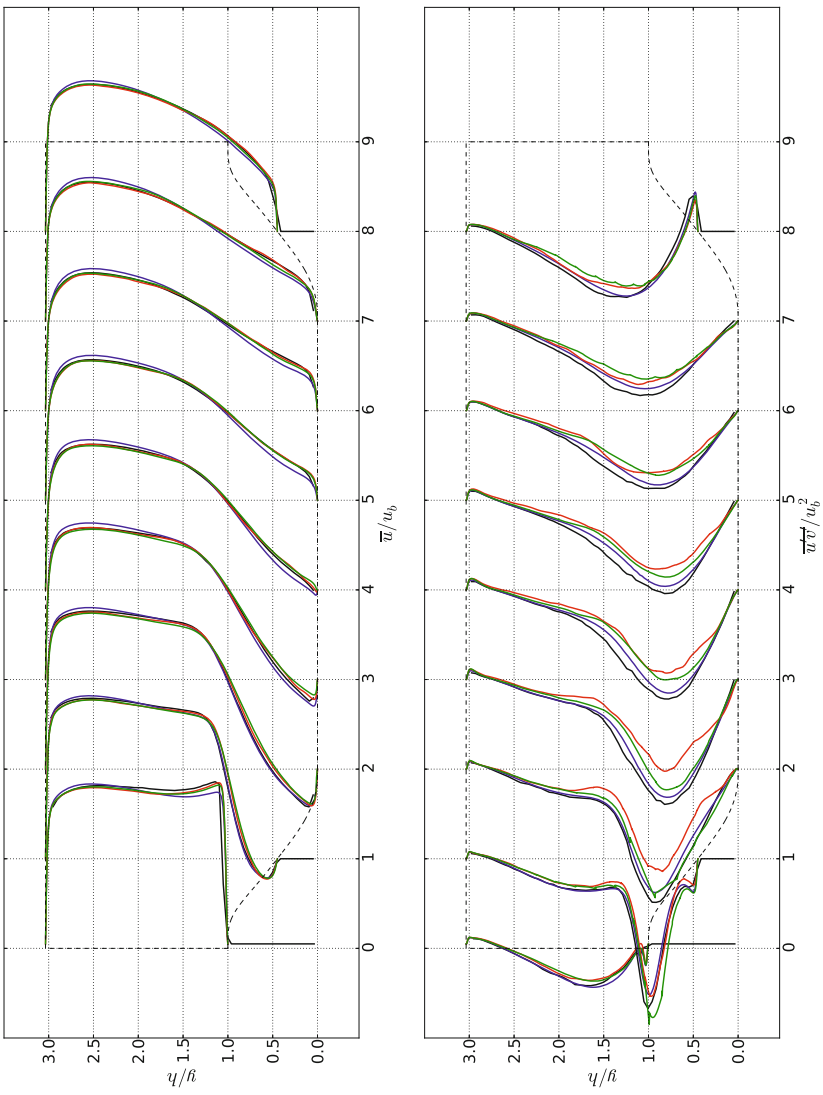


Fig. 24. ILES of the 2D periodic hill flow at $Re_b = 10595$. Averaged velocity $\overline{u}/u_b$ (top) and turbulent intensity $\overline{u'v'}/u_b^2$ (bottom). Red line: Argo (DGM, baseline mesh). Green line: Argo (DGM,coarse mesh). Black line: Experiment. Blue line: LESOCC (FVM).

5.2 ILES of the Transitional Flow in the T106C Cascade

This test case was included as C3.7 at the second International Workshop of High Order Methods in CFD [3]. Experimental data were provided by Prof. Tony Arts of the von Karman Institute for fluid dynamics. The results of Cenaero are the subject of a paper at the ASME Turbo Expo 2014 [21].

The flow conditions are defined with respect to the isentropic exit conditions, corresponding to the ratio of inlet total to exit static pressure, in particular $M_{2,is} = 0.59$ and $Re_{2,is} = 80.000$. These conditions lead to a laminar separation, followed by transition in the shear layer and reattachment zone. The boundary conditions impose total pressure and temperature, as well as flow angle at the inlet, while static pressure is imposed at the outlet. The flow is assumed to be periodic in both the pitchwise and spanwise directions. The spanwise extent is chosen to be 20 % of the axial chord C_{ax}. The working fluid is air, assumed to be a perfect gas with gas capacity ratio $\gamma = 1.4$, whereas dynamic viscosity and conductivity are considered constant. Finally, in view of the low level of turbulence in the wind tunnel, the inlet was assumed to be free of turbulence The physical values of the measurement conditions are available from the test case description of the workshop [3].

Figs 25 illustrates the computational domain and the mesh used for the computations. It is obtained by extrusion of an curved two-dimensional unstructured mesh on the periodic plane. This mesh has been generated by the open source curved grid generator Gmsh [19]. The high-order DGM has shown to be able to handle large transitions without loosing accuracy or robustness. Hence an aggressive mesh refinement strategy has been used, resulting in large mesh size transitions going out of the wake and the blade boundary layer. The mesh is essentially refined on the second half of the blade. Additional refinements are included to capture the turbulent structures originating from the separated shear layer and the vortex shedding off the trailing edge. The two-dimensional mesh is extruded on 20 regular layers, leading to $262k$ hexahedra. Using a fourth order solution ($p = 3$), this gives $16.8M$ *degrees of freedom (dof).*

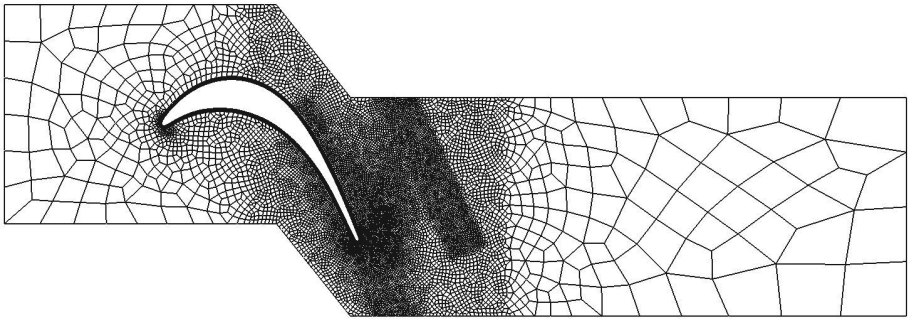


Fig. 25. Global view of the computational domain and mesh

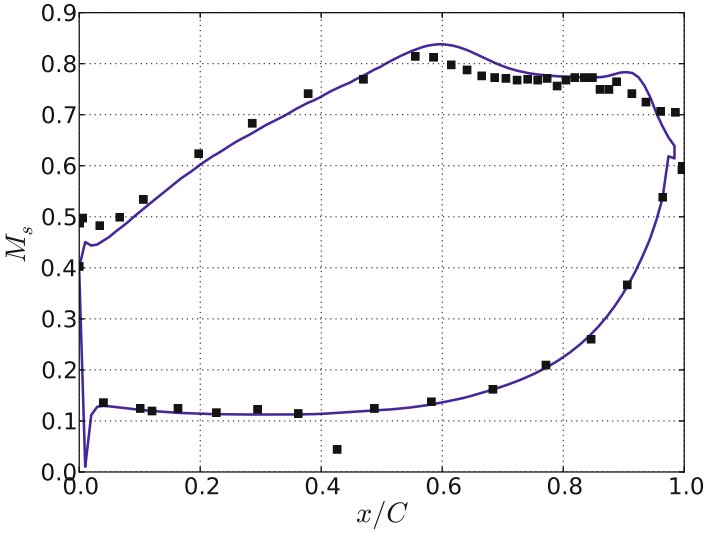
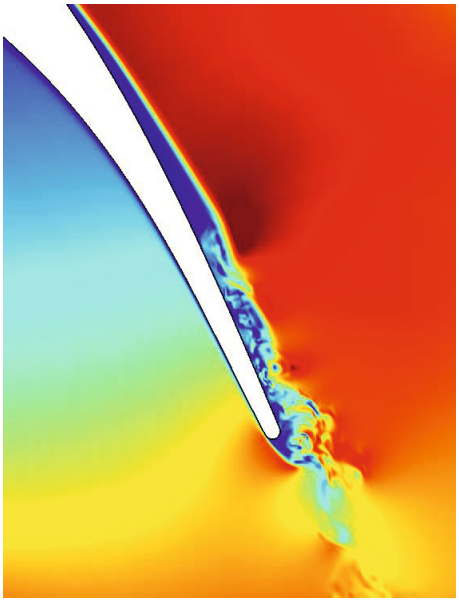


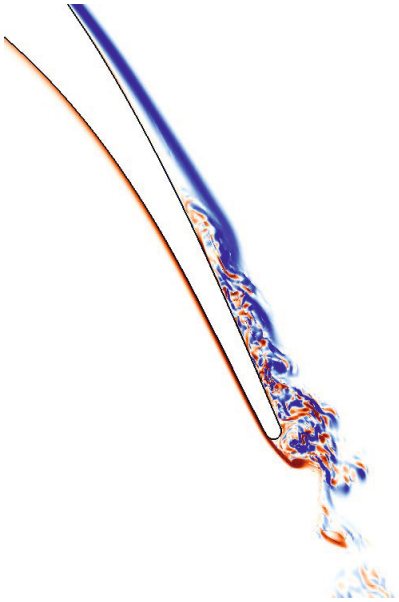
Fig. 26. Isentropic Mach number

Fig 26 shows the resulting averaged isentropic Mach number distribution along the blade chord and the measurements performed at the VKI. Those results have been presented at the second international workshop on high-order methods for CFD [3], together with those of several other participants. It was seen that all of the computations differ in the same way from the experiment at the front suction side (see also the corresponding test case section further on). LES computations, performed with a standard finite volume code at Cenaero and RANS computations at VKI, showed a similar discrepancy. Based on the remarkable consistency between the different computations and codes, it was conjectured that there was potentially a small difference in the flow conditions and the cascade geometry. This mismatch between the computational and the experimental setup is currently investigated at the VKI in collaboration with Cenaero.

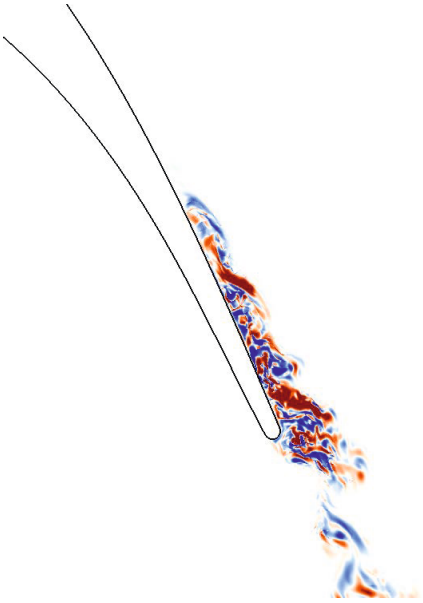
Fig. 28 shows typical snapshots of the Mach number field and the spanwise components of the vorticity field. The transition in the shear layer and reattachment zone of the separation bubble is clearly visible by the appearance of in-plane vorticity components. It occurs near 80% of the axial chord C_{ax}, whereas the separation occurs already around 50% C_{ax}.



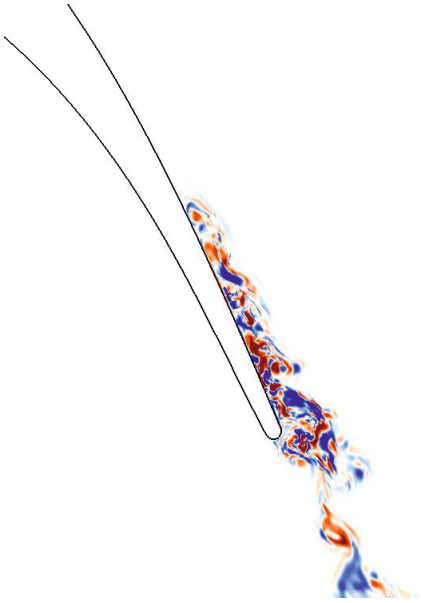
(a) Mach number



(b) Spanwise vorticity

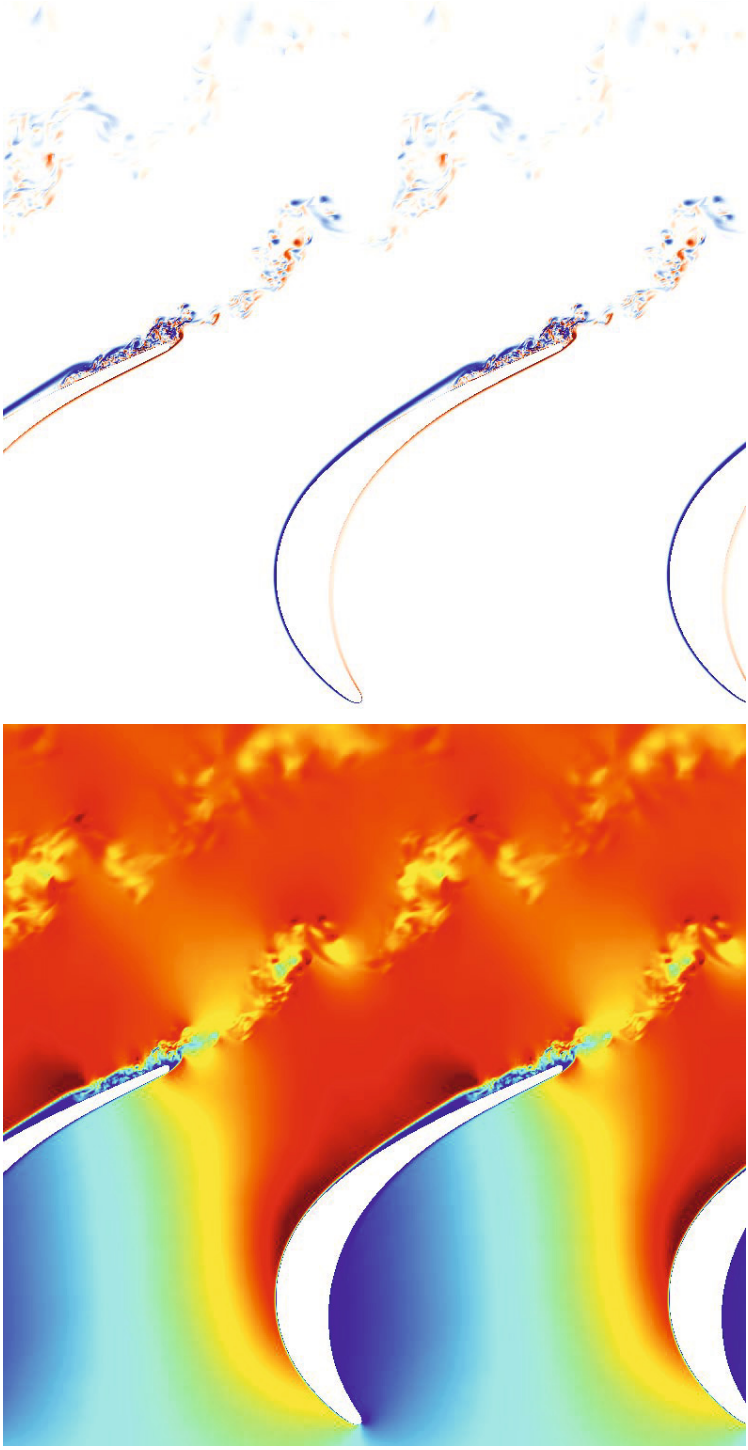


(c) Axial vorticity



(d) Pitchwise vorticity

Fig. 27. Flow field near the the trailing edge



(a) Mach number (b) Spanwise vorticity

Fig. 28. Global overview of the flow field on the spanwise periodic plane. The passage has been duplicated for clarity.

5.3 ILES of the JEAN Nozzle

This test case consists of the LES of a turbulent isothermal and subsonic jet exhausting from a nozzle. The exhaust Mach number corresponding to the isentropic expansion from inlet total to free stream static pressure is $M_{is} = 0.75$. Based on these isentropic conditions and exhaust diameter D, the Reynolds number is $Re_{D,is} = 5 \times 10^4$. Experimental data have been obtained during the research project JEAN [26], and consist of the variations of the velocity average and fluctuations along the nozzle axis, and along radial profiles at three axial positions $x/D = 1$, $x/D = 2.5$, and $x/D = 5$ from the jet exit. It should be noted that the experimental data were obtained at a much higher Reynolds number of $Re_{D,is} = 9 \times 10^5$, assuming that the impact of the Reynolds number is already negligeable for the computational conditions $Re_{D,is} = 5 \times 10^4$.

The mesh is obtained by full rotational extrusion of a two-dimensional unstructured mesh along an axi-radial cut across the domain, illustrated in Fig. 29. Due to solver limitations, a structured region is used on the axis to avoid the formation of pyramid elements. The mesh is composed of 582464 hexahedra and 23808 prisms. Using the fourth order interpolation ($p = 3$) roughly 38M *dof* are used. The far-field boundaries of the domain are located far away from the nozzle, *i.e.* at $x/D = 300$ and $r/D = 100$, and a very coarse mesh is used to destroy turbulent flow structures before they can interact with the boundary. Due to the rapid increase in mesh size, this large domain does not increase the number of *dof*, and hence computational cost, significantly. The resolution near

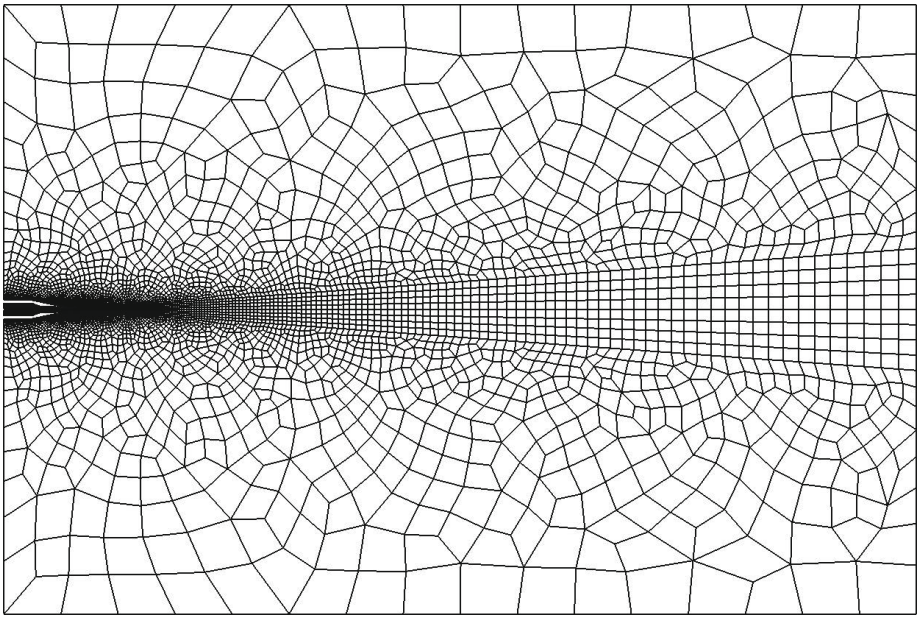


Fig. 29. Section of the JEAN nozzle mesh. Global view.

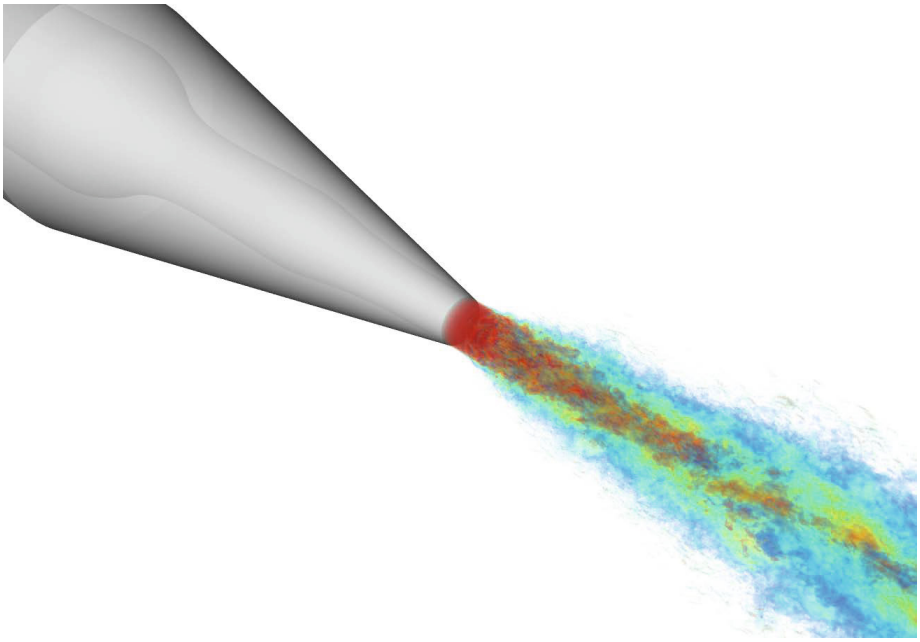


Fig. 30. JEAN nozzle. Volume rendering of the velocity.

the solid walls is fine enough to perform a wall-resolved ILES, as $y^+ < 4$ everywhere (studies have shown that accurate results can be obtained with DGM for $y^+ < 8$). Therefore stick conditions were imposed on the nozzle walls.

The geometry of the nozzle and a volume rendering of the obtained velocity are shown in Fig. 30 Fig. 31 shows a section of the mesh close to the jet exit. The boundary layer mesh is prolonged into the domain at the exit of the nozzle to capture the shear layers and the Kelvin-Helmholtz instabilities downstream. Two refinement boxes are placed in the wake of the jet, allowing to capture the complex turbulent interactions up to $x/D = 14$ downstream from the nozzle exit (up to $x/D = 5$ for the finest refinement box).

Total pressure and temperature are imposed at the inlet of the nozzle, whilst free stream boundaries are imposed elsewhere. The initial condition corresponds to the freestream conditions. To prevent the creation of strong acoustic waves during the transient phase, ramping functions are given at the inlet for the total conditions. It is important to remark that due to lack of data, the total pressure and temperature were assumed constant across the inlet surface, and that therefore no boundary layer profile was therefore imposed. For the same reasons, no inlet turbulence was imposed. The total values imposed at the inlet, the ambient states imposed at the freestream boundaries and the other physical parameters of the simulation were specified by Onera, and are summarized in table 4. We should remark that in order to obtain the imposed Reynolds number,

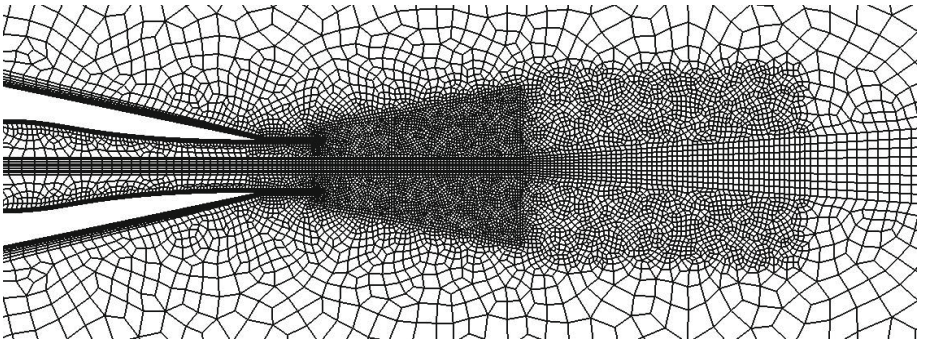


Fig. 31. Section of the JEAN nozzle mesh. Zoom on the nozzle exit.

Table 3. Physical parameters of the simulation

Ambient pressure	p_0	100380 $[Pa]$
Ambient temperature	T_0	283.15 $[K]$
Total pressure	p_T	144400 $[Pa]$
Total temperature	T_T	314.15 $[K]$
Exit Mach number	M_{is}	0.74
Dynamic viscosity	μ	3.12757×10^{-4} $[kg/m^2s]$
Conductivity	κ	0.48 $[W/mK]$
Jet diameter	D	0.05 $[m]$
Exit Reynolds number	$Re_{D,is}$	49272.44

the viscosity and conductivity were artificially increased with respect to standard values for air.

The transient of the computation corresponding to the inlet profile ramping is computed using third order accurate interpolants ($p = 2$). Using this relatively modest resolution ($\sim$ 16M dof), the flow is already fully turbulent featuring very small vortical structures in the wake of the jet as seen in Fig. 32. Once the target total conditions are reached in the nozzle, the computation is restarted using the fourth order accurate interpolants ($p = 3$). After a short transient, typically a few throughflow periods, a statistically stationary regime is obtained. The increased resolution can clearly be seen in Fig 33, showing much smaller and stronger turbulent structures.

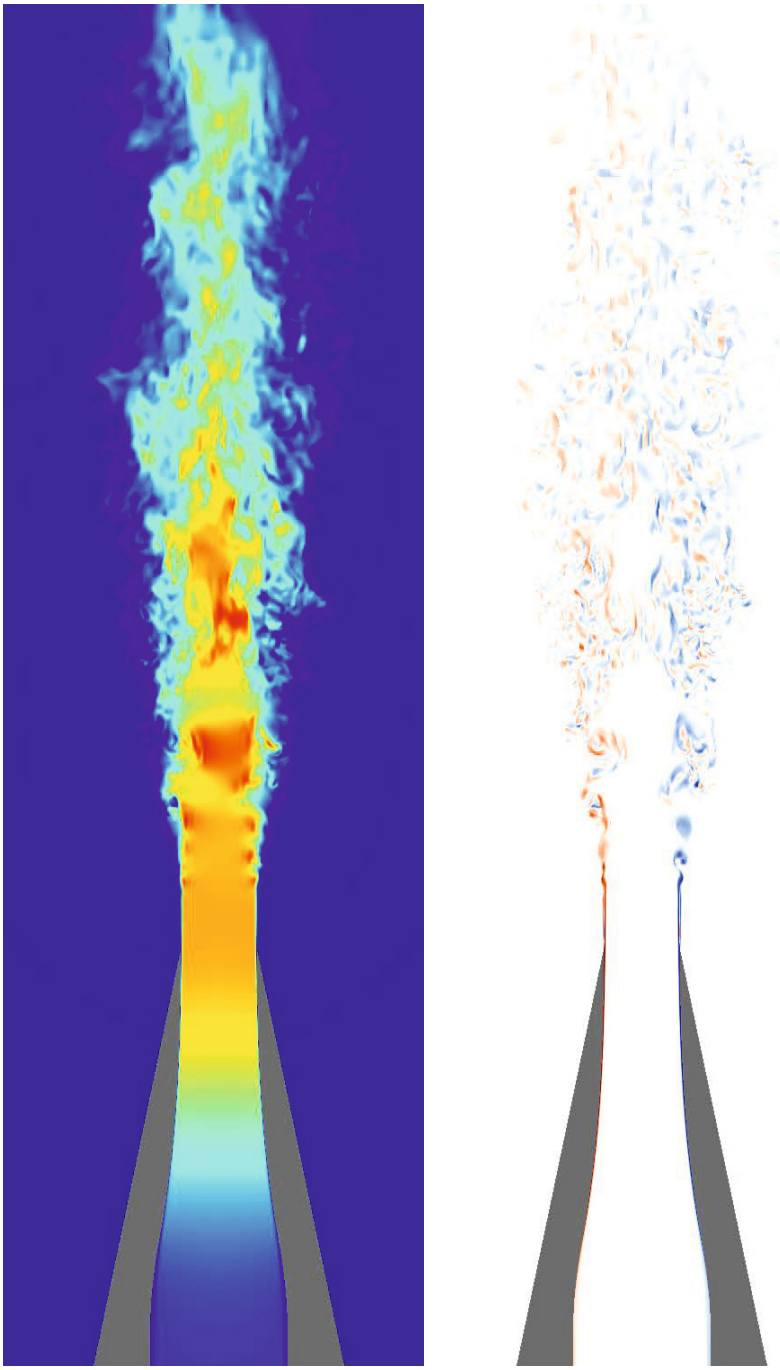


Fig. 32. Snapshot of the Mach number (top) and the z-component of the vorticity (bottom) fields using third order solution ($p = 2$).

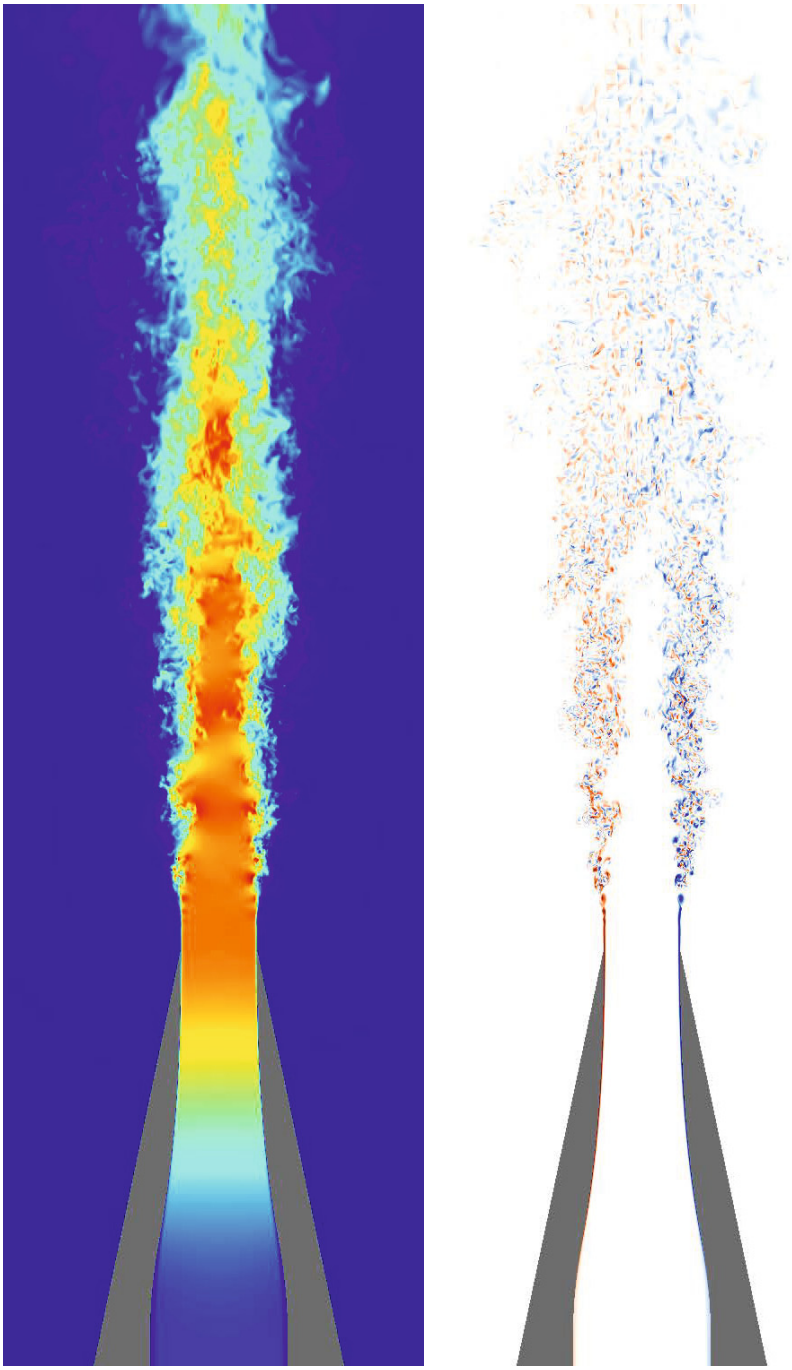


Fig. 33. Snapshot of the Mach number (top) and the z-component of the vorticity (bottom) fields using fourth order solution ($p = 3$).

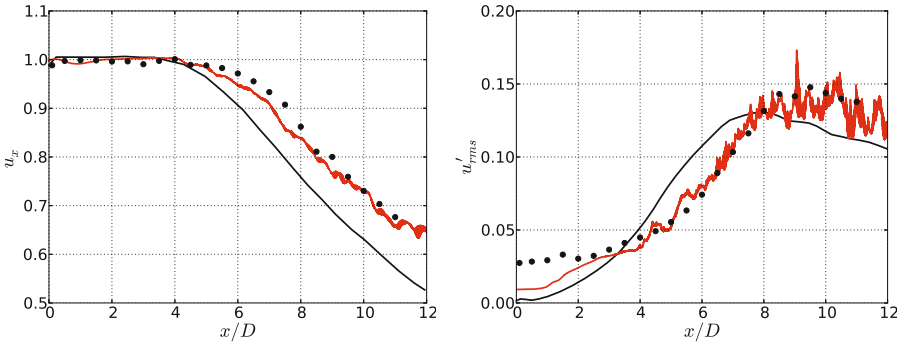


Fig. 34. JEAN Nozzle. Mean axial velocity along the nozzle axis. Experimental results (black circle), LES of Andersson *et al.* [9] (black line) and the DGM/ILES computation (red line).

Experimental data are available along the axis and at different axial positions as well as the LES results of Andersson *et al.* [9]. All the following results are scaled with the obtained velocity at the centre of the nozzle exit. Fig. 34 shows the mean axial velocity $u_x(x)$ and the mean axial fluctuations $u'_{rms}(x)$ along the nozzle axis. The results are close to the experiment, with the end of the potential core located around $x/D = 4$. The drop of velocity in the wake is also globally well represented. The absence of inlet turbulence is reflected in the velocity fluctuations, resulting in an under-prediction at the nozzle exit plane. Nevertheless, the computed fluctuations match the experimental values well further downstream, and improve upon the results by Andersson *et al.* [9].

Fig. 35 shows the mean axial velocity and the mean axial RMS fluctuations at the axial positions $x/D = 1$, $x/D = 2.5$, and $x/D = 5$. At each station, the solution is also averaged in rotation to accelerate the convergence. It can be seen that the diffusion of the jet is slightly more important in the computations, which can be expected as they are performed at a relatively low Reynolds number compared to the experiment. Especially for the fluctuations, a wider curve is obtained together with a more pronounced peak at the first station. Again, the effects of the inlet turbulence can clearly be seen at the first station where the fluctuations in the core are three times more important in the experiment. Qualitatively, these effects are in line with the results obtained by Andersson *et al.* [9]. Globally the agreement with experiments is fair and improved with respect to [9]. Further downstream oscillations of the average and correlations are observed, showing that there more time would be required to fully converge the statistics.

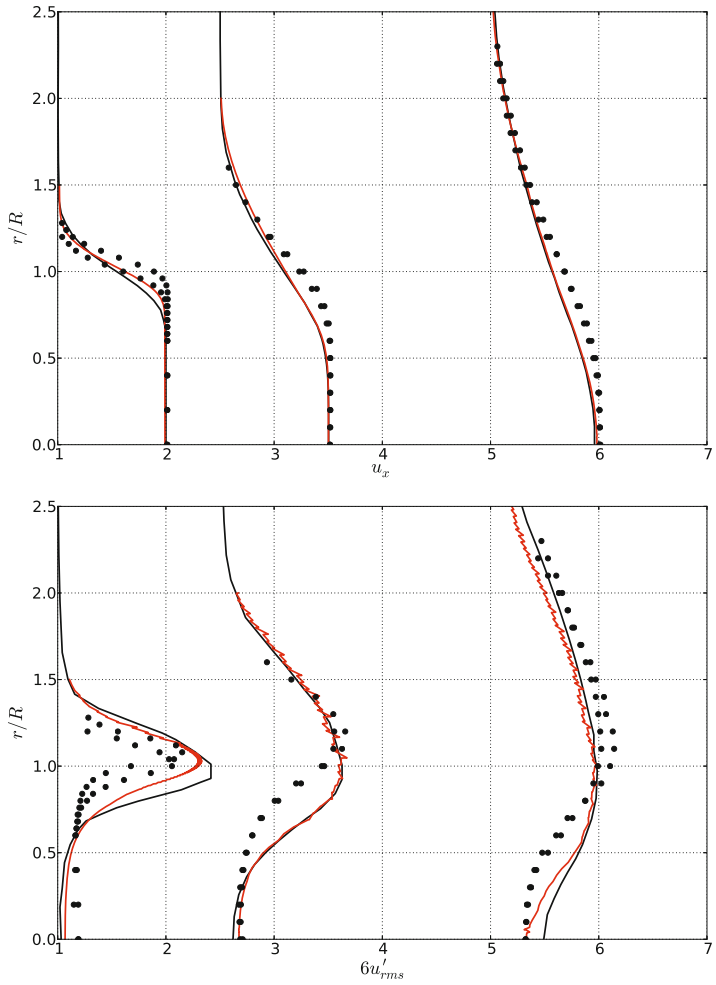


Fig. 35. JEAN Nozzle. Mean axial velocity and mean axial RMS fluctuations at the axial stations $x/D = 1$, $x/D = 2.5$, and $x/D = 5$. Experimental results (black circle), LES of Andersson *et al.* [9] (black line) and DGM/ILES (red line). Curves are shifted and scaled for clarity.

6 Conclusions

The potential of the DGM code Argo for industrial scale-resolving simulations was demonstrated. DGM, from relatively low orders onward ($p \leq 3$), provides spatial accuracy comparable to state of the art academic solvers for DNS. Excellent strong scalability was obtained on world class HPC infrastructure, using up to 130.000 cores. Finally, it was shown that the DGM/ILES provides a very good level of accuracy on free and wall bounded flow benchmarks. The viability

of the method was further demonstrated on more practical applications, namely a subsonic isothermal jet and a transitional turbine cascade.

During the cascade computations it proved difficult to match computational with experimental conditions, in all probability due to the extreme sensitivity of transitional flows to the flow conditions. As the deviations are consistent with computations performed by other institutes, amongst others during the workshop on high order methods, it is presumed that the computational setup does not reflect adequately the conditions in the wind tunnel. Further potential discrepancies include lateral wall effects in the tunnel, or insufficient spanwise extent of the computational grid. This issue will in all probability be a recurring difficulty for spanwise periodic simulations of transitional cascade or wing flows.

Acknowledgments. The authors gratefully acknowledge funding through the FP6 European research project *Adigma* and the *European Regional Development Fund* under contract N° EP1A122030000102. The authors further thank the *Partnership for Advanced Computing in Europe (PRACE)* and the *Jülich Supercomputing Centre (JSC)* for their help and the computational time on the BG/P machine JuGENE in the industrial pilot project *noFUDGE*, and on the BG/Q machine JuQUEEN during the two-year access project *PAdDLES*. Final thanks go out to Prof. Arts of the von Karman Institute for providing experimental data for the T106C case.

References

1. ERCOFTAC QNET-CFD Wiki, `http://uriah.dedi.melbourne.co.uk/w/index.php/Main_Page`
2. First International Workshop on High-Order CFD Methods. Nashville, Tennessee, January 7-12 (2012), `http://zjwang.com/hiocfd.html`
3. Second International Workshop on High-Order CFD Methods. Cologne, Germany, May 27-28 (2013), `http://www.dlr.de/as/desktopdefault.aspx/tabid-8170/13999_read-35550/`
4. Altmann, C., Beck, A.D., Hindenlang, F., Staudenmaier, M., Gassner, G.J., Munz, C.-D.: An Efficient High Performance Parallelization of a Discontinuous Galerkin Spectral Element Method. In: Keller, R., Kramer, D., Weiss, J.-P. (eds.) Facing the Multicore-Challenge III. LNCS, vol. 7686, pp. 37–47. Springer, Heidelberg (2013)
5. Andersson, N., Eriksson, L.-E., Davidson, L.: Large-Eddy Simulation of a Mach 0.75 jet. In: AIAA paper 2003-3312, 9th AIAA/CEAS Aeroacoustics Conference, pp. 2003–3312 (May 2003)
6. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. SIAM Journal of Numerical Analysis 39, 1749–1779 (2002)
7. Bijl, H., Carpenter, M.H., Vatsa, V.N., Kennedy, C.A.: Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow. Journal of Computational Physics 179, 313–329 (2002)
8. Breuer, M., Peller, N., Rapp, C., Manhart, M.: Flow over periodic hills - numerical and experimental study over a wide range of reynolds numbers. Computers and Fluids 38, 433–457 (2009)

9. Bricteux, L., Duponcheel, M., Winckelmans, G.: A multiscale subgrid model for both free vortex flows and wall-bounded flows. Physics of Fluids 21, 105102 (2009), doi:10.1063/1.3241991
10. Benocci, C., Pinelli, A.: The role of the forcing term in the large-eddy simulation of equilibrium channel flow. Engineering Turbulence Modeling and Experiments 1, 287–296 (1990)
11. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods in Fluid Dynamics. Springer (1988)
12. Carton de Wiart, C.: Cenaero's results on test case C3.6: DNS and LES of the flow over a 2D periodic hill. In: 2nd International Workshop on High-Order CFD Methods, Köln, May 27-28 (2013)
13. Carton de Wiart, C., Hillewaert, K., Bricteux, L., Winckelmans, G.: Implicit LES of free and wall bounded turbulent flows based on the discontinuous Galerkin/symmetric interior penalty method. Accepted with Minor Revisions for Int. J. Numer. Meth. Fluids (2014)
14. Carton de Wiart, C., Hillewaert, K., Duponcheel, M., Winckelmans, G.: Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number. Int. J. Numer. Meth. Fluids 74, 469–493 (2014), doi:10.1002/fld.3859
15. Cockburn, B.: Discontinuous Galerkin Methods for Convection-Dominated Problems. In: GI-BIFOA Internationale Fachtagung: Informationszentren in Wirtschaft und Verwaltung. Lecture Notes in Computational Science and Engineering, vol. 9, pp. 69–224. Springer (1974)
16. de Wiart, C.C.: Towards a discontinuous Galerkin solver for scale-resolving simulations of moderate Reynolds number flows, and application to industrial cases. PhD thesis, Ecole polytechnique de Louvain/iMMC (May 2014)
17. Deville, M.O., Fischer, P.F., Mund, E.H.: High-Order Methods for Incompressible Fluid Flow. Cambridge University Press (2002)
18. Fröhlich, J., Mellen, C.P., Rodi, W., Temmerman, L., Leschziner, M.: Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. J. Fluid Mech. 526, 19–66 (2005)
19. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)
20. Hesthaven, J.S., Warburton, T.: Nodal Discontinuous Galerkin Methods; Algorithms, Analysis and Applications. Text in Applied Mathematics. Springer (2008)
21. Hillewaert, K., Carton de Wiart, C., Verheylewegen, G., Arts, T.: Assessment of a high-order discontinuous Galerkin method for the direct numerical simulation of transition at low Reynolds number in the T106C high-lift low pressure turbine cascade. In: Proceedings of the ASME Turbine Technical Conference and Exhibition, number GT2014-26739. ASME Turbo Expo (2014)
22. Hillewaert, K.: Development of the Discontinuous Galerkin Method for high-resolution, large scale CFD and acoustics in industrial geometries. PhD thesis, Ecole polytechnique de Louvain/iMMC (February 2013)
23. Hoyas, S., Jimenez, J.: Reynolds number effects on the Reynolds-stress budgets in turbulent channels. Physics of Fluids 20, 101511 (2008), doi:10.1063/1.3005862
24. Hu, F.Q., Hussaini, M.Y., Rasetarinera, P.: An analysis of the discontinuous galerkin method for wave propagation problems. Journal of Computational Physics 151(2), 921–946 (1999)

25. Huynh, H.T.: A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In: 18th AIAA Computational Fluid Dynamics Conference, number AIAA-2007-4079 (2007)
26. Jordan, P., Gervais, Y., Valiére, J.-C., Foulon, H.: Final results from single point measurements, Project deliverable D3.4, JEAN - EU 5th Framework Programme, G4RD-CT2000-00313 (2002)
27. Karniadakis, G.E., Sherwin, S.J.: Spectral/hp element methods for computational fluid dynamics. Oxford University Press (2005)
28. Lodato, G., Castonguay, P., Jameson, A.: Discrete filter for large-eddy simulation using high-order spectral difference methods. International Journal for Numerical Methods in Fluids 72(2), 231–258 (2012)
29. Manhart, M., Friedrich, R.: DNS of a turbulent boundary layer with separation. Intl. J. Heat Fluid Flow 23(5), 572–581 (2002)
30. Moser, R.D., Kim, J., Mansour, N.N.: Direct Numerical Simulation of turbulent channel flow up to $Re_\tau = 590$. Physics of Fluids 11(4), 943–945 (1998)
31. Rapp, C., Pfleger, F., Manhart, M.: New experimental results for a les benchmark case. In: Armenio, V., Guerts, B., Fröhlich, J. (eds.) Direct and Large-Eddy Simulations VII (2009)
32. Shahbazi, K.: An explicit expression for the penalty parameter of the interior penalty method. Journal of Computational Physics 205, 401–407 (2005)
33. Sun, Y., Wang, Z.J., Lia, Y.: High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids. Communication in Computational Physics 2(2), 310–333
34. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer (1999)
35. Tucker, P.: Computation of unsteady turbomachinery flows: Part 1 - progress and challenges. Progress in Aerospace Sciences 47, 522–545 (2011)
36. Tucker, P.: Computation of unsteady turbomachinery flows: Part 2 - LES and hybrids. Progress in Aerospace Sciences 47, 546–569 (2011)
37. Vasilyev, O.V.: High order finite difference schemes on non-uniform meshes with good conservation properties. Journal of Computational Physics 157(2), 746–761 (2000)
38. Wang, W.: A non-body conformal grid method for simulations of laminar and turbulent flows with a compressible large eddy simulation solver. PhD thesis, Iowa State University (2009)
39. Wang, Z.-J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., Kroll, N., May, G., Persson, P.-O., van Leer, B., Visbal, M.: High-order CFD methods: Current status and perspectives. International Journal for Numerical Methods in Fluids 72(8), 811–845 (2013)
40. Winckelmans, G.S., Jeanmart, H., Carati, D.: On the comparison of turbulence intensities from large-eddy simulation with those from experiment or direct numerical simulation. Physics of Fluids 14(5), 1809 (2002), doi:10.1063/1.1466824

A High-Order Finite-Volume Method with Block-Structured Local Grid Refinement

J. Kok and H. van der Ven

National Aerospace Laboratory NLR,
Flight Physics and Loads Department,
Amsterdam, The Netherlands
{johan.kok,harmen.van.der.ven}@nlr.nl

Abstract. Time-dependent vortex-dominated flows are computed accurately with a high-order finite-volume method on structured grids. In order to attain the required grid resolution in the vortex region, block-wise local grid refinement is employed. A new topology-based block-refinement algorithm allows the efficient generation of such block-wise refined meshes. The high-order finite-volume method is extended with high-order interpolation to deal with the partially continuous grids at block interfaces that result from the refinement. Results are presented for three applications: one time-accurate RANS computation of a helicopter flow case and two hybrid RANS–LES computations of strongly separated flows.

Keywords: CFD, high-order method, finite-volume method, local grid refinement.

1 Introduction

Boundary-conforming block-structured finite volume solvers have several advantages. The block-structured mesh allows for accurate and efficient discretization of the flow equations. In particular, high-order methods can be devised by enlarging the stencil with only a moderate increase of computational effort. Using a single boundary-conforming mesh obviates the need for three-dimensional interpolation algorithms which are for instance necessary for Chimera methods. These advantages come at a price, though, as on the one hand the grid generation process is not fully automated and on the other hand generating an efficient grid in terms of number of grid cells is hampered by geometrical and topological restrictions, especially for complex geometries.

For time-dependent vortex-dominated flows, for example, the mesh should be as uniform as possible in the vortex regions. Within the multi-block structured framework such grids are difficult or even impossible to generate, unless some freedom in the grid structure is introduced. Here, this freedom consists of block-wise refinement of the mesh, where the grid in a block may be refined in one or more direction(s) by a factor 2^n. This results in partially continuous grids at block interfaces, where the ratio of mesh widths may be 2^n:1. Block-wise refined

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_14

meshes have the potential of attaining uniform grid spacing in the vortex regions. To generate these block-wise refined meshes efficiently, a topology-based block-refinement algorithm has been developed which refines the topology in such a way that the resulting block-refined mesh is as uniform as possible in, and only in, the vortex region.

The multi-block flow solver ENSOLV employs a low-dispersion symmetry-preserving fourth-order finite-volume method [7, 10]. This allows the accurate capturing of vortices, for example in helicopter rotor computations and in LES or hybrid RANS–LES of turbulent flow. This high-order method must be extended to deal with the partially continuous grids at block interfaces, resulting from the block-wise grid refinement. A new algorithm has been developed that can be both second-order and fourth-order accurate, and that can even be applied to completely discontinuous grids. Here, this algorithm is described as applicable to partially continuous grids. Furthermore, the algorithm is verified for a basic test case, consisting of the convection of an isentropic vortex in a uniform flow.

This paper presents results for the application of local grid refinement to both helicopter flow simulations and X-LES computations. The helicopter flow concerns the baseline case of the HART-II experiment. The X-LES method [8, 10] is a DES-type [12] hybrid RANS–LES method. Two cases are considered: the flow over a bump in a square duct and the flow over a delta wing at high angle of attack. For both cases, due to the simple geometry, a baseline grid is available that is practically uniform in the separated flow region where an LES is performed. Therefore, local grid refinement is not needed to obtain the desired grid resolution in the region of interest. Instead, the grid outside the regions of interest is coarsened to investigate whether a reduction of computational costs can be obtained in this way, without loss of accuracy.

2 Basic Method Description

The flow solver ENSOLV has the following capabilities which are relevant to the subject of this paper.

A fourth-order, symmetry-preserving, low-dispersion finite-volume scheme [7] is used to discretize convection. A central (instead of upwind) discretization is used, so that the method contains no numerical dissipation. The finite-volume method maintains its properties (fourth-order accurate, low numerical dispersion, no numerical dissipation) on non-uniform, curvilinear grids. On uniform, Cartesian grids, it is equivalent to the DRP scheme of Tam and Webb [13]. An important property of the symmetry-preserving finite-volume method is that it ensures that kinetic energy is conserved by convection (see e.g. Verstappen and Veldman [17]). This improves the stability properties of the scheme significantly, requiring only a low level of artificial diffusion, if any at all. For compressible flow, sixth-order artificial diffusion is added to the equations, maintaining the fourth-order accuracy [10].

For aeroelastic simulations, the flow solver is coupled in a strong way (exchange at each time step) to a modal representation of the structural model. At

each time step, the full 3D grid must be deformed in order to remain boundary conforming with the aeroelastically deformed geometry. This is done using a combination of a 3D volume spline (block vertices and edges) and transfinite interpolation (block faces and interior). The fourth-order finite-volume method has been extended to remain fourth-order accurate as well as fully conservative on such deforming grids.

The X-LES method [8] is a DES-type hybrid RANS–LES method based on the k–ω turbulence model. A single set of turbulence-model equations is used to model both the Reynolds stresses in RANS zones and the subgrid stresses in LES zones. The X-LES method in particular is based on the TNT k–ω model. The method switches to LES when the RANS length scale ($l = \sqrt{k}/\omega$) exceeds the LES length scale ($C_1\Delta$, with Δ the filter width and $C_1 = 0.05$). The RANS length scale is then replaced by the LES length scale in the expression for the eddy viscosity as well as in the expression for the dissipation of turbulent kinetic energy. The filter width Δ is defined at each grid point as the maximum of the mesh size in all directions.

To improve the capturing of free shear layers, two modifications have been added to the X-LES method: a stochastic subgrid-scale (SGS) [9] model and a high-pass filtered SGS model [10]. For the present computations, only the high-pass filtered SGS model has been employed. In this SGS model, the SGS stresses are computed from the velocity fluctuations instead of the instantaneous velocity, where the velocity fluctuations are obtained by applying a temporal high-pass filter to the velocity field.

To prevent the X-LES method from inadvertently switching to LES inside attached boundary layers (resulting in so-called shear-stress depletion), the shielding function of DDES [14] is included.

3 Block-Structured Local Grid Refinement

3.1 Algorithm

Block-wise refined meshes have the potential of attaining uniform grid spacing in the vortex regions. However, the grids may not be that efficient, since the grid resolution may be too small in some regions. This may happen inside the vortex region for a block with significant stretching: if the coarse cells satisfy the resolution requirement, the fine cells will be too small. It may also happen outside the vortex region when the refined block is only partially contained in the vortex region. These examples show that an efficient grid with block-wise refinement can only be obtained if the topology of the mesh is modified.

The new topology-based local grid refinement algorithm is described briefly below. The interested reader may find more details in Van der Ven et al. [15]. The algorithm consists of the following steps:

1. uniform refinement of the block topology, splitting all blocks into subblocks of a fixed block size (measured as the number of cells within a block); typically the target block size is 8^3 or 16^3;

2. further topology-refinement of those blocks which are targeted for grid refinement; the subblock topology is such that each grid-refined subblock is as close as possible to the fixed block size used in step 1;
3. based on the user-defined maximum refinement ratio (2:1, 4:1, etc.) blocks bounding the refinement region are targeted for refinement, and step 2 is repeated for those blocks;
4. generate the block grids within the refined topology.

In step 2 a refinement sensor is used which indicates how the grid within a block should be refined. The sensor consists of two parts: 1) the target mesh width; 2) the region where this mesh width should be attained. The target mesh width should be attained for all cells within a block. The refinement region can be specified in different ways: distance to a geometric object; specification of specific blocks; or a region described by a simple geometric object such as a sphere, cylinder, or cube. As the aim of the topology refinement is a uniform mesh, the grid refinement within a block is allowed to be anisotropic.

The algorithm is demonstrated for the NACA0012 airfoil, using a refinement region defined by the distance to a line segment. This example serves as an illustration of the algorithm only; in the next section the algorithm will be applied to the isolated HART II rotor.

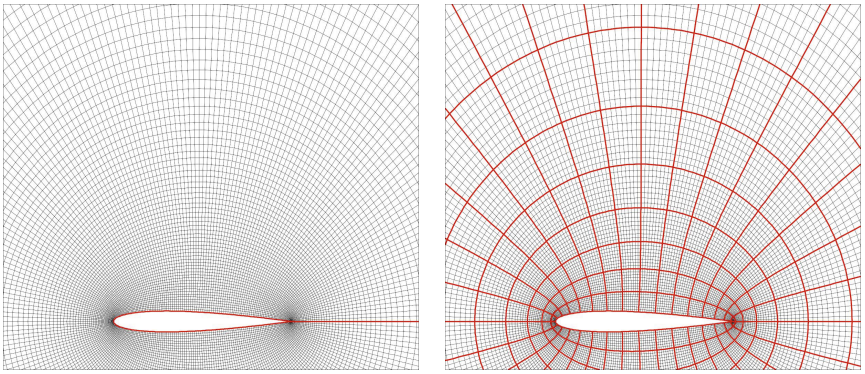
The original mesh and topology is shown in Figure 1a. With a specified block size of 8^2, the refined topology of step 1 of the algorithm is shown in Figure 1b. The line segment used to define the refinement region is shown in Figure 1c. A region at a distance of 10% chord to this segment is defined as the refinement region. Within this region a mesh width of 0.001 chord should be attained. Figure 1d shows the refined topology which is the result of step 2 of the algorithm. Note that at this stage the grid has not been refined yet and that some of the blocks in the refined topology consist of a single cell of the original mesh. Subsequently, the user-defined refinement ratio is applied. Figure 1e shows the refined topology based on a maximum ratio of 2:1. In the last step of the algorithm the refined grid is generated, which is shown in Figure 1f.

It is worthy to note that the final topology consists of 982 blocks. Clearly, it is unfeasible to expect from a user to generate such topologies by hand.

3.2 HART II Rotor

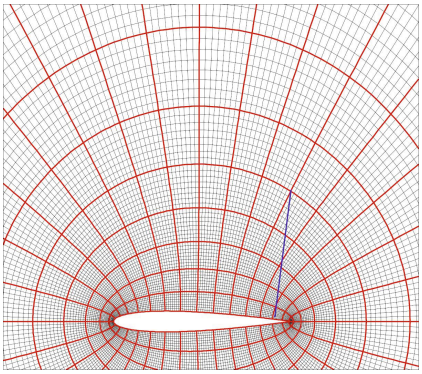
For rotor flows the convection of the tip vortices in the wake of the blades is important to capture the blade-vortex interaction. As the vortices move through the wake, the easiest way of obtaining a mesh with sufficient resolution in the vortex regions is to uniformly refine the mesh in a cylinder around the rotor blades. So the definition of the refinement region in this case is a cylinder centred at the rotor hub with radius equal to the rotor radius and sufficient height to contain the vortices.

The target mesh width is $0.01R$, where R is the rotor radius. This corresponds to 16% chord, which is a rather coarse resolution, but nevertheless leads to a refined mesh of 26 million cells, whereas the original mesh had 13 million cells.

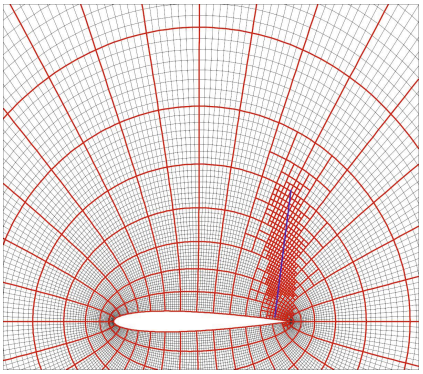


(a) Original mesh and topology

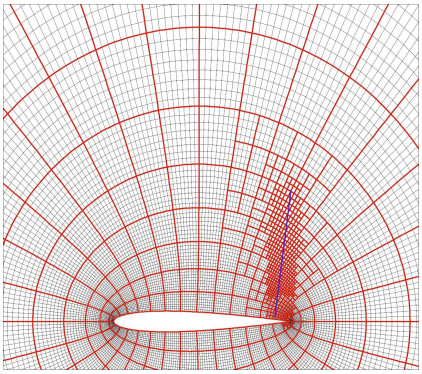
(b) Refined topology, result of step 1



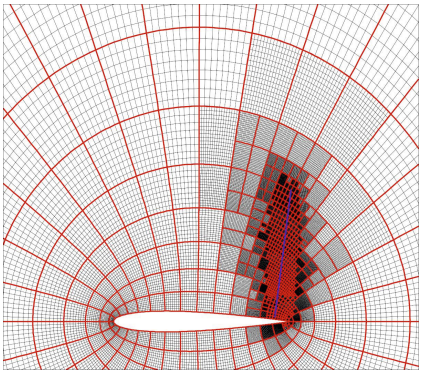
(c) The sensor



(d) Refined topology, result of step 2



(e) Refined topology, result of step 3



(f) Refined grid, result of step 4

Fig. 1. Illustration of the grid generation algorithm. Block boundaries are shown in red; grid lines in black; the line segment used in the sensor is shown in blue.

Figure 2a shows a detail of the original mesh in a horizontal plane through the hub. The same detail of the refined mesh is shown in Figure 2c. The uniform resolution in the wake is clearly visible. The imposed refinement ratio of 2:1 is also visible in the resolution outside the rotor disk. Figures 2b and 2d show the refinement in a vertical plane through the hub bisecting the rotor disk between two blades.

Figures 2e and 2f demonstrate that the target mesh width is actually obtained in the specified refinement region. The figure shows the ratio of the maximum mesh width in a cell over the target mesh width. As can be seen the ratio is less than one in the refinement region, demonstrating the correct functioning of the algorithm. The figure also shows that the algorithm is relatively efficient: the region where the target mesh width is attained is not that much bigger than the refinement region.

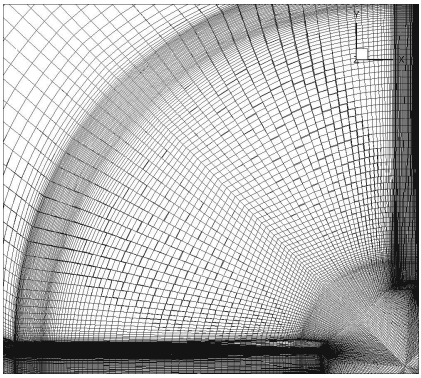
4 High-Order Method at Partially Continuous Grids

4.1 Algorithm

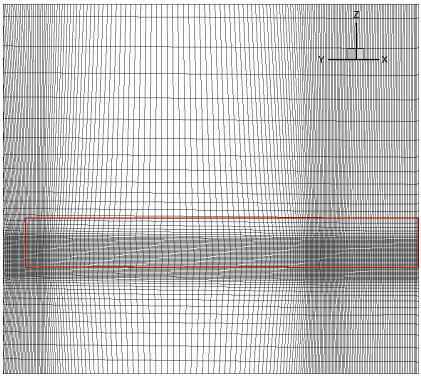
ENSOLV employs a cell-centred finite-volume method. For the second-order scheme, a five-point stencil is used in each computational direction. For the fourth-order scheme, a seven-point stencil is used. To couple the flow computations in different blocks of the multi-block grid, dummy cells are used. Considering the size of the stencil, the second-order scheme requires two rows of dummy cells along each interface, and the fourth-order scheme requires three rows. The flow solution at the cell-centres of the dummy cells are determined by interpolating the conservative flow variables of the adjacent block. The interpolation algorithm is described for the case of a 2:1 ratio in mesh size, both along and normal to the interface.

A generic interpolation algorithm has been developed, applicable both to partially continuous and full discontinuous grids. Main requirement has been that the algorithm is robust and accurate for time-accurate computations, maintaining the properties of the high-order finite-volume method as much as possible. This was obtained by initially dropping the requirement that the scheme is fully conservative at the interface. Future extension towards a fully conservative scheme, however, is possible.

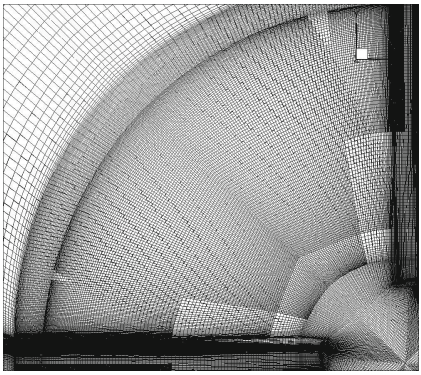
First, the second-order variant is described. In this case, the two rows of dummy-cell values are obtained by linear interpolation in computational space, which is second-order accurate. The interpolation is performed in each computational direction separately. A first option is to apply this interpolation only along the interface, as shown in figure 3, ignoring the jump in mesh size in the direction normal to the interface. This has as advantage that the symmetry of the discretization normal to the interface is maintained, requiring two rows of source cells on each side of the interface. Furthermore, this scheme is generic enough to be applied to a fully discontinuous grid (with appropriately defined interpolation coefficients). A second option is to interpolate the solution in the normal direction as well (after first interpolating along the interface), as shown



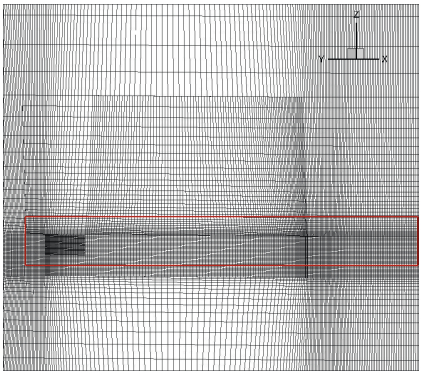
(a) Original mesh at horizontalplane



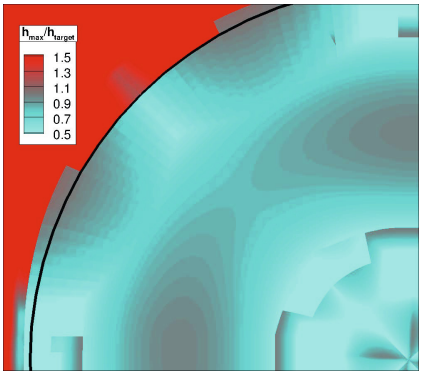
(b) Original mesh at vertical plane



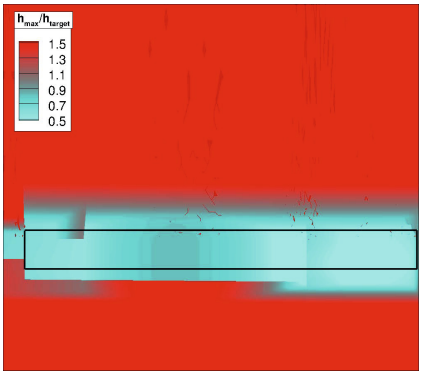
(c) Refined mesh at horizontalplane



(d) Refined mesh at vertical plane



(e) Mest width ratio at horizontal plane



(f) Mest width ratio at vertical plane

Fig. 2. Original and refined mesh for the HART II rotor at a horizontal and a vertical plane through the hub; the border of the refinement region is shown in red. Ratio of the maximum mesh width in a cell over the target mesh width in the same two planes; the border of the refinement region is shown in black.

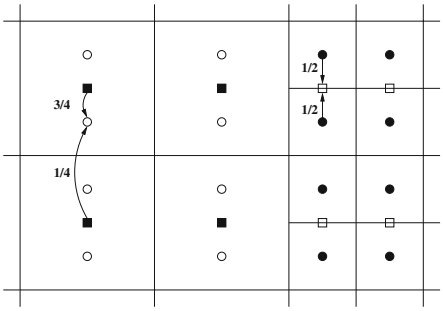


Fig. 3. Stencil to determine dummy-cell values using second-order interpolation along interface only (solid symbols: source points (cell centres); open symbols: target points (dummy cells))

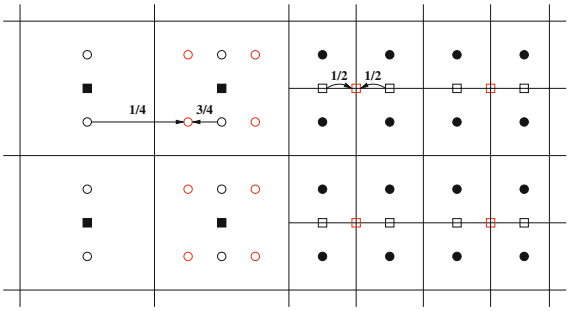


Fig. 4. Stencil to determine dummy-cell values using second-order interpolation both along and normal to interface (solid symbols: source points (cell centres); open black symbols: intermediate points after interpolation along interface; open red symbols: target points (dummy cells))

in figure 4. Note that this scheme is now asymmetric, requiring four instead of two rows of source cells from the fine grid.

This generic algorithm can be readily extended to fourth-order accuracy by increasing the order of the interpolation from second to fourth. This entails a larger interpolation stencil. Furthermore, three rows of dummy cells are needed for the fourth-order finite-volume method. Again, the first option consists of interpolating along the interface only, as shown in figure 5, maintaining the symmetry of the scheme normal to the interface. Now, three rows of source cells are required on each side of the interface. The second option, consisting of subsequently interpolating in normal direction as well, is shown in figure 6. This algorithm requires seven rows of source cells of the fine grid. Thus, the fine-grid block is required to have a minimum dimension of seven cells.

The second option is more accurate than the first, because it takes the jump in mesh size in normal direction into account. It is less robust, however, due to the loss of symmetry. This reduced robustness is felt the strongest when a

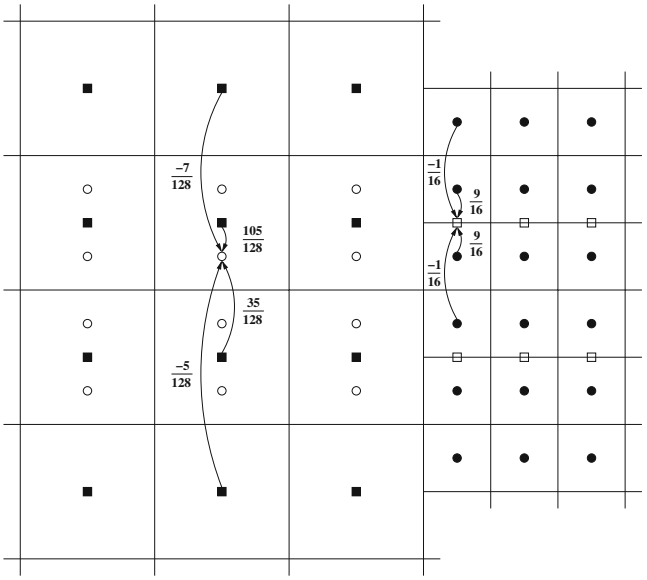


Fig. 5. Stencil to determine dummy-cell values using fourth-order interpolation along interface only (solid symbols: source points (cell centres); open symbols: target points (dummy cells))

multi-grid scheme is applied to solve the non-linear system of equations (per time step). Therefore, in the multi-grid scheme, the first option is applied on the coarse grid levels and the second option on the finest level only. This has as additional advantage that on the coarse levels, the blocks have a minimum grid size of only three cells instead of seven cells.

4.2 Verification

In order to test the capability of the high-order finite-volume scheme to accurately capture vortices without significant dissipation or dispersion on locally refined grids, the convection of a 2D isentropic vortex in a uniform flow is considered. This test case has been used to verify to high accuracy of the low-dispersion symmetry-preserving fourth-order finite-volume method on curvilinear grids [7]. Computations are performed on a 10×10 multi-block grid with local grid refinement in a checker-board pattern, as illustrated in figure 7. This figure also shows the initial the downstream boundary. The number of cells in the coarse blocks equals 20×20 and in the fine blocks 40×40. The equations are integrated in time by a low-storage 4-stage Runge–Kutta scheme.

The interpolation algorithm at partially continuous grid interfaces is verified for the fourth-order finite-volume method. Figure 8 shows the final temperature field of the vortex with three approaches: using second-order interpolation along the interface only, using fourth-order interpolation along the interface only, and

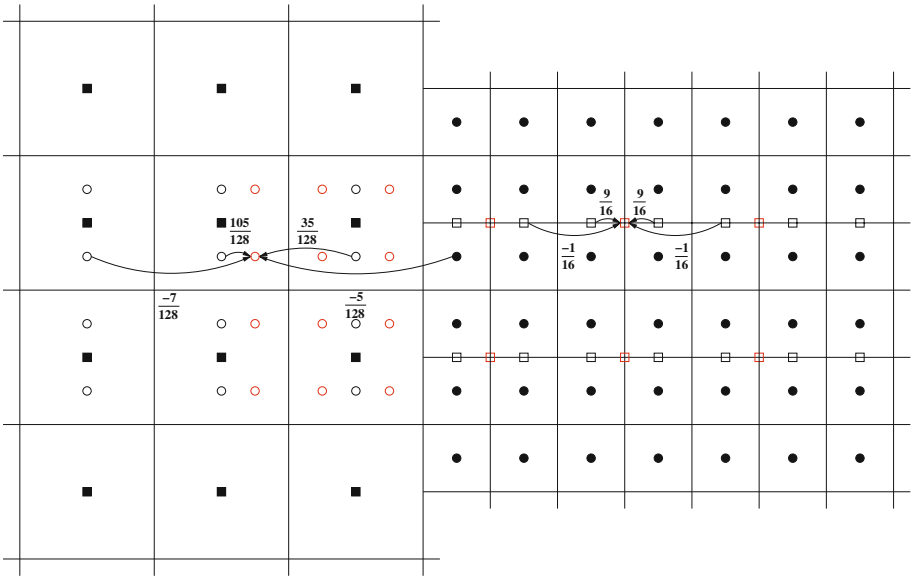


Fig. 6. Stencil to determine dummy-cell values using fourth-order interpolation both along and normal to interface (solid symbols: source points (cell centres); open black symbols: intermediate points after interpolation along interface; open red symbols: target points (dummy cells))

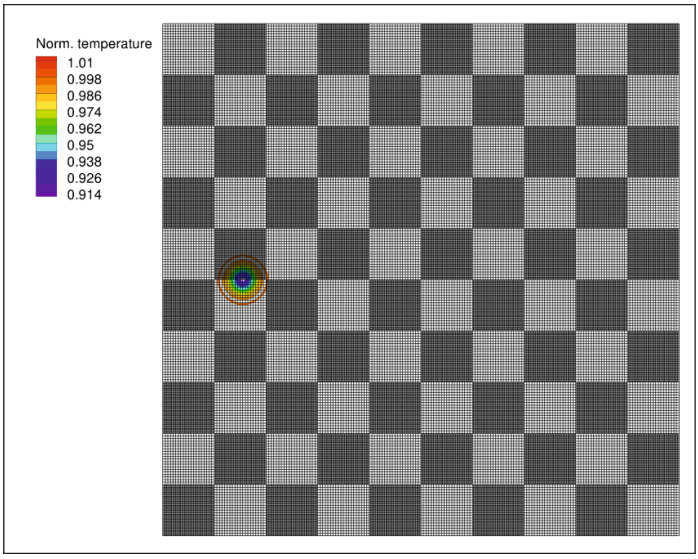


Fig. 7. Initial temperature field of strong isentropic vortex on locally refined grid

using fourth-order interpolation both along and normal to the interface. First of all, for all results the vortex is located in the correct position and only small differences with the analytic solution can be seen. This was expected, as the fourth-order finite-volume method was particularly designed to have low dispersion and dissipation in order to accurately capture vortices (and waves). The differences between the first two approaches (with interpolation only along the interface) are only small, both showing some small disturbances compared to the analytic solution and a weak increase of the temperature in the vortex core. The highest accuracy is obtained when the fourth-order interpolation is applied both along and normal to the interface. In that case, the solution is practically identical to the analytic solution (the differences in the vortex core essentially due to the limited accuracy of the plotting program). Note that these computations are stable with a very low level of sixth-order artificial diffusion ($k^{(6)} = \frac{1}{8}$).

5 Applications

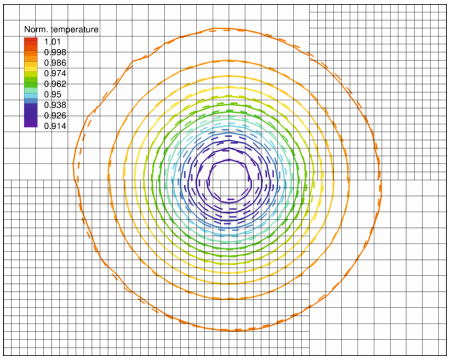
5.1 HART II Rotor

As a first application, simulations are performed for an isolated four-bladed rotor in one of the flow conditions of the HART-II experiment. The HART-II experiment is described in detail in Van der Wall et al. [16]. The chosen flow condition is a slow descent flight where multiple BVI events take place during a rotor revolution. The rotor radius of the windtunnel model is 2 meters. Rotational frequency is 109.12 rad/s. Forward speed of the rotor is 32.9 m/s. Effective shaft angle is 4.5 degrees (tilted backward).

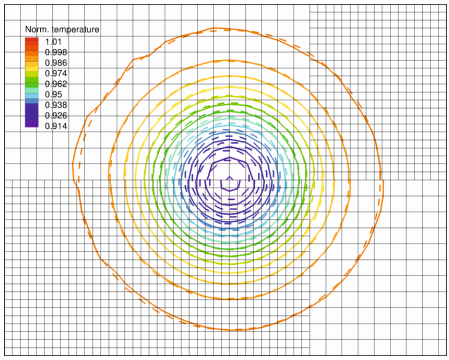
The turbulent flow is modeled using the Reynolds-averaged Navier-Stokes equations with the TNT k-ω turbulence model [6]. All simulations have been run using the fourth-order accurate finite volume scheme. Simulations have been performed on the original mesh of 13 million elements, and the locally refined mesh of 26 million elements. The discretization algorithm on locally refined meshes is described above, with second order interpolation at the partially continuous grid interfaces. The time step corresponds to 0.5° azimuth for all simulations.

For the simulation on the original mesh, rigid blades are considered. For the simulation on the refined mesh, both rigid and flexible blades are considered. In all simulations the rotor is trimmed to the experimental time-averaged forces. The aeroelastic simulation setup is described in detail in the chapter on aeroelastic test cases in this book [11], pages 649–660. In the current paper the vortex resolution of the two simulations is compared.

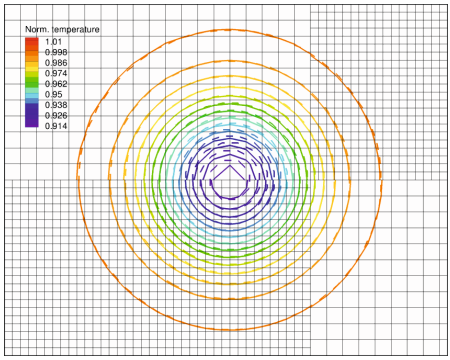
Figure 9a shows the instantaneous iso-contour of the Q-criterion for the original mesh. The Q-criterion distinguishes flow under shear stress from rotational flow, and is used here to focus on the tip vortices rather than on the boundary layer vorticity. The value of Q is equal to $0.72(U_\infty/R)^2$; the vorticity is scaled with $2U_\infty/R$. Figure 9b shows the instantaneous iso-contour of the Q-criterion for the refined mesh at the same values of Q and vorticity. The increase in vortex resolution is evident. Hence, the local grid refinement performs as expected.



(a) Second-order interpolation along interface only

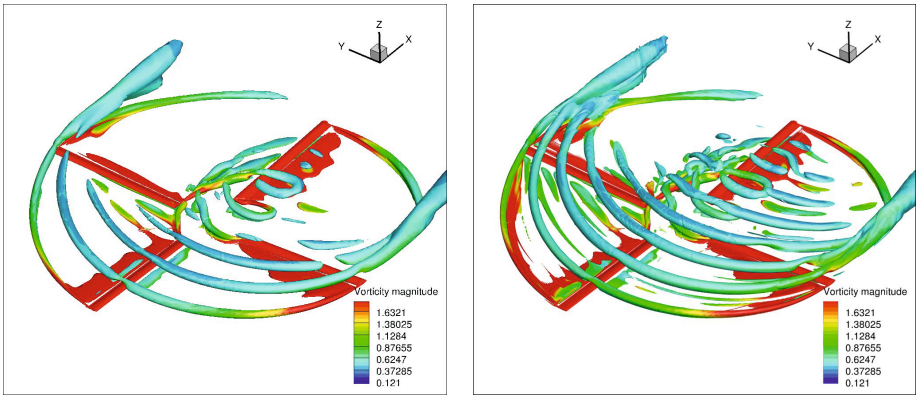


(b) Fourth-order interpolation along interface only



(c) Fourth-order interpolation along and normal to interface

Fig. 8. Convection of strong isentropic vortex on locally refined grid with fourth-order finite-volume method: final temperature field (dashed lines: analytic solution; solid lines: computed solution)



(a) Original mesh (b) Refined mesh

Fig. 9. Instantaneous iso-contour of the Q-criterion (blade at 0° azimuth) coloured with vorticity magnitude. Advancing side of the rotor is on the top-left side of the figures (positive y), retreating on the bottom-right side (negative y).

Continuing with the simulation using flexible blades, the instantaneous iso-contour of the Q-criterion is shown on the refined mesh for the trimmed solution in Figure 10. Compared to the simulation on the original grid, the lift of the blades on the advancing side (azimuth angle between 90 and 180 degrees) is smaller. Hence the strength of the tip vortex shed in this quadrant is less. In order to visualise the vortex system the value of Q is reduced by a factor of two with respect to the previous figures.

5.2 Results for Bump in Square Duct

As a second application, the turbulent, separated flow over a rounded bump in a square duct is considered. Experiments have been performed within the EU-project DESider by ONERA [1] in a hydraulic channel. The duct has a height of 0.3 m, a width of 0.5 m, and a length of 2.367 m. The bump has a height of 0.138 m, starts at the inflow plane at x = -0.367 m, and ends at x = 0 m. At the entrance, velocity profiles from the experiment are prescribed, which have a centre velocity of approximately 7 m/s. Furthermore, water with a density of 997 kg/m^3 and a dynamic viscosity of $0.89{\cdot}10^{-3}$ Pa·s is considered. As a compressible flow solver is used for solving this incompressible flow, an inflow Mach number of $M = 0.1$ is chosen.

Computations have been performed on a baseline grid of $142 \times 60 \times 76 = 647{,}520$ cells and on a grid with LGR (local grid refinement) with 355,560 cells, see figure 11. To be more precise, the latter grid actually contains local grid coarsening: the grid is coarsened by a factor 2 in all directions outside the flow separation region where the computation is in LES mode (i.e., outside the red surface in figure 11b), saving about 45% of grid points. A time step has been

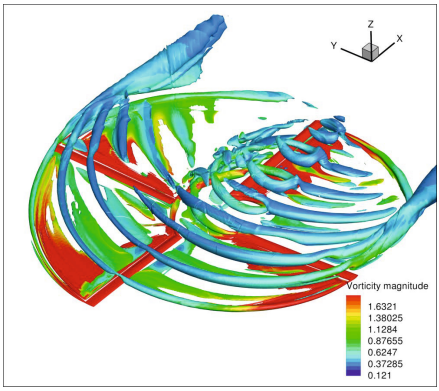


Fig. 10. Instantaneous iso-contour of the Q-criterion (blade at 0° azimuth) coloured with vorticity magnitude for the refined mesh and flexible blades. Advancing side of the rotor is on the top-left side of the figure (positive y), retreating on the bottom-right side (negative y).

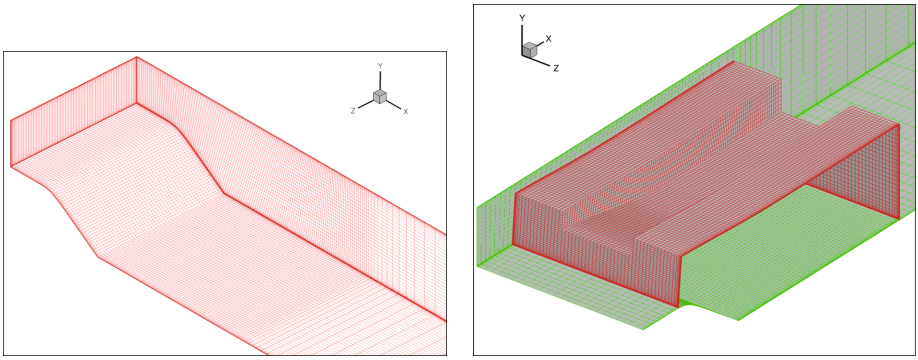
used equal to $\Delta t = 1.89{\cdot}10^{-4}$ s. A total of 50,000 time steps has been taken and flow statistics have been computed over the last 45,000 steps.

Figure 12 gives an instantaneous impression of the flow in terms of the Q-criterion, revealing the vortical structures in the separated flow region. For the grid with LGR, the figure shows that the separated flow is mostly contained within the fine-grid region, although occasionally a vortex may escape, in particular in the centre of the tunnel.

The mean pressure coefficient along the bottom wall is shown for three stations in figure 13. Compared to the experiment, the separation region is too small: the pressure recovers upstream of the experiment. This finding is consistent with the DES results from the DESider project [5]. More important here is the comparison between the computations on the baseline (BSL) grid and the grid with LGR. They are found to lie close together, with the LGR computation having a slightly smaller separation region.

The resolved turbulence level is compared on both grids in terms of the RMS value of the x-component of velocity (figure 15). The level of resolved turbulence in the shear layer is slightly lower on the grid with LGR. This may be related to the slightly smaller separation region.

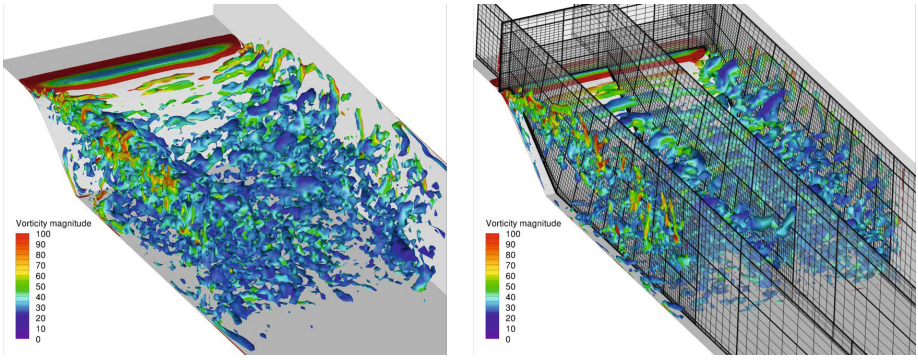
A possible explanation for the differences between the solutions on the baseline grid and the grid with LGR is that the refined grid region has been defined too narrow around the separated flow. Although for the time-averaged flow solution, the separation region is well within the refined region, this is not the case instantaneously. In particular, at the downstream boundary (large) vortical structure leave the refined region and are then suppressed on the coarser grid. Occasionally, this may also happen at the downstream part of the upper boundary of the refined region. This may reduce the level of resolved turbulence. Furthermore, at the upstream boundary, the incoming, attached boundary layer



(a) Baseline grid ($142 \times 60 \times 76$ cells)

(b) Grid with LGR: coarsened outside red surface, i.e., outside LES region

Fig. 11. Baseline grid and grid with LGR for bump in square duct



(a) Baseline grid

(b) Grid with LGR

Fig. 12. Instantaneous isocontours of Q-criterion, coloured with vorticity magnitude

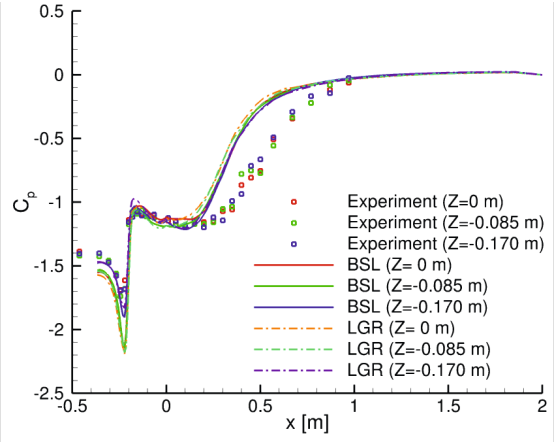


Fig. 13. Mean pressure coefficient along bottom wall

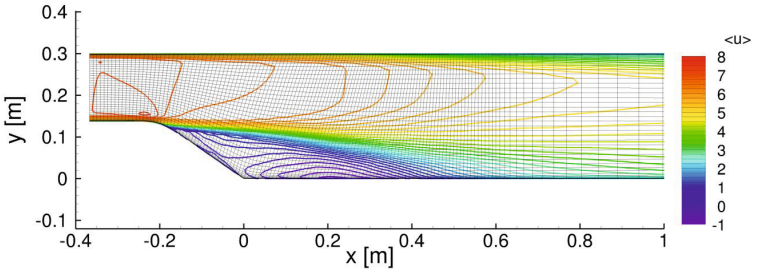
is also represented on a coarser grid and this may affect the initial development of the shear layer after separation.

5.3 Results for Delta Wing at High Angle of Attach

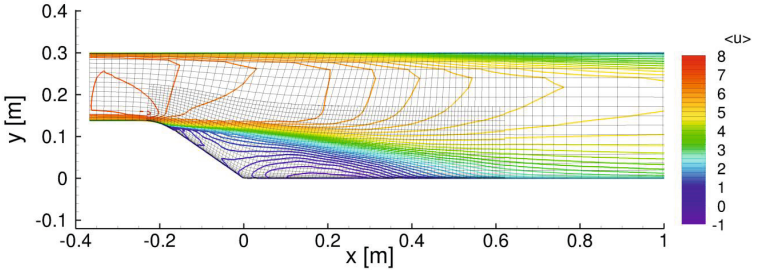
As a third application, the flow around a delta wing with a sharp leading edge at high angle of attack and high Reynolds number is considered. This flow is characterized by the main vortex developing above the wing. The vortex is formed as the shear layer emanating from the leading edge rolls up, starting immediately at the apex. At high Reynolds numbers, the shear layer rapidly becomes unstable and a turbulent vortex is formed. At a sufficiently high angle of attack, the vortex breaks down: the high axial velocity in the vortex core drops rapidly to a value close to zero.

The NASA delta wing geometry of Chu and Luckring [2] is considered, for which experiments that include measurements of velocity fluctuations have been performed by Furman and Breitsamter [3, 4]. Vortex breakdown occurs at the flow conditions $M = 0.07$, $Re_{\text{mac}} = 1{\cdot}10^6$, and $\alpha = 23°$ (with the Reynolds number based on the mean aerodynamic chord c_{mac}).

A baseline multi-block structured grid has been generated (within the ATAAC project), consisting of 22 blocks and 6.3 million grid cells (figure 16a). The grid has a conical structure over a large part of the wing: the grid covering the main vortex is essentially isotropic at each chordwise station (outside the boundary layer) and the mesh width grows in all directions (including the streamwise direction) together with the main vortex, going from approximately $0.003c_{\text{mac}}$ to $0.011c_{\text{mac}}$. In other words, the grid resolution relative to the main vortex is kept constant. Only in a small region near the apex, the conical structure is not fully maintained, avoiding a grid singularity. The far-field boundary is located at three root chord lengths from the wing. To study grid sensitivity, also a finer

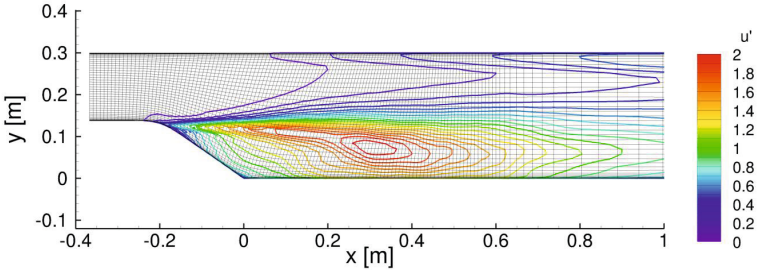


(a) Baseline grid

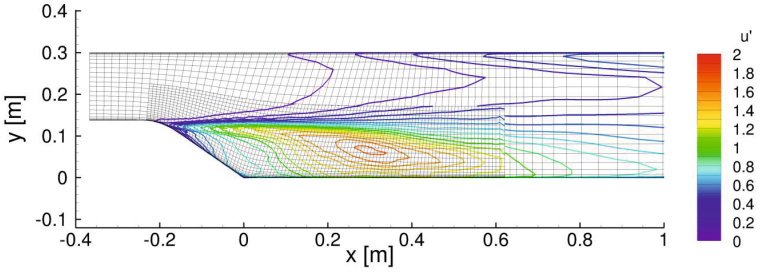


(b) Grid with LGR

Fig. 14. Mean x-component of velocity in plane $z = 0$

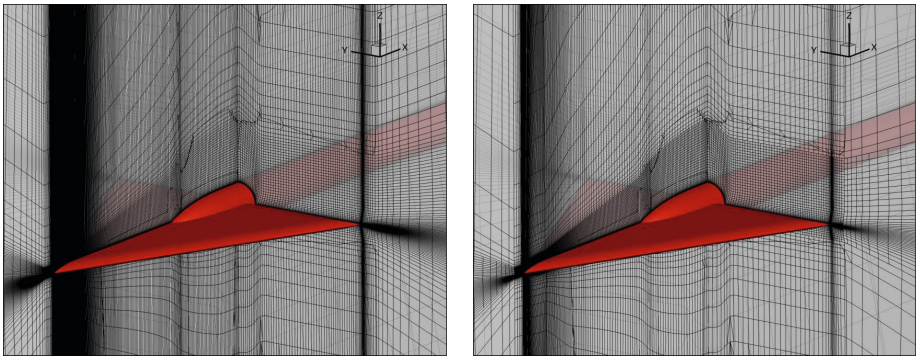


(a) Baseline grid



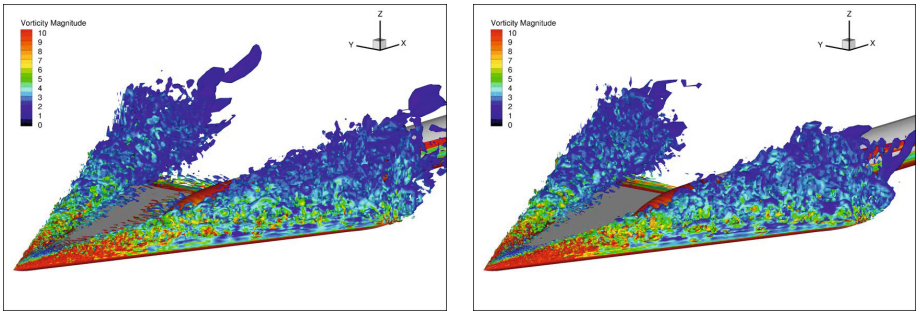
(b) Grid with LGR

Fig. 15. RMS of x-component of velocity in plane $z = 0$



(a) Baseline grid (b) Grid with LGR

Fig. 16. Grids for VFE2 delta wing



(a) Baseline grid (b) Grid with LGR

Fig. 17. Instantaneous isocontours of Q-criterion, coloured with vorticity magnitude

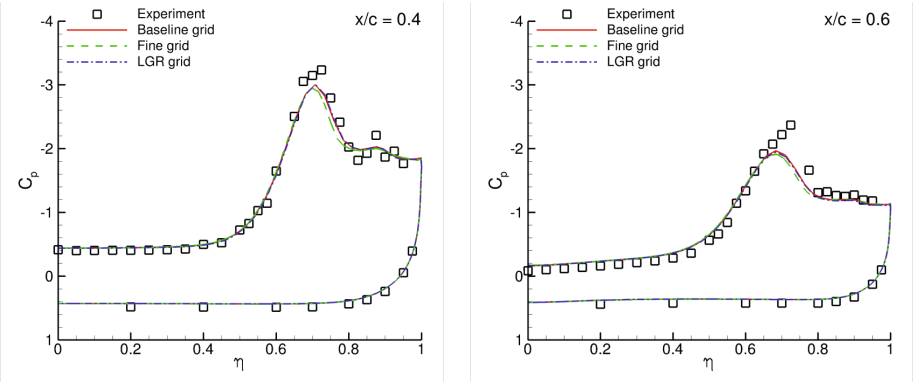
grid with the mesh width reduced by a factor 2/3 in all directions (21.4 million grid cells) has been generated.

As for the previous test case, an LGR grid with local coarsening is created (figure 16b), containing 3.7 million grid cells (reduction of 41%). The refined grid region contains the complete wing, including the boundary layers, as well as the main vortices above the wing.

X-LES computations have been performed with a time step of $3.75{\cdot}10^{-4}$ CTU (convective time units: $c_{\mathrm{mac}}/u_{\infty}$) on the baseline and LGR grids and a time step of $2.5{\cdot}10^{-4}$ CTU on the fine grid. The flow statistics have been obtained by averaging over 21 CTU for the baseline grid, 18 CTU for the LGR grid, and 8 CTU for the fine grid (after transients of about 6 CTU).

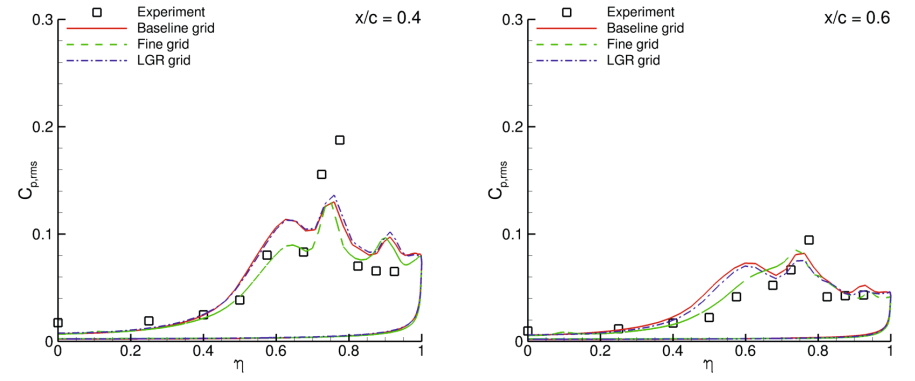
An impression of the instantaneous results is given in figure 17 in terms of isocontours of the Q-criterion. Similar fine-scale turbulent structures can be observed on the baseline and LGR grids.

The mean and RMS values of the pressure coefficient are given at two representative chordwise station in figures 18 and 19. For the mean values, there are



(a) $x/c = 0.40$ (b) $x/c = 0.60$

Fig. 18. Mean pressure coefficient at five chordwise stations



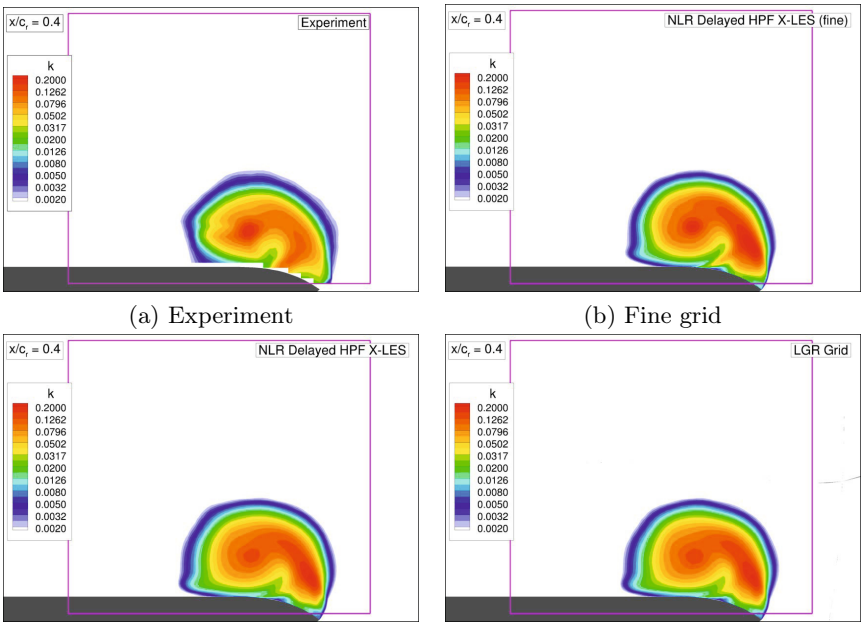
(a) $x/c = 0.40$ (b) $x/c = 0.60$

Fig. 19. RMS of pressure coefficient at five chordwise stations

no differences visible between the baseline and LGR grids, while there are only small differences with the fine grid. Larger, but still relatively small, differences are seen for the RMS values.

Finally, a comparison of the level of resolved turbulence is shown in figure 20 in terms of the resolved turbulent kinetic energy at a single station (40% root chord). Practically the same levels of resolved turbulence are obtained on the baseline and LGR grids, and also similar levels are seen on the fine grid and in the experiment.

Thus, the impact of coarsening the grid away from the region of interest is found to be small for this case; smaller than for the previous case. The main reason is that it is easier to fully encompass the region of interest (i.e., the separated flow captured with LES) in the refined grid region for an external flow than for the internal flow of the previous case.



(a) Experiment (b) Fine grid

(c) Baseline grid (d) Grid with LGR

Fig. 20. Resolved turbulent kinetic energy at chordwise station $x/c = 0.4$

6 Conclusions

An automatic topology-based grid refinement algorithm has been developed and demonstrated with the specific aim of obtaining block-structured meshes with uniform resolution in user-specified subdomains.

A generic algorithm has been developed to deal with partially continuous grids in a fourth-order finite-volume method. The stability and accuracy of the fourth-order finite-volume method have been verified for the convection of an isentropic vortex on a locally refined grid. The highest accuracy is obtained when the dummy-cell values are computed by fourth-order interpolation both along and normal to the partially continuous grid interfaces.

The simulation of an isolated helicopter rotor in forward flight clearly demonstrated improved vortex resolution on the locally refined mesh. This is essential for capturing blade-vortex interaction.

X-LES computations were performed for two tests cases on grids using local grid refinement (LGR). Compared to the baseline grids, the LGR grids were coarsened away from the separated flow regions that were captured with LES, saving 40 to 45% of the grid points. For the bump in a square duct, the results on the LGR grid showed a slightly smaller separation region than on the baseline grid. Possibly, the refined grid region was chosen too tightly around the region of interest. For the delta wing, no significant differences were found between the LGR and baseline grids. Thus, local grid refinement can be used to reduce the

computational costs of X-LES computations, provided the grid is coarsened well outside the region of interest, which is easier to achieve for external flows than for internal flows.

References

1. Aupoix, B.: The DESider bump experiment for hybrid modelling validation. In: EWA Workshop, Prague, June 5-6 (2007)
2. Chu, J., Luckring, J.M.: Experimental surface pressure data obtained on 65° delta wing across Reynolds number and Mach number ranges. TM 4645, NASA (1996)
3. Furman, A., Breitsamter, C.: Turbulent and unsteady flow characteristics of delta wing vortex systems. In: 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, January 7-10. AIAA 2008-381 (2008)
4. Furman, A., Breitsamter, C.: Experimental investigations on the VFE-2 configuration at TU Munich. In: Understanding and Modeling Vortical Flows to Improve the Technology Readiness Level for Military Aircraft, ch. 21, NATO RTO, 2009. RTO-TR-AVT-113 (2009)
5. Haase, W., Braza, M., Revell, A.: DESider – A European Effort on Hybrid RANS-LES Modelling. NNFM, vol. 103. Springer, Heidelberg (2009)
6. Kok, J.C.: Resolving the dependence on freestream values for the k–ω turbulence model. AIAA Journal 38(7), 1292–1294 (2000)
7. Kok, J.C.: A high-order low-dispersion symmetry-preserving finite-volume method for compressible flow on curvilinear grids. Journal of Computational Physics 228, 6811–6832 (2008) NLR-TP-2008-775
8. Kok, J.C., Dol, H.S., Oskam, B., van der Ven, H.: Extra-large eddy simulation of massively separated flows. In: 42nd AIAA Aerospace Sciences Meeting, Reno, NV, January 5-8. AIAA paper 2004-264 (2004)
9. Kok, J.C., van der Ven, H.: Destabilizing free shear layers in X-LES using a stochastic subgrid-scale model. In: Peng, S.-H., Doerffer, P., Haase, W. (eds.) Progress in Hybrid RANS-LES Modelling. NNFM, vol. 111, pp. 179–189. Springer, Heidelberg (2010)
10. Kok, J.C., van der Ven, H.: Capturing free shear layers in hybrid RANS–LES simulations of separated flow. In: Third Symposium Simulation of Wing and Nacelle Stall, Braunschweig, Germany, June 21-22 (2012) NLR-TP-2012-333
11. Kotecki, K., van der Ven, H., Kok, J., Hausa, H., Nowak, M., Stankiewicz, W., Roszak, R., Morzyński, M.: Aeroelastic testcases in IDIHOM project. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach. NNFM, vol. 128, pp. 649–660. Springer, Heidelberg (2015)
12. Spalart, P.R.: Detached-eddy simulation. Annual Review of Fluid Mechanics 41, 181–202 (2009)
13. Tam, C.K.W., Webb, J.C.: Dispersion-relation-preserving finite difference schemes for computational acoustics. Journal of Computational Physics 107, 262–281 (1993)
14. Travin, A.K., Shur, M.L., Spalart, P.R., Strelets, M.K.: Improvement of delayed detached-eddy simulation for LES with wall modelling. In: Wesseling, P., Oñate, E., Périaux, J. (eds.) ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands, September 5-8 (2006)

15. van der Ven, H., Boelens, O.J., Kok, J.C., Prananta, B.B., van Rooij, M.P.C.: Simulation of a rotor in forward flight using topology-based refinement of multi-block structured meshes. In: 38th European Rotorcraft Forum, Amsterdam, the Netherlands, September 4-6 (2012) NLR-TP-2012-485
16. van der Wall, B.G.: 2nd HCC aeroacoustics rotor test (HART II) Part I: Test documentation. Technical Report IB 111-2003/19, DLR (2003)
17. Verstappen, R.W.C.P., Veldman, A.E.P.: Symmetry-preserving discretization of turbulent flow. Journal of Computational Physics 187(1), 343–368 (2003)

Aghora: A High-Order DG Solver for Turbulent Flow Simulations

F. Renac, M. de la Llave Plata, E. Martin,
J.-B. Chapelier, and V. Couaillier

ONERA The French Aerospace Lab, 92320 Châtillon Cedex, France

Abstract. This paper presents details of the solver Aghora for the simulation of unsteady compressible turbulent flows. Different modelling levels are used: Reynolds averaged Navier-Stokes equations coupled with turbulence transport equations, variational multi-scale formulation of large-eddy simulation, and direct numerical simulation. The space discretization is based on a high-order discontinuous Galerkin method with representation of curved boundaries. High-order explicit and implicit Runge-Kutta methods are used for the time integration. The performance of the solver will be assessed in various examples of compressible turbulent flow numerical simulation in three space dimensions.

Keywords: discontinuous Galerkin method, compressible flows, turbulent flows, direct numerical simulation, large-eddy simulation, variational multiscale method.

1 Introduction

Discontinuous Galerkin (DG) methods are high-order finite element discretizations which were introduced in the early 1970's for the numerical simulation of the first-order hyperbolic neutron transport equation [34, 41]. In recent years, these methods have become very popular for the solution of nonlinear convection dominated flow problems [16, 17, 32]. The success of these methods lies in their high-order of accuracy and flexibility thanks to their high degree of locality. These properties make the DG method well suited to parallel computing [7], *hp*-refinement [24], *p*-multigrid [38], unstructured meshes [1], the application of boundary conditions, *etc.*

However, high-order DG methods also suffer from a number of drawbacks which make their application to realistic engineering problems a challenging task: robustness issues due to spurious oscillations in the vicinity of discontinuities associated to Gibbs phenomenon [20]; time step restriction due to the so-called Courant-Friedrichs-Levy (CFL) condition for stability of the numerical scheme [3, 15]; high computational cost and large memory requirements induced by the large number of degrees of freedom (DOFs) in practical application; *etc.* The European project IDIHOM aims at providing practical solutions to these shortcomings and demonstrating the capabilities of general high-order methods to solve a wide range of complex industrial flows. In this context, this paper

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_15

summarizes the contribution from Onera and demonstrates the capability of the Aghora DG code to deal with a number of challenging applications.

The paper is organized as follows. Section 2 presents the model problem and the numerical approach for the space-time discretization used in the Aghora code. Domain decomposition strategies are introduced in § 2.5. The overall performance of the method is assessed by several numerical experiments in § 3. Results are obtained for 3D steady and unsteady flow problems by solving the compressible Navier-Stokes (N-S) equations with Reynolds averaged Navier-Stokes (RANS), large-eddy simulation (LES) or direct numerical simulation (DNS) modelling. Finally, the conclusions of this work are summarized in § 4.

2 Model Problem and Discretization

2.1 Compressible Navier-Stokes Equations

The discussion in this paper focuses on the discretization of the compressible N-S equations for gas dynamics with a DG method. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain where $d = 3$ is the space dimension and consider the following problem

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{f}_c(\mathbf{u}) - \nabla \cdot \mathbf{f}_v(\mathbf{u}, \nabla \mathbf{u}) = 0, \qquad \text{in } \Omega \times (0, \infty), \tag{1a}$$

$$\mathbf{u}(\cdot, 0) = \mathbf{u}_0(\cdot), \quad \text{in } \Omega, \tag{1b}$$

with appropriate boundary conditions prescribed on $\partial\Omega$. The vector

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix} \tag{2}$$

represents the conservative variables with ρ the density, $\mathbf{v}$ in $\mathbb{R}^d$ the velocity vector and $E = \mathrm{p}/(\gamma - 1)\rho + \|\mathbf{v}\|_2^2/2$ the total specific energy where p denotes the static pressure, $\gamma = C_p/C_v > 1$ is the ratio of specific heats, and $\|\cdot\|_2$ denotes the Euclidean norm. The nonlinear convective and diffusive fluxes in (1) are defined by

$$\mathbf{f}_c(\mathbf{u}) = \begin{pmatrix} \rho \mathbf{v}^\top \\ \rho \mathbf{v}\mathbf{v}^\top + \mathrm{p}\mathbf{I} \\ (\rho E + \mathrm{p})\mathbf{v}^\top \end{pmatrix}, \quad \mathbf{f}_v(\mathbf{u}, \nabla \mathbf{u}) = \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \mathbf{v}^\top \boldsymbol{\tau} - \mathbf{q}^\top \end{pmatrix}, \tag{3}$$

where $\boldsymbol{\tau} = \mu(-\frac{2}{3}(\nabla \cdot \mathbf{v})\mathbf{I} + \nabla \mathbf{v} + \nabla \mathbf{v}^\top)$ denotes the viscous stress tensor with μ the kinematic viscosity coefficient defined by the Sutherland's law, and $\mathbf{q} = -k \nabla T$ is the Fourier's heat conduction law with $k = \mu C_p / Pr$ the thermal conductivity, $Pr = 0.72$ the Prandtl number and T the temperature.

In this work, we will also consider the compressible RANS equations coupled with two different turbulence models. Both the two-equation $k - \omega$ turbulence model of Wilcox [51] and the one-equation turbulence model of Spalart-Allmaras [8] are implemented in Aghora. Here, we only focus on the one-equation

turbulence model of Spalart-Allmaras. In the RANS formulation, the transport equations (2) are time-averaged and the viscous stress tensor and heat flux vector are supplemented with the Reynolds stress tensor and turbulent heat flux vector. Using the Boussinesq assumption, we obtain

$$\boldsymbol{\tau} = (\mu + \mu_t)\Big(-\frac{2}{3}(\nabla \cdot \mathbf{v})\mathbf{I} + \nabla \mathbf{v} + \nabla \mathbf{v}^\top \Big),$$
$$\mathbf{q} = -\Big(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\Big) C_p \nabla T,$$

where $Pr_t = 0.9$ is the turbulent Prandtl number and $\mu_t = \rho\tilde{\nu} f_{v1}$ represents the turbulent dynamic viscosity coefficient with

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\rho\tilde{\nu}}{\mu}.$$

The working variable $\tilde{\nu}$ is governed by

$$\partial_t \rho\tilde{\nu} + \nabla \cdot (\rho\tilde{\nu}\mathbf{v}) - \frac{1}{\sigma}\nabla \cdot \big((\mu + \rho\tilde{\nu})\nabla\tilde{\nu}\big) = c_{b1}\tilde{S}\tilde{\nu} + \frac{c_{b2}\rho}{\sigma}\nabla\tilde{\nu} \cdot \nabla\tilde{\nu} - c_{w1} f_w \frac{\rho\tilde{\nu}^2}{\eta^2}. \quad (4)$$

Here $\tilde{S} = \|\nabla \times \mathbf{v}\|_2 + \overline{S}$, $\overline{S} = \tilde{\nu} f_{v2}/\kappa^2\eta^2$, η is the distance to the nearest wall, and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g\Big(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\Big)^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2\eta^2}.$$

The reader is referred to [8] for details on the coefficients and values of the constants involved in these expressions.

2.2 The DG-VMS Approach for Large-Eddy Simulation

Due to the wide range of length and time scales involved in turbulence, direct numerical simulation of turbulent flows is computationally very expensive, and is restricted to low-to-moderate Reynolds numbers. The LES technique has been established as an alternative approach to DNS for high-Reynolds-number flows. The ultimate goal of LES is to compute explicitly the large-scale motion, while modelling the effect of the small scales on the coarse resolved field.

In this context, the DG method appears to have a number of advantages over other approaches, such as the finite-difference (FD) and the finite-volume (FV) methods, for the solution of problems of practical interest, which often involve complex geometries and require the specification of realistic boundary conditions.

The high order of accuracy provided by DG discretizations leads to an important reduction of the numerical dissipation and dispersion errors when the

polynomial degree is sufficiently high, as shown by several authors (see e.g. Ainsworth [2]). This property is of major importance in the simulation of unsteady turbulence by means of DNS and LES, for which the presence of numerical errors can have a detrimental effect on the solution [32].

Another attractive aspect of the DG method is the possibility to enforce boundary conditions weakly, which was shown by Collis [18] to improve the quality of the solution. Besides, the order of accuracy of the numerical scheme is conserved at the boundary, in contrast to most FD and FV methods in which the order of the numerical scheme has to be reduced when approaching the boundary.

In order to obtain accurate solutions of the LES equations, it is essential to develop modelling approaches able to capture the main features of the interscale dynamics. Multilevel LES methods accomplish this by simulating the most relevant part of these dynamics explicitly in the computational domain. They rely on the decomposition of the flow into different frequency bands, each associated with a particular range of length scales.

The Variational Multiscale (VMS) method was first proposed by Hughes in 1995 [25,26], in the general context of computational mechanics. The application of VMS to the problem of turbulence was published a little later [27,28].

It is based on variational projection instead of spatial filtering, as is done in traditional LES. In VMS, the exact solution of the N-S equations, $\mathbf{u}$, is decomposed into a large-scale, $\overline{\mathbf{u}}$, a small-scale, $\widetilde{\mathbf{u}}$, and an unresolved component, $\mathbf{u}'$, such that $\mathbf{u} = \overline{\mathbf{u}} + \widetilde{\mathbf{u}} + \mathbf{u}'$. The large and small scales combine to provide the resolved scales, denoted by $\overline{\widetilde{\mathbf{u}}}$. The solution space is thus partitioned as

$$\mathcal{V} = \overline{\mathcal{V}} \oplus \widetilde{\mathcal{V}} \oplus \mathcal{V}', \tag{5}$$

and the weighting function space as

$$\mathcal{W} = \overline{\mathcal{W}} \oplus \widetilde{\mathcal{W}} \oplus \mathcal{W}'. \tag{6}$$

The scale separation is achieved by means of a variational or Galerkin projection of the N-S equations onto the solution space. This results in a system of equations for the large and small scales in the space-time domain Ω, which can be formally written as

$$V\left(\phi, \overline{\widetilde{\mathbf{u}}}\right)_\Omega = \overline{\mathbf{M}}\left(\overline{\phi}, \overline{\widetilde{\mathbf{u}}}\right)_\Omega + \widetilde{\mathbf{M}}\left(\widetilde{\phi}, \overline{\widetilde{\mathbf{u}}}\right)_\Omega, \quad \forall \phi \in \mathcal{W}. \tag{7}$$

where $V\left(\phi, \overline{\widetilde{\mathbf{u}}}\right)_\Omega$ denotes a variational form of the N-S equations with weighting functions $\phi = \overline{\phi} + \widetilde{\phi} + \phi'$. On the right-hand side of (7), $\overline{\mathbf{M}}$ and $\widetilde{\mathbf{M}}$ represent the model terms acting on the large and small scales, respectively.

In general $\overline{\mathbf{M}} = 0$, so that no direct modelling is involved in the large-scale equations, while the small-scale equations usually entail very simple models such as the Smagorinsky model [44] or the structure function model [36]. This choice is based on the assumption that the spectral gap between the large and the unresolved scales is wide enough so that the effects of the latter on the former may be neglected.

Few attempts to combine the VMS approach with a DG discretization have been reported in the literature. Collis et al. [18] and Ramakrishnan et al. [40] are among the first authors to use a DG-VMS approach to perform LES of the plane channel flow configuration. In § 3.5, we present first results of the application of a modal DG-VMS approach to the LES of the Taylor-Green vortex at $Re = 3000$.

2.3 Discontinuous Galerkin Formulation

The DG method consists in defining a discrete weak formulation of the problem (1). The domain Ω is partitioned into a shape-regular mesh Ω_h consisting of non-overlapping and nonempty elements κ of characteristic size h. We further define the sets $\mathcal{E}_i$ and $\mathcal{E}_b$ of interior and boundary faces in Ω_h and set $\mathcal{E}_h = \mathcal{E}_i \cup \mathcal{E}_b$.

We look for approximate solutions in the function space of piecewise polynomials $\mathcal{V}_h^p = \{\phi \in L^2(\Omega_h) : \ \phi|_\kappa \circ F_\kappa \in \mathcal{P}^p(I^d), \ \forall \kappa \in \Omega_h\}$, where $\mathcal{P}^p(I^d)$ denotes the space of functions formed by tensor products of polynomials and of either total or partial degree at most p on the master element I^d where $I = [-1, 1]$. Each physical element κ is the image of I^d through the mapping F_κ. A Gram-Schmidt orthonormalization procedure is used to construct the orthonormal basis from an initial monomial basis in order to get a diagonal mass matrix for the explicit time integration. We refer to [5, 42] for details on this procedure. The numerical integration is either performed in I^d, or by using optimal quadratures specific to each type of element [45].

The numerical solution of equation (1) or of equations (1) and (4) is sought under the form

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{l=1}^{N_p} \phi_\kappa^l(\mathbf{x}) \mathbf{U}_\kappa^l(t), \quad \forall \mathbf{x} \in \kappa, \ \kappa \in \Omega_h, \ \forall t \geq 0, \tag{8}$$

where $(\mathbf{U}_\kappa^l)_{1 \leq l \leq N_p}$ are the DOFs in the element κ. The subset $(\phi_\kappa^1, \ldots, \phi_\kappa^{N_p})$ constitutes a hierarchical and orthonormal modal basis of $\mathcal{V}_h^p$ restricted onto the element κ and N_p is its dimension.

The semi-discrete form of system (1a) then reads: find $\mathbf{u}_h$ in $[\mathcal{V}_h^p]^{d+2}$ such that for all v_h in $\mathcal{V}_h^p$ we have

$$\int_{\Omega_h} v_h \partial_t \mathbf{u}_h dV + \mathcal{L}_c(\mathbf{u}_h, v_h) + \mathcal{L}_v(\mathbf{u}_h, v_h) = 0. \tag{9}$$

Hereafter, we will use the notations $\{\phi\} = (\phi^+ + \phi^-)/2$ and $[\![\phi]\!] = \phi^+ - \phi^-$ which denote the average and jump operators defined for a given interface e in $\mathcal{E}_i$. Here, ϕ^+ and ϕ^- are the traces of any quantity ϕ on the interface e taken from within the interior of the element κ^+ and the interior of the neighboring element κ^-, respectively (see Fig. 1).

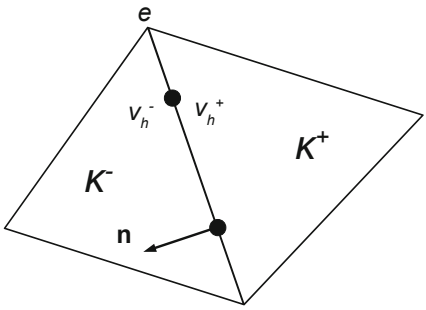


Fig. 1. Inner and exterior elements κ^+ and κ^- and definition of traces $v_h^\pm$ on the interface e and of the unit outward normal vector $\mathbf{n}$

The discretization of the convective terms in equation (9) reads

$$\begin{aligned}\mathcal{L}_c(\mathbf{u}_h, v_h) = &- \int_{\Omega_h} \mathbf{f}_c(\mathbf{u}_h) \cdot \nabla_h v_h dV \\ &+ \int_{\mathcal{E}_i} [\![v_h]\!] \mathbf{h}_c(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) dS \\ &+ \int_{\mathcal{E}_b} v_h^+ \mathbf{f}_c\big(\mathbf{u}_b(\mathbf{u}_h^+, \mathbf{n})\big) \cdot \mathbf{n} dS,\end{aligned}$$

where $\mathbf{n}$ denotes the unit outward normal vector to an element κ^+ (see Fig. 1) and $\mathbf{u}_b$ is an appropriate operator which allows to impose the boundary conditions on $\mathcal{E}_b$. The numerical flux $\mathbf{h}_c$ may be chosen to be any monotone Lipschitz continuous function satisfying consistency and conservativity properties (see [17] for instance). In numerical examples of § 3 we use the local Lax-Friedrichs flux $\mathbf{h}_c(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) = \{\mathbf{f}_c(\mathbf{u}_h)\} + \alpha [\![\mathbf{u}_h]\!]/2$ with $\alpha = \max\{\rho_s\big(\nabla_{\mathbf{u}}(\mathbf{f}_c(\mathbf{u}) \cdot \mathbf{n})\big) : \ \mathbf{u} = \mathbf{u}_h^\pm\}$ where $\nabla_{\mathbf{u}}(\mathbf{f}_c \cdot \mathbf{n})$ denotes the Jacobian matrix of the convective fluxes in the direction $\mathbf{n}$ and ρ_s its spectral radius.

For the discretization of the diffusive terms in (9), we use the BR2 method of Bassi and coworkers [11]:

$$\begin{aligned}\mathcal{L}_v(\mathbf{u}_h, v_h) = &\int_{\Omega_h} \mathbf{f}_v(\mathbf{u}_h, \nabla_h \mathbf{u}_h + \mathbf{R}_h) \cdot \nabla v_h dV \\ &- \int_{\mathcal{E}_i} [\![v_h]\!] \big\{\mathbf{f}_v\big(\mathbf{u}_h, \nabla_h \mathbf{u}_h + \eta_{BR2} \mathbf{r}_h^e\big)\big\} \cdot \mathbf{n} dS \\ &- \int_{\mathcal{E}_b} v_h^+ \mathbf{f}_v\big(\mathbf{u}_b(\mathbf{u}_h^+, \mathbf{n}), \nabla \mathbf{u}_b(\mathbf{u}_h^+, \mathbf{n}) + \eta_{BR2} \mathbf{r}_h^e\big) \cdot \mathbf{n} dS,\end{aligned}$$

where Neumann boundary conditions are enforced in the expression of $\mathbf{f}_v$ on $\mathcal{E}_b$ through the operator $\nabla \mathbf{u}_b(\mathbf{u}_h^+, \mathbf{n})$ and η_{BR2} is a user-defined parameter for stabilization of the method. The so-called global lifting operator $\mathbf{R}_h$ is defined in

a weak sense as the sum of local lifting operators $\mathbf{r}_h^e$: $\mathbf{R}_h = \sum_{e \in \mathcal{E}_h} \mathbf{r}_h^e$, where $\mathbf{r}_h^e$ has a support on the elements adjacent to e in $\mathcal{E}_i$ and results from the solution of the following problem defined on the internal faces: for all v_h in $\mathcal{V}_h^p$ we have

$$\int_{\kappa^+ \cup \kappa^-} v_h \mathbf{r}_h^e dV = -\int_e \{v_h\} [\![\mathbf{u}_h]\!] \mathbf{n}^\top dS.$$

A similar expression consistent with the boundary conditions of the problem is used on $\mathcal{E}_b$. We refer to [10, 11] for further details on this approach.

A similar method is used for the discretization of equation (4) for the turbulent variable where a Roe flux is applied as approximate Riemann solver and the BR2 method is used for the discretization of viscous fluxes and source terms.

Finally, in the case of non-smooth solutions, a shock-capturing term is added to the semi-discrete form of the equations (9). We use either the residual-based methods from [22, 23] or the entropy-based method from [21].

2.4 Time Discretization

The semi-discrete equation (9) is discretized in time by means of either explicit or implicit integration. Strong stability preserving Runge-Kutta schemes [46, 49] are used for the explicit integration in time. Implicit time integration is performed by either backward difference formulae or by explicit first stage, single diagonal coefficient, diagonally implicit Runge-Kutta (ESDIRK) schemes [6, 31]. The sparse structure of the implicit matrix and the large number of unknowns motivate the choice of a Krylov subspace method to solve the linear system at each time step. We use a restarted GMRES method [48] with block Jacobi or block ILU(0) preconditioners for its resolution. For computational efficiency, we use a Jacobian-free strategy where the matrix-vector product required by the GMRES algorithm is approximated by finite differences. Then, an approximate Jacobian based on mixed precision algorithm [4] is used as preconditioner to reduce the strong computational cost and memory requirement associated to a high-order DG method.

2.5 Hybrid Parallel Approaches

Domain decomposition methods based on an MPI strategy are usually applied to deal with the large CPU time and memory requirements of the DG method. However, such an approach may suffer from several limitations due to large number of DOFs per element, high receive-send message frequencies, or the synchronization problems at very large scale. The combination of two parallelization paradigms, MPI and OpenMP, has a number of benefits with respect to MPI-based strategies [29, 33, 35], such as memory savings, better load balancing, and coarser granularity for a sustained parallel efficiency at large scale. On top of

that, the flexibility of hybrid approaches leads to a better code adequacy to the target architecture hardware specificities, such as non-uniform memory access or heterogeneity (Nvidia GPU accelerators, Intel Xeon Phi coprocessors).

Relying on a classic domain decomposition method, a first parallel MPI strategy executes two global operations for each global time step calculation (MPI_Gather and MPI_Bcast). Five point-to-point operations are done per time step or Runge-Kutta stage during the computations of fluxes (MPI_Issend, MPI_Irecv, and MPI_Wait routines). Moreover, non-blocking synchronous send mode overlaps communications with calculations.

Two hybrid MPI/OpenMP strategies are considered in this work. On the one hand, a "Fine-Grained" approach consists in adding OpenMP work-sharing directives in a non-intrusive way at the level of loops. Nevertheless, this results in non-negligible extra-costs in elapsed time as the threads are dynamically created and destroyed in a fork-join pattern inside each loop. On the other hand, a "Coarse-Grained" approach generates threads only once at the beginning of a parallel region. The multithreading support level provided by the MPI library offers the OpenMP threads the capability to call MPI in various different ways. This last approach has been implemented into the Aghora code, where the parallel region succeeds now in encapsulating the whole iterative scheme. To achieve interesting performances, a particular attention should be paid not only to the workload sharing between threads, but also to the load-balancing and the synchronizations between threads and processes.

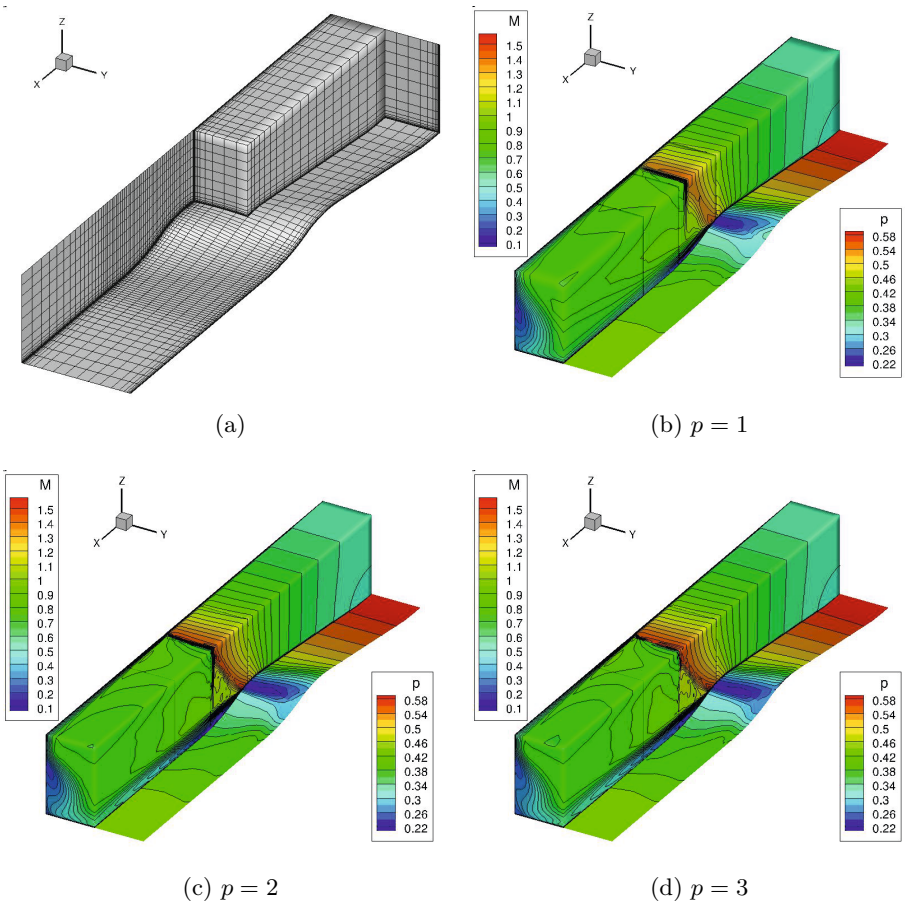
The loops on the elements and the faces of the mesh applied in the implementation of the space discretization, as well as the data copy into send buffers and from receive buffers, are parallelized using OpenMP. To avoid data race conditions, extra arrays per element are allocated in order to ensure reproductibility of the numerical results regardless of the number of threads.

3 Numerical Experiments

In this section we present numerical experiments for the N-S equations (1) and RANS equations (1),(4) in three-space dimensions to illustrate the performance of the method.

3.1 Three-Dimensional Transonic Channel

This test case has been first proposed in [12] for the analysis of shock wave boundary layer interaction and consists in the flow induced by a swept bump at the lower wall of a transonic channel (see Fig. 2a). Adiabatic wall conditions are specified on the bottom and side walls and a symmetry condition is imposed at the top of the channel. This latter boundary condition constitutes a slight modification compared to the experimental set-up. The inflow is subsonic with uniform distributions of total pressure $\mathrm{p}_{i,0}$ and total temperature $T_{i,0}$. At the outlet, the static pressure is set to $\mathrm{p}/\mathrm{p}_{i,0} = 0.64$. The Reynolds number based on the channel half-height and stagnation conditions is $Re = 1.69 \times 10^6$. In



(a) (b) $p = 1$

(c) $p = 2$ (d) $p = 3$

Fig. 2. Cut view and geometry definition of the 3D transonic channel (a); Mach number and wall pressure distributions (b-d)

this configuration, a shock wave appears in the divergent part of the duct and interacts with the fully turbulent boundary layers.

We use the RANS equations coupled with the Spalart-Allmaras turbulence model. The domain is discretized with a mesh consisting of $46,208$ quadratic hexahedra. The non-dimensional values of the first cell height is in the range $3 \leq y^+ \leq 4$ upstream of the flow separation (see Fig. 2a).

The Mach number contours and pressure distributions at the bottom wall are displayed in Figs. 2(b-d) for approximation orders $1 \leq p \leq 3$. The solution is plotted by subdividing each element into eight subelements with values of the solution reconstructed at the centres of the subelements. The results highlight the lambda shock structure and the large region of separated flow induced by the shock-boundary layer interaction. Note that the results obtained with

third- and fourth-order numerical schemes are qualitatively similar and clearly better compared to those obtained with the second-order scheme.

3.2 The NASA Rotor 37

The capability of performing a simulation in a rotating frame of reference has been implemented into the Aghora DG solver. By considering a relative-frame / relative-variable formulation, this involves the addition of rotational source terms to the momentum and energy equations as well as the ability to consider rotational periodicity and adapted boundary conditions. These turbomachinery-specific developments have been validated on a transonic axial compressor, the NASA Rotor 37.

The NASA Rotor 37 was designed and studied experimentally at the NASA Lewis Research Center (today NASA Glenn) in the late 1970's [1]. Since then, a great number of numerical studies on the NASA Rotor 37 have been reported in the literature.

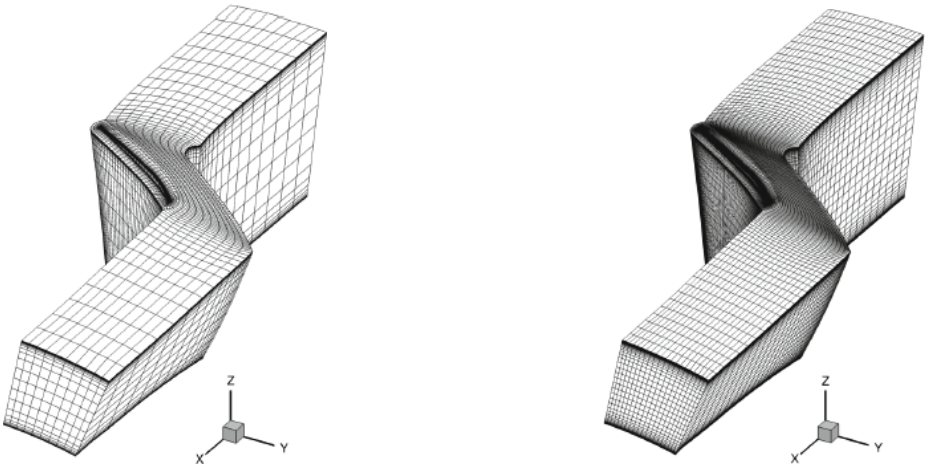
The rotor has a design pressure ratio of 2.106 at a mass flow of 20.19 kg/s. The measured choking mass flow is of 20.93 kg/s. The design wheel speed is 17,188 rpm, which leads to a nominal tip speed of 454 m/s. The computations are aimed at a point corresponding to 98% of the choking mass flow rate.

The inlet conditions are prescribed by specifying the absolute total pressure $\mathrm{p}_{i,0}$, the absolute total temperature $T_{i,0}$, and the direction of the flow, which is assumed to be axial. The inflow turbulence intensity is of 3% and the turbulent to molecular viscosity ratio is of the order of 10. At the outlet a static pressure profile satisfying the radial equilibrium condition is imposed. The walls are assumed to be adiabatic.

Two grid sizes have been considered, a fine mesh consisting of 672,896 points and a coarse mesh with half the number of points of the fine mesh in each direction, with a total of 87,769 points. These grids are shown in Fig. 3. On each grid, two different simulations have been performed with polynomial degrees $p = 1$ and $p = 2$, respectively, using the Spalart-Allmaras RANS model and the shock-capturing technique developed in [22]. The simulations are converged until the relative error in the mass flow rate between the inlet and the outlet has reached an acceptably low value of the order of 10^{-5}.

Figure 4 shows the iso-contours of the Mach number and the static pressure for the different computations on a plane located at mid-span. We can see from these plots that as the polynomial degree is increased ($p = 1$ to $p = 2$) the flow solution on the coarse mesh converges towards the second-order ($p = 1$) solution on the fine mesh. It should be noted that these plots have been obtained on the volume mesh (solution at the cell centres) which does not take into account the high-order character of the solution. A high-order representation of these iso-contours is expected to accentuate this trend.

Figure 5 presents the radial profiles of adiabatic efficiency, η_{ad}, absolute total pressure ratio, $\mathrm{p}_i/\mathrm{p}_{i,0}$, and absolute total temperature ratio, $T_i/T_{i,0}$, at a station



(a) Coarse grid: 87,769 points. (b) Fine grid: 672,896 points.

Fig. 3. Computational grids used in NASA Rotor 37 computations

located at $x = 0.1067$m downstream of the blade. The adiabatic efficiency is defined as

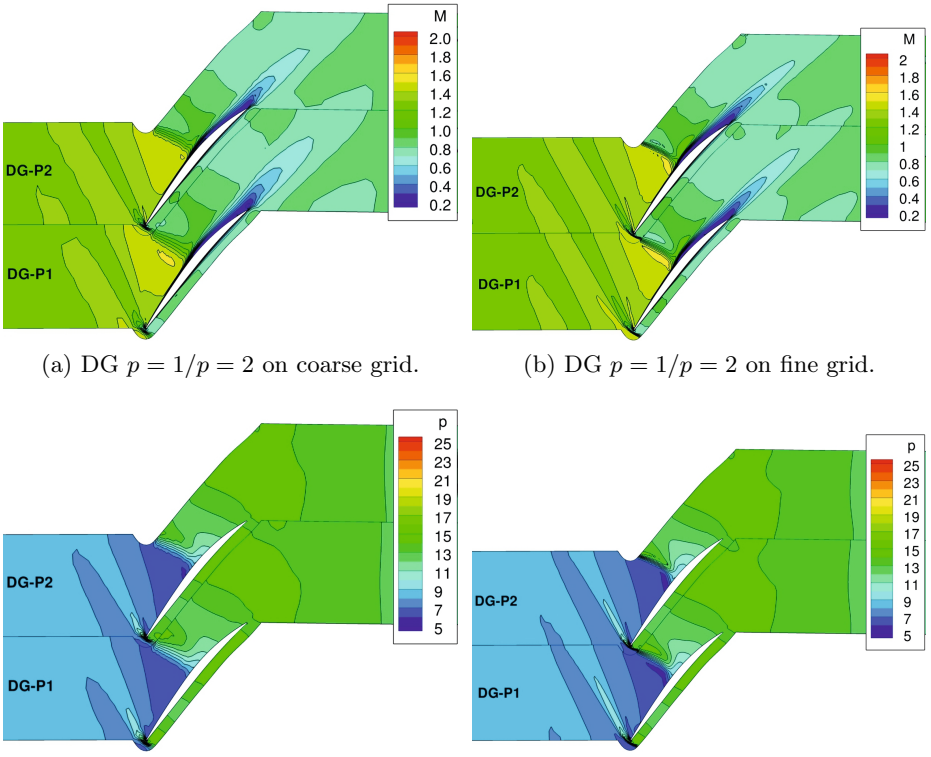
$$\eta_{ad} = \frac{(\mathrm{p}_i/\mathrm{p}_{i,0})^{\frac{\gamma-1}{\gamma}} - 1}{(T_i/T_{i,0})}.$$

These profiles are obtained by averaging in the azimuthal direction (across the annulus). Once more, we can see that the profiles provided by the 3rd-order simulation ($p = 2$) on the coarse grid are very similar to those obtained from the 2nd-order simulation ($p = 1$) on the fine grid. The gain in terms of DOFs is over 50% when considering the $p = 2$ simulation on the coarse with respect to the fine grid simulation $p = 1$.

The differences found between the simulation results and the experimental data are in line with what is usually found in the literature. A detailed study on the possible origins of these differences was done by the AGARD Working Group 26 and can be found in [19].

3.3 Jean Nozzle

This test case is concerned with the flow induced by a turbulent isothermal and subsonic jet exhausting from a nozzle and has been experimentally investigated during the European project JEAN for prediction of noise by jet flows [30]. Figure 6 presents the geometry of the nozzle with curved inner walls. A close-up view of the mesh in the nozzle region is also provided. This mesh was provided by UCL, it consists of $179,180$ quadratic prisms and hexahedra. The jet is isothermal with a static temperature in the exit plane of $T_j/T_{i,0} = 1$ where $T_{i,0}$ is the total temperature of the jet. The Mach number, based on the jet



(a) DG $p = 1/p = 2$ on coarse grid.

(b) DG $p = 1/p = 2$ on fine grid.

(c) DG $p = 1/p = 2$ on coarse grid.

(d) DG $p = 1/p = 2$ on fine grid.

Fig. 4. DG computations of the Rotor 37 test case. Iso-contours of the Mach number (upper panels) and the static pressure (lower panels) at mid-span.

velocity and temperature on the exit plane, is $M = u_j/(\gamma r T_j)^{1/2} = 0.75$. The Reynolds number, based on u_j and the diameter of the nozzle at the exit, is set to $Re = \rho_j u_j D/\mu(T_j) = 5 \times 10^4$. In these conditions, the flow is subsonic and fully turbulent. We impose the total pressure, total temperature and velocity direction at the inlet of the settling chamber. No-slip and adiabatic conditions are applied at the walls, and non-reflecting boundary conditions are applied at the free artificial boundaries. The RANS equations coupled with the Spalart-Allmaras turbulence model are also used for this test case.

Figure 7 presents the iso-contours of the Mach number and μ_t/μ ratio on a vertical plane crossing the jet axis obtained at the steady-state. We clearly see the structure of the potential core of the jet and the development of turbulence in the mixing layer downstream of the nozzle lips. Unfortunately, the mesh extent is not large enough and spurious reflection may have affected the quality of the results.

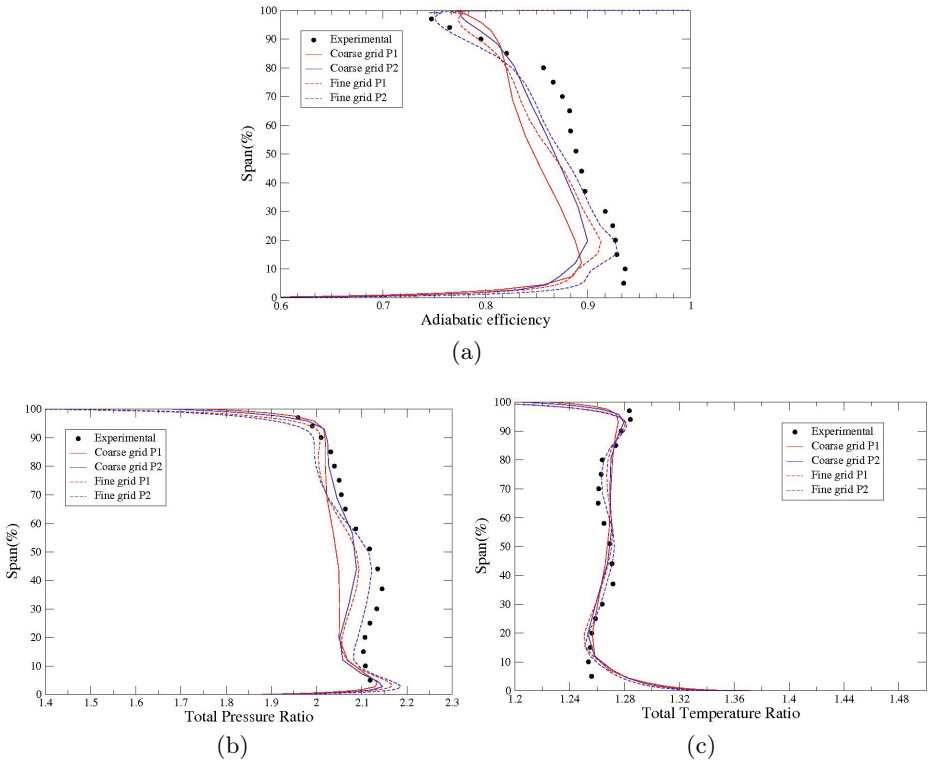
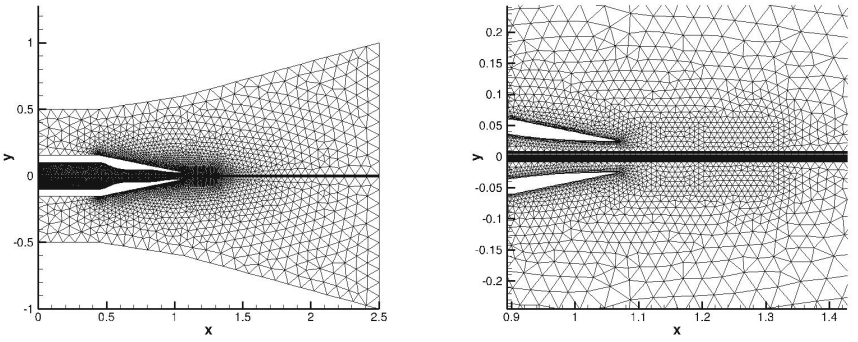


Fig. 5. DG computations of the Rotor 37 test case. Spanwise profiles of (a) the adiabatic efficiency, (b) total pressure ratio and (c) total temperature ratio.

3.4 *hp*-Adaptive DNS of Plane Channel Flow

A remarkable property of DG approximations is that the spatial resolution can be conveniently adapted, not only by local mesh refinement (h-refinement), but also by locally increasing the polynomial degree within the elements (p-refinement). Thanks to the compact character of the numerical scheme, DG methods can easily handle complex geometries and irregular meshes with hanging nodes, which simplifies the implementation of h-refinement techniques. The overall flexibility provided by the DG approach therefore makes this type of discretization a very appealing tool for the simulation of inhomogeneous turbulent flows using multiscale methods.

This property is of great interest for the simulation of wall-bounded flows for which the spatial resolution needs to be adapted in the proximity of the wall in order to accurately capture the physics of the near-wall turbulence. As an example, in the case of the turbulent plane channel flow at high Reynolds numbers, it is well known that the size of the turbulent structures near the wall, the so-called streaks, in the streamwise and spanwise directions is small with respect to those developing in the external region. By varying the polynomial



(a) (b)

Fig. 6. Mesh for the Jean nozzle test case: cut of the mesh in a vertical plane crossing the jet axis (a) and zoom in the nozzle region (b)

Table 1. Resolution details for the computations of the turbulent channel flow at $Re_\tau = 590$

Numerical method	# DOFs	# Integration points	degree
DG	7.02×10^5	1.12×10^6	$p = 4, 5, 7$
Spectral	3.79×10^7	1.28×10^8	-

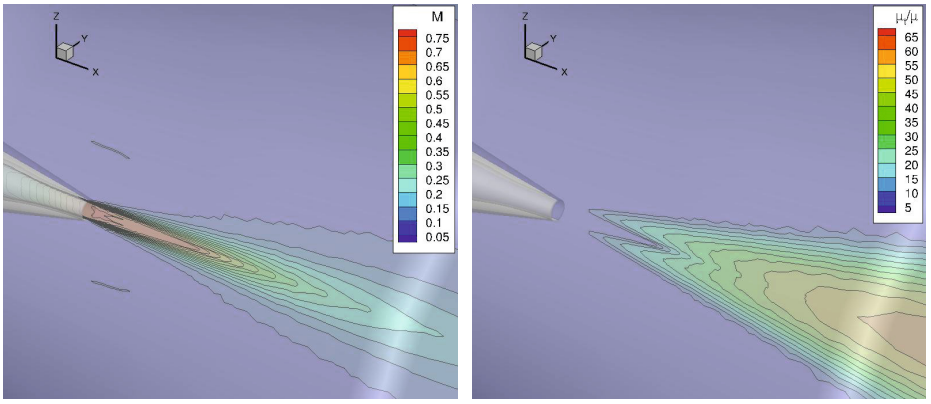
degree of the discretization in the direction normal to the wall, it is thus possible to achieve an adequate resolution for the fine-scale structures while keeping the number of DOFs low with respect to a non-adapted simulation.

In order to illustrate the flexibility of DG-based *hp*-adaptation to deal with the physics of the turbulence, we have performed an *hp*-adapted DNS simulation of the turbulent plane channel configuration at $Re_\tau = 590$.

Figure 8 shows the statistically-averaged profiles of the mean streamwise velocity and the fluctuating velocity. We observe an excellent agreement with the reference solution, despite the significant reduction in number of DOFs in the DG simulation with respect to the reference DNS, as can be seen from Table 1.

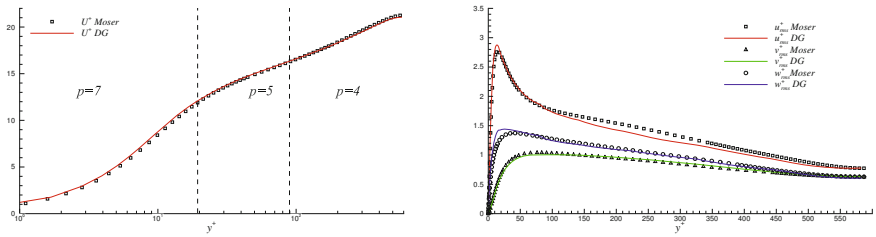
3.5 VMS-LES of the Taylor-Green Vortex at $Re = 3000$

VMS-LES Results. The Taylor-Green vortex is a model problem for the study of freely decaying turbulence in a periodic box. The physical characteristics of the flow, such as the anisotropy of the Reynolds stresses and the transition from a laminar to a turbulent state due to the presence of strong vortex stretching, are representative of realistic turbulent flows. This configuration is therefore a



(a) (b)

Fig. 7. Jean nozzle test case: Mach number (a) and μ_t/μ ratio (b) contours obtained with a third-order approximation



(a) Mean velocity profile. (b) Fluctuating velocity profiles.

Fig. 8. Mean and fluctuating velocity profiles for the turbulent channel flow at $Re_\tau = 590$

good candidate for the assessment of the DG-VMS approach for the simulation of turbulence. A detailed description of the TGV problem can be found in references [8, 9, 50].

Here, we present the results from the application of a modal DG-VMS approach to LES of the TGV at $Re = 3000$. In particular, the evolution of the mean statistical quantities and the turbulent kinetic energy spectra is studied.

As regards the choice of the model term in the small-scale equations, a small-small approach is adopted in the structure function model [36], namely,

$$\widetilde{\mathbf{M}}\left(\tilde{\phi}, \bar{u}, \tilde{u}, \hat{u}\right) = \left(\tilde{\phi}, 2\nu_t \tilde{S}_{ij}\right)_{\Omega}, \tag{10}$$

where the eddy viscosity is given by

$$\nu_t = C_{SF}\tilde{\Delta}\sqrt{\tilde{F}_2(\tilde{\Delta})}, \quad \tilde{F}_2(\tilde{\Delta}) = \langle \|\tilde{\mathbf{u}}(\mathbf{x}) - \tilde{\mathbf{u}}(\mathbf{x}-\mathbf{r})\|^2 \rangle, \quad \|\mathbf{r}\| = \tilde{\Delta}.$$

Table 2. Resolution details for the computations of the Taylor-Green vortex at $Re = 3000$

Numerical method	# DOFs	# Integration points	# Elements / degree
DG	72^3	84^3	12^3 / $p = 5$
Spectral	384^3	576^3	-

The model parameters (the model constant C_{SF}, and filter width, $\tilde{\Delta}$) will depend on the particular *hp*-discretization used. The evaluation of $C_{SF}\tilde{\Delta}$ is based on the hypothesis that the cut-off wavenumber of the given discretization is located in the inertial range and that the energy transfers are from the resolved to the residual motions, being these transfers balanced by the dissipation (see Pope [39]).

The DG solutions are compared with reference data generated using a Fourier pseudo-spectral code [13]. In Fig. 9 we can observe the good correlation between the modal DG-VMS and the Fourier pseudo-spectral (PS) reference DNS of the Taylor-Green vortex at Re = 3000. The DG-VMS computation is based on a polynomial degree $p = 5$ and a number of elements 12^3. This corresponds to a number of DOFs of 72^3 with respect to 384^3 in the reference DNS. Table 2 summarizes the resolution details of the these computations.

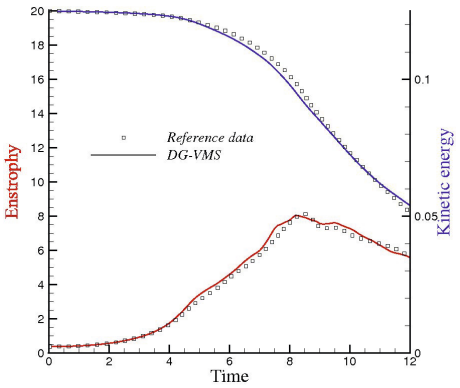


Fig. 9. Time evolutions of the enstrophy and kinetic energy for the Taylor-Green vortex at Re=3000

Strong Scalability Analysis In order to compare the behaviour of the implemented parallel strategies (see § 2.5), a strong scalability analysis is performed by partitioning the original mesh in the three axis directions by a same factor. This ensures that the amount of data exchanged by messages is identical in all

directions for a given partition. For the MPI strategy, the partition leads to one MPI process per core, whereas, in the hybrid strategy, the partition leads to one MPI process per processor (eight-core pack). Each MPI process from the hybrid approach relies on the multithreading support level 1 where only the thread master can call the MPI library, and will deal with 8 times more elements than in the MPI strategy.

Numerical experiments were performed on the Curie supercomputer into the TGCC by CEA on nodes composed of two eight-core Intel Sandy Bridge E5-2680 processors in the context of a PRACE project. The compilation environment relies on the Intel compiler Ifort (v11.1.072), with the Intel MPI distribution (v4.1.0.030) or the Bull MPI distribution (v1.1.16.5). Different DNS of the Taylor-Green vortex have been performed on a uniform mesh with 336^3 hexahedra and different polynomial degree $1 \leq p \leq 4$.

Figure 10 depicts the parallel behaviour of the initial MPI strategy. On $21,952$ cores, the receive-send message frequency is of about 14 messages per second for $p = 1$. This means that the interconnection network strongly deteriorates the performance. The frequency decreases to an average of 6 messages per second for $p = 2$ and results in a good scalability at large scale as the time spent in the computational phase becomes larger than the time spent in the communication phase. Using $p = 2$ on $21,952$ cores, we observe from the comparison of obtained and theoretical speedup that the MPI strategy offers an efficiency of about 88%.

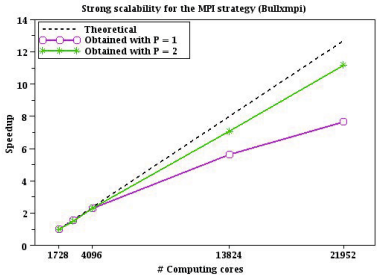
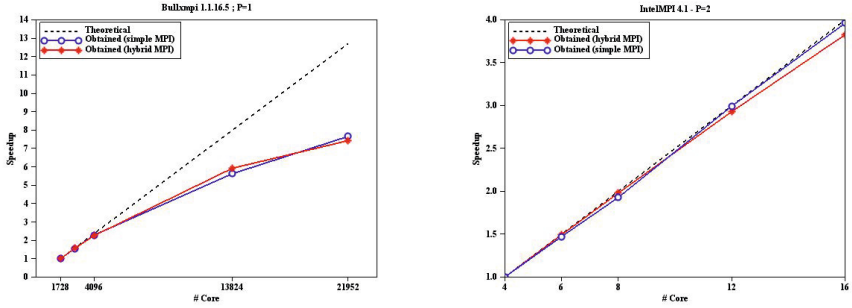


Fig. 10. Effect of the polynomial degree p on the speedup using the MPI strategy

Figure 11 highlights the parallel behaviour of both strategies running a computation involving a large number of computing nodes (Fig. 11a) and only one computing node (Fig. 11b). For the intra-node case, hybrid results come from one MPI process per processor with an increasing number of OpenMP threads starting from 2 up to 8. The strong scaling curves are quite similar regardless of the polynomial degree or the number of computing cores. First experiments (results not shown here) demonstrate a gain in elapsed time for $p = 4$ with the hybrid strategy of about 13% on $13,824$ computing cores and about 19% on $21,952$ computing cores. A polynomial degree $p = 4$ leads to an increase of the algorithmic complexity resulting in more floating point operations in shared memory. The effect of different support levels of the MPI library on the performances wiil be adressed in future investigations.



(a) At large scale ; $p = 1$ (b) Intra-node ; $p = 2$

Fig. 11. Comparison of the parallel behaviour of the two strategies

4 Concluding Remarks

The present results obtained on three-dimensional and fully turbulent test cases clearly demonstrate the ability of the DG Aghora code to produce very accurate solutions with respect to low-order simulations such as standard second-order finite volume methods. Important savings in terms of number of DOFs are also gained for a comparable level of accuracy. The code allows different physical modellings from RANS equations with one-equation and two-equation turbulence models to LES, VMS-LES and DNS approaches. This results in a wide range of applications such as unsteady flows, transonic flow regime, turbomachinery applications or complex geometries with high-order boundary representation.

So far, important developments have been successful in improving the efficiency of the Aghora code. Local mesh and local order variations demonstrated the flexibility of the method and subsequent CPU time savings. Likewise, domain decomposition strategies present strong potentiality of code scalability. Among others, future work will consist in further enhancement of the code with the objective of considering large-scale problems and taking benefits of today's computer architecture. Time implicit integration with efficient linear solvers of reduced memory requirements and CPU time cost are currently under development. Moreover, *hp*-adaptative methods for VMS-LES will enable high-accuracy turbulence simulations of complex configurations at reasonable costs. A particular attention will also be paid to the implementation of the code adapted to modern supercomputing plaforms in order to fully exploit all available computational ressources: multi-core processors, Nvidia GPU accelerators, Intel Xeon Phi coprocessors, etc. Heterogeneous multi-core architectures indeed require specific adaptations of the programming models and it is thus crucial to develop highly flexible parallel solutions. These enhancements will enable the accurate description of complex large-scale turbulent flows of industrial relevance.

References

1. Abgrall, R., Shu, C.-W.: Development of residual distribution schemes for the discontinuous Galerkin schemes: the scalar case with continuous elements, Commun. Comput. Phys. 5, 376–390 (2009)
2. Ainsworth, M.: Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. J. Comput. Phys. 198(1), 106–130 (2004)
3. Apoung Kamga, J.-B., Després, B.: CFL condition and boundary conditions for DGM approximation of convection-diffusion. SIAM J. Numer. Anal. 44, 2245–2269 (2006)
4. Baboulin, M., Buttari, A., Dongarra, J., Kurzak, J., Langou, J., Langou, J., Luszczek, P., Tomov, S.: Accelerating scientific computations with mixed precision algorithms. Computer Physics Communications 180, 2526–2533 (2009)
5. Bassi, F., Botti, L., Colombo, A., Di Pietro, D.A., Tesini, P.: On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. J. Comput. Phys. 231, 45–65 (2012)
6. Bijl, H., Carpenter, M.H., Vatsa, V.N., Kennedy, C.A.: Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: laminar flow. J. Comput. Phys. 179, 313–329 (2002)
7. Biswas, R., Devine, K., Flaherty, J.E.: Parallel adaptive finite element methods for conservation laws. Appl. Numer. Math. 14, 255–284 (1994)
8. Brachet, M., Meiron, D., Orszag, S., Nickel, B., Morf, R., Frisch, U.: Small-scale structure of the TaylorGreen vortex. J. Fluid Mech. 130(41), 1452 (1983)
9. Brachet, M., Meneguzzi, M., Vincent, A., Politano, H., Sulem, P.: Numerical evidence of smooth self-similar dynamics and possibility of subsequent collapse for three-dimensional ideal flows. Phys. Fluids 4, 2845 (1992)
10. Bassi, F., Crivellini, A., Rebay, S., Savini, M.: Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k-\omega$ turbulence model equations. Comput. Fluids 34, 507–540 (2005)
11. Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., Savini, M.: A High-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere, R., Dibelius, G. (eds.) Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium (1997)
12. Cahen, J., Couaillier, V., Délery, J., Pot, T.: Validation of code using turbulence model applied to three-dimensional transonic channel. AIAA J 33, 671–679 (1995)
13. Cant, R.S.: Fergus, a user guide. technical report. Cambridge University Engineering Department (1999)
14. Chapelier, J.-B., de la Llave Plata, M., Renac, F., Lamballais, E.: Evaluation of a high-order discontinuous Galerkin method for the DNS of turbulent flows. Comp. Fluids 95, 206–226 (2014)
15. Chavent, G., Cockburn, B.: The local projection P^0P^1-discontinuous Galerkin finite element method for scalar conservative laws. M2AN Math. Model. Anal. Numer. 23, 565–592 (1989)
16. Cockburn, B., Shu, C.W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: general framework. Math. Comp. 52, 411–435 (1989)
17. Cockburn, B., Shu, C.W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Computing 16, 173–261 (2001)

18. Collis, S.S.: Discontinuous Galerkin methods for turbulence simulation. In: Proceedings of the Summer Program, CTR, pp. 155–167 (2002)
19. Dunham, J.: CFD Validation for Propulsion System Components. AGARD Advisory Report 355 (1998)
20. Gottlieb, D., Shu, C.W.: On the Gibbs phenomenon and its resolution. SIAM Rev. 39, 644–668 (1997)
21. Guermond, J.-L., Pasquetti, R., Popov, B.: Entropy viscosity method for nonlinear conservation laws. J. Comput. Phys. 230, 4248–4267 (2011)
22. Hartmann, R., Houston, P.: Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations. J. Comput. Phys. 183, 508–532 (2002)
23. Hartmann, R.: Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. Int. J. Numer. Meth. Fluids. 72, 883–894 (2013)
24. Houston, P., Süli, E.: *hp*-adaptive discontinuous Galerkin finite/ element methods for first-order hyperbolic problems. SIAM J. Sci. Comput. 23, 1226–1252 (2001)
25. Hughes, T.J.R.: Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid-scale models, bubbles and the origins of stabilized methods. Comput. Methods Appl. Mech. Eng. 127, 387–401 (1995)
26. Hughes, T.J.R., Stewart, J.R.: A space-time formulation for multiscale phenomena. J. Comput. Appl. Math. 74, 217–229 (1996)
27. Hughes, T.J.R., Mazzei, L., Oberai, A.A., Wray, A.A.: The multiscale formulation of large-eddy simulation: Decay of homogeneous isotropic turbulence. Phys. Fluids 13, 505–512 (2001)
28. Hughes, T.J.R., Oberai, A.A., Mazzei, L.: Large-eddy simulation of turbulent channel flows by the variational multiscale method. Phys. Fluids 13, 1784–1799 (2001)
29. Jin, H., Jespersen, D., Mehrotra, P., Biswas, R., Huang, L., Chapman, B.: High performance computing using MPI and OpenMP on multi-core parallel systems. Parallel Computing 37, 562–575 (2011)
30. Jordan, P., Gervais, Y., Valire, J.-C., Foulon, H.: Final results from single point measurements, Project deliverable D3.4, JEAN EU 5th Framework Programme, G4RD-CT-2000-00313 (2002)
31. Kværnø, A.: Singly diagonally implicit Runge-Kutta methods with an explicit first stage. BIT Numerical Mathematics 44, 489–502 (2004)
32. Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.): ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications. NNFM, vol. 113. Springer, Heidelberg (2010)
33. Lavallée, P.-F., Wautelet, P.: Hybrid MPI-OpenMP Programming, CNRS IDRIS Lessons (2013)
34. Lesaint, P., Raviart, P.-A.: On a finite element method for solving the neutron transport equation. In: de Boor, C. (ed.) Mathematical Aspects of Finite Elements in Partial Differential Equations, pp. 89–123. Academic Press, New York (1974)
35. Lhner, R., Baum, J.D.: Handling tens of thousands of cores with industrial/legacy codes: Approaches, implementation and timings. Computers & Fluids 85, 53–62 (2013)
36. Metais, O., Lesieur, M.: Spectral Large-Eddy Simulation of isotropic and stably stratified turbulence. J. Fluid Mech. 239, 157–194 (1992)
37. Moser, R.D., Kim, J., Mansour, N.N.: Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. Phys. Fluids 11, 943 (1999)

38. Nastase, C.R., Mavriplis, D.J.: High-order discontinuous Galerkin methods using an *hp*-multigrid approach. J. Comput. Phys. 213, 330–357 (2006)
39. Pope, S.B.: Turbulent flows. Cambridge University Press (2000)
40. Ramakrishnan, S., Collis, S.S.: The Local Variational Multi-Scale Method for Turbulence Simulation. Sandia Report SAND2005-2733 (2005)
41. Reed, W.H., Hill, T.R.: Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, NM (1973)
42. Remacle, J.F., Flaherty, J.E., Shephard, M.S.: An Adaptive Discontinuous Galerkin Technique with an Orthogonal Basis Applied to Compressible Flow Problems. SIAM Review 45, 55–73 (2003)
43. Reid, L., Moore, R.D.: Design and overall performance of four highly loaded, high-speed inlet stages for an advanced high-pressure-ratio core compressor. NASA Technical Report 1337 (1978)
44. Smagorinsky, J.: General circulation experiments with the primitive equations: I. The basic equations. Mon. Weather Rev. 91, 99–164 (1963)
45. Solin, P., Segeth, K., Dolezel, I.: Higher-order finite element methods. Chapman & Hall/CRC Press (2003)
46. Spiteri, R.J., Ruuth, S.J.: A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. 40, 469–491 (2002)
47. Spalart, P.R., Allmaras, S.R.: A one-equation turbulence model for aerodynamic flows. La Rech. Aéro. 1, 5–21 (1994)
48. Saad, Y., Schultz, M.: A generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM Journal Sci. Stat. Comput. 7, 856–869 (1986)
49. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. 77, 439–471 (1988)
50. Taylor, G., Green, A.: Mechanism of the production of small eddies from large ones. Proc. R. Soc. A 158(895), 499 (1937)
51. Wilcox, D.C.: Reassessment of the Scale-Determining Equation for Advanced Turbulence Models. AIAA J 26, 1299–1310 (1988)

Implementation of High-Order Discontinuous Galerkin Method for Solution of Practical Tasks in External Aerodynamics and Aeroacoustics

I.S. Bosnyakov[1], S.V. Mikhaylov[1], A.N. Morozov[1], V.Y. Podaruev[1], A.I. Troshin[1], V.V. Vlasenko[1], A.V. Wolkov[1], A. Garcia–Uceda[2], and C. Hirsch[2]

[1] Central Institute of aero- and hydrodynamics (TsAGI), Zhukovsky, Russia
bosnyakov@tsagi.ru
[2] NUMECA Int., Brussels, Belgium

Abstract. For solution of 3D stationary RANS equations, closed by EARSM turbulence model, high order Discontinuous Galerkin method (degree of basic polynomials $K = 2, 3$ with 1st order implicit smoother is proposed. Modifications, which were introduced in the method to achieve stability and fast convergence, are described. The method is enhanced by the use of improved Gauss quadrature rules and of *h-p* multigrid multigrid acceleration and is implemented into NUMECA FINE^{TM}/Hexa code in version for massive parallel calculations. For solution of 3D non-stationary Isentropic Linearized Euler Equations within the perturbation approach in aeroacoustics, explicit high order Discontinuous Galerkin method is implemented. Calculations of various tests, including U3, U2 and A14, demonstrate the efficiency of developed methods.

Keywords: Discontinuous Galerkin, RANS, EARSM, implicit smoother, high-order method, multigrid, isentropic linearized Euler euations, external aerodynamics, aeroacoustics.

1 Introduction

Recently, the interest of many investigators turns to high accuracy order methods that permit to obtain good results for more rough calculation grids in comparison with traditional finite-volume methods of the second accuracy order.

One of the most perspective approaches to high-accuracy approximation both for structured and for non-structured grids is Galerkin's method with discontinuous basis functions (DG). This methods combines the advantages of finite-volume methods (such as possibility to describe the conservation laws in a flow and taking into account the direction of information propagation) with advantages of finite-element methods (such as independence upon quality of computational grid and compact stencil). In addition, DG approach provides clear way to construct compact numerical schemes of arbitrarily high accuracy order.

For the first time, this method has been proposed in [1] for solution of equation, which describes neutron transition, and its first analysis has been presented

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_16

in the paper [2]. Numerical solution of 2D Euler and Navier-Stokes equations for triangular non-structured grids using this method has been presented, for the first time, in the papers [3,4]. The most full theoretical description of this method with solution of 1D and 2D model problems is presented in the papers [5, 6].

The first 3D realizations of the method concern aeroacoustics problems, where linearized Euler equations were solved [7–10]. Now, the number of 3D DG realizations for non-linear conservation laws isn't too large. In published articles, one can find an information only about some examples of successful realization of the method. For example, DG using for tetrahedral grids has been described in the paper [11]. In the case of structured hexahedral grids, 3D DG algorithm has been realized in the papers [12,13]. One of the first successful DG implementation for non-structured hexahedral grids is presented in the paper [14].

In our opinion, hexahedral grids have an advantage in comparison with tetrahedral grids because the first ones provide covering the computational domain and have the lower number of inner sides. It diminishes the number of arithmetic operations.

Practical DG realization for solution of Euler and Navier-Stokes nonlinear equations meets many difficulties. Because of this reason, the method hasn't obtained wide propagation yet. Nevertheless, Discontinuous Galerkin in combination with such technologies as parallelization and *h-p* multigrid is considered now as one of the most relevant high-order numerical methods capable both to increase the predictive accuracy for flows exhibiting separation and re-attachment and at the same time to diminish the computational cost.

Within IDIHOM project, TsAGI has implemented high-order implicit Discontinuous Galerkin method for solution of full 3D stationary Reynolds-averaged Navier-Stokes equations (RANS), closed by an explicit algebraic Reynolds stress model (EARSM), into the widely-known FINE^{TM}/Hexa code produced by NUMECA International. The method is applicable to stationary aerodynamics problems and designed to be unconditionally stable. The main specific feature of the scheme is the use of first order implicit smoother in combination with high-order-in-space approximation of the explicit operator. Optimized Gauss quadrature rules are applied for the definition of volume and surface integrals with the aim to reduce the number of arithmetic operations. The implemented method includes possibilities of using *h-p* multigrid approach and parallelized for massive multiprocessor calculations. 3D hexahedral unstructured DG solver, which has been developed by TsAGI in collaboration with NUMECA, may be disseminated among NUMECA users in the nearest future. This solver has been validated on the test problems characteristic for external aerodynamics, including test cases U.2, U.3, and A.3.

Additionally, and again in collaboration with NUMECA, TsAGI has adapted the previously developed high-order Linearized Euler Solver for solution of aeroacoustic problems and applied it for the calculation of test case A.14.

This paper describes the main ideas of the implemented high-order DG solvers and also presents the most characteristic results of their validation.

2 Equations of Discontinuous Galerkin Method

Systems of partial differential equations, which are used in aerodynamics and aeroacoustics, can be represented in following general form:

$$\Gamma \frac{\partial \mathrm{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathrm{Q}, \mathbf{G}) = \mathrm{S}(\mathrm{Q}, \mathbf{G}). \tag{1}$$

Let's designate the dimension of this system (quantity of equations) by N. Roman capital regular letters (e.g., Q) will designate vectors associated with the set of variables vectors of N dimension), roman bold letters (e.g., $\mathbf{F}$) will designate the vectors with dimension 3 that are associated with 3 spatial directions x, y, z. Elements of these vectors can be any objects – numbers, vectors, operators. In the equation system (1), Q is the vector of primitive variables, $\Gamma = \partial \mathrm{U}/\partial \mathrm{Q}$ is Jacobian of transformation from vector Q to the vector of conservative variables U; $\mathbf{F} = [\mathrm{F}_x; \mathrm{F}_y; \mathrm{F}_z]^{\mathrm{T}}$ is the set of fluxes of the vector U; S is a vector of source terms; $\mathbf{G} \equiv \nabla \mathrm{Q} = [\mathrm{G}_x; \mathrm{G}_y; \mathrm{G}_z]^{\mathrm{T}}$ is the gradient of vector Q.

In DG method, numerical solution of system (1) in each cell of computational grid is represented as a linear combination of local polynomial basis functions $\varphi_j(\mathbf{x})$:

$$\mathrm{Q} = \sum_{j=1}^{K_f} \mathrm{q}_j \varphi_j(\mathbf{x}). \tag{2}$$

Coefficients of this expansion, $\mathrm{q}_j(t)$, are the main unknown values in DG method. Each of them is vector of N dimension with components corresponding to primitive variables.

The following set of basis functions is chosen: $\varphi_j(x, y, z) = \xi^{\alpha_j} \eta^{\beta_j} \zeta^{\gamma_j}$, where ξ, η, ζ are coordinates in a local frame, connected with grid lines of a hexahedral cell. Axes of this coordinate frame pass through the centers of the cell's opposite faces, and values of coordinates in these centers are ± 1. Cartesian and local coordinate frames are related by linear transformation

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} x - x_c \\ y - y_c \\ z - z_c \end{bmatrix}.$$

Here (x_c, y_c, z_c) are coordinates of the cell center. Matrix $\mathbf{J} = \partial(x; y; z)/\partial(\xi, \eta, \zeta)$ is calculated once at the beginning of calculation and is stored in the memory. Degree of polynomials $(\alpha_j + \beta_j + \gamma_j)$ is varied from 0 to K. Quantity of basic functions K_f is related with maximal degree of basic polynomial K by the formula $K_f(K) = (K + 1)(K + 2)(K + 3)/6$.

Representation of numerical solution in the form (2) may be treated as polynomial reconstruction of the gas parameters distribution within the computational cell. Theoretically, DG method based on polynomials of degree K should provide solution of the system (1) with accuracy order $(K + 1)$.

To determine the dependence of coefficients q_j upon time, each equation of system (1) is multiplied by basis functions and is integrated over the cell volume Ω:

$$\int\limits_{\Omega} \left(\Gamma \frac{\partial \mathrm{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathrm{Q},\mathbf{G}) \right) \varphi_i(\mathbf{x}) d\Omega = \int\limits_{\Omega} \mathrm{S}(\mathrm{Q},\mathbf{G})\, \varphi_i(\mathbf{x}) d\Omega, \quad i = 1, ..., K_f. \quad (3)$$

Applying transformation $(\nabla \cdot \mathbf{F})\, \varphi = \nabla \cdot (\mathbf{F}\, \varphi) - \mathbf{F}\, \nabla \varphi$ and using Gauss-Ostrogradsky's formula $\int\limits_{\Omega} (\nabla \cdot \mathbf{A})\, d\Omega = \oint\limits_{\Sigma} (\mathbf{A} \cdot \mathbf{n})\, d\Sigma$ (Σ is the cell surface, $\mathbf{n}$ is the outer unit normal to the surface element $d\Sigma$), one can rewrite (3) as follows:

$$\int\limits_{\Omega} \Gamma \frac{\partial \mathrm{Q}}{\partial t}\, \varphi_i(\mathbf{x}) d\Omega + \oint\limits_{\Sigma} \mathrm{F}_n\, \varphi_i(\mathbf{x}) d\Sigma = \int\limits_{\Omega} \mathrm{F}_i\, d\Omega + \int\limits_{\Omega} \mathrm{S}\, \varphi_i(\mathbf{x}) d\Omega, \quad (4)$$

where $\mathrm{F}_n = (\mathbf{F} \cdot \mathbf{n}) = \mathrm{F}_x\, n_x + \mathrm{F}_y\, n_y + \mathrm{F}_z\, n_z$, $\mathrm{F}_i \equiv (\mathbf{F} \cdot \nabla \varphi_i(\mathbf{x}))$.

The first term in the left part of the equation (4) is connected with rate of Q temporal change in the given cell. The second term in the left part is connected with integration over the cell surface and describes the contribution of Q fluxes through the cell side. The right part includes the terms connected with integration over the cell volume. These terms can be named as source terms. However, during the IDIHOM project the authors have found that the term $\int\limits_{\Omega} (\mathbf{F} \cdot \nabla \varphi_i(\mathbf{x}))\, d\Omega$ is better to be considered as a result of DG-approximation of flux terms; so, it should be approximated using the same principles as the flux term $\oint\limits_{\Sigma} (\mathbf{F} \cdot \mathbf{n})\, \varphi_i(\mathbf{x}) d\Sigma$. The second term in the right part, $\int\limits_{\Omega} \mathrm{S}\, \varphi_i(\mathbf{x}) d\Omega$, is a classical source term, depending on the unknown function Q and producing exponentially-varying modes in the solution of (4).

It is important to note that the choice of polynomials $\varphi_j(x, y, z) = \xi^{\alpha_j} \eta^{\beta_j} \zeta^{\gamma_j}$ as basis functions allows to avoid the stability problems in the strongly distorted cells, because the mass matrix $M_{ij} = \int\limits_{\Omega} \varphi_i \varphi_j d\Omega$ (that arises after substitution (2) into (4)) doesn't degenerate in the distorted cells.

3 Optimization of Quadrature Rules for Integration in DG Method

The traditional approach to calculate integrals in equations of DG method (4) is based on the use of Gauss quadrature nodes. Summation of integrand function values at these nodes with defined weights provides exact solution for polynomial representation of integrand function. The higher polynomial order, the larger number of Gauss nodes is to be used. As the DG method requires computation both of volume and surface integrals, it is necessary to use extremely large number of the nodes, both on faces of control volume and within it. Determination of formulas with given accuracy for integration with minimal number of quadrature nodes is an actual mathematical problem. Its solution permits to diminish the

calculation time. One of the methods to enhance solver performance is to find an optimal distribution of quadrature nodes so it can be used simultaneously in surface integrals computation and in control volume as well. Such approach could permit to diminish total number of nodes for given integration accuracy and, consequently, reduce number of arithmetic operations. For the first time such approach has been presented in the paper [5] in the case of description of DG algorithm for 2D grid with rectangular cells and for piecewise quadratic approximation of solution.

Let's consider a function $f(x)$ that is continuous and integrable within the region $\Theta \subset \mathbb{R}^n$ ($n = 2$ for surface integrals and $n = 3$ for volume integrals). Below we shall use the term *quadrature rule* for the following formula:

$$\int\limits_{\Theta} f(x)\,dx \cong \sum_{j=1}^{N} c_j f(x_j), \quad i = 1, ..., K_f, \tag{5}$$

where the points $x_j \subset \Theta$ shall be called as *Gauss nodes*. Quadrature rule (5) has m-property, if it gives exact value of integral for any $f(x)$ that is polynomial of degree not higher than m. This property is satisfied under the following condition:

$$\begin{bmatrix} x_1^{\alpha_1} & ... & x_N^{\alpha_1} \\ \cdots & \cdots & \cdots \\ x_1^{\alpha_\psi} & ... & x_N^{\alpha_\psi} \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ ... \\ c_N \end{bmatrix} = \begin{bmatrix} f_1 \\ ... \\ f_\psi \end{bmatrix}, \tag{6}$$

where $f_i = \int\limits_{\Theta} x^{\alpha_i} dx$ and $\alpha^{(1)}$, ..., $\alpha^{(\psi)}$ are all possible integer-valued vectors, consisted of n non-negative components with sum not higher than m. Quantity of such vectors is equal to $\psi = M(n, m)$, where $M(s, t) \equiv \frac{(s+t)!}{s! \cdot t!}$.

The task is to find the set of Gauss nodes and corresponding coefficients c_j, which provide the m-property under the following additional requirements:

- integration domain is a cube (for $n = 3$) or square ($n = 2$) with sides $[-1, 1]$;
- number of Gauss nodes should be as low as possible;
- volume and surface integrals for the computational cell should use as many common Gauss nodes as possible.

The latter two requirements are explained by desire to minimize the quantity of nodes, where it is necessary to calculate the integrands.

The review of existing rules may be found at site [15]. Sufficient theoretical ideas for quadrature rule problem can be found in the books [16,17]. Also, the book [18] is of important applied significance. In particular, it is shown in [18] that minimal quantity of Gauss nodes for quadrature formula with m-property is equal to

$$N_{\min} = \chi = M\,(n,\, k)\,, \tag{7}$$

where $k = [m/2]$. Formula (7) means that quantity of Gauss points N may be less than quantity of equations in system (6). In the same book, it is also shown that quadrature rule with minimal quantity of Gauss nodes $N = \chi$ has

$(2k+1)$-property, if and only if these points are common roots of all orthogonal polynomials of degree $(k+1)$.

Some methods presented in [18] and [17] permit to obtain quadrature formulas with the least number of nodes. One of such methods is connected with transformation of rotation groups. Resulting formulas with minimal number of nodes are presented in [18] and [15]. However, the quadrature rules with rotational symmetry don't seem optimal since one have to use separate nodes for calculating volume and surface integrals. Therefore an attempt to construct new rules was undertaken.

The following "straightforward" strategy was applied. Some quantity of Gauss nodes $N < \psi$ were taken in the volume and on the surface. Their coordinates x_j were substituted into system (6). If the system (6) had no solution, some new nodes were added or removed until a solution is obtained or until the number of nodes became too large. In the choice of Gauss nodes, the preference was given to the common roots of all orthogonal polynomials of degree $(k+1)$ and to points on the cell surface (that may be used both for volume integrals and for surface integrals). Considerations of symmetry and some additional intuitive ideas were also applied.

As a result, in the case of $K = 1$, the quadrature rule for volume integral with property $m = 5$ and with $N = 15$ nodes has been found. The combination of this formula with formulas for all surface integrals of the cell uses 21 nodes. This combination is essentially better than analogous formulas in the current DG realization.

An attempt to find a new quadrature formulas for the cases $K = 2$ and $K = 3$ has been performed. For that, orthogonal bases with polynomials of higher degrees have been created. But a more efficient system of nodes hasn't been found. Detailed investigation of the already existing quadrature formulas has permitted to find appropriate series of nodes for $K = 2$ and $K = 3$. Thus, the following combinations of quadrature formulas have been proposed for use in DG method:

	For sides (Σ) $n = 2$	For volumes (Ω) $n = 3$
$K = 1$	$m = 3$, $N = 5$	$m = 5$, $N = 15$
$K = 2$	$m = 5$, $N = 7$	$m = 7$, $N = 31$
$K = 3$	$m = 7$, $N = 14$	$m = 9$, $N = 57$

As a test for comparison of working times of initial and modified programs, a calculation of inviscid compressible subsonic flow around a sphere has been chosen. Ambient flow velocity was constant and equal to 50 m/sec and it has been directed along x axis. Pressure and temperature of ambient flow were equal to 10^5 Pa and 293 K, correspondingly. In accordance with these parameters, Mach number of ambient flow was equal to 0.15 and gas density – to 11.9 kg/m^3.

Non-structured grids generated around a quarter of unit radius sphere have been used for the calculations. Grids have been built using automatic mesh generator NUMECA/HEXPRESS. Grid cells were hexahedrons. Symmetry condition has been chosen at the block boundaries intersected the sphere. Normal velocity at the sphere surface was equal to zero. Ambient flow conditions were given at outer boundaries. The calculation domain was a parallelepiped $[20, 20] \times [0, 20] \times [0, 20]$ and consisted of single block.

The calculations have been performed for $K = 1, 2, 3$. For each K, an individual grid has been chosen to have approximately equal number of degrees of freedom. Fragments of these free grids are presented in Figure 1.

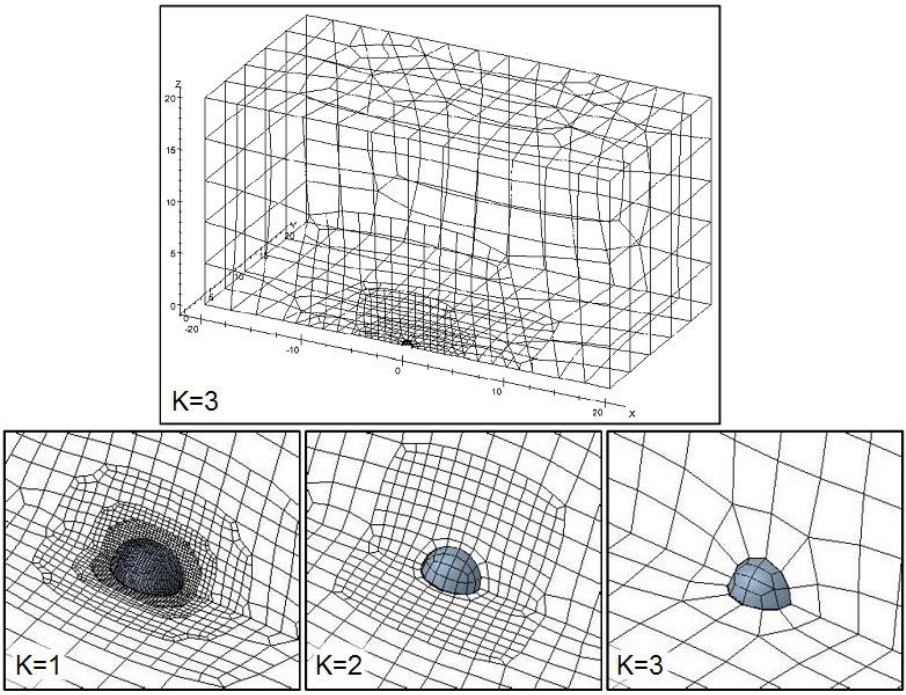


Fig. 1. Grids around sphere for testing the efficiency of quadrature rules

Two approaches to improve the algorithm work efficiency have been considered: 1) diminishing the volume of calculations due to using alternative formulas with lower number of Gauss nodes; 2) diminishing the calculations due to using common nodes of quadrature formulas both for volume and for surface integrals (so-called economy regime). It should be noted that economy regime has been efficiently realized only for inner faces whereas outer sides are calculated without economy.

The obtained results for calculation times for fixed number of iterations ($N = 2000$) are presented in the following table.

polynomial degree		$K = 1$	$K = 2$	$K = 3$	$K = 3^*$
calculation time, sec	initial quadrature formulas	2016	1758	2924	2859
	new quadrature formulas	1991	1439	2436	2388
	new formulas in economy regime	1985	–	2402	2420
acceleration due to lower number of quadrature nodes, %		1.2	18.1	16.7	15.3
acceleration due to using common quadrature nodes, %		0.3	–	1.1	1.2
total acceleration, %		1.5	18.1	17.8	16.5

It is obvious that the new formulas provides a reliable gain that more than 15%. The algorithm with using part of Gauss nodes for calculating of both volume and surface integrals, doesn't result in visible economy of arithmetic operations.

4 Implicit Numerical Method for Solution of Stationary RANS Equations

Within IDIHOM project, TsAGI has developed an implicit numerical method to find stationary solutions of full 3D RANS equations, closed by non-Boussinesq model of turbulence – EARSM model, based on [19]. Such turbulence models provide good compromise between the calculation efficiency and its accuracy. As well as in two-equation Boussinesq models, EARSM models use only two additional differential equations, so that the memory requirements don't grow and the time of computation increases unessentially. But more general (than in Boussinesq approach) epxressions for Reynolds stresses allow to represent additional effects like the anisotropy of turbulent transport and the influence of the streamline curvature. In particular, it can improve the description of separation zones [19].

Additional differential equations of the chosen EARSM model are the equation for the turbulence kinetic energy k and the equation for the natural logarithm of characteristic frequency of turbulent fluctuations $\tilde{\omega} = \ln(\omega/\omega_0)$. Logrаithmic formulation of the second equation allows to satisfy the condition $\omega > 0$ and to improve the quality of polynomial reconstruction in near-wall cells, where the parameter ω varies by several orders of magnitude. Value of $\omega_0 = 1$ Hz was chosen arbitrarily.

In the case of RANS equations closed by this EARSM model, system dimension $N = 7$, the vector of primitive variables $\mathrm{Q} = [\rho;\, u;\, v;\, w;\, p;\, k;\, \tilde{\omega}]^{\mathrm{T}}$, and the vector of conservative variables $\mathrm{U} = [\rho;\, \rho\,u;\, \rho\,v;\, \rho\,w;\, \rho\,E;\, \rho\,k;\, \rho\,\tilde{\omega}]^{\mathrm{T}}$. The vector

F_k of fluxes along x_k axis ($x_1 = x$, $x_2 = y$, $x_3 = z$) and the vector of source terms S (see (1)) have the following form:

$$\mathrm{F}_k = \begin{bmatrix} \rho u_k \\ \rho u_i u_k + p\delta_{ik} + \tau_{ik} + \rho R_{ik} \\ \rho E u_k + p u_k + (\tau_{ik} + \rho R_{ik})\, u_i - (\frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t}) c_p \frac{\partial T}{\partial x_k} \\ \rho k u_k - (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_k} \\ \rho \tilde{\omega} u_k - (\mu + \sigma_\omega \mu_t) \frac{\partial \tilde{\omega}}{\partial x_k} \end{bmatrix}, \; \mathrm{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \rho k (\tilde{P} - \beta^* \omega_0 e^{\tilde{\omega}}) \\ D_\omega + \rho(\gamma P - \beta \omega_0 e^{\tilde{\omega}}) \end{bmatrix}.$$

Here and below the rule about summation over repeating indices is used.

Closing relations and the values of constants:

$$p = \rho R T, \quad \tau_{ij} = -\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_m}{\partial x_m} \delta_{ij} \right), \quad E = c_v T + \frac{u_k u_k}{2} + k,$$

$$R = 287 \frac{\mathrm{J}}{\mathrm{kg} \cdot \mathrm{K}}, \quad c_p = \frac{g}{g-1} R, \;\; c_v = \frac{1}{g-1} R, \;\; g = 1.4, \;\; \mathrm{Pr} = 0.72, \;\; \mathrm{Pr}_t = 0.9,$$

$$\mu = \mu_r \left(\frac{T}{T_r} \right)^{3/2} \frac{T_r + T_s}{T + T_s}, \quad \mu_r = 1.884 \cdot 10^{-5} \, \frac{\mathrm{kg}}{\mathrm{m} \cdot \mathrm{s}}, \quad T_r = 293 \, \mathrm{K}, \quad T_s = 110 \, \mathrm{K},$$

$$\mu_t = \max(\rho \frac{k}{\omega_0} e^{-\tilde{\omega}}, 0), \; P = -a_{ij} \frac{\partial u_i}{\partial x_j}, \; \tilde{P} = \min(P, 10\beta^* \omega_0 e^{\tilde{\omega}}), \; R_{ij} = k a_{ij}, \quad (8)$$

$$a_{ij} = \frac{2}{3}\delta_{ij} + \beta_1^* \tilde{S}_{ij} + \beta_3 (\tilde{\Omega}_{ik}\tilde{\Omega}_{kj} - \frac{1}{3} II_{\tilde{S}} \delta_{ij}) + \beta_4 (\tilde{S}_{ik}\tilde{\Omega}_{kj} - \tilde{\Omega}_{ik}\tilde{S}_{kj}) + \\ + \beta_6 (\tilde{S}_{ik}\tilde{\Omega}_{kl}\tilde{\Omega}_{lj} + \tilde{\Omega}_{ik}\tilde{\Omega}_{kl}\tilde{S}_{lj} - \frac{2}{3} IV \delta_{ij} - II_{\tilde{\Omega}} \tilde{S}_{ij}), \quad (9)$$

$$\tilde{S}_{ij} = \frac{\tau_t}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \; \tilde{\Omega}_{ij} = \frac{\tau_t}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \; \tau_t = \max \left(\frac{e^{-\tilde{\omega}}}{\beta^* \omega_0}, \, 6 \cdot \sqrt{\frac{\mu e^{-\tilde{\omega}}}{\beta^* \rho k \omega_0}} \right), \quad (10)$$

$$II_{\tilde{S}} = \tilde{S}_{ij}\tilde{S}_{ji}, \quad II_{\tilde{\Omega}} = \tilde{\Omega}_{ij}\tilde{\Omega}_{ji}, \quad IV = \tilde{S}_{ik}\tilde{\Omega}_{kj}\tilde{\Omega}_{ji}, \quad \beta_1 = -\frac{N}{Q}, \quad \beta_1^* = \min(\beta_1, -0.15),$$

$$\beta_3 = -\frac{2 \cdot IV}{N \cdot Q_1}, \quad \beta_4 = -\frac{1}{Q}, \quad \beta_6 = -\frac{N}{Q_1}, \quad Q = \frac{N^2 - 2II_{\tilde{\Omega}}}{A_1}, \quad Q_1 = \frac{Q}{6}(2N^2 - II_{\tilde{\Omega}}),$$

$$N = C_1' + \frac{9}{4}\sqrt{2\beta^* II_{\tilde{S}}}, \quad D_\omega = (\mu + \sigma_\omega \mu_t) \frac{\partial \tilde{\omega}}{\partial x_k} \frac{\partial \tilde{\omega}}{\partial x_k}, \quad A_1 = 1.245, \quad C_1' = 1.8,$$

$$\sigma_k = \sigma_\omega = 0.5, \quad \beta = 0.075, \quad \beta^* = 0.09, \quad \gamma = \frac{\beta}{\beta^*} - \frac{\sigma_\omega \kappa^2}{\sqrt{\beta^*}}, \quad \kappa = 0.41.$$

In comparison with [19], the described here turbulence model doesn't include the blending function for transition from near-wall turbulence to free turbulence, because the considered flows are characterized by the dominance of near-wall trubulence. On the same reason, the equation for $\tilde{\omega}$ doesn't contain the so-called "cross diffusion" [20]. It was found that turning off the cross diffusion allows to

smooth the k and $\tilde{\omega}$ distributions at the outer boundaries of turbulent regions, that improves the convergence of calculations. One more modification is omitting the limiter of P in equation for $\tilde{\omega}$; this modification increases the stability of calculations. The fourth modification is the use of limiter for coefficient β_1, that allows the model to keep its diffusive properties in regions with non-equilibrium turbulence, where the coefficient β_1 can become too far from its equilibrium value (-0.18).

To obtain stationary solutions of the RANS equations closed by EARSM model, the implicit numerical method is applied. Let's introduce the increment operator for an arbitrary value b. This operator corresponds to transition from the time layer n to the time layer $(n+1)$: $\Delta b \equiv b^{n+1} - b^n$. Primitive variables at the known time layer are represented as an expansion over basis functions (3): $\mathrm{Q}^n = \sum\limits_{j=1}^{K_f} \mathrm{q}_j^n \varphi_j(\mathbf{x})$, and primitive variables at an arbitrary moment t are represented as $\mathrm{Q}(\mathbf{x},t) = \mathrm{Q}^n + \Delta\mathrm{Q}(\mathbf{x},t)$, where

$$\Delta\mathrm{Q}(\mathbf{x},t) = \sum_{j=1}^{K_f} \Delta\mathrm{q}_j(t)\varphi_j(\mathbf{x}). \tag{11}$$

The coefficients $\Delta\mathrm{q}_j$ are to be found from the solution of approximate analogue of the equation system (4). The matrix Γ in non-stationary term of equation (4) is taken from the known time layer and the time derivative is approximated with the first accuracy order:

$$\Gamma\frac{\partial \mathrm{Q}}{\partial t} \approx \Gamma^n \cdot \frac{\mathrm{Q}^{n+1} - \mathrm{Q}^n}{\tau} = \Gamma^n \cdot \frac{\Delta\mathrm{Q}}{\tau}. \tag{12}$$

The fluxes through the cell face with the number s (that belongs to the surface Σ) are approximated at the unknown time layer $(n+1)$. The fluxes can be represented as a sum of convective and diffusive fluxes: $\mathrm{F}_n(\mathrm{Q}_s, \mathbf{G}_s) = \mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s) + \mathrm{F}_n^{\mathrm{diff}}(\mathrm{Q}_s, \mathbf{G}_s)$.

The following linearized representation is used for convective fluxes $\mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s)$:

$$\mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s^{n+1}) \approx \mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s^n) + \left(\frac{\partial \mathrm{F}_n^{\mathrm{conv}}}{\partial \mathrm{U}}\right)^n \left(\frac{\partial \mathrm{U}}{\partial \mathrm{Q}}\right)^n \Delta\mathrm{Q}_s = \\ = \mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s^n) + A_n\Gamma_s^n\Delta\mathrm{Q}_s, \tag{13}$$

where $A_n \overset{def}{=} \frac{\partial \mathrm{F}_n^{\mathrm{conv}}}{\partial \mathrm{U}}(\mathrm{Q}_s^n) = \frac{\partial \mathrm{F}_x^{\mathrm{conv}}}{\partial \mathrm{U}} n_x + \frac{\partial \mathrm{F}_y^{\mathrm{conv}}}{\partial \mathrm{U}} n_y + \frac{\partial \mathrm{F}_z^{\mathrm{conv}}}{\partial \mathrm{U}} n_z$. It is a square matrix of size $N \times N$. This matrix is calculated using the parameters at the known time layer n. The method to calculate both the values $\mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s^n)$ and $A_n\Gamma_s^n\Delta\mathrm{Q}_s$ is based on the solution of Riemann problem about decay of an arbitrary discontinuity. Roe's linearized solution [21] is used:

$$\mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s^n) = \frac{\mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_L^n) + \mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_R^n)}{2} - |A_n|\frac{\mathrm{U}(\mathrm{Q}_R^n) - \mathrm{U}(\mathrm{Q}_L^n)}{2}, \tag{14}$$

where Q_L and Q_R are the gas parameters in the vicinity of Gauss node from the two sides of the cell face. The matrix $|A_n| = T|\Lambda|T^{-1}$, where the columns of matrix T are the right eigenvectors of A_n, and $|\Lambda|$ is diagonal matrix with the modules of A_n eigenvalues on the diagonal. Components of the matrix A_n, and accordingly components of T and $|\Lambda|$, are calculated at the known time layer n using Roe averagings $\bar{\mathrm{V}}_i = \frac{\sqrt{\rho_L}\cdot(\mathrm{V}_i)_L + \sqrt{\rho_R}\cdot(\mathrm{V}_i)_R}{\sqrt{\rho_L}+\sqrt{\rho_R}}$, where $\mathrm{V}_i = \{u,\, v,\, w,\, H,\, k,\, \tilde{\omega}\}$, $H = E + p/\rho$. To understand the structure of the term $\Gamma_s^n \Delta\mathrm{Q}_s$, let's consider Roe solution for the vector of conservative parameters U at the cell face:

$$\mathrm{U}_s = \frac{\mathrm{U}(\mathrm{Q}_L) + \mathrm{U}(\mathrm{Q}_R)}{2} - \operatorname{sign} A_n \frac{\mathrm{U}(\mathrm{Q}_R) - \mathrm{U}(\mathrm{Q}_L)}{2}, \tag{15}$$

where $\operatorname{sign} A_n = T\cdot\operatorname{sign}\Lambda\cdot T^{-1}$ and $\operatorname{sign}\Lambda$ is diagonal matrix containing the signs of A_n eigenvalues. It is assumed that (15) is valid for each time moment within one time step; but the matrix $\operatorname{sign} A_n$ is taken to be constant and is calculated at the known time layer n. Then the term $\Gamma_s^n \Delta\mathrm{Q}_s$ can be represented as follows:

$$\Gamma_s^n \Delta\mathrm{Q}_s \approx \Delta\mathrm{U}_s = \frac{\Delta\mathrm{U}_L + \Delta\mathrm{U}_R}{2} - \operatorname{sign} A_n \frac{\Delta\mathrm{U}_R - \Delta\mathrm{U}_L}{2} \approx$$
$$\approx I_n^+ \Gamma(\mathrm{Q}_L^n)\Delta\mathrm{Q}_L + I_n^- \Gamma(\mathrm{Q}_R^n)\Delta\mathrm{Q}_R, \tag{16}$$

where matrices $I_n^+ = \frac{I+\operatorname{sign} A_n}{2}$ and $I_n^- = \frac{I-\operatorname{sign} A_n}{2}$ are introduced, I is a unit matrix. Let's substitute the expansions $\Delta\mathrm{Q}_L$ and $\Delta\mathrm{Q}_R$ by basis functions: $\Delta\mathrm{Q}_L = \sum\limits_{j=1}^{K_f} \Delta\mathrm{q}_{jL}\varphi_{jL}(\mathbf{x})$ and $\Delta\mathrm{Q}_R = \sum\limits_{j=1}^{K_f} \Delta\mathrm{q}_{jR}\varphi_{jR}(\mathbf{x})$ into (16). Here φ_{jL} are basis functions for the cell to the left from the given face and φ_{jR} are basis functions for the cell to the right from the given face. One of these cells coincides with the current cell and the other cell is neighboring one. Finally we get

$$\Gamma_s^n \Delta\mathrm{Q}_s = \sum_{j=1}^{K_f} \left(\varphi_{jL}(\mathbf{x}) I_n^+ \Gamma(\mathrm{Q}_L^n)\Delta\mathrm{q}_{jL} + \varphi_{jR}(\mathbf{x}) I_n^- \Gamma(\mathrm{Q}_R^n)\Delta\mathrm{q}_{jR}\right). \tag{17}$$

So, the approximation of $\mathrm{F}_n^{\mathrm{conv}}(\mathrm{Q}_s^{n+1})$ (see (13)) depends on the values of $\Delta\mathrm{q}_j$ in the current cell and in the neighboring cell from the other side of the face s.

Diffusive fluxes are also linearized:

$$\mathrm{F}_n^{\mathrm{diff}}(\mathrm{Q}_s^{n+1},\, \mathbf{G}_s^{n+1}) \approx \mathrm{F}_n^{\mathrm{diff}}(\mathrm{Q}_s^n,\, \mathbf{G}_s^n) + J_{nx}\Delta(\mathrm{G}_x)_s + J_{ny}\Delta(\mathrm{G}_y)_s + J_{nz}\Delta(\mathrm{G}_z)_s, \tag{18}$$

where $J_{nk} = \left(\partial \mathrm{F}_n^{\mathrm{diff}}/\partial \mathrm{G}_k\right)_s^n$. Parameters at the cell face $\mathrm{Q}_s^{\mathrm{n}}$ appear in diffusive fluxes (18) in expressions for coefficients of molecular and turbulent diffusion. They are calculated using symmetric formula

$$\mathrm{Q}_s^n = \frac{1}{2}\left(\mathrm{Q}_L^n + \mathrm{Q}_R^n\right). \tag{19}$$

These "diffusive" values of parameters at the cell face differ from "convective" values that may be extracted from (15). The use of two different approximations

for parameters at the cell face corresponds to different nature of convection and diffusion. Convective fluxes are connected with directional propagation of information (along characteristics) and are approximated by asymmetric formulas like (15). Diffusive fluxes describe symmetric propagation of information and are approximated by symmetric formulas like (19). It is important to note that the term $(\tau_{ik} + \rho R_{ik})u_i$ in the energy equation is treated as diffusive flux, and u_i in this term are also averaged according to (19). Components of matrices J_{nk} (see (18)) are also taken from (19).

Special feature of DG method is that gradients $\mathbf{G}$ cannot be calculated by strict differentiation of the parameter distribution in the cells [22]. Discontinuities of parameter distributions on the cell faces lead to local loss of gradients approximation. For the determination of gradients with high order of accuracy, several methods have been proposed [4, 23, 24]. Comparison of these methods on the basis of scalar model equation [25] have shown the advantages of the so-called BR2 method [23]. It is this method of gradients approximation that is used here for $\mathbf{G}$ and $\Delta\mathbf{G}$. In this method, gradients are represented as follows:

$$\mathbf{G}(\mathbf{x}) = \sum_{j=1}^{K_f} \mathrm{q}_j \nabla\varphi_j(\mathbf{x}) + \sum_s \mathbf{R}_s(\mathbf{x}). \tag{20}$$

The first term in (20) is obtained by strict differentiation of the parameter distribution in the current cell. $\mathbf{R}_s(\mathbf{x})$ is a correction because of discontinuous behavior of the function $\mathrm{Q}(\mathbf{x})$ at the face with the number s. $\mathbf{R}_s(\mathbf{x})$ is nullified at all faces of the current cell except the face with the number s. The equation for $\mathbf{R}_s(\mathbf{x})$ can be found in [23]; finally, $\mathbf{R}_s(\mathbf{x})$ depends on values of q_j in the current cell and in the neighboring cell from the other side of face s. For gradients at this face at the known time layer n, $\mathbf{G}_s^n$, and for their increments, $\Delta\mathbf{G}_s$, symmetric formulas are used:

$$\mathbf{G}_s^n = \frac{1}{2}\left(\mathbf{G}_L^n + \mathbf{G}_R^n\right), \quad \Delta\mathbf{G}_s = \frac{1}{2}\left(\Delta\mathbf{G}_L + \Delta\mathbf{G}_R\right). \tag{21}$$

$\mathbf{G}_L$ and $\mathbf{G}_R$ are calculated using (20). But due to nullification of the contributions from all faces except the current face, $\mathbf{G}_s$ depends only on the values of q_j in the current cell and in the neighboring cell from the other side of the face s. As a result, the approximation of $\mathrm{F}_n^{\mathrm{diff}}(\mathrm{Q}_s^{n+1})$ (see (18)) depends on the values of $\Delta\mathrm{q}_j$ in the current cell and in the neighboring cell from the other side of the face s.

It is necessary to note that the *h-p* multigrid technology, which is used this work, requires to perform some iterations with 1^{st} order of accuracy ($K = 0$). Original BR2 method looses approximation at $K = 0$ and gives two times smaller values of gradients at cell faces. To repair these errors, BR2 gradients at the cell faces were doubled. Gradients inside the cells (which are used in source terms of turbulence model) were not modified.

The fluxes that are appeared in the term $\int\limits_{\Omega} \mathrm{F}_i \, d\Omega$ (see (4)) are also approximated at the unknown time layer $n+1$. At that, the same principles are used as in approximation of the term $\oint\limits_{\Sigma} \mathrm{F}_n \, \varphi_i d\Sigma$:

$$\mathrm{F}_i(\mathrm{Q}^{n+1}, \mathbf{G}^{n+1}) \approx \mathrm{F}_i^{\mathrm{conv}}(\mathrm{Q}^n) + \mathrm{F}_i^{\mathrm{diff}}(\mathrm{Q}^n, \mathbf{G}^n) + A_i \Gamma^n \Delta \mathrm{Q} + \\ + R_{ix} \Delta \mathrm{G}_x + R_{iy} \Delta \mathrm{G}_y + R_{iz} \Delta \mathrm{G}_z, \quad (22)$$

where $A_i \overset{def}{=} \frac{\partial \mathrm{F}_i^{\mathrm{conv}}}{\partial \mathrm{U}}(\mathrm{Q}_s^n) = \frac{\partial \mathrm{F}_x^{\mathrm{conv}}}{\partial \mathrm{U}} \frac{\partial \varphi_i}{\partial x} + \frac{\partial \mathrm{F}_y^{\mathrm{conv}}}{\partial \mathrm{U}} \frac{\partial \varphi_i}{\partial y} + \frac{\partial \mathrm{F}_z^{\mathrm{conv}}}{\partial \mathrm{U}} \frac{\partial \varphi_i}{\partial z}$, $R_{ik} = \left(\frac{\partial \mathrm{F}_i^{\mathrm{diff}}}{\partial \mathrm{G}_k} \right)_s^n$. Contrary to the cell sides, distributions of parameters inside the cell is assumed to be continuous. So, the values of Q^n and components of matrices A_i, Γ^n and R_{ik} are taken from corresponding Gauss nodes at the known time layer n. Gradients $\mathbf{G}^n$ and their increments $\boldsymbol{\Delta}\mathbf{G}$ are calculated using BR2 method [23]. According to (20), they contain contributions from the current cell and from all adjacent neighboring cells.

It is important to construct unconditionally stable approximation for the source term $\int\limits_{\Omega} \mathrm{S} \varphi_i \, d\Omega$ in (4). Let's linearize the source terms near the known time layer n. At that, gradients appearing in the source terms are fixed at the known time layer:

$$\mathrm{S}(\mathrm{Q}, \mathbf{G}) \approx \mathrm{S}(\mathrm{Q}, \mathbf{G}^n) \approx \mathrm{S}(\mathrm{Q}^n, \mathbf{G}^n) + (\partial \mathrm{S}/\partial \mathrm{Q})^n \cdot (\mathrm{Q} - \mathrm{Q}^n), \quad (23)$$

where $\partial \mathrm{S}/\partial \mathrm{Q}$ is Jacobian matrix of source terms. It is known [26] that source terms, linearly depending upon U, cause exponentially time-varying solution modes like $\exp(\lambda_i t)$, where λ_i are eigenvalues of $\partial \mathrm{S}/\partial \mathrm{Q}$. Let's rewrite (23) as follows:

$$\mathrm{S}(\mathrm{Q}, \mathbf{G}) \approx \mathrm{S}(\mathrm{Q}^n, \mathbf{G}^n) + (\partial \mathrm{S}/\partial \mathrm{Q})^n \cdot (\mathrm{Q} - \mathrm{Q}^n) = \\ = \mathrm{S}(\mathrm{Q}^n, \mathbf{G}^n) + T \Lambda T^{-1} \cdot (\mathrm{Q} - \mathrm{Q}^n) = \mathrm{S}(\mathrm{Q}^n, \mathbf{G}^n) + \\ + T \Lambda^+ T^{-1} \cdot (\mathrm{Q} - \mathrm{Q}^n) + T \Lambda^- T^{-1} \cdot (\mathrm{Q} - \mathrm{Q}^n).$$

Diagonal matrix Λ consists of eigenvalues λ_i of the matrix $(\partial \mathrm{S}/\partial \mathrm{Q})^n$. T is the matrix, which columns are right eigenvectors of the matrix $\partial \mathrm{S}/\partial \mathrm{Q}$. Diagonal matrix Λ^+ contains only positive eigenvalues λ_i; other eigenvalues are replaced by zeroes; in diagonal matrix Λ^- only negative eigenvalues λ_i are kept. Let's introduce the matrices $J_{\mathrm{S}}^+ = T \Lambda^+ T^{-1}$ and $J_{\mathrm{S}}^- = T \Lambda^- T^{-1}$. Then the source term $\mathrm{S}^-(\mathrm{Q}) = J_{\mathrm{S}}^- \cdot (\mathrm{Q} - \mathrm{Q}^n)$ causes exponentially increasing solution modes and the source term $\mathrm{S}^-(\mathrm{Q}) = J_{\mathrm{S}}^- \cdot (\mathrm{Q} - \mathrm{Q}^n)$ causes exponentially decreasing solution modes.

The implicit approximation $\mathrm{S}^-(\mathrm{Q}) \approx \mathrm{S}^-(\mathrm{Q}^{n+1})$ is unconditionally stable for description of the exponentially decreasing solution modes. On the contrary, the explicit approximation $\mathrm{S}^+(\mathrm{Q}) \approx \mathrm{S}^+(\mathrm{Q}^n)$ is unconditionally stable for description of the exponentially increasing solution modes. Hence, if the matrix $\partial \mathrm{S}/\partial \mathrm{Q}$ has both positive and negative eigenvalues, the following approximation is unconditionally stable:

$$\mathrm{S}(\mathrm{Q}, \mathbf{G}) \approx \mathrm{S}(\mathrm{Q}^n, \mathbf{G}^n) + J_{\mathrm{S}}^- \Delta \mathrm{Q}. \quad (24)$$

Finally, let's substitute (12), (13), (18), (22), and (24) into (4). Using representation of $\Delta \mathrm{Q}$ in the form (11), we finally obtain the following system of equations:

$$\sum_{j=1}^{K_f} \left(\int\limits_{\Omega} \Gamma^n \, \varphi_i(\mathbf{x}) \varphi_j(\mathbf{x}) d\Omega \right) \frac{\Delta \mathrm{q}_j}{\tau} + \\ + \left\{ - \sum_{j=1}^{K_f} \left(\int\limits_{\Omega} A_i \Gamma^n \, \varphi_j(\mathbf{x}) d\Omega + \int\limits_{\Omega} J_{\mathrm{S}}^{-} \, \varphi_i(\mathbf{x}) \varphi_j(\mathbf{x}) d\Omega \right) \Delta \mathrm{q}_j + \right. \\ + \oint\limits_{\Sigma} A_n \Gamma_s^n \Delta \mathrm{Q}_s \, \varphi_i(\mathbf{x}) d\Sigma - \int\limits_{\Omega} \left(R_{ix} \Delta \mathrm{G}_x + R_{iy} \Delta \mathrm{G}_y + R_{iz} \Delta \mathrm{G}_z \right) \, d\Omega + \\ \left. + \oint\limits_{\Sigma} \left(R_{nx} \Delta (\mathrm{G}_x)_s + R_{ny} \Delta (\mathrm{G}_y)_s + R_{nz} \Delta (\mathrm{G}_z)_s \right) \, \varphi_i(\mathbf{x}) d\Sigma \right\} = \\ = \int\limits_{\Omega} \left(\mathrm{F}_i(\mathrm{Q}^n, \mathbf{G}^n) + \mathrm{S}(\mathrm{Q}^n, \mathbf{G}^n) \, \varphi_i(\mathbf{x}) \right) \, d\Omega - \oint\limits_{\Sigma} \mathrm{F}_n(\mathrm{Q}_s^n, \mathbf{G}_s^n) \, \varphi_i(\mathbf{x}) d\Sigma. \quad (25)$$

Let's name as an *implicit* operator the terms in the left part of the equation (25) that depend upon the parameter increments (between the known time layer and the unknown time layer). Let's name as an *explicit* operator the terms in the right part of the equation (25) that depend upon the parameters at the known time layer. If the implicit operator terms inside the braces are removed, then an explicit scheme of the first accuracy order in time is obtained. This scheme is stable in the case of restricted time steps τ only. The implicit operator terms inside the braces are intended to provide unconditional stability of the scheme, i.e. to smooth perturbations because of instability of explicit operator. So, these terms (inside the braces) may be named as an *implicit smoother*.

In approaching the stationary solution, the implicit operator tends to zero. Only the stationary solution, which is defined by the explicit operator, is practically interesting. Therefore, the explicit operator approximation has to provide the given high accuracy order in space. The implicit operator has the first accuracy order in time only and it has to vanish, when the stationary solution is achieved. Therefore, all simplifications of implicit operator approximation are admissible, including the simplifications that decrease the spatial accuracy order of this operator.

The formula (25) is a system of linear algebraic equations for searching $\Delta \mathrm{q}_j$. For each cell, (25) is a set of $K_f \cdot N$ equations, because the vector $\Delta \mathrm{q}_j$ consists of N components, and because the equation (25) is obtained from (1) by multiplication by basis function φ_i $(i = 1, ..., K_f)$ and by following integration over the cell. Let's introduce the vectors of size $K_f \cdot N$ for each cell:

$$\overrightarrow{\Delta \mathrm{q}} \equiv \left((\Delta \mathrm{q}_1)_\rho ; (\Delta \mathrm{q}_2)_\rho ; ... ; \left(\Delta \mathrm{q}_{K_f} \right)_\rho ; ... ; (\Delta \mathrm{q}_1)_{\tilde{\omega}} ; (\Delta \mathrm{q}_2)_{\tilde{\omega}} ; ... ; \left(\Delta \mathrm{q}_{K_f} \right)_{\tilde{\omega}} \right)^{\mathrm{T}},$$

$$\overrightarrow{\mathrm{R}} \equiv \Big((\mathrm{R}_1)_\rho ; (\mathrm{R}_2)_\rho ; ... ; \left(\mathrm{R}_{K_f}\right)_\rho ; (\mathrm{R}_1)_u ; (\mathrm{R}_2)_u ; ... ; \left(\mathrm{R}_{K_f}\right)_u ; ... ; (\mathrm{R}_1)_{\tilde{\omega}} ; (\mathrm{R}_2)_{\tilde{\omega}} ; ... ; \left(\mathrm{R}_{K_f}\right)_{\tilde{\omega}} \Big)^{\mathrm{T}},$$

where each component of the vector $\overrightarrow{\mathrm{R}}$ is the right-hand side (explicit operator) of corresponding equation in the set (25). After substituting the approximations for $\Gamma_s^n \Delta \mathrm{Q}_s$ (17) and for $\Delta(\mathrm{G}_k)_s$ (see (21) and (20)) into (25), one may finally reduce (25) to the following form:

$$D(c)\overrightarrow{\Delta \mathrm{q}}(c) + \sum_s H_s(c)\,\overrightarrow{\Delta \mathrm{q}}\,(c_s(c)) = \overrightarrow{\mathrm{R}}(c), \tag{26}$$

where c is the current cell, $c_s(c)$ is the neighboring cell adjacent to the current cell c from the other side of the face s. $D(c)$ and $H_s(c)$ are square matrices of size $(K_f \cdot N) \times (K_f \cdot N)$ which are calculated using parameters at the known time layer n.

Initially it was supposed to solve the equation system (26) using an iterative blockwise Gauss-Seidel method [27]. According to TsAGI experience with finite-volume methods, this method provides an acceptable convergence rate of iterations at low CPU and memory costs per iteration.

5 Achievement of Stability and Fast Convergence of the Implicit Method for Stationary RANS Equations

Essential efforts have been paid to provide stability and fast convergence of the described above implicit DG method. Improvements, which have been introduced into the code to achieve these results, may be subdivided into corrections of turbulence model and modifications of numerical scheme.

The following corrections have been made in turbulence model to raise the stability of calculations:

1. Turbulent time scale τ_t (9) can be rewritten as follows:

$$\tau_t = \max\left(\frac{e^{-\tilde{\omega}}}{\beta^*\omega_0},\ 6\cdot\sqrt{\frac{\mu e^{-\tilde{\omega}}}{\beta^*\rho k\omega_0}}\right) = \frac{e^{-\tilde{\omega}}}{\beta^*\omega_0}\cdot\max\left(1,\ C_\tau\right), \quad C_\tau = 6\cdot\sqrt{\frac{\mu e^{\tilde{\omega}}\beta^*\omega_0}{\rho k}}.$$

   In calculation of τ_t the coefficient C_τ is replaced by $1000\cdot\tanh(C_\tau/1000)$, so that its value is softly bounded by 1000. This trick smoothes the behavior of source terms in the viscous sublayer of turbulent boundary layer.
2. Coefficients β_i, $i = 1, 3, 4, 6$ (formula (10)) are replaced by $10\cdot\tanh(\beta_i/10)$, so that their values are softly held within $(-10, 10)$.
3. Components of undimensional tensors $\tilde{S}_{ij}$ and $\tilde{\Omega}_{ij}$ are damped using the smoothing factor D_s:

$$\tilde{S}_{ij} = \tilde{S}_{ij}\cdot D_S, \quad \tilde{\Omega}_{ij} = \tilde{\Omega}_{ij}\cdot D_S, \quad D_S = \frac{10}{\sqrt{II_{\tilde{S}}}}\tanh\left(\frac{\sqrt{II_{\tilde{S}}}}{10}\right).$$

4. In calculation of the Reynolds stress tensor components (8), value of k is bounded from below by zero: $R_{ij} = \max(k, 0)\, a_{ij}$.
5. If in some Gauss point the conditions $S_{\rho k} = \rho k(\tilde{P} - \beta^* \omega_0 e^{\tilde{\omega}}) < 0$ and $k < 0$ are satisfied simultaneously, then the source term in equation for ρk is nullified. This correction prevents the following shift of k towards the negative area.

Now let's proceed to the modifications introduced in the numerical method. Numerical effects connected with these modifications were studied in series of inner tests, based on 2D flow with parameters $p = 1$ atm, $T = 300$ K, $V = 270$ m/s along the flat plate with length 1 m. Two computational grids, which were used for these calculations, have been built using automatic mesh generator NUMECA/HEXPRESS. They are shown in Figure 2. Upper picture show the general view that was the same for both grids. The plate was placed at the middle of the lower boundary; outer boundaries were placed at the distance 15 m from the plate. Below one may see the details of grids near the plate leading edge. First grid was designed for calculations of laminar boundary layer on the basis of Navier-Stokes equations. It consisted of 1239 cells; vertical size of the smallest cells (near the plate) was equal to $2 \cdot 10^{-3}$ m. Second grid was designed for calculations of turbulent boundary layer on the basis of RANS equations. It consisted of 5185 cells; vertical size of the smallest cells (near the plate) was equal to $2 \cdot 10^{-6}$ m. In addition, some calculations were performed on the basis of Euler equations. In Euler tests, the first ("laminar") grid was used; slip boundary condition was imposed on the lower boundary. In Euler tests, initial field contained Gauss-shaped perturbation of density with characteristic radius 1.0 m and with amplitude 10% of the density of undisturbed flow (in the center of computational domain). The task was to simulate the drift of this perturbation with the flow. In Navier-Stokes and RANS tests, uniform initial field was imposed, no-slip boundary condition was used on the lower boundary, and the task was to obtain the stationary boundary layer on the plate. In RANS tests, turbulence characteristics of the initial flow were equal to $k = 11$ m^2/s^2, $\omega = 10^4$ Hz.

Even in Euler tests it was found that system of linear algebraic equations (26) cannot be solved by Gauss-Seidel method (residual either did not decrease or even grown during iterations). Attempts to apply the blockwise Jacoby method [27] lead to analogous results. To get the convergence, system (26) was modified as follows:

$$D(c)\overrightarrow{\Delta \mathrm{q}}(c) + C_H \cdot \sum_s H_s(c)\, \overrightarrow{\Delta \mathrm{q}}\,(c_s(c)) = \overrightarrow{\mathrm{R}}(c). \tag{27}$$

It has been found that Gauss-Seidel iterations converge at $C_H \lesssim 0.8$, and Jacoby iterations converge up to $C_H \lesssim 0.99$. But in the test about laminar boundary layer the use of Gauss-Seidel method with $C_H \lesssim 0.8$ did not allow to get stationary flow (though the linear system (27) solved at each time step with high accuracy). To the contrary, Jacoby method with $C_H \lesssim 0.99$ converged to stationary flow in several tens of time steps (in calculation with global time stepping with Courant number CFL $= 10^6$). Dependence of the convergence rate

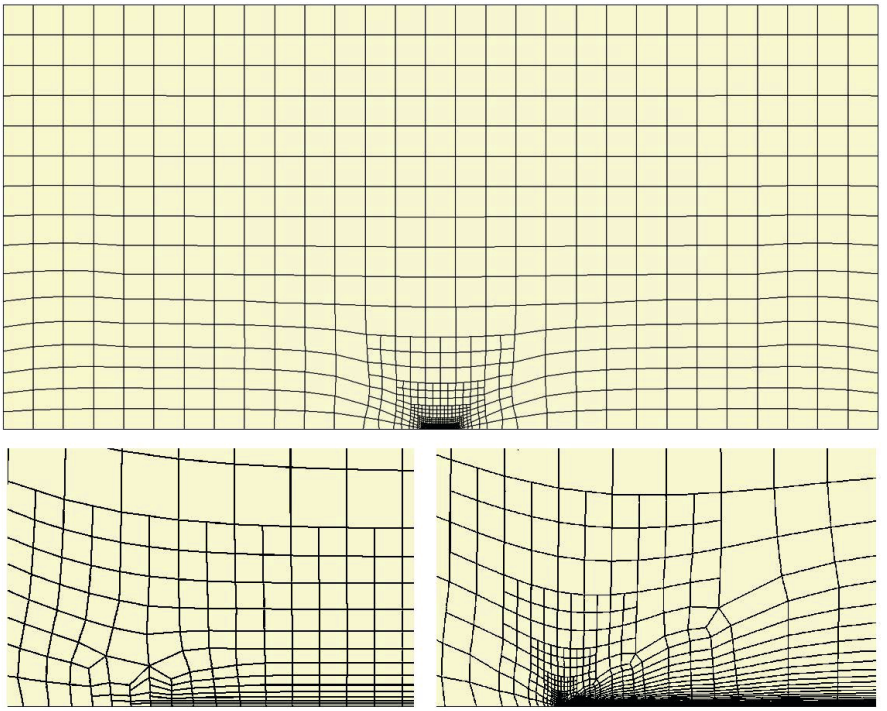


Fig. 2. Grid for test calculations and its details around the plate leading edge

upon C_H is shown in Figure 3. On the basis of these results, it has been decided to use Jacoby method with $C_H = 0.99$.

In the Euler tests with drift of Gauss-shaped perturbation of density, the analysis of the linear system solution convergence (within one time step) has also been performed. It has been found that during several initial iterations the residual may grow strongly (more than 1000 times) with respect to its initial level, and only after that it begins to diminish. This effect was found both for Gauss-Seidel method and for Jacoby method – see Figure 4, a. This result means that restriction of the linear system solver iterations by some constant small quantity (i.e. 6 iterations are widely used in practice with finite-volume methods) may result in moving away from the linear system solution. Instead, it can be recommended to continue iterations till the moment, when the residual will diminish by the given value from initial level. This approach increases the necessary quantity of iterations (up to several hundreds per time step), but guarantees the approaching to solution of linear system. Experiments have shown that the reasonable choice is stopping the iterations when the residual is diminished to 10^{-2} of its initial value. Figure 4, b shows that following growth of the linear system solution accuracy doesn't influence on the accuracy of global convergence to stationary flow, but increases the time of computation.

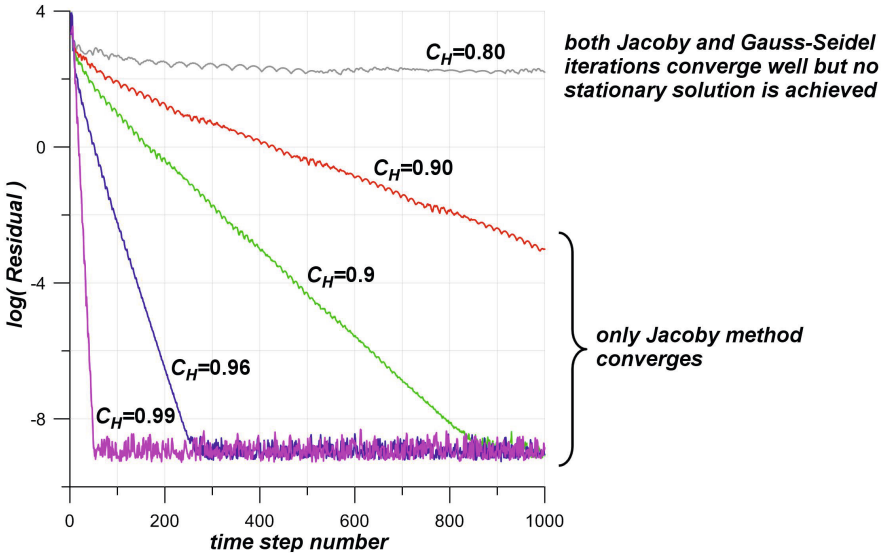


Fig. 3. Dependence of the convergence rate upon the damping coefficient C_H

Following tests have been performed for the task of turbulent boundary layer at the flat plate. Calculations appeared to be stable, but without convergence to stationary state. To resolve this problem, another coefficient has been introduced into the linear system (27). Let's rewrite (27) in the following form:

$$\left(\left(D - D^{\text{diff}}\right) + D^{\text{diff}}\right)\cdot\overrightarrow{\Delta \text{q}}(c) + C_H \cdot \sum_s \left(\left(H_s - H_s^{\text{diff}}\right) + H_s^{\text{diff}}\right)\overrightarrow{\Delta \text{q}}\left(c_s(c)\right) = \overrightarrow{\text{R}}(c).$$

Here the matrices D^{diff} and H_s^{diff} describe the contribution of diffusive (both molecular and turbulent) fluxes in the matrices $D(c)$ and $H_s(c)$, accordingly. It was found that convergence can be improved essentially by multiplying the "diffusive" matrices by some coefficient $C_{\text{vis}} > 1$:

$$\begin{aligned}\left(\left(D - D^{\text{diff}}\right) + C_{\text{vis}}\, D^{\text{diff}}\right)\cdot\overrightarrow{\Delta \text{q}}(c) + \\ + C_H \cdot \sum_s \left(\left(H_s - H_s^{\text{diff}}\right) + C_{\text{vis}}\, H_s^{\text{diff}}\right)\overrightarrow{\Delta \text{q}}\left(c_s(c)\right) = \overrightarrow{\text{R}}(c). \quad (28)\end{aligned}$$

During the convergence, the increased diffusive fluxes provide additional damping of perturbations. But if the stationary state is achieved ($\overrightarrow{\Delta \text{q}} = 0$), solution does not depend on C_{vis}. On the basis of experience, value $C_{\text{vis}} = 3$ may be recommended. At lower values of C_{vis} there is no convergence to stationary solution, and higher values of C_{vis} result in deceleration of the convergence to stationary state – see Figure 5.

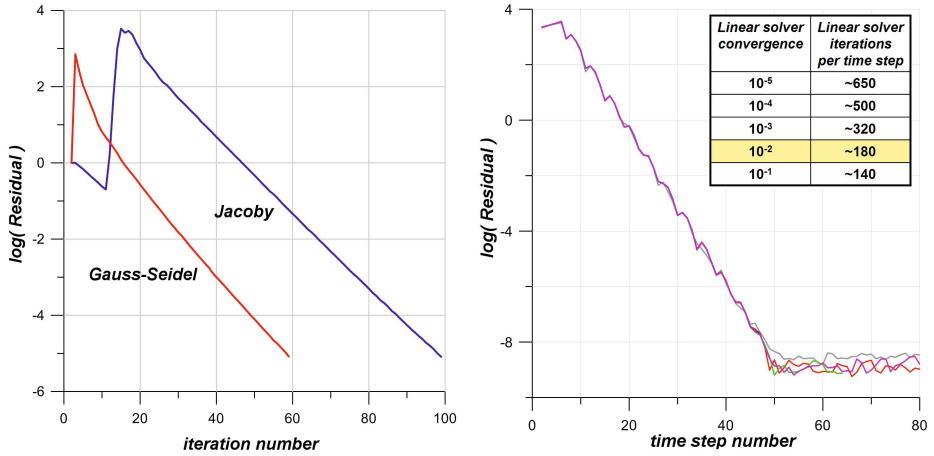


Fig. 4. a) convergence of linear system solution, b) global convergence to stationary flow

RANS calculations have also shown that it is impossible to obtain stationary solution in the case of global time stepping with high values of Courant number CFL. Analysis shows that instability is caused by source terms of the turbulence model. The critical value of CFL is 1000. But such value of maximal Courant number over the whole computational domain is insufficient for calculations of flows with turbulent boundary layers, because in this case the local value of CFL above the boundary layer appears to be about 10^{-3}, and the flow develops too slowly. So, it has been decided to perform all calculations with local time stepping, with local CFL 1000 in all cells. Experience have shown that the convergence of calculation with local time stepping with CFL 1000 may be not worse than the the convergence of calculation with global time stepping with maximal CFL 10^6.

6 Multigrid Algorithm for DG Method

The used multigrid method is based on the well-known full approximation storage multigrid scheme (see [28]). The main idea of classical multigrid method (h-multigrid) consists in fast transmission of information between different parts of computational domain. For this purpose, the equation solution is performed using the set of sequentially refining grids. The solution for the coarsest grid is used to solve the equation for the finer grid and vice versa the solution for the finest grid is used as an initial approach to solve equations for the coarse grid. At that, the equations for the coarse grid are modified to compensate the approximation loss in large cells.

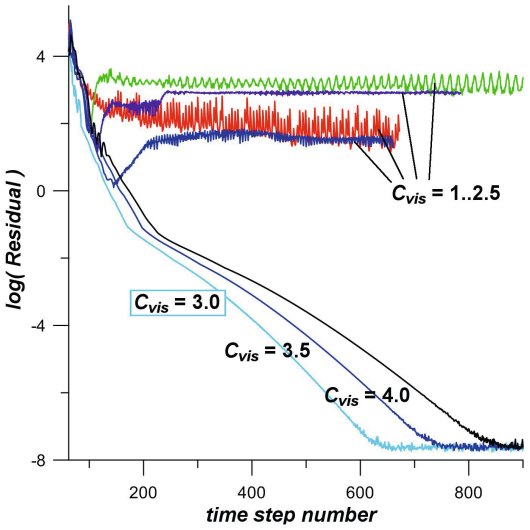


Fig. 5. Dependence of the convergence rate upon the coefficient C_{vis}

The polynomial multigrid method (p-multigrid) is developed on the basis of the paper [29]. Unlike h-multigrid, the equations are solved using the same level of grid but with the most order of basis functions. The maximal order K increases or diminishes similarly to enlargement or reduction of cells in grids.

Let's define a sequence of levels of a multigrid method $l = 1, ..., L$, such that the level $l+1$ has a higher number of cells or a higher number of basis functions (degrees of freedom) than the level l. At the highest (finest) level L, the equation system (26), (27), (28) may be represented as follows:

$$\mathbf{M}\Delta\mathbf{q}_L + \mathbf{R}_L\left(\mathbf{q}_L\right) = 0, \tag{29}$$

where the vector $\Delta\mathbf{q}$ is join of vectors $\overrightarrow{\Delta \mathrm{q}}$ for all cells of computational domain and $\mathbf{R}$ is join of vectors $\overrightarrow{\mathrm{R}}$ for all cells. Dimension of vectors $\Delta\mathbf{q}$ and $\mathbf{R}$ is $N_{\text{cells}} \cdot K_f \cdot N$. Matrix $\mathbf{M}$ has dimension $(N_{\text{cells}} \cdot K_f \cdot N) \times (N_{\text{cells}} \cdot K_f \cdot N)$. Equation system (29) can be solved using the method described in the Section 4. The implicit smoother provides correction of the solution at the highest level L: $\tilde{\mathbf{q}}_L = \mathbf{q}_L + \Delta\mathbf{q}_L$.

Let's write the grid equations at the coarser (lower) multigrid level $l = L - 1$:

$$\mathbf{M}\Delta\mathbf{q}_l + \mathbf{R}_l\left(\mathbf{q}_l\right) = \mathbf{F}_l. \tag{30}$$

The right-hand side appeared in this equation allows us to take into account the error because of the approximation accuracy loss when passing to the lower grid level, where the equations are written either on a coarser grid or using polynomials of a lower degree. To determine the right-hand side, let's consider

this system of equations on two sequential levels: level l (the equation (30)) and the upper level $l+1$:

$$\mathbf{M}\Delta\mathbf{q}_{l+1} + \mathbf{R}_{l+1}\left(\mathbf{q}_{l+1}\right) = \mathbf{F}_{l+1}. \tag{31}$$

The iterative scheme must be organized to guarantee that the solution transferred from an upper level to a lower one, $\mathbf{q}_l = I_{l+1}^l\left[\mathbf{q}_{l+1}\right]$, is also a solution of the equation (30) at the lower level:

$$\mathbf{M}I_{l+1}^l\left[\mathbf{q}_{l+1}\right] + \mathbf{R}_l\left(I_{l+1}^l\left[\mathbf{q}_{l+1}\right]\right) = \mathbf{F}_l. \tag{32}$$

Transferring the equation (31) to the lower level, we have:

$$I_{l+1}^l\left[\mathbf{M}\Delta\mathbf{q}_{l+1}\right] + \tilde{I}_{l+1}^l\left[\mathbf{R}_{l+1}\left(\mathbf{q}_{l+1}\right)\right] = \tilde{I}_{l+1}^l\left[\mathbf{F}_{l+1}\right]. \tag{33}$$

Here $I_{l+1}^l\left[\cdot\right]$ and $\tilde{I}_{l+1}^l\left[\cdot\right]$ are the operators for transfer of solution and residual from higher multigrid level to lower one. Algorithms for these operators are given below.

Subtracting the equation (33) from (32), we obtain the recurrent formula for the right-hand side:

$$\mathbf{F}_l = \mathbf{R}_l\left(I_{l+1}^l\left[\mathbf{q}_{l+1}\right]\right) + \tilde{I}_{l+1}^l\left[\mathbf{F}_{l+1} - \mathbf{R}_{l+1}\left(\mathbf{q}_{l+1}\right)\right]. \tag{34}$$

At the highest level $\mathbf{F}_{l=L} = 0$. Low-frequency error on an upper multigrid level is represented as high frequency error on an lower multigrid level and can be easily eliminated by one of several smoothing iterations. As a result, we have the following correction of solution: $\tilde{\mathbf{q}}_l = \mathbf{q}_l^{(0)} + \Delta\mathbf{q}_l$, where initial approximation for solution at the coarser level is result of its transfer from the upper level: $\mathbf{q}_l^{(0)} = I_{l+1}^l\left[\mathbf{q}_{l+1}\right]$.

The equations (30) are repeatedly solved at each level including the lowest one – the zero level at which the coarsest grid is used and the piecewise constant reconstruction of the solution is performed (the single basis function, $\varphi(x) \equiv 1$, is used). Then, on the basis of the obtained set of solutions, correction at higher multigrid levels is performed in accordance with following rule:

$$\mathbf{q}_\mathbf{l}^{(\mathrm{new})} = \tilde{\mathbf{q}}_\mathbf{l} + \mathbf{q}_{\mathrm{corr}}, \tag{35}$$

where $\mathbf{q}_{\mathrm{corr}} = \left\{I_{l-1}^l\left[\tilde{\mathbf{q}}_{l-1} - I_l^{l-1}\left[\tilde{\mathbf{q}}_l\right]\right]\right\}$ and $I_{l-1}^l\left[\cdot\right]$ is operator of solution transfer from the lower multigrid level to the higher one.

The expressions in the braces are averaged according to following rule:

$$\left\{\theta\right\}_i = \frac{\sum\limits_{f=1}^{N_f} \frac{1}{2}\left(\theta_i + \theta_n\right) S_f}{\sum\limits_{f=1}^{N_f} S_f},$$

where i is a cell index, n is an index of neighboring cell that is separated from the cell i by the face f. The summation is performed over all faces of all cells. S_f is the area of face f.

The algorithm also involves the damping procedure for solution correction:

$$\mathbf{q}_{\text{corr}} = \mathbf{q}_{\text{corr}} D \frac{|\tilde{\mathbf{q}}_l|}{\max\left(|\tilde{\mathbf{q}}_l|, \left|I_l^{l-1}[\tilde{\mathbf{q}}_l]\right|, |\tilde{\mathbf{q}}_{l-1}|, \varepsilon\right)},$$

where $D = 0.25...0.75$, $\varepsilon = 10^{-20}$.

When the solution is transferred from a finer grid to a coarser one (h-multigrid), the following operators of solution and residual transfer are used:

$$I_{l+1}^l[\mathbf{q}_{l+1}] = \frac{\sum\{\mathbf{q}_{l+1}\}\,\Omega^{l+1}}{\sum\Omega^{l+1}}, \quad \tilde{I}_{l+1}^l[\mathbf{R}_{l+1}] = \sum\{\mathbf{R}_{l+1}\}.$$

Here the summation is performed over all the cells of level $l+1$ that belong to a large cell of level l and Ω is the cell volume. The back transfer of solution from a coarser grid to a finer one is performed using the interpolation operator:

$$I_l^{l+1}[\mathbf{q}_l] = \mathbf{q}_l.$$

The operators of the solution transfer between the levels with different numbers of basis functions in the reconstructed solution (p-multigrid) are obtained from the condition that the solutions $\mathbf{q}_l$ and $\mathbf{q}_{l-1}$ should be equivalent (in the framework of DG approach) in the considered cell. The number of basis functions at these levels is determined by relations $M = K_f\left(K^l\right)$, $m = K_f\left(K^{l-1}\right)$. The equivalence of solutions means that the following integral conditions, which are obtained by multiplying the difference between solutions by the basis functions φ_j of the level, where the solution is transferred, are fulfilled: $\int\limits_\Omega (\mathbf{q}_l - \mathbf{q}_{l-1})\,\varphi_j d\Omega = 0$. Therefore, the operators I_l^{l+1} and I_{l+1}^l can be represented as follows:

$$I_l^{l+1}[\mathbf{q}_l] = [\mathbf{M}_{pp}]^{-1}\mathbf{M}_{pq}\mathbf{q}_l, \quad I_{l+1}^l[\mathbf{q}_{l+1}] = [\mathbf{M}_{qq}]^{-1}\mathbf{M}_{qp}\mathbf{q}_{l+1}, \tag{36}$$

where components of mass matrices are calculated as follows:

$$\mathbf{M}_{qq}^{i,j} = \int\limits_\Omega \varphi_i^{l-1}\varphi_j^{l-1}d\Omega, \quad \mathbf{M}_{pp}^{i,j} = \int\limits_\Omega \varphi_i^l\varphi_j^l d\Omega,$$

$$\mathbf{M}_{qp}^{i,j} = \int\limits_\Omega \varphi_i^{l-1}\varphi_j^l d\Omega, \quad \mathbf{M}_{pq}^{i,j} = \int\limits_\Omega \varphi_i^l\varphi_j^{l-1}d\Omega.$$

The operator for transferring the residual in relation (34), generally speaking, may differ from the operator (36):

$$\tilde{I}_{l+1}^l[\mathbf{R}_{l+1}] = \mathbf{M}_{qp}[\mathbf{M}_{pp}]^{-1}\mathbf{R}_{l+1}.$$

To demonstrate the efficiency of this multigrid method in combination with the implicit method, described in Section 4, series of tasks about inviscid compressible gas flow around sphere at $M = 0.15$ was calculated for sequence of

enclosed grids using piecewise-linear, quadratic and cubical polynomial basis functions. The cells of non-structured grids are hexahedrons. The calculations with high accuracy order (quadratic and cubical polynomial basis functions) were performed with taking into account the curvature of hexahedron sides adjacent to the sphere.

Figure 6 presents a typical history of solution residual convergence in DG scheme ($K = 3$) depending on time. Here, the calculation using h-p multigrid scheme (6-level for sequence of 3 grids) with implicit smoother is drawn by the red line. The history of solution residual convergence without multigrid method is drawn by the blue line. It is obvious that multigrid algorithm with implicit smoother removes oscillating character of convergence and essentially diminishes the calculation time.

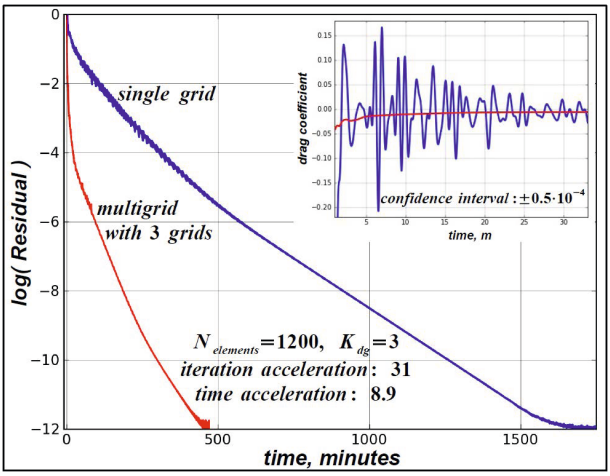


Fig. 6. DG scheme calculation ($K = 3$) of inviscid flow around the sphere ($M = 0.15$)

7 Parallel Implementation of 3D High-Order DG Algorithm for RANS Equations

Parallel version of 3D High-Order DG code on the base of NUMECA FINE^{TM}/ Hexa [29,30] has been developed. It includes the implicit smoother (see Sections 4, 5), h-p multigrid (Section 6), and new Gauss quadrature rules (Section 3).

The initial implementation of multigrid method in the DG was significantly different from that in NUMECA. This is due to the fact that DG is very sensitive to the accuracy of geometrical parameters. Therefore, each face of merged cell is considered as a sum of the appropriate faces in its parent cells. And, accordingly, the fluxes of conservative parameters through the face of the combined cells are calculated as the sum of fluxes through its constituent faces. Thus, multigrid in DG needed to be developed on the base on own algorithm of parallelization. As

mentioned above, due to restrictions in access to the code NUMECA, the direct development of a parallel algorithm was not possible. So, the deep refactoring of DG code was made with the aim to come as close to the available NUMECA code as possible. The main efforts were aimed at modification of h-multigrid for the DG to make it like NUMECA. Such changes have been made possible after the decision was made to use only the first order method at coarse grid. As a result of refactoring, parallelization of code is reduced to inserting barriers and block exchanges between sub-domains [31]. Generally, it uses standard NUMECA calls.

Let's consider a flow of the inviscid gas around a sphere at low Mach number. The computational grid consists of 8980 cells. It is built using automatic mesh generator NUMECA/HEXPRESS. Freestream parameters are following: $p = 1$ atm, $T = 293$ K, $V = 50$ m/sec ($M \approx 0.15$). Parallel efficiency of DG with implicit smoother in solution of this task on 4-cores i7 computer is demonstrated in Figure 7. The ideal acceleration is equal to quantity of processors. One may see the weak dependence of the efficiency on the accuracy order of the scheme. This is due to the fact that calculation time per iteration of the implicit scheme is essentially higher than the time of exchanges between subdomains. Accordingly, in the case of running on multiple processors the losses depend only on differences between sizes of subdomains.

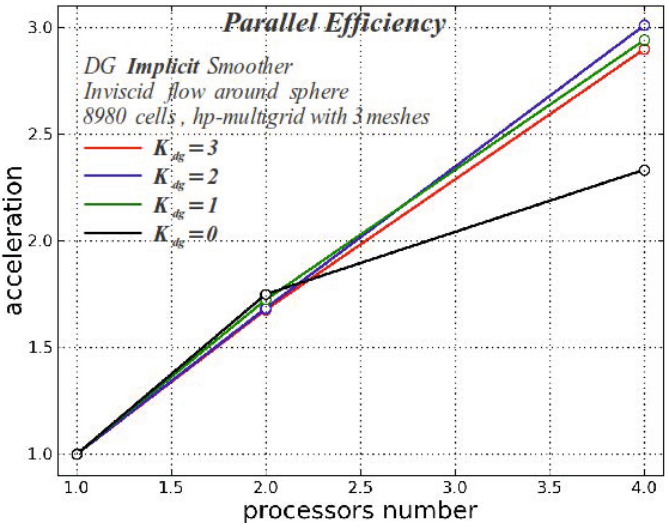


Fig. 7. Parallelization efficiency for DG scheme with implicit smoother

8 Calculations of Tests U.3, U.3, and A.3 Using Implicit DG Solver for RANS Equations

When the preliminary testing was finished, implicit DG solver for RANS equations has been applied to calculation of the IDIHOM project's standard tests:

U.3 (subsonic flow around three-element airfoil L1T2 [32] at high angle of attack), U.2 (transonic flow around wing ONERA M6 [33]), and A.3 (subsonic flow around three-element wing [34]). In these tests, accuracy order of numerical method and the code performance (running speed) were estimated.

Geometry of high-lift airfoil L1T2 [32] may be seen in Figure 8. Flow with following parameters was considered: Mach number $M_\infty = 0.197$, angle of attack $\alpha = 20.18°$, chord based Reynolds number $\mathrm{Re} = 3.52 \times 10^6$. Turbulence parameters of the ambient flow: $k_\infty = 0.68\ \mathrm{m}^2/\mathrm{s}^2$ ($\mathrm{Tu}_\infty = \sqrt{\frac{2}{3}\frac{k_\infty}{V_\infty^2}} \times 100\% = 1.0\%$), $\omega_\infty = 4.5 \times 10^4$ Hz (at the chosen value of k_∞, this value of Ω_∞ corresponds to $\mu_{t\infty} \approx \mu_\infty$).

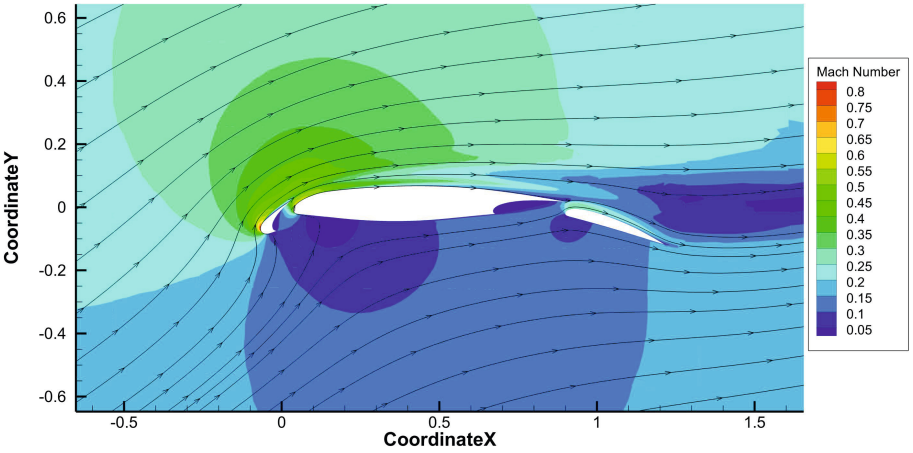


Fig. 8. General view of flow around airfoil L1T2 (test U.3)

A set of three unstructured hexahedral grids have been constructed:

- fine grid: 20854 cells, height of the first near-wall cell $h_{\min} \sim 1.7 \times 10^{-5}$ m, increment of cell size growth (geometric ratio) GR = 1.21;
- medium grid: 7432 cells, $h_{\min} \sim 3.7 \times 10^{-5}$ m, GR = 1.46;
- coarse grid: 2836 cells, $h_{\min} \sim 8.9 \times 10^{-5}$ m, GR = 2.14.

In the vicinity of solid surfaces, curvature of grid lines have been taken into account [35]. Cell edges are considered as elements of parabolas. End points and central points of the cell edges belong to solid surfaces.

Outer boundary of computational domain is placed at the distance of 10 chords from the airfoil, and boundary condition at this boundary is based on analysis of Riemann invariants. To take into account the velocity circulation, additional velocity field $\boldsymbol{V}_\Gamma(\boldsymbol{r})$ is superimposed at the boundary point $\boldsymbol{r}$:

$$\boldsymbol{V}_\Gamma(\boldsymbol{r}) = \frac{\Gamma\boldsymbol{\tau}}{2\pi\,|\boldsymbol{r} - \boldsymbol{r}_c|},$$

where $\boldsymbol{r}_c$ is the central point of the airfoil, $\boldsymbol{\tau}$ is unit vector normal to $\boldsymbol{r} - \boldsymbol{r}_c$, and Γ is the prescribed circulation value.

Solid surfaces are treated as adiabatic no-slip walls.

To achieve fully-2D computation, set of 2D basis functions have been used.

Calculations were performed with local time stepping. Each computation started from local Courant number CFL = 1. If after current iteration the new flowfield appeared to be physical, CFL incremented 1.5 times; if it was non-physical, the numerical solution was returned to state before the iteration, and the iteration was repeated with $2 - 10$ times smaller value of CFL. In all calculations the maximal value of Courant number was restricted by the value 1000, except the calculation with $K = 3$ on the fine grid, where CFL was restricted by the value 100. In all calculations, deep convergence to stationary flow has been achieved (residual has decreased by $8-15$ orders of magnitude from initial level). Calculations have been parallelized; from 4 to 48 processor cores were used.

The following table presents the obtained lift and drag coefficients C_L and C_D, as well as characteristics of the code performance, including maximal RAM storage, normalized run-time to converge variable CL within tolerance 0.04 and variable CD within tolerance 0.001 – $\bar{T}_{\text{run}} = \frac{T_{\text{run}} \cdot N_{\text{CPU}}}{T_{\text{bench}}}$ and unit normalized run-time $\bar{\bar{T}}_{\text{run}} = \bar{T}_{\text{run}}/N_{\text{DOF}}$. Here N_{CPU} is quantity of processor cores, T_{bench} is run-time for calculation of the benchmark test task using the benchmark code (TauBench, based on the code Tau by DLR) on one core of the same computer, and N_{DOF} is number of degrees of freedom (quantity of cells multiplied by the number of basis functions).

Grid	DG	N_{CPU}	RAM, Gb	Results for C_L			Results for C_D		
				C_L	$\bar{T}_{\text{run}}$	$\bar{\bar{T}}_{\text{run}}$	C_D	$\bar{T}_{\text{run}}$	$\bar{\bar{T}}_{\text{run}}$
coarse	$K = 1$	$4^{a)}$	0.1	3.0888	35	4.11×10^{-3}	0.1762	46	5.41×10^{-3}
coarse	$K = 2$	$4^{a)}$	0.2	3.4292	214	1.26×10^{-2}	0.0971	214	1.26×10^{-2}
coarse	$K = 3$	$12^{b)}$	0.5	3.5219	486	1.71×10^{-2}	0.1066	721	2.54×10^{-2}
medium	$K = 1$	$4^{a)}$	0.3	3.6475	126	5.65×10^{-3}	0.1323	179	8.03×10^{-3}
medium	$K = 2$	$12^{b)}$	0.6	3.9724	781	1.75×10^{-2}	0.0712	998	2.24×10^{-2}
medium	$K = 3$	$16^{b)}$	1.4	3.9455	3067	4.13×10^{-2}	0.0659	3670	4.94×10^{-2}
fine	$K = 1$	$4^{a)}$	0.6	3.8463	826	1.32×10^{-2}	0.1234	1079	1.72×10^{-2}
fine	$K = 2$	$8^{b)}$	1.5	3.9944	6590	5.27×10^{-2}	0.0681	8093	6.47×10^{-2}
fine	$K = 3$	$48^{b)}$	3.8	3.9844	152789	0.733	0.0641	196629	0.943

[a)] PC Intel Core i7-2600, 3.4-3.8 GHz

[b)] 4 nodes cluster (16 cores each) Intel Xeon E5-2670, 2.6-3.3 GHz, 1Gbit Ethernet

Efficiency of high-order calculations is illustrated by Figure 9. Here one can see that convergence to values of coefficients C_L and C_D to reference values, obtained in fully-converged finite-volume calculations by DLR, is achieved more quickly with $K = 2$, 3 than with $K = 1$. However, by now $K = 3$ is slower than $K = 2$.

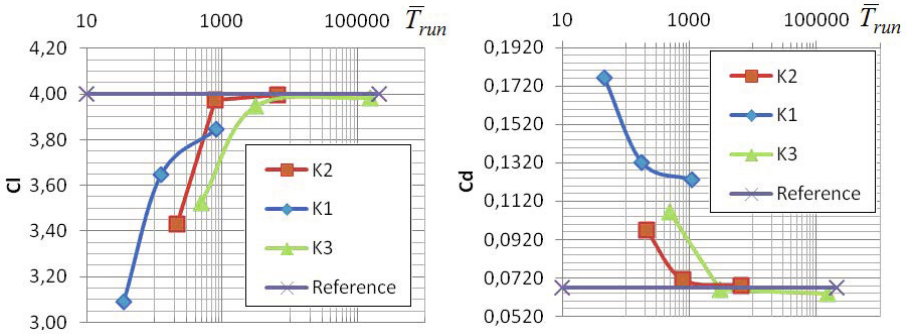


Fig. 9. Grid convergence as function of normalized run-time (test U.3)

Figure 10 shows grid convergence with the growth of N_{DOF}. Along vertical axes, error in determination of integral coefficients in comparison with asymptotically derived values C_L^* and C_D^* is plotted. High-order methods ($K = 2$, 3) are converged faster than low order ($K = 1$). For the drag coefficient, convergence for $K = 3$ is faster than for $K = 2$.

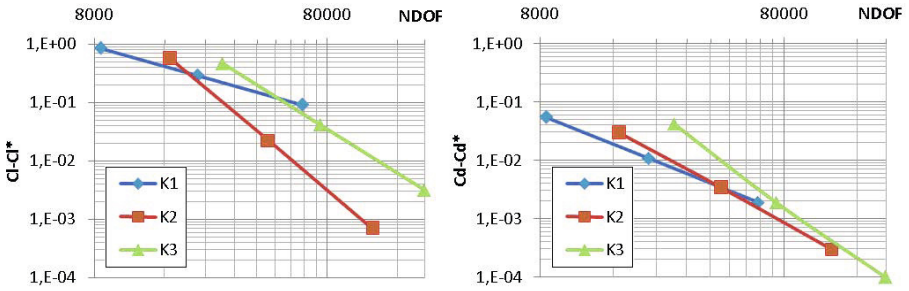


Fig. 10. Grid convergence of lift and drag coefficients, as function of N_{DOF} (test U.3)

In the next table the convergence order, estimated on the basis of obtained values of C_L and C_D, is shown.

K	Convergence order based on	
	C_L	C_D
1	2.2	3.4
2	6.7	4.4
3	5.0	6.5

Calculations of transonic flow around the wing ONERA M6 [33] have been performed for the following flow regime (TEST 2308 from [33]): Mach number $M_\infty = 0.8395$, angle of attack $\alpha = 3.06°$, mean aerodynamic chord-based Reynolds number $\mathrm{Re} = 11.72 \times 10^6$, stagnation temperature $T_0 = 300$ K. Turbulence parameters of the ambient flow: $k_\infty = 10\ \mathrm{m}^2/\mathrm{s}^2$ ($\mathrm{Tu}_\infty \approx 1\%$), $\omega_\infty = 6.7 \times 10^5$ Hz (at the chosen value of k_∞, this value of ω_∞ corresponds to $\mu_{t\infty} \approx \mu_\infty$).

A set of three unstructured hexahedral grids has been constructed:

- fine grid: 204062 cells, $h_{\min} \sim 8 \times 10^{-5}$ m, GR = 2.0;
- medium grid: 75987 cells, $h_{\min} \sim 1 \times 10^{-4}$ m, GR = 3.0;
- coarse grid: 38219 cells, $h_{\min} \sim 3 \times 10^{-4}$ m, GR = 4.0.

In the vicinity of solid surfaces, curvature of grid lines have been taken into account. The same boundary conditions, as in test for airfoil L1T2, were used.

This stage of testing is not completely finished yet. Some problems with stability are not resolved. In particular, only calculations with $K = 1$ and $K = 2$ have reached the convergence (with decrease of residual by $5 - 6$ orders of magnitude from initial level), except the calculation with $K = 2$ on a medium grid, where amplitude of C_L and C_D oscillations is about $3 - 4$ counts. By now, we have failed to achieve stable calculations with $K = 3$.

The following tables summarize current preliminary results for ONERA M6 wing:

Grid	DG	C_L	C_D
coarse	$K = 1$	0.26678	0.02090
medium	$K = 1$	0.27952	0.01824
fine	$K = 1$	0.28217	0.01634
coarse	$K = 2$	0.24974	0.01699
medium	$K = 2$	0.26619	0.01327
fine	$K = 2$	0.27285	0.01260

K	Convergence order based on	
	C_L	C_D
1	7.4	2.6
2	4.7	7.9

These results are illustrated by Figure 11. Efficiency of DG method ($K = 1, 2$) is compared with results obtained using 2^{nd} accuracy finite-volume method (calculations with NUMECA FINE^{TM}/Open 2.11 code with Spalart-Allmaras turbulence model). Higher efficiency of finite-volume calculation in comparison with DG $K = 1$ method may be explained by the fact that calculation with the use of EARSM turbulence model is inevitably slower than calculation on the basis of one-equation Boussinesq Spalart-Allmaras model. One may also see that lower-order methods appear to be more efficient if low accuracy (error about 10^{-3}) is acceptable. But if high accuracy (error lower than 10^{-4}) is required, DG $K = 2$ appears to be more efficient than low-order methods.

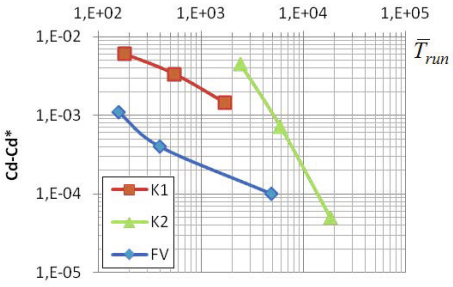


Fig. 11. Grid convergence of lift and drag coefficients, as function of normalized run-time (test U.2)

A.3 subsonic high-lift external aerodynamics test case is based on the materials from the first AIAA CFD High Lift Prediction Workshop (HLPW) [34]. The configuration is taken from the NASA Trap Wing model, a three-element 3D high lift configuration with slat, main wing and flap attached to a wind tunnel body. In the present paper, the Configuration 1 is considered with slat at 30° and flap at 25° deflection.

Computations have been performed for the following flow regime: Mach number $M_\infty = 0.2$, angle of attack $\alpha = 13°$, mean aerodynamic chord-based Reynolds number $\mathrm{Re} = 4.3 \times 10^6$, stagnation temperature $T_0 = 300$ K. Turbulence parameters of the ambient flow: $k_\infty = 0.7$ m^2/s^2 ($\mathrm{Tu}_\infty = 1\%$), $\omega_\infty = 3.7 \times 10^4$ Hz (at the chosen value of k_∞, this value of ω_∞ corresponds to $\mu_{t\infty} \approx \mu_\infty$).

Both reference FV computations and high-order DG computations have been performed on structured hexahedral multiblock grids. Using grid generator ICEM CFD 14.0, a set of five grids has been constructed. Their characteristics are presented in the following table:

Grid ID	Number of cells	Height of the first near-wall cell (m)	Increment of cell size growth
1	32 771	1.0×10^{-4}	3.00
2	121 014	5.0×10^{-5}	2.10
3	262 168	3.0×10^{-5}	1.80
4	2 097 334	1.0×10^{-5}	1.30
5	16 778 752	6.0×10^{-6}	1.15

Grids #1 and #5 are presented in Figure 12. The extra coarse grid #1 near the wing tip is shown in Figure 13. Outer boundary of computational domain is placed at the distance of 124 chords from the model. There, boundary condition is based on Riemann invariants analysis. Solid surfaces are treated as adiabatic no-slip walls. For the reference computations, the EWT-ZEUS solver developed in TsAGI has been used. The solver employs RANS equation system closed by SST turbulence model. Implicit second-order FV method based on MUSCL approach for convective fluxes has been applied. Integral characteristics of flow (C_L and C_D) obtained on different grids are presented in Figure 14. Accuracy order of this numerical scheme estimated based on the results of the finest three grids is equal to 2.77 (based on C_L) and 1.15 (based on C_D).

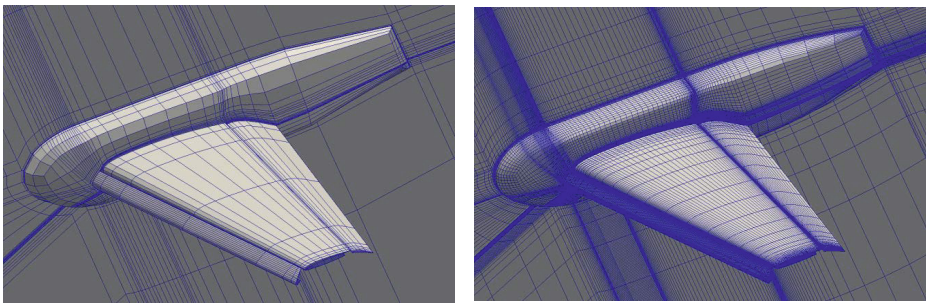


Fig. 12. Grids for high-lift 3D test: a) grid #1 and b) grid #5

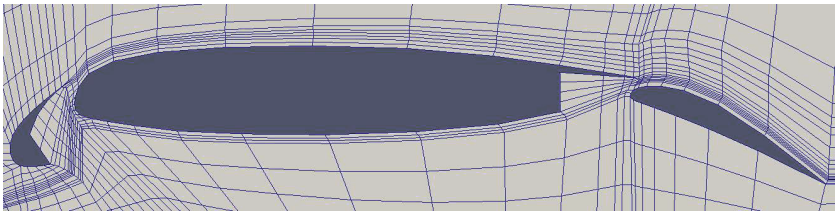


Fig. 13. Cross-section of grid #5 for high-lift 3D test

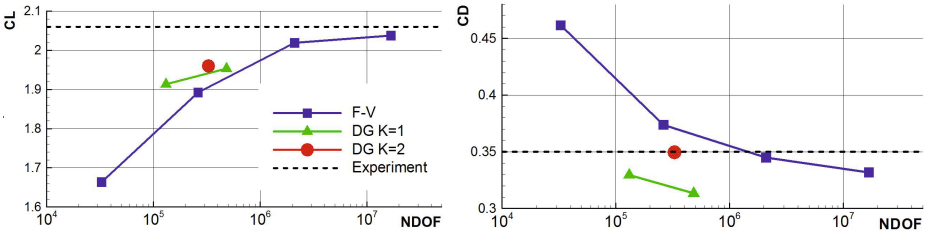


Fig. 14. a) C_L and b) C_D characteristics of HLPW model vs. NDOF for FV and DG computations

Unfortunately, we experienced convergence problems in DG computations of this test case. After long initial period of convergence, instability started to grow which finally led to computation crash. We believe these problems to be associated with the presence of stretched and twisted hexahedral cells in the grid emerged due to our efforts to generate extremely coarse grid around this geometrically complex configuration. Using the non-orthogonal shape functions defined in the physical space results in ill-conditioned mass matrices in such highly distorted cells. It leads to errors accumulation by the numerical scheme, which causes sudden divergence of solution. We plan to continue our research to eliminate this source of errors by revising the local shape functions.

Despite the crash of all test runs, DG computations on extra coarse grid #1 with $K = 1$ and $K = 2$ and on medium grid #2 with $K = 1$ have rather long convergence period so that the forces are almost stabilized. These results are also presented in Figure 14. It is seen that with equivalent NDOFs, the accuracy of DG results is higher than that obtained with the FV method. Figure 15 presents the pressure distributions on the model surface obtained in second-order FV computation on fine grid #5 and in third-order DG $K = 2$ computation on coarse grid #1. It is clearly seen that despite the significant difference in the density of the grids, the flow patterns are close to each other. Comparison of pressure distributions (Figure 16) obtained in section $z/b = 0.65$ with experimental data and with results of the reference computations shows that the data of DG $K = 2$ computations on coarse grid #1 approach the data of reference computation (grid #5) closer than DG $K = 1$ computations on grids #1 and #2.

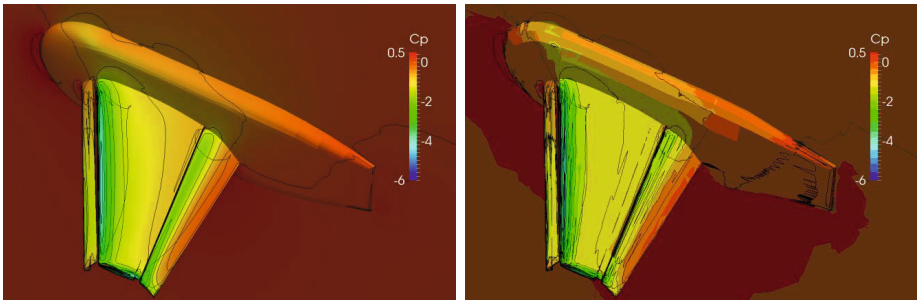


Fig. 15. Pressure distributions on the model surface obtained in computations using a) second-order FV approach on the finest grid #5 and b) third-order DG $K = 2$ on coarse grid #1

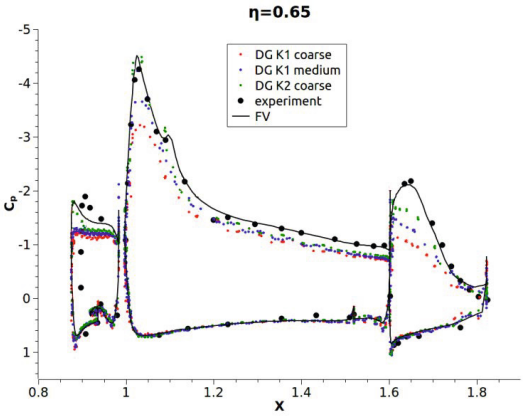


Fig. 16. Pressure distributions in wing section of HLPW model

In the other computations (medium grid with $K = 2$, fine grid with $K = 1$ and $K = 2$), the divergence develops too fast thus not allowing to obtain any results and to perform the assessment of accuracy order.

Results of these tests show that high-order methods (DG with $K > 1$) on coarse grids allow to obtain solution of quality, comparable with quality of solutions, obtained using low order method (DG with $K = 1$) on much finer grids (with 10 or more higher quantity of cells). Differences between the expected and obtained convergence orders are possibly can be explained by following effects: 1) grids are still too coarse, and convergence rate is still far from asymptotic value; 2) nesting of cells on different grids is not good. Stability problems in transonic calculation of flow around ONERA M6 wing are explained by the presence of shock waves in flow. Authors has successful experience in calculation of low Mach number flow around ONERA M6 wing. To stabilize calculation, it is necessary to find appropriate monotonizing procedure; but such study was not performed within IDIHOM project.

9 Explicit Numerical Method for Solution of Isentropic Linearized Euler Equations

Another direction of TsAGI activity withtin IDIHOM project was devoted to development of DG solver for solution of tasks from aeroacoustics. Acoustic tasks are characterized by the necessity to simulate propagation of very small perturbations over non-uniform aerodynamic flowfields. This implies that numerical method should handle multiscale tasks with low dispersion and low dissipation errors for small scales. This requirement leads necessarily to high-order numerical schemes. In addition, very detailed numerical grids and rather large computational domains should be considered to calculate accurately the near acoustic field and to have possibility to impose correctly boundary conditions for calculations of the far acoustic field. DG method suited well to solution of such tasks, because it is numerical method of arbitrarily high accuracy order that is practically independent upon grid quality and also highly parallelizable.

There are several approaches for solving acoustic tasks. The approach, which was considered in this work, is known as "perturbation method" [36]. It requires relatively not much computational costs and consists of three general steps. At the first step a basic aerodynamic flowfield is calculated independently using Reynolds averaged Navier-Stokes equations. Then near acoustic field is simulated directly as generation and propagation of small acoustic perturbations over the aerodynamic field. This direct numerical simulation of sound is performed on the basis of linearized Euler equations. And finally far acoustic field is estimated using special methods (Kirchhoff or Ffowcs-Williams Hawkings methods), which consider radiation of sound by control surface over practically uniform aerodynamic field. Acoustic characteristics of the control surface are taken from the calculation of near acoustic field.

TsAGI work within IDIHOM project was concentrated on direct simulation of sound propagation in near acoustic field. Theoretically, it is possible to neglect the influence of viscosity on sound propagation and to consider this process as adiabatic. To minimize essentially the quantity of arithmetic operations, it was decided to calculate sound propagation on the basis of Isentropic Linearized Euler Equations (ILEE). This equation system may be obtained from full linearized Euler equations, if we replace the last differential equation (for energy) by isentropic relation $p' = c_a^2 \rho'$.

In the case of ILEE equations, the system (1) dimension $N = 4$, primitive variables $\mathrm{Q} = [\rho'; u'; v'; w']^{\mathrm{T}}$ (perturbations of density and velocity), and conservative variables $\mathrm{U} = [\rho'; \rho_a u' + \rho' u_a; \rho_a v' + \rho' v_a; \rho_a w' + \rho' w_a]^{\mathrm{T}}$ (index a corresponds to basic aerodynamic field that is taken from RANS calculations). The vector F_k of fluxes along x_k axis ($x_1 = x$, $x_2 = y$, $x_3 = z$) and the vector of source terms S (see (1)) have the following form:

$$\mathrm{F}_k = \begin{bmatrix} U'_k \\ U'_i u_{ak} + \rho_a u_{ai} u'_k + c_a^2 U'_1 \delta_{ik} \end{bmatrix}, \quad \mathrm{S} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Because acoustic tasks have essentially non-stationary character, it was decided to use explicit DG method. Using (2), one may rewrite (4) as follows:

$$\sum_{j=1}^{K_f} \left(\int\limits_{\Omega} \Gamma\, \varphi_i(\mathbf{x})\varphi_j(\mathbf{x}) d\Omega \right) \frac{d\mathrm{q}_j}{dt} + \oint\limits_{\Sigma} \mathrm{F}_n\, \varphi_i(\mathbf{x}) d\Sigma = \int\limits_{\Omega} \mathrm{F}_i\, d\Omega,\ i = 1, ..., K_f. \quad (37)$$

Let's introduce the vector of size $K_f \cdot N$ for each cell:

$$\overrightarrow{\mathrm{q}} \equiv \left((\mathrm{q}_1)_\rho; (\mathrm{q}_2)_\rho; ...; \left(\mathrm{q}_{K_f}\right)_\rho; (\mathrm{q}_1)_u; (\mathrm{q}_2)_u; ...; \left(\mathrm{q}_{K_f}\right)_u; ...; (\Delta \mathrm{q}_1)_w; (\mathrm{q}_2)_w; ...; \left(\mathrm{q}_{K_f}\right)_w \right)^{\mathrm{T}}.$$

Then (37) may be rewritten in matrix form:

$$M \frac{d\overrightarrow{\mathrm{q}}}{dt} = \overrightarrow{\mathrm{b}} \quad \text{or} \quad \frac{d\overrightarrow{\mathrm{q}}}{dt} = \overrightarrow{\mathrm{R}}, \quad (38)$$

where $M_{s\cdot i,\, m\cdot j} = \int\limits_{\Omega} \Gamma_{sm}\, \varphi_i(\mathbf{x})\varphi_j(\mathbf{x}) d\Omega$, $(\overrightarrow{\mathrm{b}})_{s\cdot i} = \int\limits_{\Omega} (\mathrm{F}_i)_s\, d\Omega - \oint\limits_{\Sigma} (\mathrm{F}_n)_s\, \varphi_i(\mathbf{x}) d\Sigma$, $\overrightarrow{\mathrm{R}} = M^{-1}\overrightarrow{\mathrm{b}}$. Matrix M has dimension $(K_f \cdot N) \times (K_f \cdot N)$ To get high accuracy order in time, the time integration of differential equation (38) is performed by the explicit m-stage Runge-Kutta (RK) method:

$$\begin{cases} \overrightarrow{\mathrm{q}}^{(0)} = \overrightarrow{\mathrm{q}}^{n}, \\ \overrightarrow{\mathrm{q}}^{(1)} = \overrightarrow{\mathrm{q}}^{n} + \alpha_1 \tau\, \mathrm{R}^{(0)}, \\ \cdots \\ \overrightarrow{\mathrm{q}}^{(m)} = \overrightarrow{\mathrm{q}}^{n} + \alpha_m \tau\, \mathrm{R}^{(n-1)}, \\ \overrightarrow{\mathrm{q}}^{(n+1)} = \overrightarrow{\mathrm{q}}^{(m)}. \end{cases}$$

In the current work 4-stage RK method with the following coefficients is used:

$$\alpha_1 = 1/4 \quad \alpha_2 = 1/3 \quad \alpha_3 = 1/2 \quad \alpha_4 = 1.$$

Fluxes $\mathrm{F}_n(\mathrm{Q}_s^k)$ through the cell face s are determined using simplified algorithm [37], based on Roe solution of the Riemann problem. Let's designate states of flow on the left and on the right of the cell face s by indices "L" and "R". It will be assumed that $|\mathrm{Q}_a^R - \mathrm{Q}_a^L| \ll |\mathrm{Q}_a|$. This assumption allows using two important simplifications [37] in classical Roe method: 1) primitive variable approach and 2) use of approximate Roe matrix, which coincides with Jacobian of ILEE, computed using arithmetic averages of aerodynamic flows from the left and from the right of the cell boundary. So, the flux $\mathrm{F}_n(\mathrm{Q}_s^{\mathrm{n}})$ may be calculated using formula

$$\mathrm{F}_n(\mathrm{Q}_s^{(k)}) = \frac{\mathrm{F}_n(\mathrm{Q}_L^{(k)}) + \mathrm{F}_n(\mathrm{Q}_R^{(k)})}{2} - d_n^{(k)},$$

where the dissipative term $d_n^{(k)} = T^{(k)}|\Lambda^{(k)}|\left(T^{(k)}\right)^{-1}\Gamma^{(k)}\frac{\mathrm{Q}_R^{(k)}-\mathrm{Q}_L^{(k)}}{2}$. The columns of matrix T are the right eigenvectors of A_n, $|\Lambda|$ is diagonal matrix with the modules of A_n eigenvalues on the diagonal. Components of the matrix $\Gamma = \partial \mathrm{U}/\partial \mathrm{Q}$, of the matrix $A_n = \partial \mathrm{F}_n/\partial \mathrm{U}$, and accordingly components of T and $|\Lambda|$, are calculated at the known time layer of RK procedure (k) using arithmetic averagings $\bar{\mathrm{V}}_i = \left((\mathrm{V}_i)_L + (\mathrm{V}_i)_R\right)/2$, where $\mathrm{V}_i = \{u_a, v_a, w_a, c_a\}$, $c_a = \sqrt{\gamma\, p_a/\rho_a}$ is speed of sound. Matrix A_n has the following form:

$$A_n = \begin{bmatrix} 0 & n_x & n_y & n_z \\ -u_a V_{an} + c_a^2 n_x & u_a n_x + V_{an} & u_a n_y & u_a n_z \\ -v_a V_{an} + c_a^2 n_y & v_a n_x & v_a n_y + V_{an} & v_a n_z \\ -w_a V_{an} + c_a^2 n_z & w_a n_x & w_a n_y & w_a n_z + V_{an} \end{bmatrix},$$

where $V_{an} = u_a n_x + v_a n_y + w_a n_z$. It worth to note that matrix T is not symmetric with respect to n_x, n_y, n_z and can degenerate, e.g. if one of these values is zero. But the product $d_n = T|\Lambda|T^{-1}\frac{\mathrm{Q}_R-\mathrm{Q}_L}{2}$ is symmetric and does not degenerate [37].

10 Calculations of Acoustic Tests Using Explicit DG Solver for ILEE Equations

DG solver for ILEE equations, which have been created on the basis of the explicit method from Section 9, has been verified on a series of inner tests: the evolution of the Gaussian acoustic pulse in a uniform flow, the propagation of the acoustic sine wave and signals with continuous Π-like spectrum over a uniform flow. We shall consider only test about sine wave that is of major importance. It allows to determine the grid spatial resolution necessary for accurate simulation of small perturbations propagation.

Let's consider a one-dimensional aerodynamic flow in a tube with smooth slip walls. The tube length equals to $L = 200$ and the tube is located to the right from the cross-section $x = 0$. The basic aerodynamic flow field is a stationary uniform flow, defined as follows:

$$p_a = \frac{1}{\gamma},\ \ \rho_a = 1\ \ \left(\Rightarrow c_a = \sqrt{\gamma\frac{p_a}{\rho_a}} = 1\right),\ \ u_a = M_a = 0.5,\ \ v_a = w_a = 0.$$

The base field is disturbed by a sinusoidal perturbation of a small amplitude and unit wavelength at the left boundary. Only Riemann invariant z_3, that is constant along the characteristic $dx/dt = u_a + c_a$, is perturbed. Consequently, the exact solution exists as a sinusoidal perturbance that propagates along the x axis with the constant velocity $(u_a + c_a)$ without fading.

Exact analytical solution for the whole flow (with acoustic disturbances):

$$\begin{cases} \rho_{\text{exact}}(x,t) = \left[\dfrac{1}{\gamma\, z_0(x,t)} \left(\dfrac{\gamma-1}{4} \left(z_+(x,t) - z_-(x,t) \right) \right)^2 \right]^{\frac{1}{\gamma-1}}, \\ u_{\text{exact}}(x,t) = \dfrac{1}{2} \left(z_+(x,t) + z_-(x,t) \right), \\ v_{\text{exact}}(x,t) = w_{\text{exact}}(x,t) = 0, \\ p_{\text{exact}}(x,t) = \dfrac{1}{\gamma} \left(\dfrac{\gamma-1}{4} \left(z_+(x,t) - z_-(x,t) \right) \right)^2 \rho_{\text{exact}}(x,t). \end{cases} \tag{39}$$

where $\varepsilon = 10^{-3}$ is amplitude of the sine wave. The fluctuations are defined as follows: $\rho' = \rho - \rho_a$, $u' = u - u_a$, $v' = v - v_a$, $w' = w - w_a$. Both initial solution and boundary conditions at each time moment are taken from the exact solution (39).

Calculations are performed on grids with 3, 4, 6 and 8 cells per wave length $\lambda = 1$ respectively. The results obtained on two equivalent grids (with the same N_{DOF}) at time $t = 60$ with $K = 3$, are compared in group and with the exact solution. Courant number is 0.1 for all computations.

Usually in acoustics specially optimized dispersion-relation-preserving (DRP) scheme of the 4^{th} order in space in aggregate with low dissipative and dispersion RK procedure developed by Tam et al. [39] is used. So, in our paper we also provided comparison of the developed DG scheme with Tam scheme which uses 4-stage low-dispersion and dissipation RK procedure of the 2^{nd} order in time.

The representative solution shape and its comparison with the exact solution are given in Figure 17, a. The numerical scheme dissipation monotonically attenuates the perturbation amplitude. The perturbation covers distance $(u_a + c_a)t = 1.5 \cdot 60 = 90$. Therefore, there is no boundary influence to the right of $x = 90$ and the signal amplitude fades uniformly with x. In the region [0; 90] one can observe the spatial sweep of signal fade-out.

Figure 17, b shows the results of solutions comparison. The solution envelopes are shown. It is necessary to note that the exact solution oscillates in the range $[-0.0005; 0.0005]$. The considered Tam scheme gives acceptable level of dissipation (decrease of the amplitude is not higher than 20%) on the grid with 16 cells per wavelength. DG scheme with $K = 2$ gives better level of dissipation (less than 10%) on the grid with 6 cells per wavelength, and DG scheme with $K = 2$ – even at 3 cells per wavelength! At that, DG scheme $K = 3$ with 4 cells per wavelength and DG $K = 2$ with 8 cells per wavelength provide dissipation level less than 2%. It is worth to add that the solutions obtained with DG scheme give no visible displacement in phase.

For this 1D test, order of grid convergence of DG $K = 3$ solver has been estimated. For this purpose, calculations have been performed on grids with 4, 8, 16, 32 and 64 cells per wavelength of the sine wave. In several control points, which were placed along the computational domain with step $\Delta x = 30$, dependences $p(t)$ were registered. After the calculation, RMS value of pressure perturbations

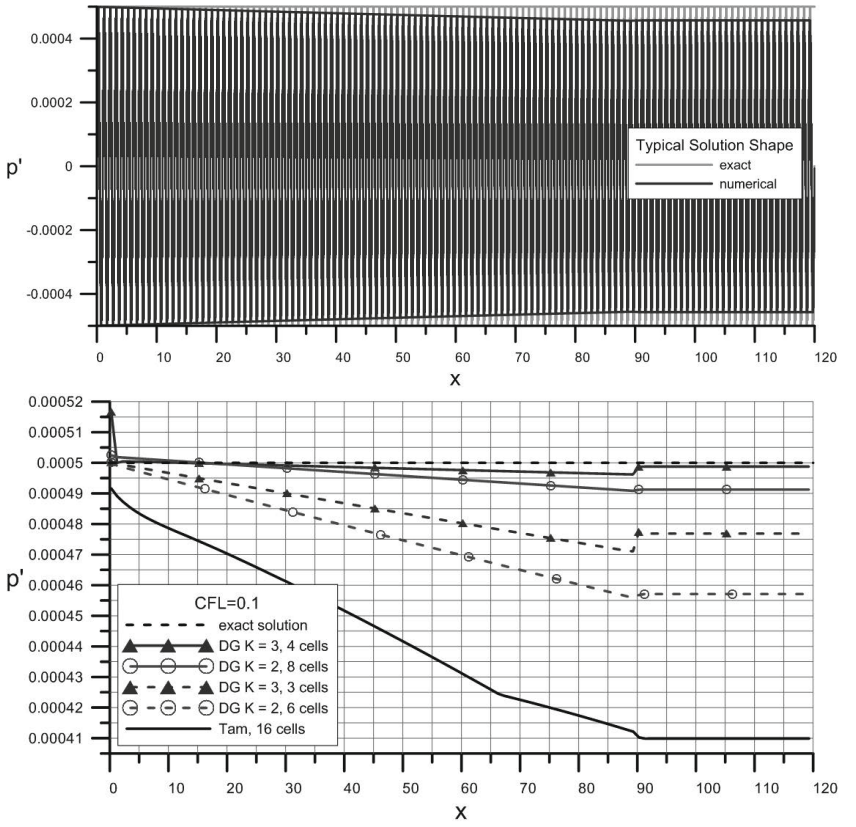


Fig. 17. Solution of task about 1D sine wave propagation at $t = 60$: a) shape of solution, obtained using DG scheme, vs. exact solution; b) solution envelopes for different grids

over the predefined physical time period $t \in [t_0; t_0 + T]$ was determined according to formula:

$$\Sigma \equiv \sqrt{\overline{p'^2}} = \sqrt{\frac{1}{T}\int\limits_{t_0}^{t_0+T} p^2(t)dt - \left(\frac{1}{T}\int\limits_{t_0}^{t_0+T} p(t)dt\right)^2}. \tag{40}$$

For this test, values $t_0 = 50$, $T = 10$ have been chosen. The order of convergence has been estimated using classical Richardson extrapolation that assumes that the parameter Σ (40) depends on the cell size h in accordance with formula

$$\Sigma(h) = \Sigma_{\text{exact}} + Ah^m, \tag{41}$$

where Σ_{exact} is the exact solution, A is some constant and m is the convergence order. Two approaches were considered. In the first approach, sequence of 3

grids was used, and Richardson extrapolation was used to estimate both the exact solution Σ_{exact} and the convergence order m. In the second approach, exact value of RMS for the sine wave was used as Σ_{exact}, and the convergence order m was estimated from the sequence if 2 grids. Results are presented in the following table. Results for different control points have appeared to be practically coinciding, so this table contains data only for each sequence of grids.

Grid sequence (cells per wavelength)	4/8/16	8/16/32	16/32/64	4/8	8/16	16/32	32/64
order m	undetermined	3.68	3.94	undetermined	3.70	3.84	2.91

One may see that the grid with 4 cells per wavelength appeared too coarse, so that it is still impossible to neglect the contribution of higher degrees of h in the dependence $\Sigma(h)$. Results for other sequences of grids are pretty close to the expected convergence order $m = 4$. The only exception is the sequence of two finest grids – 32/64 cells per wavelength, where the convergence order appeared to be close to 3. It is possible that at these grids the difference between $\Sigma(h)$ and Σ_{exact} becomes so small that the accuracy of the order m determination degrades because of truncation errors.

Finally, let's consider the results obtained for the standard test of IDIHOM project – test A14. This test is based on experiment with propagation of sound perturbations along the wing flap configuration, which was performed within EU/FP7 VALIANT project (40). Geometry of wing and flap may be seen in Figure 18. The following flow regime was considered: velocity $u = 50$ m/s, angle of attack $\alpha = 0°$, temperature $T = 298$ K. Reynolds number based on the flap chord was equal to 3.3×10^6. In the experiment, initial turbulence was generated by turbulizing grid placed at the left boundary of the domain that is shown in Figure 18.

Calculations of this test were performed in 2D (flat) formulation. 2D basis functions were used in DG method. Initially, RANS calculation has been performed on appropriate grid with detailed resolution of turbulent boundary layer (for uniform inflow). The converged flowfield obtained in RANS calculation has been used as basic (aerodynamic) flowfield.

Acoustic calculations have been performed using DG method $K = 3$ for four quasi-uniform grids with quasi-quadratic cells with 4, 6, 9 and 14 cells per the shortest wavelength of the incoming sound.

In the main series of calculations, stochastic perturbations were continuously introduced from the inflow boundary. Stochastic perturbations with nearly uniform flat distribution (in $1/3^{\text{rd}}$ octave presentation) within the frequency range $100 - 12000$ Hz were used. Instant field of pressure perturbations in the time moment $t = 0.00615$ sec is shown in Figure 19. A similar field for the case of

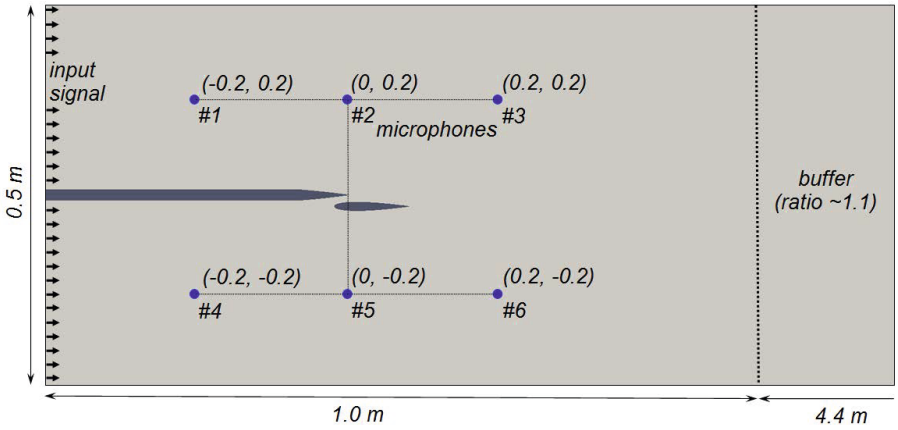


Fig. 18. Geometry of computational domain and position of control points for test A14

pulse propagation with the same frequency characteristics as in stochastic case is shown in Figure 20. One may see that incoming sound waves propagate (practically without damping) along the computational domain. Their diffraction and reflection from the flap tip may be seen.

After that, an attempt has been made to estimate the convergence order using the same procedure as in test with 1D propagation of a shock pulse. Dependencies $p(t)$ were registered in six control points that are shown in Figure 18. RMS value of pressure perturbations Σ was determined using (40). Using sequences of Σ for 3 grids, convergence order has been estimated. But this attempt has appeared to be unsuccessful. In some points the convergence order was undetermined, in other points it was far from the expected values, and in different points these values were very different.

To understand this result, additional series of calculations have been performed. Now the flat sine waves of maximal frequency (12000 Hz) were introduced continuously through the left boundary. Maximal frequency was chosen to have the maximal approximation errors and to avoid the influence of truncation errors on the results of Richardson extrapolation.

Attempt to estimate the convergence order has been performed again. It appeared to be unsuccessful, too – in spite of the fact than in the case with 1D propagation of the same wave on comparable grids we had a success. Analysis of the results allows to assume that the reason of the unsuccess in the test A14 is the interference of the main flat waves (propagating to the left) with the waves, reflected from the flap tip and from the upper on lower boundaries of computational domains. Even if all these waves have the same frequency, the behavior of signal in control points depends not only upon the grid resolution but also upon the phase shift between the interfering waves. As a result, formula (41) is inapplicable.

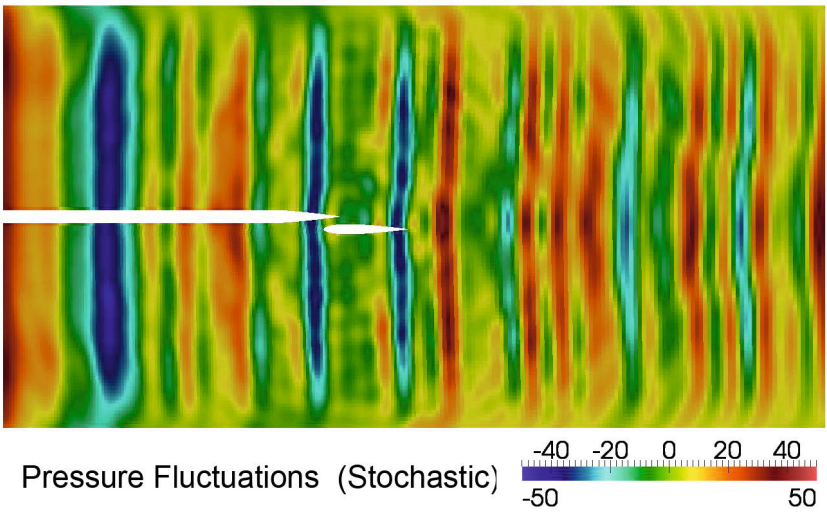


Fig. 19. Propagation of stochastic perturbations around the wing/flap configuration

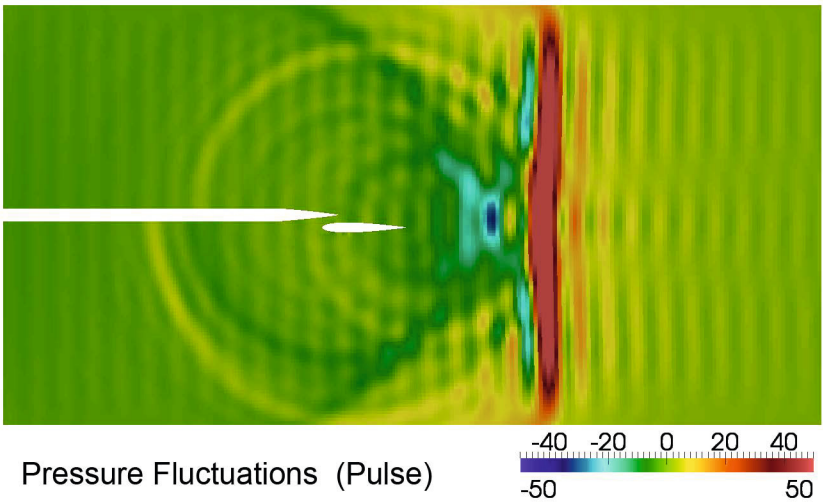


Fig. 20. Propagation of a pulse around the wing/flap configuration

11 Conclusions

1. High-order Discontinuous Galerkin method ($K = 2 - 3$) with 1st order implicit smoother is proposed for solution of stationary RANS equations, closed by non-Boussinesq turbulence model of EARSM class. The method is enhanced by the use of improved Gauss quadrature rules and of h-p multigrid acceleration and is implemented into NUMECA FINE^{TM}/Hexa code

in version for massive parallel calculations. Stable and efficient calculations of fully-subsonic flows are achieved with local time stepping with Courant number $CFL \leq 1000$. Stability of the method in calculations of flows with shock waves should be the subject of additional study. The use of h-p multigrid may diminish the calculation time by the order of magnitude. Parallel efficiency of method is close to 75% of ideal value.

2. Explicit high-order Discontinuous Galerkin method ($K = 2 - 3$) is implemented for solution of unsteady Isentropic Linearized Euler equations for solution of aeroacoustical tasks. Method with $K = 3$ allows to calculate propagation of sound waves on grids with only 3 cells per wavelength. Grid convergence order of the method is close to the expected value $m = 4$, if tasks without interference of waves with different frequency and phase shift are considered. In task with interference of waves, standard approach to determination of the convergence order seems to be inapplicable.

References

1. Reed, W.H., Hill, T.R.: Triangular mesh methods for the neutron transport equation. Technical Report LA–UR–73–479. Los Alamos Scientific Laboratory (1973)
2. Le Saint, P., Raviart, P.: On a finite element method for solving the neutron transport equation. In: de Boor, C. (ed.) Mathematical Aspects of Finite Elements in Partial Differential Equations, pp. 89–145. Academic Press, New York (1974)
3. Bassi, F., Rebay, S.: High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations. J. Comput. Phys. 138, 251–285 (1997)
4. Bassi, F., Rebay, S.: A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations. J. Comput. Phy. 131, 267–279 (1997)
5. Cokburn, B.: Discontinuous Galerkin methods for convection-dominated problems. In: Barth, T., Deconik, H. (eds.) High-Order Methods for Computational Physics. LNCSE, vol. 9. Springer, Berlin (1999)
6. Cockburn, B., Shu, C.-W.: Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. J. Sci. Comp. 16(3) (2001)
7. Atkins, H.L., Lockard, D.P.: A high-order method using unstructured grids for the aeroacoustic analysis of realistic aircraft configurations. AIAA Paper 99-1945 (1999)
8. Remaki, M., Habashi, W.G., Ait-Ali-Yahia, D., Jay, A.: A 3D discontinuous Galerkin method for multiple pure tone noise problems. AIAA Paper 2002-0229 (2002)
9. Reymen, Y., De Roeck, W., Rubio, G., Baelmans, M., Desmet, W.: A 3D discontinuous Galerkin method for aeroacoustic propagation. In: 12th International Congress on Sound and Vibration, Lisbon (2005)
10. Peyret, C.: Hp Discontinuous Galerkin Method for Computational Aeroacoustics. AIAA Paper 2007-3475 (2007)
11. Luo, H., Baum, J.D., Löhner, R.: A fast, p-multigrid discontinuous Galerkin method for compressible flows at all speeds. AIAA Paper 2006-110 (2006)
12. Bassi, F., Botti, L., Colombo, A., Crivellini, A., Franchina, N., Ghidoni, A., Rebay, S.: Very High-Order Accurate Discontinuous Galerkin Computation of Transonic Turbulent Flows on Aeronautical Configutations. ADIGMA. Results of a collaborative research project funded by the European Union (2006–2009)

13. Couaillier, V., Renac, F., Le Pape, M.C.: Hybrid Multigrid DG/FV Methods for Viscous Turbulent Flows. ADIGMA. Results of a collaborative research project funded by the European Union (2006–2009)
14. Wolkov, A., Hirsch, C., Leonard, B.: Discontinuous Galerkin Method on Unstructured Hexahedral Grids for 3D Euler and Navier-Stokes Equations. AIAA Paper 2007-4078 (2007)
15. http://nines.cs.kuleuven.be/research/ecf
16. Sobolev, S.L.: Introduction into theory of cubature formulas, p. 808. Nauka, Moscow (1974)
17. Sobolev, S.L., Vaskevitch, V.L.: Cubature formulas, p. 484. Publishing Company of Institute of Mathematics, Novosibirsk (1996)
18. Mysovskih, I.P.: Interpolational cubature formulas, p. 336. Nauka, Moscow (1981)
19. Menter, F.R., Garbaruk, A.V., Egorov, Y.: Explicit algebraic Reynolds stress models for anisotropic wall-bounded flows. In: Proc. EUCASS 2009, Versailles, France, p. 14 (2009)
20. Wilcox, D.C.: Turbulence modeling for CFD, 2nd edn. DCW Industries, La Cañada (1998)
21. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys. 43(2), 357–372 (1981)
22. Wang, Z.J.: High-order methods for the Euler and Navier-Stokes equations on unstructured grids. Prog. Aerosp. Sci. 43, 1–41 (2007)
23. Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., Savini, M.: A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Proc. 2nd European Conference on Turbomachinery – Fluid Dynamics and Thermodynamics, pp. 99–108. Technologisch Instituut, Antwerpen (1997)
24. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection diffusion system. SIAM J. Numer. Anal. 35(6), 2440–2463 (1998)
25. Troshin, A.I., Vlasenko, V.V., Wolkov, A.V.: Comparison of three viscous terms approximations for discontinuous Galerkin method by solving model convection-diffusion problems. In: Proc. European congress on computational methods in applied sciencies and engineering (ECCOMAS 2012), Vienna (2012)
26. Oran, E.S., Boris, J.P.: Numerical simulation of reactive flow, 2nd edn. Cambridge Univ. Press (2001)
27. Ortega, J.M.: Introduction to parallel and vector solution of linear systems. Plenum Press (1988)
28. Brandt, A.: Multilevel adaptive computations in fluid dynamics. AIAA Journal 18(10) (1980)
29. Wolkov, A.V.: Application of the Multigrid Approach for Solving 3D Navier-Stokes Equations on Hexahedral Grids Using the Discontinuous Galerkin Method. Comp. Math. and Math. Phys. 50(3), 495–508 (2010)
30. Wolkov, A., Hirsch, C., Leonard, B.: Discontinuous Galerkin Method on Unstructured Hexahedral Grids for 3D Euler and Navier-Stokes Equations. AIAA Paper 2007-4078 (2007)
31. http://www.mcs.anl.gov/research/projects/mpi
32. AGARD: A selection of experimental test cases for the validation of CFD codes. AGARD-AR-303, vol. 1, 139 p. (1994)
33. AGARD: Pressure distributions on the ONERA-M6-WING at transonic Mach numbers. AGARD-AR-138, Appendix B-1 (1979)
34. http://hiliftpw.larc.nasa.gov/index.html

35. Hirsch, C., Wolkov, A., Leonard, B.: Discontinuous Galerkin method on unstructured hexahedral grids. AIAA Paper 2009-177, 16 p. (2009); Comm. Pure Appl. Math. 33, 609–633 (1980)
36. Vlasenko, V., Bosniakov, S., Mikhailov, S., Morozov, A., Troshin, A.: Computational approach for investigation of thrust and acoustic performances of present-day nozzles. Prog. Aerosp. Sci. 46, 141–197 (2010)
37. Vlasenko, V., Wolkov, A.: Hirsch, Ch.: Computationally effective Discontinuous Galerkin scheme for linearized Euler equations. In: Proc. West-East High Speed Flow Field Conference, Moscow (2007)
38. Tam, C.K.W., Webb, J.C.: Dispersion-relation-preserving schemes for computational acoustics. J. Comput. Phys. 107, 262–281 (1993)
39. Roger, M., Lemoine, B., Legriffon, I., Terracol, M., Manoha, E.: Final report on Task 2.3 – Flap test case. Contributions of ECL and ONERA. Deliverable D18–02, Contract No. ACP8-GA-2009-233680-VALIANT (July 2011)

High-Order Residual Distribution and Error Estimation for Steady and Unsteady Compressible Flow

M. Vymazal[1], L. Koloszar[1], S. D'Angelo[1], N. Villedieu[1], M. Ricchiuto[2], and H. Deconinck[1]

[1] Von Karman Institute for Fluid Dynamics, Aeronautics and Aerospace Department, Chaussée de Waterloo 72, Rhode-St-Genèse, Belgium
[2] INRIA Bordeaux, France
{vymazal,koloszar,dangelo,deconinck}@vki.ac.be, mario.ricchiuto@inria.fr

Abstract. In the first part, an extension of upwind residual distribution schemes for high-order accurate solution of hyperbolic problems is introduced, based on the use of spatially varying distribution matrices . Following this, the application to adjoint-based error estimation for steady compressible flow is presented. Finally the resolution of acoustic wave propagation by a space-time residual distribution is discussed. The accuracy of the methodology is demonstrated on several test cases.

Keywords: CFD, high-order, residual distribution, adjoint error.

1 Residual Distribution with Variable β-Coefficients

We propose a new formulation of the residual distribution ($\mathcal{RD}$) method for steady hyperbolic conservation laws. Traditionally, the residual is distributed in parts to the degrees of freedom by means of piecewise-constant distribution coefficients per element. To retain the upwind character of the scheme, the calculation of these distribution coefficients required a sub-division of the element for high-order approximations. The sub-division of 3D or very high-order elements [1,2] was an inefficient, complex and error-prone process.

In this new formulation, we avoid the sub-division by defining variable non-linear distribution coefficients over the whole high-order element. These coefficients are anisotropic and depend on the hyperbolic operator. The new $\mathcal{RD}$ form makes the extension to generic 2D and 3D high–order elements much simpler. Moreover, it does not suffer from the undamped modes in $k = 1$ quadrilaterals, as found in [3].

The $\mathcal{RD}$ variant proposed here is in fact a Petrov-Galerkin Finite Element method. However, this method does retain many of the characteristics of the $\mathcal{RD}$ framework: a globally continuous discrete solution defined on Lagrangian P_k elements; a residual computed by integration of the differential operator and a very compact stencil; stabilization via a generalization of the $\mathcal{RD}$ multidimensional upwinding.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_17

1.1 Nomenclature

We seek the steady solution $\mathbf{u}$ of a system of hyperbolic conservation laws

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}}(\mathbf{u}) = 0 \tag{1}$$

in a domain Ω, where $\Omega \in \mathbb{R}^d$ and $d = 2$ or $d = 3$. The symbol $\boldsymbol{\mathcal{F}}$ denotes a tensor of inviscid fluxes: $\boldsymbol{\mathcal{F}}(\mathbf{u}) = (\mathbf{F}_1(\mathbf{u}), \mathbf{F}_2(\mathbf{u}), \mathbf{F}_3(\mathbf{u}))$ in three dimensions. We denote by $\mathcal{T}^d_{h,k}$ a d-dimensional triangulation of the domain Ω, whereby each element of the triangulation is a Lagrangian P_k Finite Element equipped with $N = \frac{(k+1)(k+2)}{2}$ degrees of freedom in 2D and $N = \frac{(k+1)(k+2)(k+3)}{6}$ degrees of freedom in 3D, respectively. In the following paragraphs, we will assume $d = 2$ and drop the superscript d as the description is identical in two and three dimensions.

For any given function $\mathbf{u}(\mathbf{x}, t), \mathbf{x} \in \Omega$, we define the spatial Finite Element approximation

$$\mathbf{u}^{h,k}(\mathbf{x}, t) = \sum_{i \in \mathcal{T}_{h,k}} \varphi_i^{h,k}(\mathbf{x}) \mathbf{u}_i(t), \tag{2}$$

where $\mathbf{u}_i(t)$ is the value of $\mathbf{u}$ at node i, i.e. $\mathbf{u}_i(t) = \mathbf{u}(\mathbf{x}_i, t)$, while $\varphi_i^{h,k}(\mathbf{x})$ denotes the k-th order Lagrangian basis function of node i.

Since the system (1) is hyperbolic, an arbitrary linear combination of its flux Jacobians can be decomposed as

$$\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{u})}{\partial \mathbf{u}} \cdot \mathbf{c} = c_1 \frac{\partial \mathbf{F}_1}{\partial \mathbf{u}} + c_2 \frac{\partial \mathbf{F}_2}{\partial \mathbf{u}} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1} \qquad \forall \mathbf{c} = (c_1, c_2)^T \in \mathbb{R}^2, \tag{3}$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues of $(\partial \boldsymbol{\mathcal{F}}(\mathbf{u})/\partial \mathbf{u} \cdot \mathbf{c})$ and $\mathbf{R}$ contains its right eigenvectors. Using the notation above, we define the following matrices:

$$\mathbf{K}^+_{\mathbf{c}} = \mathbf{R}\boldsymbol{\Lambda}^+\mathbf{R}^{-1} \qquad \mathbf{K}^-_{\mathbf{c}} = \mathbf{R}\boldsymbol{\Lambda}^-\mathbf{R}^{-1} \tag{4}$$

with $\boldsymbol{\Lambda}^+_{ii} = \max(0, \boldsymbol{\Lambda}_{ii})$ and $\boldsymbol{\Lambda}^-_{ii} = \min(0, \boldsymbol{\Lambda}_{ii})$. The subscript $\mathbf{c}$ stresses the fact that the Jacobian was projected onto direction $\mathbf{c}$.

1.2 Schemes

The standard multidimensional upwind scheme consists of three steps:

1. Compute the element residual:

$$\boldsymbol{\phi}^K = \int_K \nabla \cdot \boldsymbol{\mathcal{F}} \, d\mathbf{x} = \oint_{\partial K} \boldsymbol{\mathcal{F}} \cdot \mathbf{n} \, dl \tag{5}$$

2. Compute the contribution of the residual to each element node by evaluating the *distribution matrices* $\boldsymbol{\beta}_i^K$:

$$\boldsymbol{\phi}_i^K = \boldsymbol{\beta}_i^K \boldsymbol{\phi}^K, \tag{6}$$

where $\boldsymbol{\beta}_i^K$ is a function of a suitable linearized state $\mathbf{u}_*^h$ in the element. For example, introducing upwind parameters

$$\mathbf{K}_{\mathbf{n}_j} = \frac{1}{2}\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{u}_*^h)}{\partial \mathbf{u}} \cdot \mathbf{n}_j, \quad (\mathbf{n}_j \text{ is inward normal pointing towards node } j)$$

we can write the distribution matrices for the *LDA* scheme as

$$\boldsymbol{\beta}_i^{LDA} = \mathbf{K}_{\mathbf{n}_i}^+ \cdot \Big[\sum_{j\in K} \mathbf{K}_{\mathbf{n}_j}^+\Big]^{-1} \tag{7}$$

3. Evolve $\mathbf{u}^h$ in pseudo-time in order to solve the system until the steady state is reached

$$\frac{\partial \mathbf{u}_i^h}{\partial \tau} + \sum_{K, i\in K} \boldsymbol{\phi}_i^K = 0 \tag{8}$$

The drawback of this approach is that for higher-order approximation, one is forced to divide the element into linear sub-elements (triangles or tetrahedra in 3D) in order for the inward normals to be well defined. As the number of degrees of freedom increases, so does the number of sub-elements, rendering the method inefficient and complex to implement. Moreover, the definition of inward normals on tensor product elements is not straightforward.

For this reason, the requirement on the distribution matrix $\boldsymbol{\beta}$ to be constant per (sub-)element is relaxed and we consider the following $\mathcal{RD}$ prototype for computation of nodal residual $\boldsymbol{\phi}_i^K$, instead of (6):

$$\boldsymbol{\phi}_i^K = \int_K \boldsymbol{\beta}_i(\mathbf{u}^h, \nabla\varphi, \mathbf{x})(\nabla \cdot \boldsymbol{\mathcal{F}}(\mathbf{u}^h))\, d\mathbf{x} \tag{9}$$

On P_1 elements, the matrices $\boldsymbol{\beta}_i^K$ remain constant per element and the 'constant' and 'variable distribution coefficient' $\mathcal{RD}$ variants are equivalent under certain additional assumptions.

- *LDA*
  The distribution matrix for node i of element K is given by

$$\boldsymbol{\beta}_i^{LDA}(\mathbf{u}^h, \nabla\varphi, \mathbf{x}) = \mathbf{K}_{\nabla\varphi_i}^+ \cdot \Big[\sum_{j\in K} \mathbf{K}_{\nabla\varphi_j}^+\Big]^{-1} \tag{10}$$

- *SUPG*
  Let $\mathbf{D}_{\varphi_i}$ denote a diagonal matrix with entries $\mathbf{D}_{ii} = \varphi_i$. Then the SUPG system distribution matrix reads

$$\boldsymbol{\beta}_i^{SUPG}(\mathbf{u}^h, \nabla\varphi, \mathbf{x}) = \mathbf{D}_{\varphi_i} + \mathbf{K}_{\nabla\varphi_i} \cdot \Big[\sum_{j\in K} \mathbf{K}_{\nabla\varphi_j}^+\Big]^{-1} \tag{11}$$

  Nodal residuals for the LDA and SUPG schemes are given by (9) with $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^{LDA}$ or $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^{SUPG}$, respectively.

- *N scheme*

$$\boldsymbol{\phi}_i^N = \int_K \mathbf{K}^+_{\nabla\varphi_i}(\mathbf{u}_i^h - \mathbf{u}_*^h)\,d\mathbf{x} \tag{12}$$

and $\mathbf{u}_*$ is defined by the identity

$$\nabla\cdot\boldsymbol{\mathcal{F}}(\mathbf{u}) = \sum_{j\in K}\mathbf{K}^+_{\nabla\varphi_j}(\mathbf{u}_j^h - \mathbf{u}_*^h) \tag{13}$$

The N scheme has positive coefficients and can be recast as *LDA* with a dissipation term:

$$\boldsymbol{\phi}_i^N = \boldsymbol{\phi}_i^{LDA} + \sum_{j\in K}\int_K \mathbf{K}^+_{\nabla\varphi_i}\Big[\sum_{m\in K}\mathbf{K}^+_{\nabla\varphi_m}\Big]^{-1}\mathbf{K}^+_{\nabla\varphi_j}(\mathbf{u}_i^h - \mathbf{u}_j^h)\,d\mathbf{x} \tag{14}$$

- Nonlinear *blended* scheme
The amount of extra dissipation term in the N scheme can be controlled by a nonlinear switch $\theta(\mathbf{u})\in[0,1]$ which is activated around discontinuities:

$$\boldsymbol{\phi}_i^B = \boldsymbol{\phi}_i^{LDA} + \theta(\mathbf{u}^h)\sum_{j\in K}\int_K \mathbf{K}^+_{\nabla\varphi_i}\Big[\sum_{m\in K}\mathbf{K}^+_{\nabla\varphi_m}\Big]^{-1}\mathbf{K}^+_{\nabla\varphi_j}(\mathbf{u}_i^h - \mathbf{u}_j^h)\,d\mathbf{x} \tag{15}$$

1.3 Boundary Conditions

We follow the approach of [4] to weakly impose boundary conditions. Integrating (9) twice by parts (once forward and once backward) yields

$$\boldsymbol{\phi}_i^K = \int_K \boldsymbol{\beta}_i(\mathbf{u}^h,\mathbf{x})(\nabla\cdot\boldsymbol{\mathcal{F}}(\mathbf{u}^h))\,d\mathbf{x} + \int_{\partial\Omega}\varphi_i(\boldsymbol{\mathcal{F}}(\mathbf{u}^b) - \boldsymbol{\mathcal{F}}(\mathbf{u}^h))\cdot\mathbf{n}\,dl \tag{16}$$

The quantity $\mathbf{u}^b$ is determined by boundary conditions. In the second integral, the upwind test function $\boldsymbol{\beta}$ was replaced by standard Lagrange shape function in order to distribute the flux correction $\boldsymbol{\mathcal{F}}(\mathbf{u}^b)-\boldsymbol{\mathcal{F}}(\mathbf{u})$ in central manner. Practical computer implementation proceeds as follows: first all nodal residuals in all mesh elements are computed and distributed. In the second step, all elements K such that their face $\Gamma = K\cap\partial\Omega\neq\emptyset$, receive additional residuals

$$\boldsymbol{\phi}_i^\Gamma = \int_\Gamma \varphi_i(\boldsymbol{\mathcal{F}}(\mathbf{u}^b) - \boldsymbol{\mathcal{F}}(\mathbf{u}^h))\cdot\mathbf{n}\,dl \qquad \forall i\in\Gamma \tag{17}$$

The correction fluxes for typical boundary conditions are listed below.

- *Wall*

$$\boldsymbol{\mathcal{F}}(\mathbf{u}^b)\cdot\mathbf{n} = (0,p\mathbf{n},0)^T,\ \text{where } p \text{ is pressure on the wall} \tag{18}$$

- *Subsonic inlet and outlet*

$$(\boldsymbol{\mathcal{F}}(\mathbf{u}^b) - \boldsymbol{\mathcal{F}}(\mathbf{u}^h))\cdot\mathbf{n} = \mathbf{K}^-_{\mathbf{n}}(\mathbf{u}^h)(\mathbf{u}^b - \mathbf{u}^h) \tag{19}$$

For higher-order approximation, equal-order interpolation is used for both the geometry and the solution. The geometry of a curvilinear element is defined by an transformation to a parent element in $\xi-\eta$ space.

1.4 Applications

Figure 1 presents inviscid computation on P_2 and P_3 elements around the NACA0012 airfoil at a freestream Mach number $M = 0.5$ and angle of attack $\alpha = 2°$ and demonstrates the capability of the method to run on both unstructured simplex meshes and structured or hybrid meshes composed of Lagrangian elements.

Figure 2 shows the inviscid transonic flow around the Onera M6 wing computed on a mesh provided by Warsaw Institute of Technology. The solution was obtained by the SUPG scheme on P_2 elements.

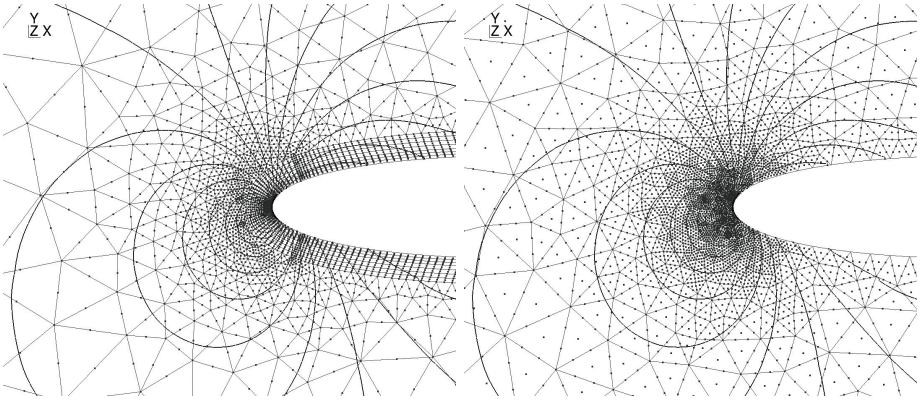


Fig. 1. Subsonic flow around NACA 0012 - Mach isolines on P_2 mesh with mixed element types (left) and on P_3 triangular mesh (right)

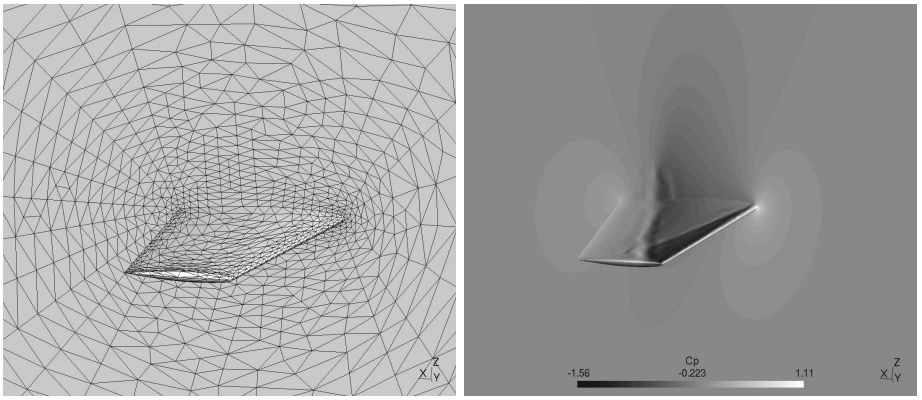


Fig. 2. Onera M6 test case (freestream Mach number $M = 0.8395$, angle of attack $\alpha = 3.06$) - P_2 mesh and isolines of pressure coefficient

2 *A Posteriori* Error Estimation

In many applications there is particular interest in specific quantities of the solution. In aerodynamics, for example, these include physically relevant integral quantities (like the drag or lift of an airfoil) or single density or pressure values on the profile of the airfoil. Consequently, there is considerable interest in constructing *a posteriori* error estimates for such derived quantities so as to improve the reliability and efficiency of numerical computations.

So, if we call define the quantity of interest as a functional $J(\boldsymbol{u})$, then, given a positive tolerance TOL, we consider the design of an adaptive algorithm with the stopping criterion as follows

$$|J(\boldsymbol{u}) - J(\boldsymbol{u}_h)| \leq \mathsf{TOL}. \tag{20}$$

where $J(\boldsymbol{u}_h)$ is the numerical evaluation of the target quantity. To this aim, we purvey an *a posteriori* error bound, $\mathcal{E}(\boldsymbol{u}_h)$, controlling the numerical solution measured in terms of the target functional, in order to be able to write

$$|J(\boldsymbol{u}) - J(\boldsymbol{u}_h)| \approx \mathcal{E}(\boldsymbol{u}_h) \leq \mathsf{TOL}. \tag{21}$$

This error estimation is obtained by using duality arguments and the corresponding adjoint solution, $\boldsymbol{z}$, that let us capture information about the local error propagation. Here below, we will describe the error analysis and adaptivity based on a hyperbolic adjoint argument using the general approach developed by Becker and Rannacher [5], later recalled in [7] and expanded in [9].

2.1 Adjoint-Based Error Estimation

We first introduce some notation about duality based *a posteriori* error estimation for a given target quantity, see [6]. Consider the nonlinear problem

$$Nu = 0 \quad \text{in } \Omega \qquad\qquad Bu = 0 \quad \text{on } \Gamma, \tag{22}$$

where N is a nonlinear differential operator and B a boundary operator. We define a nonlinear target functional

$$J(u) = \int_{\Omega} j_{\Omega}(u)\, d\boldsymbol{x} + \int_{\Gamma} j_{\Gamma}(Cu)\, ds, \tag{23}$$

where $j_{\Omega}(\cdot)$ and $j_{\Gamma}(\cdot)$ may be nonlinear with derivatives j'_{Ω} and j'_{Γ}, respectively, while C is a differential boundary operator on Γ with derivative C'. This target functional (23) is call *compatible* with the primal problem (22) if the following compatibility condition holds

$$\big(N'[u]w, z\big)_{\Omega} + \big(B'[u]w, (C'[u])^* z\big)_{\Gamma} = \big(w, (N'[u])^* z\big)_{\Omega} + \big(C'[u]w, (B'[u])^* z)\big)_{\Gamma}, \tag{24}$$

where $(N'[u])^*$, $(B'[u])^*$ and $(C'[u])^*$ denote the adjoint operators with respect to $N'[u]$, $B'[u]$ and $C'[u]$. Hence, from (24), we can determine the continuous adjoint problem associated to (22) and (23)

$$(N'[u])^* z = j'_\Omega[u] \quad \text{in } \Omega \qquad\qquad (B'[u])^* z = j'_\Gamma[Cu] \quad \text{on } \Gamma. \tag{25}$$

Therefore, through a suitable numerical discretisation based on the weak form, we recover a semi-linear form, linear in the first argument and nonlinear in its second argument, $\mathcal{N}_h(\cdot,\cdot)$, such that the nonlinear problem (22) in discrete form becomes

$$\mathcal{N}(u_h, \tilde{v}_h) = 0 \quad \forall \tilde{v}_h \in \widetilde{\mathcal{V}}_h, \tag{26}$$

where $u_h \in \mathcal{V}_h$ is the numerical solution of u belonging to a given discrete space and $\tilde{v}_h$ the test function of the numerical scheme defined over the space $\widetilde{\mathcal{V}}_h$. By the compatibility condition (24) and using infinite-dimensional trial and test spaces $\mathcal{V}$, we introduce the corresponding linearized adjoint problem as follows

$$\overline{\mathcal{M}}(w, z) = \bar{J}(w) \qquad \forall w \in \mathcal{V}. \tag{27}$$

where $\overline{\mathcal{M}}(w,z)$ and $\bar{J}(w)$ stand for the mean-value linearisation of the nonlinear operator $\mathcal{N}(u,v)$ and the target functional $J(u)$, respectively.

We denote by $\tilde{z}_h$ the adjoint solution belonging to the test discrete space $\widetilde{\mathcal{V}}_h$, used in the discrete primal problem. This solution may be computed directly by an adjoint consistent discrete problem (27) or by any suitable projection operator (i.e. interpolation, L_2 projection) whose imagine belongs to $\widetilde{\mathcal{V}}_h$.

In the wake of [8], an exact *error representation formula* comes out as follows

$$J(u) - J(u_h) = \mathcal{R}_\Omega(u_h, z - \tilde{z}_h) \equiv \sum_{\kappa \in \mathcal{K}_h} \eta_\kappa, \tag{28}$$

where $\mathcal{R}_\Omega(u_h, z - \tilde{z}_h) = -\mathcal{N}(u_h, z - \tilde{z}_h)$ and the *local adjoint-based indicator*

$$\eta_\kappa = \int_\kappa R(u_h)(z - \tilde{z}_h)\, d\boldsymbol{x} + \int_{\partial\kappa \cap \Gamma} r(u_h)(z - \tilde{z}_h)\, ds, \tag{29}$$

with the local residuals $R(u) = -Nu$ and $r(u) = -Bu$. However, this error representation formula does not indicate which elements in the mesh should be refined to reduce the measured error in the functional. To do this, an error localization procedure has to be defined to point out a local contribution of each element to the functional error. By applying the *triangle inequality*, indeed, we have

$$\left|J(u) - J(u_h)\right| \leq \mathcal{R}_{|\Omega|}(u_h, z - \tilde{z}_h) \equiv \sum_{\kappa \in \mathcal{K}_h} |\eta_\kappa|. \tag{30}$$

Unfortunately, this estimation is not still computable because of the unknown analytical solution, z. Thus, in order to finally make these error estimate computable, z must be replaced by a suitable approximation which does not affect negatively the quality of the error bound. The analytical adjoint solution must then be numerically defined on a sequence of suitable adjoint finite element spaces $\overline{\mathcal{V}}_h$, based on a finer adjoint partition $\overline{\mathcal{K}}_h$ or an higher adjoint polynomial $\bar{p}$.

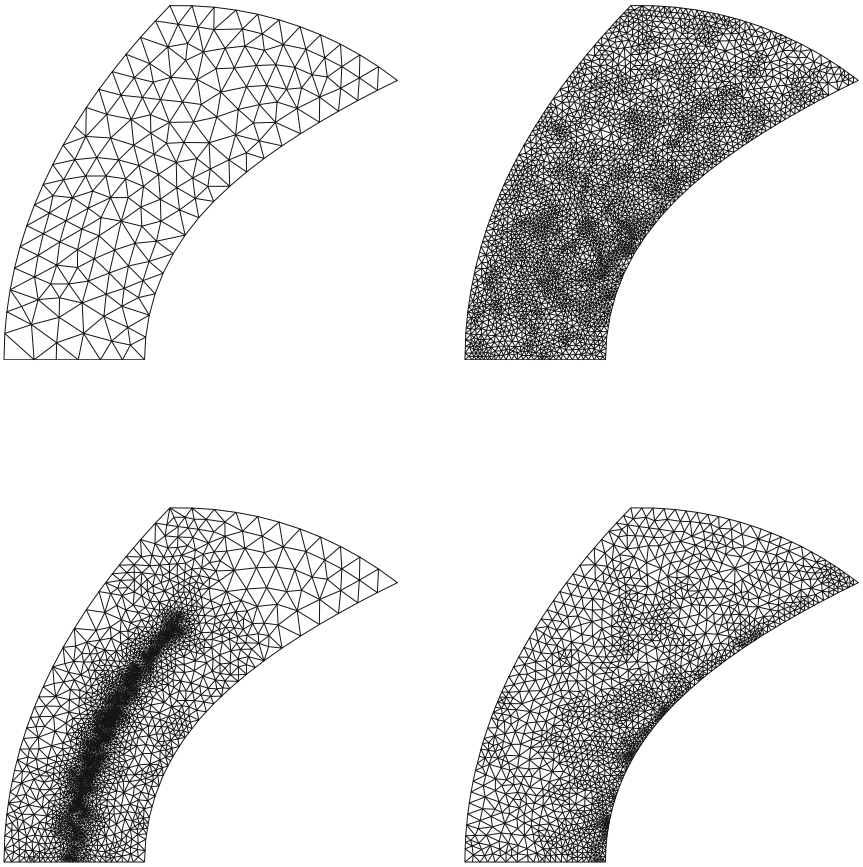


Fig. 3. Ringleb problem for pointwise and wall force target: (a) initial mesh with 317 triangles, (b) residual-based adaptation with 6312 triangles and $J(e) = 1.055\,10^{-5}$ for point target and $J(e) = 8.481\,10^{-6}$ for wall force target, (c) goal oriented adjoint-based with 5192 triangles and $J(e) = 7.600\,10^{-6}$ for point target and (d) goal oriented adjoint-based with 2853 triangles and $J(e) = 8.366\,10^{-6}$ for wall force

2.2 Numerical Example: Ringleb Problem

The Ringleb problem consists of a smooth transonic flow in a channel. The left and right boundaries are considered as reflective walls while the bottom and upper boundary are the inlet and outlet, respectively. This problem is one of the few non-trivial examples of the Euler 2D equations where a smooth analytical solution can be deduced. Hence, it becomes a interesting test to accurately prove the sharpness of the error representation for the 2D Euler equations.

Using the SUPG scheme, we choose two different target functionals, a pointwise and an integral one and we compare adaptive mesh procedures based on the residuals energy norm and on the adjoint solution. As first quantity of interest, we choose the physical variable, ρ, at a given point, such that $J(\boldsymbol{u}) = \rho(-0.4, 2.0)$. The corresponding adjoint solution will be a singularity driven backward by a spike from that point to the inlet boundary. Figure 3 plots the final meshes driven by a standard residual-based and with the adjoint-based error estimation. We note that by the use of a goal-oriented refinement, we are able to reach a final error 30% smaller with 20% less of degrees of freedom.

As a second test, we consider as a target quantity, the force magnitude applied on the right wall of the channel, whose exact value is $J(\boldsymbol{u}) = 1.10567714227773$. The standard iterative procedure is not affected by the choice of the target quantity and so, it generates the same meshes for any target considered. On the other hand, by using the adjoint-based indicators and then a goal oriented refinement, we are able to reach the same order of error magnitude, with a much coarser mesh. Moreover, here the refinement is mainly focused along the right side of the domain and in particular, where the curvature of the boundary is higher as there, both error sensitivity and force contribution are higher.

3 Space-Time $\mathcal{RD}$ Method for Linearized Euler Equations

For unsteady cases, such as acoustic wave propagation, the constant β Residual Distribution Method [11] is considered in a space-time framework. This means that time is handled as an additional space dimension. In this work the existing P_1 LEE solver was extended to be able to handle P2, quadratic elements. The formulation used in the P_2 case turned out to be very time consuming, therefore a new formula for residual calculation has been implemented.

3.1 Extension of the Space-Time Framework to P2 Elements

Consider the Linearized Euler Equations (LEE) in a uniform flow without source term:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}}(\mathbf{u}) = 0 \tag{31}$$

This can be also be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{u}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{u}}{\partial y} = 0, \tag{32}$$

where $\mathbf{u} = (\rho', \rho_0 u', \rho_0 v', p')^T$, the superscript (ρ') denotes the perturbation and the subscript (ρ_0) refers to the value of mean flow. ρ is the density, (u, v) represents the velocity components and p the pressure. Finally, c_0 is the speed of sound.

We use the approach of Ricchiuto et al. [10], [11] to extend space-time schemes to high order discretization. To provide high order accuracy both in space and in time, we combine quadratic triangular elements in space with a quadratic discretization of time. This means that each triangular element in space is equipped with 6 degrees of freedom. We split this triangle in four sub-elements $\{T_s\}_{s=1,\ldots,4}$ like on figure 4. This yields the new space-time prism of figure 5 which has three levels in time. Within this prism, the solution is approximated by:

$$\mathbf{u}^h = \sum_l H^l(t) \sum_{i \in \tau_h} \psi_i(x,y) \mathbf{u}_i^l \tag{33}$$

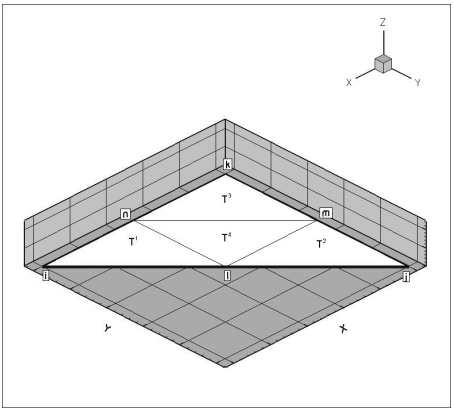


Fig. 4. Quadratic elements in space

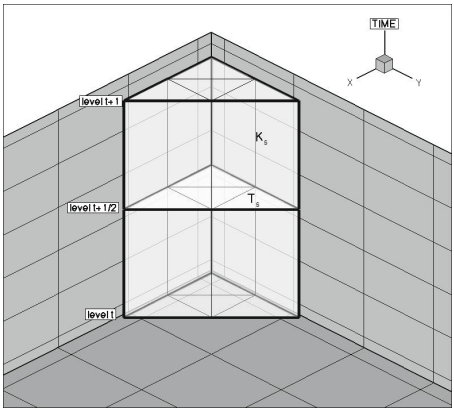


Fig. 5. Quadratic space-time element

Where $\mathbf{u}_i^l$ is the value of $\mathbf{u}^h$ at node i and time t_l : $\mathbf{u}_i^l = \mathbf{u}^h(x_i, y_i, t^l)$ and where $\psi_i(x,y)$ denotes the (mesh dependent) quadratic continuous Lagrangian basis function, and $H^l(t)$ is the 1D (time dependent) quadratic basis function. We split this prism in sub-prisms based on the sub-elements of the quadratic triangle (as illustrated on figure 5). In each of the sub-prisms we define the space-time upwind parameters as:

$$\begin{aligned} \hat{k}_i^n &= \tfrac{\Delta t}{2} k_i - \tfrac{|T_s|}{3}, \\ \hat{k}_i^{n+1} &= \tfrac{\Delta t}{2} k_i + \tfrac{|T_s|}{3} \end{aligned} \quad i \in T_s \tag{34}$$

Where k_i is the spatial upwind parameter defined by:

$$k_i = \mathbf{A} n_{i_x} + \mathbf{B} n_{i_y} \tag{35}$$

and where n_i is the inward normal to the edge facing node $i \in T_s$, and scaled with the length of the edge. At each time iteration $\mathbf{u}^{n-1}$ and $\mathbf{u}^n$ are known and

we want to compute $\mathbf{u}^{n+1}$ using the usual steps. First, we compute the residual on each sub-prism between t^{n+1} and t^n:

$$\boldsymbol{\phi}^{K_s} = \int_{t_n}^{t_{n+1}} \int_{T_s} \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{u}) \right) dx\, dy\, dt \tag{36}$$

Then, the residual is distributed to all the nodes of the sub-triangle T_s of the level:

$$\begin{aligned} \boldsymbol{\phi}_i^{n-1} &= 0 \\ \boldsymbol{\phi}_i^{n} &= 0 \\ \boldsymbol{\phi}_i^{n+1} &= \boldsymbol{\beta}_i^{T_s} \boldsymbol{\phi}_i^{K_s} \end{aligned}$$

Finally, we assemble the nodal contributions and we use pseudo-time iterations to solve the final system: (with C_i a CFL-dependent stability constant):

$$\mathbf{u}_i^{n+1,\tau+1} = \mathbf{u}_i^{n+1,\tau} + \frac{\Delta\tau}{C_i} \left(\sum_{T_s, i \in T_s} \boldsymbol{\phi}_i^{K_s} \right)^{\tau} \tag{37}$$

3.2 High Order Discretization of Source Terms

In the following we will focus on the way to discretize the source terms. Let us consider the complete LEE:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{u}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{u}}{\partial y} + \mathbf{H} = \mathbf{S}, \tag{38}$$

$$\mathbf{H} = \begin{bmatrix} (\rho' u_0 + \rho_0 u') \frac{\partial u_0}{\partial x} + (\rho' v_0 + \rho_0 v') \frac{\partial u_0}{\partial y} \\ (\rho' u_0 + \rho_0 u') \frac{\partial v_0}{\partial x} + (\rho' v_0 + \rho_0 v') \frac{\partial v_0}{\partial y} \\ (\gamma - 1) p' \left(\frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \right) - (\gamma - 1) \left(u' \frac{\partial p_0}{\partial x} + v' \frac{\partial p_0}{\partial y} \right) \end{bmatrix} \tag{39}$$

$\mathbf{H}$ represents the contribution due to non-uniformity in the mean flow, and $\mathbf{S}$ is the acoustic source term such as monopole, dipole... They are both treated as a combined source term $\mathbf{Q} = \mathbf{S} - \mathbf{H}$ and discretized by quadratic functions:

$$\mathbf{Q}^h = \sum_l H^l(t) \sum_{i \in T} \psi_i(x, y) \mathbf{Q}_i^l \tag{40}$$

This source term is simply added in the computation of the residual as follows:

$$\boldsymbol{\phi}^{K_s} = \int_{t_n}^{t_{n+1}} \int_{T_s} \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{u}) - \mathbf{S}^h(\mathbf{u}) \right) dx\, dy\, dt \tag{41}$$

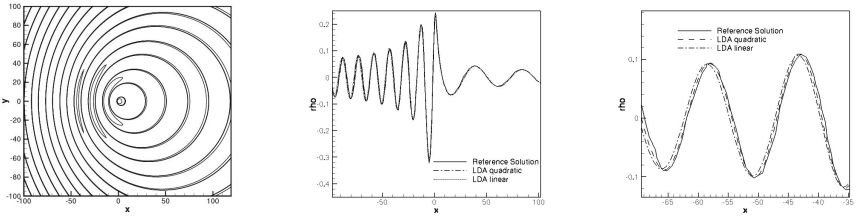
Then, the residual is distributed and treated as usual.

3.3 Comparison between 3rd and 2nd Order Schemes

Let us consider the propagation of a monopole in a uniform mean flow. The domain extends from $-100 \leq x, y \leq 100$ and the mesh composed of 201×201 nodes. The monopole is implemented by using the following source vector:

$$\mathbf{S} = f(x,y)\sin(\omega t)\begin{bmatrix}1\\0\\0\\1\end{bmatrix} \qquad f(x,y) = \varepsilon \exp[-\alpha((x-x_s)^2 + (y-y_s)^2)] \quad (42)$$

We choose the Mach number equal to $M = 0.5$ and the source is located at the center of the domain. The following parameters identify the source: the amplitude is $\varepsilon = 0.5$, the width of the source is $\alpha = \frac{\ln 2}{2}$ and the angular frequency is $\omega = \frac{2\pi}{30}$. We use this test case to compare the performances of linear (P_1) and quadratic elements (P_2). Figure 6a shows the iso-contours of pressure perturbation obtained with a quadratic discretization of the solution. In the profile we can observe two acoustic waves. The first is propagating upstream



(a) Iso-contours of pressure with values $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$

(b) Pressure profile along the axis $y = 0$

(c) Zoom on on the upwind wave

Fig. 6. Monopole source in a uniform subsonic flow (M=0.5) at t=270s

with a velocity of $1 - M$ and the wavelength is $\lambda_{up} = (1 - M)\lambda$. The second wave is propagating downstream with a velocity of $1 + M$ and the wavelength is $\lambda_{down} = (1 + M)\lambda$. On Figure 6b and Figure 6c we compare the solutions obtained with linear and quadratic elements at $t = 270s$ with the exact solution given in [12].We can see that the results of both discretizations are comparable.

Now, we want to study from the accuracy and CPU time point of view if the P_2 scheme is more efficient than the P_1 one. To do this, we keep the same test case and the same domain. We stop the simulation at t=80s and we compute the L_2-error:

$$\varepsilon = \frac{\sqrt{\sum_i \left(\mathbf{u}_i^h - \mathbf{u}(x_i, y_i, 80)\right)^2}}{\#DOF} \quad (43)$$

Table 1. Comparison of L_2 error and CPU time on P_1 and P_2 mesh

	h=2.4	h=2.0	h=1.4	h=1.2
$\Delta t(P_1)$	0.48	0.38	0.29	0.22998
CPU time (P_1)	1038s	2184s	4428s	9936s
$\varepsilon(P_1)$	1.76086 e-2	1.72714 e-2	1.6673 e-2	1.6688 e-2
$\Delta t(P_2)$	0.1	0.09	0.03	0.03
CPU time (P_2)	18177s	32604s	174090s	256668s
$\varepsilon(P_2)$	2.542 e-2	2.54619 e-2	2.5911 e-2	2.5809 e-2

In the table 1 are detailed the results obtained for mesh sizes of 2.4, 2.0, 1.4 and 1.0. We compare the error, CPU time and the time step used for both schemes on the same number of degrees of freedom. The first important thing to notice is that, usually, it is possible to use much bigger time steps with P_1 elements. This phenomenon is due to the fact that the P_2 schemes lack of stability when the mesh is not refined enough. Then, to be able to obtain a result it is necessary to drastically reduce the time step which increases a lot the computational time. Moreover, the error obtained is usually bigger with the P_2 schemes than with P_1 one.

3.4 Alternative Residual Calculation

Instead of using the quadrature rule to compute the residual over an element, a faster but approximate way to calculate the cell residual is proposed here. The cell residual, in absence of a source and diffusive term, is given by

$$\boldsymbol{\phi}^T = \int_T \nabla \cdot \boldsymbol{\mathcal{F}}(\mathbf{u})\, d\Omega \tag{44}$$

and instead of transforming this integral into a surface integral and using a quadrature rule, the flux is discretized using the FE-like discretization:

$$\boldsymbol{\phi}^T = \int_T \nabla \cdot \boldsymbol{\mathcal{F}}(\mathbf{u})\, d\Omega = \int_T \nabla \cdot \sum_{i \in T} \boldsymbol{\mathcal{F}}_i \psi_i^{P_1}(x, y)\, d\Omega \tag{45}$$

Applying the gradient operator to the shape functions yields the scaled face normals of the cells. Now, the cell residual can be rewritten as:

$$\begin{aligned} \boldsymbol{\phi}^T &= \int_T \sum_{i \in T} \boldsymbol{\mathcal{F}}_i \frac{\mathbf{n}_i}{2|T|}\, d\Omega \\ &= \frac{1}{2} \sum_{i \in T} \boldsymbol{\mathcal{F}}_i \mathbf{n}_i \\ &= \frac{1}{2} \sum_{i \in T} ((f_x)_i \cdot n_i^x + (f_y)_i \cdot n_i^y) \end{aligned}$$

This result holds for the two-dimensional case. For the three-dimensional case, the following expression for the cell residual is used:

$$\phi^T = \frac{1}{3} \sum_{i \in T} \left[(f_x)_i \cdot n_i^x + (f_y)_i \cdot n_i^y + (f_z)_i \cdot n_i^z \right]$$

The derivation shown before suggests that for a linear advection case, the new and the previous implementation of the cell residual calculation must give exactly the same results. In order to check the implementation, a 3D linear advection test case was run for the two different implementations and no difference in the cell residuals was found.

Figure 7 presents the scaling of the run-time with the number of processors. The numerical setup for this test is the fine grid having 16.7 M grid cells. The simulation is run for 5 time steps, then an average of the run-time per time step is taken. The results show that the approximate calculation with the new method is 29% faster on average.

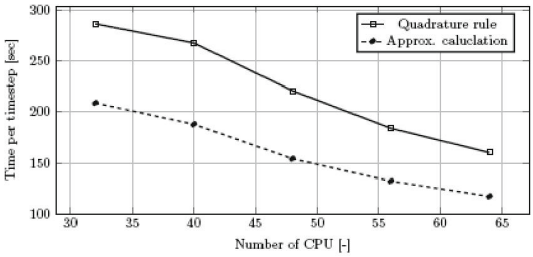


Fig. 7. Comparison of the simulation run-time for the standard and approximate calculation of the cell residual

4 Conclusions

New developments have been discussed on how to extend residual distribution schemes on P_1 elements to higher-order Finite Elements. A goal oriented a *posteriori* error estimation driving a mesh adaptation algorithm for the new class of schemes has been introduced.

Finally, the Linearized Euler Equation solver of the COOLFluiD code via space-time Residual Distribution Method was successfully extended to P_2 elements both in space and time. It was found that the P_2 method is not giving significant improvements over the P_1 implementation, but increases the simulation time significantly.

References

1. Abgrall, R., Roe, P.L.: High order fluctuation schemes on triangular meshes. Journal of Computational Physics 19, 3–36 (2003)
2. Ricchiuto, M., Villedieu, N., Abgrall, R., Deconinck, H.: On uniformly high–order accurate residual distribution schemes for advection diffusion. J. Comput. Appl. Math. 215, 547–556 (2008)

3. Rubino, D.T.: Residual distribution schemes for advection and advection–diffusion problems on quadrilateral and hybrid meshes. PhD Thesis, Politecnico di Bari (2005)
4. Abgrall, R., De Santis, D., Ricchiuto, M.: Construction of a high order Residual Distribution scheme for complex viscous flows. In: Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13 (2012)
5. Becker, R., Rannacher, R.: A Feed-Back Approach to Error Control in Finite Element Methods: Basis Analysis and Examples. J. Numer. Math. 4, 237–264 (1996)
6. Becker, R., Rannacher, R.: An optimal control approach to a-posteriori error estimation in finite element methods. Acta Numerica 10, 1–102 (2001)
7. üli, E.S., Houston, P.: Adaptive Finite Element Approximation of Hyperbolic Problems, pp. 269–344. Springer (2002)
8. Barth, T.J.: A Posteriori Error Estimation and Mesh Adaptivity for Finite Volume and Finite Element Method. NASA Ames Research Center, Moffet Field (2002)
9. Hartmann, R.: Adaptive Finite Element Methods for the Compressible Euler Equations. PhD thesis, University of Heidelberg (2002)
10. Ricchiuto, M., Abgrall, R., Deconinck, H.: Construction of very high order residual distributive schemes for unsteady scalar advection: preliminary results. 33rd computational fluid dynamics novel methods for solving convection dominated systems, vol. 5. Von Karman Institute for Fluid Dynamics (2003)
11. Koloszár, L., Villedieu, N., Quintino, T., Rambaud, P., Deconinck, H., Anthoine, J.: Residual distribution method for aeroacoustics. AIAA Journal 2011 0001-1452 49(5), 1021–1037 (2011)
12. Bailly, C., Juvé, D.: Numerical solution of acoustic propagation problems linearized Euler's equations. In: Proceedings of the 6th AIAA/CEAS Aeroacoustic Conference (2000)

Recent Progress in High-Order Residual-Based Compact Schemes for Compressible Flow Simulations: Toward Scale-Resolving Simulations and Complex Geometries

P. Cinnella, C. Content, K. Grimich, A. Lerat, and P.Y. Outtier

Laboratoire DynFluid, Arts et Métiers ParisTech, 151 bd. de l'Hôpital, 75013, Paris, France
paola.cinnella@ensam.eu

Abstract. Recent developments about the extension of high-order Residual-Based Compact schemes to unsteady flows and complex configurations are discussed, with application to scale-resolving simulations and complex turbomachinery flows.

Keywords: High-order, Residual-Based Compact Scheme, Scale-Resolving, Overset grids.

1 Introduction

This paper summarizes recent developments of a family of high-order Residual-Based Compact (RBC) schemes, initially proposed by [1, 2]. Differently from standard numerical schemes that approximate space derivatives independently in each space direction, RBC schemes seek for a compact approximation of the complete residual r, i.e. the sum of all derivatives in the governing equations. Because of this feature, RBC schemes belong to the group of so-called genuinely multidimensional schemes such as the fluctuation splitting schemes or the Residual Distribution (RD) schemes (see, e.g. [3, 4]).

Major developments described in the following focus on the extension of RBC schemes to unsteady flows and to complex geometries. Selected applications, including two application challenges among those considered in the IDIHOM project, are presented to assess the capabilities of the schemes, specifically for scale-resolving simulations and complex transonic turbomachinery flows.

2 RBC Schemes

In this Section, we recall the design principles of RBC approximations of the space derivatives for a hyperbolic system of conservation laws. For the sake of brevity, we will focus on two-dimensional problems, but there is no restriction to extend the analysis to multi-dimensional hyperbolic problems. At this stage, we treat time derivatives exactly, i.e. we focus on semi-discrete approximations in space.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_18

2.1 High-Order RBC Schemes

Let us consider the hyperbolic system of conservation laws:

$$w_t + f_x + g_y = 0 \text{ on } \mathbb{R}^2 \times \mathbb{R}^+ \tag{1}$$

with initial conditions

$$w(x, y, 0) = w_0(x, y)$$

where t is the time, x and y are Cartesian space coordinates, w is the state vector and $f = f(w)$, $g = g(w)$ are flux components depending smoothly on w. The Jacobian matrices of the flux are denoted $A = \mathrm{d}f/\mathrm{d}w$ and $B = \mathrm{d}g/\mathrm{d}w$. System (1) is approximated in space on a uniform mesh ($x_j = j\delta x, y_k = k\delta y$), with steps δx and δy of the same order of magnitude, say $\mathcal{O}(h)$, using the basic difference and average operators:

$$(\delta_1 v)_{j+\frac{1}{2},k} = v_{j+1,k} - v_{j,k} \qquad (\delta_2 v)_{j,k+\frac{1}{2}} = v_{j,k+1} - v_{j,k}$$
$$(\mu_1 v)_{j+\frac{1}{2},k} = \frac{1}{2}(v_{j+1,k} + v_{j,k}) \qquad (\mu_2 v)_{j,k+\frac{1}{2}} = \frac{1}{2}(v_{j,k+1} + v_{j,k})$$

where j and k are integers or half integers.

A *residual-based* scheme is expressed in terms of approximations of the exact residual:

$$r := w_t + f_x + g_y \tag{2}$$

More precisely, such a scheme is of the form:

$$(\tilde{r_0})_{j,k} = \tilde{d}_{j,k} \tag{3}$$

where $\tilde{r_0}$ is a space-centered approximation of r called the *main residual* and $\tilde{d}$ is a residual-based dissipation term defined as:

$$\tilde{d}_{j,k} = \frac{1}{2}[\delta_1(\Phi_1 \tilde{r_1}) + \delta_2(\Phi_2 \tilde{r_2})]_{j,k} \tag{4}$$

where $\tilde{r_1}$ and $\tilde{r_2}$, respectively defined at $j + \frac{1}{2}, k$ and $j, k + \frac{1}{2}$, are also space-centered approximations of r called the *mid-point residuals*, and Φ_1, Φ_2 are numerical viscosity matrices. These matrices depend only on the eigensystems of the Jacobian matrices A and B and on the steps δx and δy. They are designed once for all [5] and use no tuning parameters nor limiters. Since the matrices Φ_1 and Φ_2 remain $\mathcal{O}(1)$ as δx and δy tend to zero, the dissipation $\tilde{d}$ represents, to the leading order, a numerical approximation of the second-order partial differential term:

$$d = \frac{\delta x}{2}(\Phi_1 r)_x + \frac{\delta y}{2}(\Phi_2 r)_y \tag{5}$$

This leading term of the expansion, that is only first order accurate, vanishes for an exact solution ($r = 0$), so that $\tilde{d}$ is actually consistent with a high-order dissipation term that will be discussed later.

In the following, the residuals are discretized by using compact formulae such that:

$$\begin{aligned}(\tilde{r_0})_{j,k} &= r_{j,k} + \mathcal{O}(h^{2p}) \\ (\tilde{r_1})_{j+\frac{1}{2},k} &= r_{j+\frac{1}{2},k} + \mathcal{O}(h^{2p-2}) \\ (\tilde{r_2})_{j,k+\frac{1}{2}} &= r_{j,k+\frac{1}{2}} + \mathcal{O}(h^{2p-2}).\end{aligned}$$

with $p \geq 2$.

Thus, the dissipation term (4) satisfies:

$$\tilde{d}_{j,k} = d_{j,k} + \mathcal{O}(h^{2p-1})$$

and the truncation error of the semi-discrete scheme (3) is

$$\varepsilon_{j,k} = r_{j,k} + \mathcal{O}(h^{2p}) - d_{j,k} + \mathcal{O}(h^{2p-1}).$$

Since the exact residual r and the leading term d of the residual-based dissipation (5) are null for an exact unsteady solution, we finally obtain:

$$\varepsilon_{j,k} = \mathcal{O}(h^{2p-1}). \tag{6}$$

In other terms, approximating the main residual at order $2p$ and the mid-point residuals at order $2p-2$ leads to a Residual-Based Compact scheme of order $q = 2p-1$. Such a scheme is called RBCq. In the following, we focus on schemes using 5×5-point stencils at most, which corresponds to $p = 2, 3$ and 4. More precisely, RBC3 schemes can be constructed with 3×3 points only and RBC5 and RBC7 schemes with 5×5 points.

2.2 Cauchy-Stable RBC Schemes for Unsteady Flows

The main residual $\tilde{r}_0$ is approximated through a difference operator of the form:

$$(\tilde{r_0})_{j,k} = \left(\overline{D_1}\ \overline{D_2} w_t + \overline{D_2}\ \overline{N_1} \frac{\delta_1 \mu_1 f}{\delta x} + \overline{D_1}\ \overline{N_2} \frac{\delta_2 \mu_2 g}{\delta y} \right)_{j,k} \tag{7}$$

with $\overline{D_m}$ and $\overline{N_m}$ $(m = 1, 2)$ formal polynomials of the second difference operator in the m^{th} direction, of the form:

$$\overline{N_m} = I + \overline{a}\delta_m^2, \qquad \overline{D_m} = I + \overline{b}\delta_m^2 + \overline{c}\delta_m^4, \qquad \overline{a}, \overline{b}, \overline{c} \in \mathbb{R}, \tag{8}$$

where I is the identity operator and

$$(\delta_m^p f)_{j,k} = \underbrace{\delta_m(\delta_m(...(\delta_m f)))}_{p \text{ times}} \tag{9}$$

The degrees of the polynomials are chosen in such a way that the scheme stencil is limited to 5×5 space points at most. Operator (7) is obtained by replacing

space derivatives in each direction by Pade operators:

$$\begin{aligned} f_x =& (\overline{D_1})^{-1}\overline{N_1}\frac{\delta_1\mu_1 f}{\delta x} + \mathcal{O}(\delta x^{2p}) \\ g_y =& (\overline{D_2})^{-1}\overline{N_2}\frac{\delta_2\mu_2 g}{\delta y} + \mathcal{O}(\delta y^{2p}) \end{aligned} \tag{10}$$

and subsequently applying the operator $\overline{D_1}\ \overline{D_2}$ to the whole left-hand side of the equation.

The truncation error of (7) is

$$\varepsilon_{j,k} = \left[I + \mathcal{O}(h^2)\right]\left[r_{j,k} + \mathcal{O}(h^{2p})\right] \tag{11}$$

where the terms involving $r_{j,k}$ vanish for an exact unsteady solution. This represents a substantial difference of RBC schemes with respect to standard Pade approximations and avoids the inversion of linear systems per each space direction if a suitable time-integration technique is selected [1, 2, 6].

An approximation of the main residual of order $2p = 4$ on a 3×3 stencil can be obtained for the following choice of coefficients:

$$\overline{a} = 0, \quad \overline{b} = \frac{1}{6}, \quad \overline{c} = 0. \tag{12}$$

To achieve order $2p = 6$, a 5×5 point stencil is required. Setting to zero truncation error terms up to the 4th order, a one-parameter family of 6th-order approximations is obtained. In the following, we retain the choice made in [5, 7]:

$$\overline{a} = \frac{1}{10}, \quad \overline{b} = \frac{4}{15}, \quad \overline{c} = \frac{1}{90}, \tag{13}$$

which is more suitable for the extension to compressible Navier-Stokes equations.

Finally, the requirement of an order $2p = 8$ on a 5×5-point stencil leads to the unique solution

$$\overline{a} = \frac{5}{42}, \quad \overline{b} = \frac{2}{7}, \quad \overline{c} = \frac{1}{70}. \tag{14}$$

Note that, because we use purely centered operators, no damping effects are introduced at this stage. Thus, the dissipation properties of RBC schemes are actually governed by the right-hand side operator $\tilde{d}$ of Eq.(3).

As anticipated at the beginning of this Section, the dissipation operator $\tilde{d}_{j,k}$ is given by Eq.(4), and involves mid-point residual approximations. Precisely, the following difference operators are used:

$$\begin{aligned} (\tilde{r_1})_{j+\frac{1}{2},k} =& \left[N_1^{\mu}\mu_1\left(D_2 w_t + N_2\frac{\delta_2\mu_2 g}{\delta y}\right) + N_1^{\delta}D_2\frac{\delta_1 f}{\delta x}\right]_{j+\frac{1}{2},k} \\ (\tilde{r_2})_{j,k+\frac{1}{2}} =& \left[N_2^{\mu}\mu_2\left(D_1 w_t + N_1\frac{\delta_1\mu_1 f}{\delta x}\right) + N_2^{\delta}D_1\frac{\delta_2 g}{\delta y}\right]_{j,k+\frac{1}{2}} \end{aligned} \tag{15}$$

based again on the use of formal polynomials of the difference operators:

$$N_m^{\delta} = I + a^{\delta}\delta_m^2\ ,\ \ N_m^{\mu} = I + a^{\mu}\delta_m^2\ ,\ \ N_m = I + a\delta_m^2\ ,\ \ D_m = I + b\delta_m^2 + c\delta_m^4 \tag{16}$$

$$\text{with } m = 1, 2 \text{ , and } a^{\delta}, a^{\mu}, a, b, c \in \mathbb{R}$$

As mentioned previously, the residual-based dissipation $\tilde{d}_{j,k}$ is consistent with a high-order dissipation term. This term has been identified in [8] for a RBCq scheme of order $q = 2p - 1$ ($p \geq 2$) as

$$d_q = (-1)^{p-1}\kappa\{\delta x[\Phi_1(\delta x^{q-1} f_{qx} + \chi \delta y^{q-1} g_{qy})]_x + \delta y[\Phi_2(\delta y^{q-1} g_{qy} + \chi \delta x^{q-1} f_{qx})]_y\}, \tag{17}$$

$$d_q = \mathcal{O}(h^q)$$

where $(.)_{qx} = \dfrac{\partial^q(.)}{\partial x^q}$ and $(.)_{qx} = \dfrac{\partial^q(.)}{\partial x^q}$, and $\kappa > 0$ and χ depends only on the polynomials coefficients in (16).

The expression (17) for the dissipation operator has been used to establish a general condition for a RBCq scheme to be dissipative. This condition, called the χ-criterion, is given in the following theorem.

Theorem 21 [8] *The operator* (17) *is dissipative for any order* $q = 2p - 1$ *(*$p \geq 2$*), any advection direction (A, B) and any functions* Φ_1, Φ_2 *satisfying the conditions*

$$\Phi_1 A \geq 0, \quad \Phi_2 B \geq 0$$

with $\Phi_1 = sgn(A) min\left(1, \frac{1}{\alpha}\right)$, $\Phi_2 = sgn(B) min\,(1, \alpha)$ *and* $\alpha = \delta x B/(\delta y A)$, *if and only if* $\chi = 0$.

Taking into account the accuracy order and the conditions for the stencil to have a minimal extent (3×3 points for RBC3 and 5×5 points for RBC5 and RBC7), the χ-criterion leads to the following conclusions (see [8] for details). Dissipation is ensured in any situation by:

- a unique set of coefficients for RBC3:

$$a^{\mu} = 0, \qquad a^{\delta} = 0, \qquad a = 0, \qquad b = \frac{1}{6}, \qquad c = 0. \tag{18}$$

- a two-parameter family of coefficients for RBC5; this family contains the set of coefficients used in [5, 7]:

$$a^{\mu} = \frac{1}{12}, \qquad a^{\delta} = \frac{1}{6}, \qquad a = \frac{1}{10}, \qquad b = \frac{4}{15}, \qquad c = \frac{1}{90}. \tag{19}$$

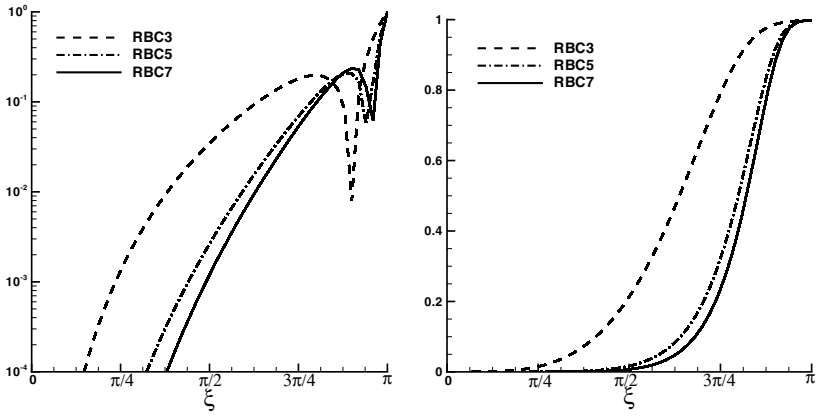
- a unique set of coefficients for RBC7:

$$a^{\mu} = \frac{1}{10}, \qquad a^{\delta} = \frac{11}{60}, \qquad a = \frac{5}{42}, \qquad b = \frac{2}{7}, \qquad c = \frac{1}{70}. \tag{20}$$

Note that coefficients for RBC3 and RBC7 are different from those given in former works [5, 7], where their calculation was based on accuracy and minimal complexity considerations only. In practice, the violation of the dissipation criterion of theorem 21 by RBC3 or RBC7 leads to a weak numerical instability for any choice of the time integration scheme.

2.3 Spectral Properties

The spectral properties of RBC schemes of different orders, i.e. the dispersion error and damping function for a given Fourier mode have been investigated in [9]. Fig. 1a shows the phase errors (in log scale) for RBC schemes of different orders as a function of the reduced wavenumber, for a linear advection problem along a grid direction (1D case). Tab. 1 provides the wave number corresponding to an error of 10^{-3}. RBCq schemes of fifth- and seventh-order accuracy exhibit a cut-off wave number ξ_c very close to $\pi/2$, the smallest resolvable wave number being close to $2\pi/5$ according to Nyquist criterion. Similarly, Fig. 1b provides the damping function $\mathcal{D}_{\xi_0^*}$. As it can be seen in Tabs. 2, RBC5 and RBC7 exhibit a damping function of less than 10^{-3} up to cut-off wave numbers of 1.03 and 1.24, respectively, which raises sharply at higher wave numbers. This shows that the intrinsic numerical dissipation of high-order RBCq schemes acts as a selective filter with a sharp cut-off at high frequency: it efficiently damps out grid-to-grid oscillations that can lead to numerical instabilities without affecting the resolved wave numbers. Also note that, coherently with the truncation error analysis, the resolvability limit of RBC schemes is essentially ruled by the dissipation and not by the dispersion error, their leading-order error term being of dissipative nature. The results summarized in Tab. 1 and 2 prove that RBC schemes of order 5 and 7 can accurately resolve a given wavelength by means of less than 7 or 6 mesh cells respectively, whereas RBC3 requires approximately 9 mesh cells to meet the prescribed accuracy requirements on dispersion errors and 16 mesh points for dissipation errors. For comparison, typical second-order schemes used in industrial codes have a resolvability of 20 to 30 points per wavelength.



(a) $\mathcal{P}_{\xi_0^*}$. (b) $\mathcal{D}_{\xi_0^*}$.

Fig. 1. 1D behavior of $\mathcal{P}_{\xi_0^*}$ and $\mathcal{D}_{\xi_0^*}$ for RBC schemes

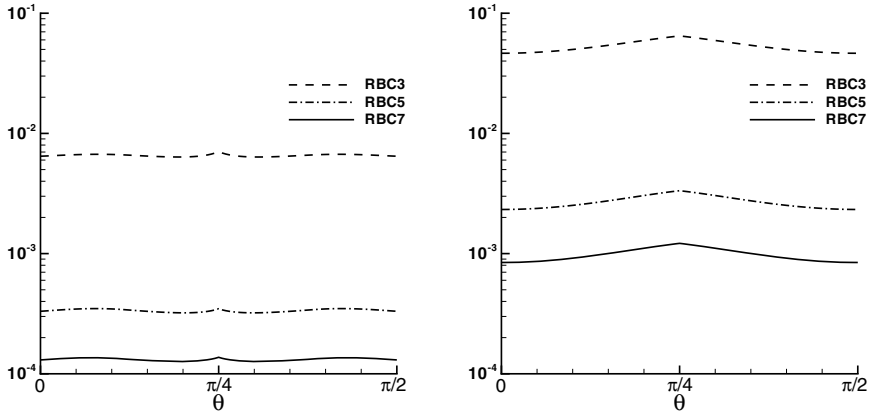
Table 1. Resolvability limit due to dispersion of RBC schemes

	ξ_c	$\lambda_c/\delta x$
RBC3	0.74	8.47
RBC5	1.39	4.53
RBC7	1.54	4.07

Table 2. Resolvability limit due to dissipation of RBC schemes

	ξ_c	$\lambda_c/\delta x$
RBC3	0.40	15.56
RBC5	1.03	6.08
RBC7	1.24	5.06

Then, spectral properties of RBCq schemes were also investigated in the case of multidimensional advection. An overview of the phase and damping error for the range of advection directions, $\theta \in [0, \pi/2]$ is provided in Fig. 2, showing the average value of $\mathcal{P}_{\xi^*_\theta}$ and $\mathcal{D}_{\xi^*_\theta}$ for $\boldsymbol{\xi} \in [-\pi/2, \pi/2]^2$. Thanks to their genuinely multi-dimensional formulation, RBC schemes provide an almost constant error level over the whole range of advection direction, with a maximum at $\theta = \pi/4$ and a minimum at $\theta = 0$ and $\theta = \pi/2$. Furthermore, error levels decrease quickly with the order of accuracy.



(a) $\underset{\boldsymbol{\xi}\in[-\pi/2,\pi/2]^2}{\text{mean}}(\mathcal{P}_{\xi^*_\theta})$ (b) $\underset{\boldsymbol{\xi}\in[-\pi/2,\pi/2]^2}{\text{mean}}(\mathcal{D}_{\xi^*_\theta})$.

Fig. 2. Average of $\mathcal{P}_{\xi^*_\theta}$ and $\mathcal{D}_{\xi^*_\theta}$ over the interval $\boldsymbol{\xi} \in [-\pi/2, \pi/2]^2$ as a function of the advection direction θ

3 Extension to Complex Geometries

RBC schemes, developed in a regular Cartesian grid framework by using finite-difference approximation operators, are extended to general curvilinear grids by means of a cell-centred finite volume formulation, such that the problem unknowns are the point-wise values of the conservative variables at cell centers. A finite volume RBC scheme can be expressed again under the general form (3)

with the dissipation operator (4), provided that the local residuals r are replaced by integrated residuals R over the computational cell (for the main residual) or over a shifted cell (for the mid-point residuals). Details can be found in [10]. For such a straightforward FV extension the nominal order of accuracy is not preserved on non-uniform curvilinear meshes. Ref. [11] shows, in the framework of non compact schemes, that nominal accuracy can still be preserved if the mesh satisfies suitable regularity conditions. The higher the scheme accuracy, the more stringent grid regularity requirements, so that nominal accuracy is generally lost for cases of practical interest. In order to ensure high-accuracy on curvilinear grids while avoiding expensive finite-element like reconstructions of the solution over une cell, two strategies can be adopted. The first-one consists in developing a high-accurate finite volume discretization making use of weighted discretization operators to take into account mesh deformations while still taking benefit of the structured grid formulation: specifically, this strategy has been used to develop a curvilinear grid extension of the RBC scheme of nominal 3rd-order accuracy. The second one consists in maximizing the regions of the computational domain that are discretized with Cartesian grids: this is achieved by developing an overset grid strategy using curvilinear grid blocks close to solid walls, while covering the rest of the domain by means of Cartesian grid blocks. The main features of both strategies are briefly described in the following.

3.1 High-Accurate Finite Volume Formulation

We consider a RBC finite volume scheme of the general form:

$$(\tilde{R}_0)_{j,k,l} = \tilde{d}_{j,k} \tag{21}$$

where $\tilde{R}_0$, called the *main residual*, is a space-centered approximation of R and $\tilde{d}$ is a residual-based dissipation term, introduced to ensure numerical stability, defined in terms of first-order differences of the residual as:

$$\tilde{d}_{j,k} = \frac{1}{2}[\delta_1(\Phi_1 \tilde{R}_1) + \delta_2(\Phi_2 \tilde{R}_2)]_{j,k} \tag{22}$$

where $\tilde{R}_1$ and $\tilde{R}_2$, respectively defined at $(j+\frac{1}{2}, k)$ and $(j, k+\frac{1}{2})$, are also space-centered approximations of R on a suitably chosen shifted control volume, are called the *mid-point residuals*, and Φ_1, Φ_2 are the previously defined dissipation matrices.

In Ref. [10], compact approximations of the main and mid-point residuals were constructed based on weighted approximation operators. With such an approach, it was possible to construct an RBC scheme of third-order accuracy on Cartesian meshes and at of second-order accuracy at least on general deformed meshes. We refer to [10] for details. Such an approach is more accurate than the straightforward finite volume counterpart of RBC schemes, while introducing only a moderate overcost in terms of CPU time and memory requirements. An analysis of the truncation error and spectral properties of this scheme, also discussed in [10] shows that this scheme is slightly more dissipative than its

finite-difference counterpart. Nevertheless, this (small) extra dissipation ensures robusteness on strongly distorted grids and is compensated by the better overall accuracy of the approximation.

3.2 Overset Grid Strategy

We define an overset mesh in D dimension(s) as a mesh composed of M ordered component grids $\{G_m\}_{m\in[1,M]}$ of the spatial domain $\Omega \subset \mathbb{R}^D$. The ordinal of the grid defines its priority in the overset assembly algorithm. An assembly algorithm is used to determine the connectivity (or localization) of points that must exchange information between component grids. Considering a point $P \in G_m$, this step consist in finding whether or not P lies within grid $G_{m'}$ and, if so, to find a base point $Q \in G_{m'}$ corresponding to the centroid of the cell of $G_{m'}$ in which lies point P. A brute force search algorithm being too costly in terms of computational effort, we apply several preconditioning steps, described in [12].

Once grid connectivities have been determined, the next step is to interpolate the solution between overset grids. For this purpose, we use Lagrange interpolation formulae based on polynomials of even order $N = q + 1$, with q the (odd) order of accuracy of the inner-point scheme in use. To achieve such interpolations, we use transfinite interpolation where the mesh is not too distorted and we use the iso-parametric mapping method where the transfinite interpolation fails. The iso-parametric mapping was introduced in Ref. [13], extended to high-order in Ref. [14] and used for aeroacoustics simulations for instance in Ref. [15].

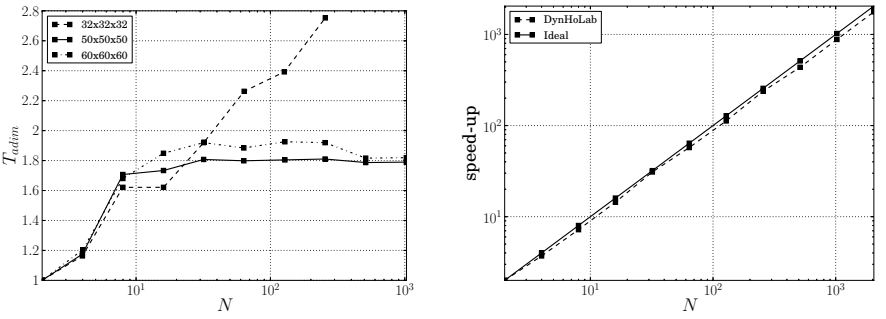
We refer to [12] for further details about the connectivity search algorithm and Lagrange interpolation techniques.

4 Implementation Details

Finite volume extensions of RBC schemes up to fifth-order accuracy have been implemented in the *elsA* code developed by ONERA. This code is used for the turbomachinery applications shown in Section 5.3. We refer to the *elsA* user manual for details about the code [17].

On the other hand, finite volume RBC schemes up to 7th-order accuracy in conjunction with the overset grid strategy have been implemented in a in-house code, named DynHoLab [16]. DynHoLab can handle different governing equations both in two and three dimensions of space. At the present stage of development, the code uses a finite volume methodology on multi-block structured meshes. DynHoLab is coded using a combination of a compiled and type-safe language (FORTRAN) with an interpreted and dynamically typed language (Python [18]). Thanks to this, it benefits from the fast and safe execution of the first one, along with easy and fast development capabilities thanks to the second one. The data-structure chosen for DynHoLab is a CGNS-tree [19] which provides a full hierarchical structure to store the data. The code framework handle parallel computations through an MPI implementation which is based on the intrinsic multi-block architecture of DynHoLab. Thus, it is possible to manage

structured as well as unstructured tessellations of the computational domain. Figure 3 illustrates the weak and strong scalability properties of the code. This is led with the cheapest scheme implemented (among high-order schemes) which correspond to the worst case for a scalability study. The code appears to be scalable up to one thousand cores using blocks of 50^3 cells. On the other hand, the code exhibit an efficiency close to 0.9 up to 2048 cores.



(a) Weak scalability: Time per iteration per cell as a function of the processing units

(b) Strong scalability: Speed-up as a function of the processing units (log-log)

Fig. 3. Scalability of DynHoLab

In both *elsA* and DynHOLab codes, the RBC schemes are used in conjunction with nominally second-order accurate approximations of the viscous terms and of the boundary conditions. The last choice is expected to have an impact on the overall convergence order of the simulations and will be improved in the future.

For steady problems, the solution is advanced through a time marching procedure. The equations are discretized in time by means of the robust first-order Euler method. To reduce computational costs and preserving code modularity, deferred-correction strategy based the Roe-Harten first-order upwind implicit operator is adopted, which is solved by means of L-U factorization. For unsteady problems, the time derivative is approximated through the second-order accurate backward linear multistep scheme (Gear's scheme). The resulting nonlinear system of equations is then solved through Newton-subiterations. The number of subiterations is such that the residual converges by at least 3 orders of magnitude at each time step.

5 Numerical Applications

In the first part of this Section, RBC schemes are applied to selected test cases. The first one is a simple inviscid transonic flow over an NACA0012 airfoil. This test case, proposed in the 1st and 2nd workshop on high-order methods [20, 21], is used for a preliminary validation of the proposed schemes and of the overset

grid strategy. Then, RBC schemes are applied to scale-resolving simulations, to demonstrate the excellent resolvability properties of these schemes. Finally, we demonstrate the robustness and the accuracy of RBC schemes for some challenging transonic turbomachinery configurations.

5.1 Preliminary Validation: Inviscid Flow over an NACA0012 Airfoil

The first test-case is the inviscid two-dimensional flows over a NACA0012 at transonic conditions $M = 0.8$ and $\alpha = 2.25^0$. The simulations are carried out using an overset grid composed by a combination of Cartesian and curvilinear grids (see Fig. 4). Precisely, the grid results from the superposition of four Cartesian grids of 200×90, 140×140, 140×140, and 140×140 cells, an O-shaped body grid of 500×20 cells, and a background polar grid of 200×36 cells. A blanking algorithm is used to remove from the computation cells that are overlapped by grids with a higher priority level, so that in practice the solution is computed by using 75647 degrees of freedom. In the following, the overset grid solution is compared to that obtained on a single-block O-shaped grid made of 601×147 grid points (88347 degrees of freedom)

Figure 5a illustrates the computed isolines of the Mach number using an RBC scheme of 3rd-order accuracy. Shocks are captured in a sharp and almost non-oscillatory way. Present results are found to be in general good agrement with those collected at the first international workshop on High-Order Methods [20], and are superposed to those obtained for a single-block computation. This is confirmed by inspection of the Mach number distribution along the airfoil wall (see Fig. 5b) and of the aerodynamic coefficients, equal to $C_l = 0.3519$ and $C_d = 0.2265 \times 10^{-1}$ respectively. Compared to single block computations, the overlapping strategy allows a more efficient distribution of the degress of freedom, and thus a better overall accuracy.

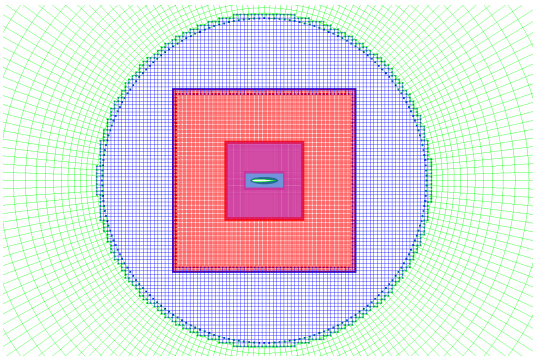
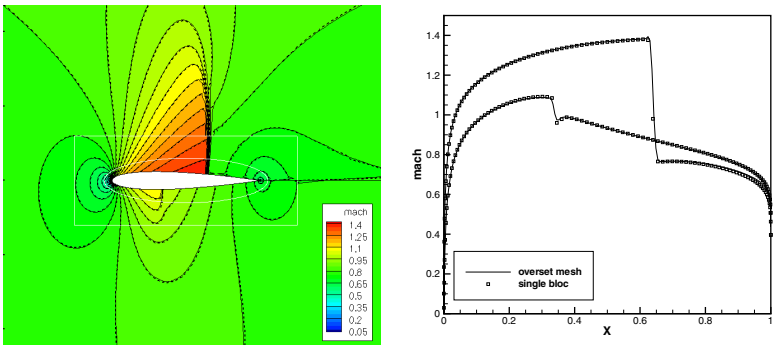


Fig. 4. Detail of the overset grid used to compute transonic flow over a NACA0012



(a) (b)

Fig. 5. Transonic flow over a NACA0012: comparison of an overset grid and a single-block grid solution. a) Mach isolines; b) wall distributions of the Mach number.

5.2 Scale-Resolving Flow Simulations

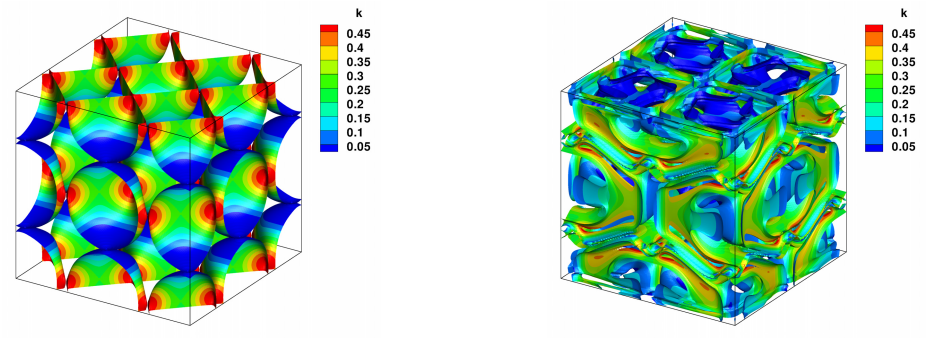
Taylor-Green Vortex. The second application, also taken from the 2nd International Workshop on High-Order CFD Methods [20,21], is used to study the resolvability properties of high-order schemes in view of subsequent application to fine-scale turbulence simulations. A three-dimensional vortex is set as an initial condition for 3D-computation in a periodic box $[0, 2\pi]^3$, then it breaks down, giving origin to smaller and smaller structures. Hereafter we carry out a series of Implicit Large Eddy Simulations (ILES) where the main role of the SGS model, i.e. energy drain at small scales, is taken by the numerical dissipation of RBC schemes.

Fig. 6 shows the evolution of an isosurface of the Q criterion at different times. The vortex is initially deformed through vortex stretching and vortex tilting mechanics, that vortex filaments blow up and the flow transitions to the turbulent regime. Finally, turbulence decays.

The initial conditions of the computation are:

$$\begin{cases} u(x,y,z,0) & = \sin(x)\cos(y)\cos(z) \\ v(x,y,z,0) & = -\cos(x)\sin(y)\cos(z) \\ w(x,y,z,0) & = 0 \\ \rho(x,y,z,0) & = 1 \\ p(x,y,z,0) & = p_0 + \dfrac{\rho}{16}(\cos(2z)+2)(\cos(2x)+\cos(2y)) \end{cases}$$

Where we choose $p_0 = 100$, a Mach number $M_0 = 0.1$, a Reynolds number $Re = 1600$ and a Prandlt number $Pr = 0.71$. For these conditions, previously computed by Brachet [22] using a spectral method, a recent reference DNS obtained on a mesh of 512^3 cells with a pseudo-spectral code over a quite long interation time is available [21].



(a) Iso surface Q=0 at $t = 0$.

(b) Iso surface Q=0 at $t = 4$.

(c) Iso surface Q=3 at $t = 8$.

(d) Iso surface Q=3 at $t = 12$.

Fig. 6. Iso surface of the Q criterion colored by k (computed with RBC5 on the 128^3 mesh). The figure show phases of the vortex break-up.

Solutions are computed on a series of structured meshes of 64^3, 128^3, 256^3 and 384^3 cells, respectively. The time step selected for each grid is such that the CFL number is approximately equal to 5, which is slightly smaller than typical CFL numbers used for industrial LES.

The absence of external forcing implies that the kinetic energy is only decaying during the computation. In the following, we look at the time derivative of the kinetic energy integrated on the whole computational domain, $-dK/dt$, i.e. the kinetic energy dissipation rate, and at the integrated enstrophy Ω. Figure 7a shows mesh convergence for RBC5. For these scheme the solution is already in reasonable agreement with the reference one when using a grid with 64^3 DOF. Note that, even if the energetically relevant scales are well captured by the simulation, the solution is still far from a converged DNS: this is illustrated by Fig. 7b, which shows the time evolution of the integrated enstrophy for the RBC5 scheme. Enstrophy is more difficult to match than the dissipation rate of the kinetic energy since it requires capturing accurately not only the velocity field but also its gradients [29] up to small flow scales. Note that, for high grid resolutions,

mesh convergence of RBC5 is slown down by numerical errors introduced by the time integration scheme. This will be improved in the next future.

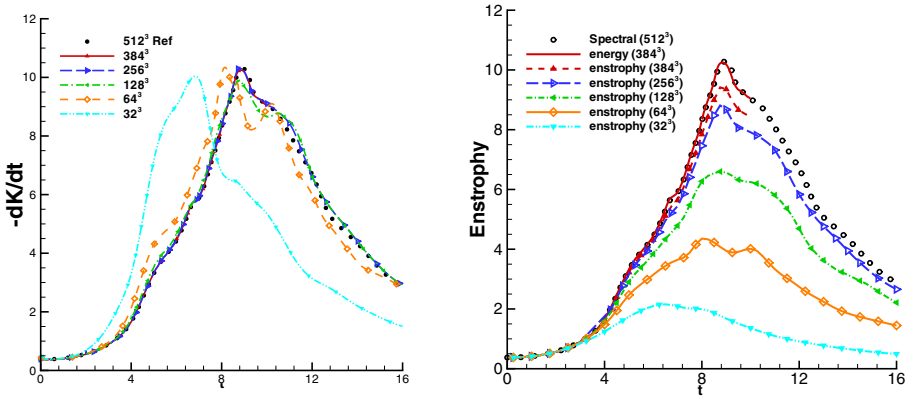


Fig. 7. Mesh convergence of the time evolution of the total energy (left) and enstrophy right). RBC5 scheme.

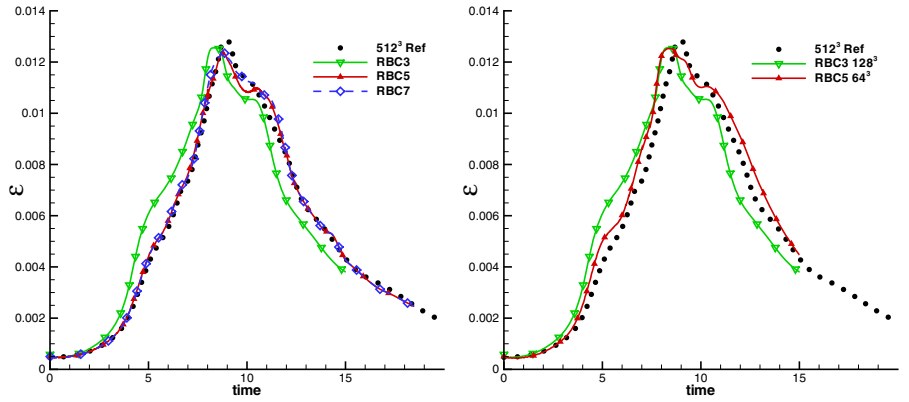


Fig. 8. Time evolution of the total energy for different RBC schemes, 128^3 grid (left). Comparison of the resolvability properties of RBC3 and RBC5 (left).

Figure 8a compares solutions provided by RBC schemes of different orders on the 128^3 grid. The relatively small differences between RBC5 and RBC7 are due to the fact that, for sufficiently high spatial resolution, numerical errors are essentially due to the second-order time discretization scheme in use, which is the same for all computations. On the other hand, RBC5 overperforms RBC3, providing the same or better accuracy on a grid of 64^3 DOF only (Fig. 8b). Even if RBC5 is about 3.25 more expensive than RBC3 (mainly due to the higher level of convergence of the residuals required at each time step), it allows

a reduction by a factor 16 of the DOF in time and space required to achieve a given resolution, leadingin turn to a reduction by roughly a factor 4 of the computational effort required to achieve a given accuracy.

2D Periodic Hill. A very large implicit large eddy simulation (ILES) of the flow over a periodic 2D-Hill was carried out using the RBC scheme of 3rd-order accuracy. The computations were conducted for an average Mach number of 0.1 and an average Reynolds number of 10595. The flow is driven by a forcing term counteracting the drag force exerted on the channel walls. The forcing term is updated at each time step as a function of the instantaneous space-averaged mass flow. The wall temperature is imposed.

The calculations were carried out on a structured grid made of $64 \times 32 \times 32$ cells, corresponding to the coarsest linear grid used for the 2nd workshop on High-Order Methods [21]

The calculations were initialized with a uniform field. The simulation took about 50 convective periods (based on the streamwise length L of the computational domain and the bulk velocity at the throat section U_b, a convective period being given by $T = L/U_b$) to reach a statistically steady state. Then, statistics were collected over 9 additional periods. Calculations were carried out by using a constant time step, taken equal to $10^{-2}T$. Fig. 9 shows the averaged fields of the streamwise velocity $< u >$, as well as mean streamlines. Figs 10 and 11 display the average streamwise velocity and shear Reynolds stress profiles at different streamwise locations. Present results are compared to the reference LES of Breuer et al. [23], obtained on a grid of about 13 million cells by using a second-order centred finite volume scheme, to the experimental data by Rapp [23] and to results obtained by using a second-order accurate central difference scheme with second-order filtering (denoted FDo2-SFo2). Present results show that RBC scheme provides results in reasonably good agreement with the reference data, at least for first-order statistics, despite the extremely coarse grid in use. The RBC solution represents a dramatic improvement over the solution provided by the standard second-order scheme, which does not even capture the correct trends, since numerical errors introduced by the filter tend to laminarize the flow. This is confimed by the fact that Reynolds stresses for this scheme are virtually null, so that they are not represented on Fig. 11.

5.3 RANS Simulations of Complex Turbomachinery Configurations

The applications discussed in this section aim at illustrating the capabilities of RBC schemes for the numerical simulation of industrially relevant configurations in turbomachinery, with focus on transonic turbomachinery with strong shock waves. The numerical simulations were carried out by using RBC schemes implemented in ONERA's code *elsA*.

Rotor37. The NASA Rotor 37 is an isolated transonic axial flow compressor rotor with 36 blades. Extensive experimental data have been collected for this

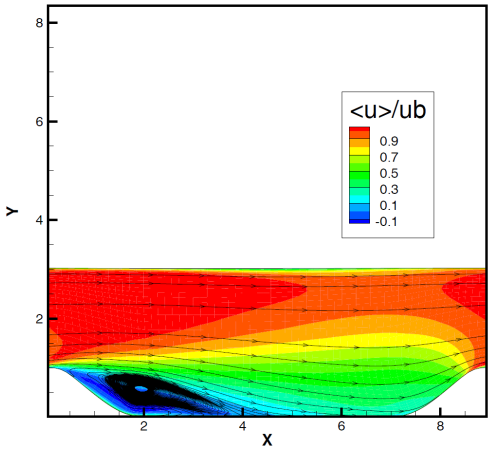


Fig. 9. ILES of the flow over a periodi 2D Hill. Mean steamwise velocity contours and mean streamlines

case with the aim of validating the predictive capabilities of numerical simulation codes and models ([24, 25]). Numerical studies available in the litterature focus mainly on the assessment of turbulence models for the Reynolds-Averaged Navier–Stokes (RANS) equations and Large Eddy Simulations (LES). RANS simulations often fail to predict the global and local flow properties of NASA Rotor 37, due to the complex physical phenomena taking place in this configuration (shocks, shock/boundary layer interactions, separation, shock/tip leakage vortex interactions). If the impact of physical models on numerical predictions of the Rotor 37 flow has been extensively studied, the effect of numerical discretizations has been paid much less attention (an exception is represented by [26]), especially for orders of accuracy greater than two, also because of the difficulty of achieving a robust behaviour of higher-order schemes for this kind of challenging configurations.

In the following, we focus on numerical simulations of the Rotor37 configuration for operating conditions corresponding to the design rotational speed and to a mass flow rate equal to 98% of the choke conditions. We solve the steady Reynolds-Averaged Navier-Stokes (RANS) equations supplemented by the one-equation Spalart-Allmaras turbulence model. Numerical computations were run using a series of three meshes made up by six structured blocks with conformal joins. The finest grid is clustered enough close to the walls so to be relevant for low-Reynolds computations. The coarsest grid uses values of y^+ greater than 10. The main properties of these meshes are given in Tab. 3. The medium and coarse grids are obtained through successive agglomerations of neighbouring cells (eight by eight) of the finest one.

Boundary conditions based on local 1D Riemann invariants are used at inlet and outlet boundaries. At the inlet, total pressure and temperature, as well as flow direction are imposed. At the outlet, the static pressure distribution is im-

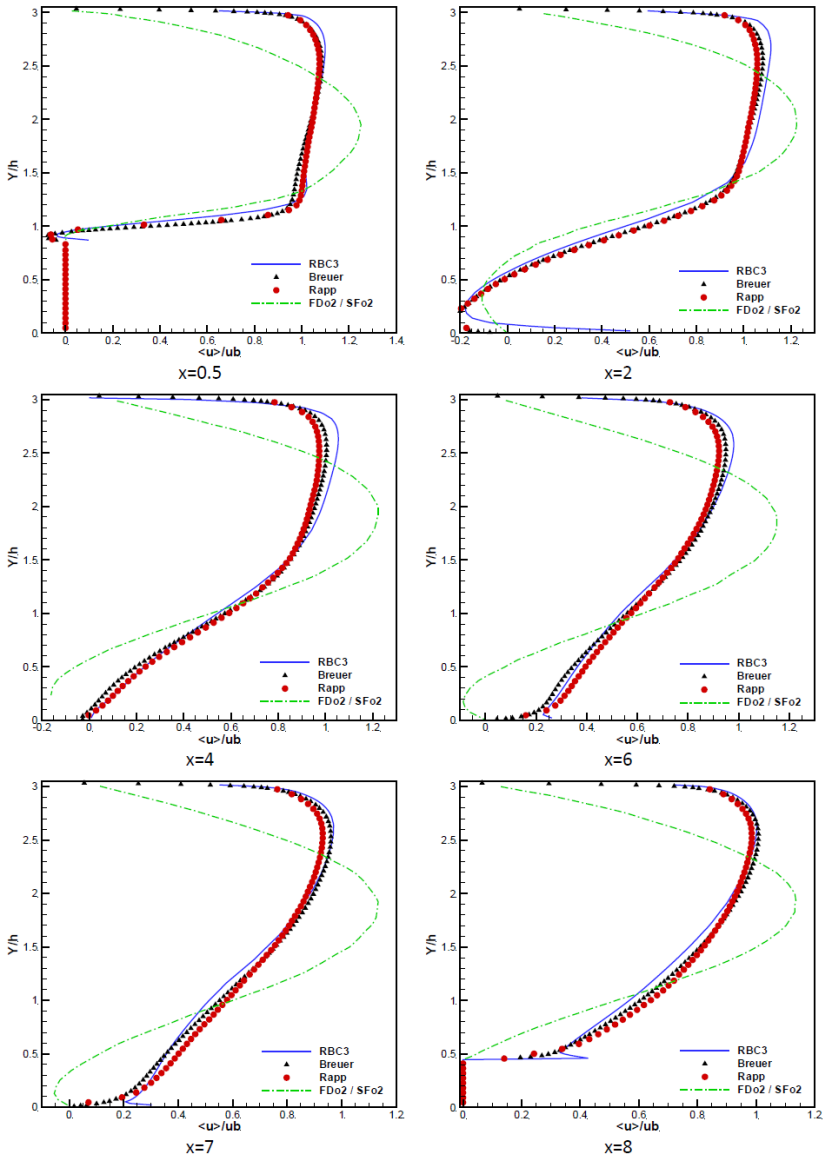


Fig. 10. ILES of the flow over a periodi 2D Hill. Mean steamwise velocity profiles

posed and iteratively adjusted in order to achieve a prescribed target mass flow. The walls are assumed to be adiabatic. Numerical computations were carried out with a CFL number equal to 5 for all cases. Computations were stopped once the relative variation in mass flow rate was lower than 0.5%.

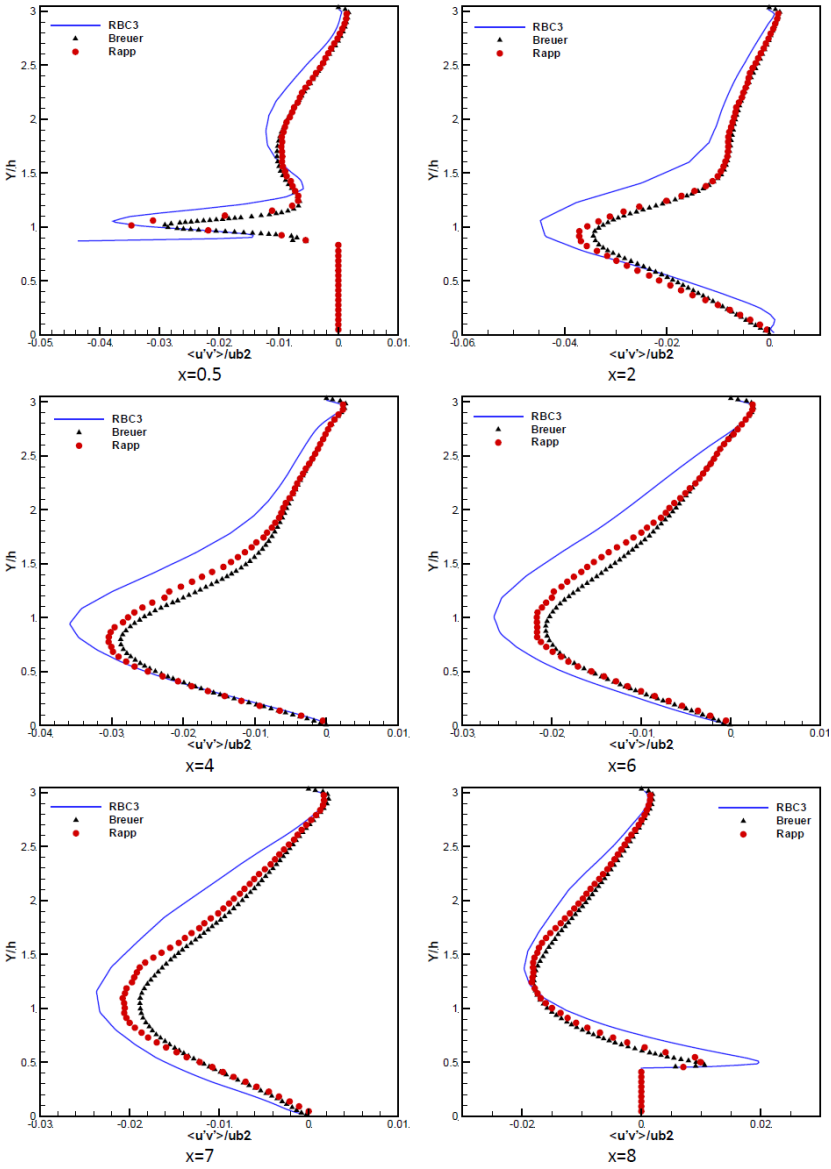


Fig. 11. ILES of the flow over a periodi 2D Hill. Reynolds shear stress profiles

Figure 12 provides an overall view of the computed flow field provided by the RBC3 scheme on the finest grid. The flow field is characterised by a bow shock upstream of the blade, as well as a passage shock leading to boundary layer separation. A complex shock structure is also present in the tip clearance. Shock sharpness and size of the separation bubble vary according to the scheme in use:

Table 3. Computational grids

Grid	Cell count	Cell layers in the tip clearance	y^+
Fine	1 480 704	24	2
Medium	185 088	12	5
Coarse	23 136	6	15

the sharpest shock and the greatest separation bubble are provided by RBC3. results are compared with those provided by the classical Jameson's scheme, the baseline solver in *elsA*. The latter tends to smear shocks and leads to a smaller separation bubble. Note also that the high-accurate scheme predicts a higher maximum value of the Mach number in the computational field: precisely, maximum values of 1.907, 1.844, and 1.828, are obtained for RBC3, RBC2, and Jameson's scheme, respectively. RBC3 and RBC2 give quite similar results, the shock at the suction side being captured slightly more sharply by RBC3.

Figure 13 displays the radial distributions of rotor performance at an axial station located 10.67 cm downstream of the blade leading edge. For this station, experimental measurements of the radial distributions of total pressure ratio, total temperature ratio, and adiabatic efficiency are available. Results provided by RBC schemes of 2nd and 3rd order accuracy are in reasonably good agreement with the experiments already on the medium grid, whereas a finer grid is necessary to obtain similar accuracy with Jameson's scheme. This is better seen in Fig. 14, which compares the radial pressure ratio distributions obtained with the three schemes on the two finer grids.

To complete the discussion, the computational cost of RBC schemes is compared to the cost of Jameson's scheme. The present computations were run on a PC using X5660 Xeon exacore processors with a clock frequency of 2.8Ghz. Computations run with the *elsA* code were parallelized on two cores using MPI. With this configuration, the computational cost per iteration and per point was of about 0.0125 sec using Jameson's scheme. The global computational cost is the CPU time required to achieve an almost converged mass flow. All of the schemes achieve the required tolerance level within almost the same number of iterations (about 3000). RBC schemes are more expensive than Jameson's scheme in terms of operation count: precisely, the computational costs per iteration and per point of RBC2 and RBC3 are about 1.8 and 2.2 times greater, respectively. Nevertheless, RBC schemes are clearly more accurate, since they provide almost grid-converged solutions using about 1/8 of the mesh cells required by the baseline scheme. RBC3 is slightly more expensive than RBC2, mainly because of the computation of weighting coefficients, but the extra cost is compensated by a somewhat better accuracy. It is concluded that, for this steady RANS case, using a well-designed, low dissipative scheme has a stronger effect in terms of improved accuracy of the solution than increasing the formal order of accuracy, so that satisfactory results can be obtained already for a RBC

scheme of 2nd-order accuracy. Conclusions are likely to be different unsteady RANS and LES computations. Further research will focus on the application of high-order RBC schemes to scale-resolving unsteady simulations.

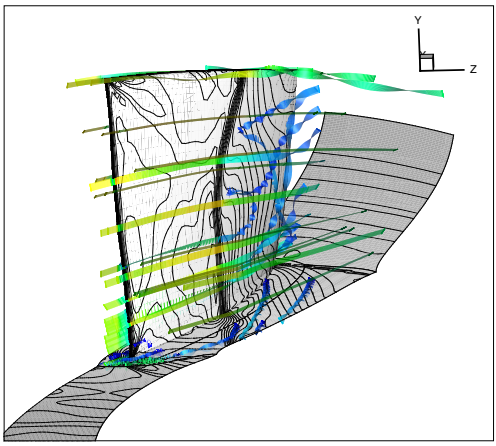


Fig. 12. Overall view of the flow field (98% choke conditions). Mach number distribution at midplane, and stream ribbons colored by the Mach number. RBC3 scheme.

High-Pressure BRITE Turbine Stage. As a final application, the RBC*i* scheme is used to compute the BRITE HP turbine stage experimentally tested in the compression tube facility CT3 of the Von Karman Institute [27] at high vane exit Mach number (pressure ratio 5.11). This case is computed in order to demonstrate the applicability of the proposed methodology to complex unsteady 3D cases, like a realistic turbine configuration including tip clearance.

The computational grid contains approximately 3 millions of cells and is composed by twelve blocks: both the rotor and the stator are discretised by an O-shaped grid around the blades and three H-shaped blocks for inlet, outlet, and inter-blade regions; the tip clearance is also discretized with an O-shaped and three H-shaped grids. Unsteady computations are initialised with steady results obtained by imposing a mixing-plane inter-stage condition. The use of chorochronic periodic boundary conditions [28] allows simulating just one blade per row. The flow is modelled through the RANS equations completed by the Spalart-Allmaras transport-equation model for turbulence. The governing equations are advanced in time via a dual-time stepping technique: the physical time step is approximately 1/100 of the rotation period and about 30 sub-iterations per time step were used to converge the solution in pseudo-time.

RBCi allows sharp capturing of shock waves and von Karman vortices in the blade wakes (see 15, Fig. 17. Fig. 15 shows a snapshot of the 3D vortex structures within the geometry with an isosurface of the Q-criterion colored by the entropy showing the 3D nature of the flow field.

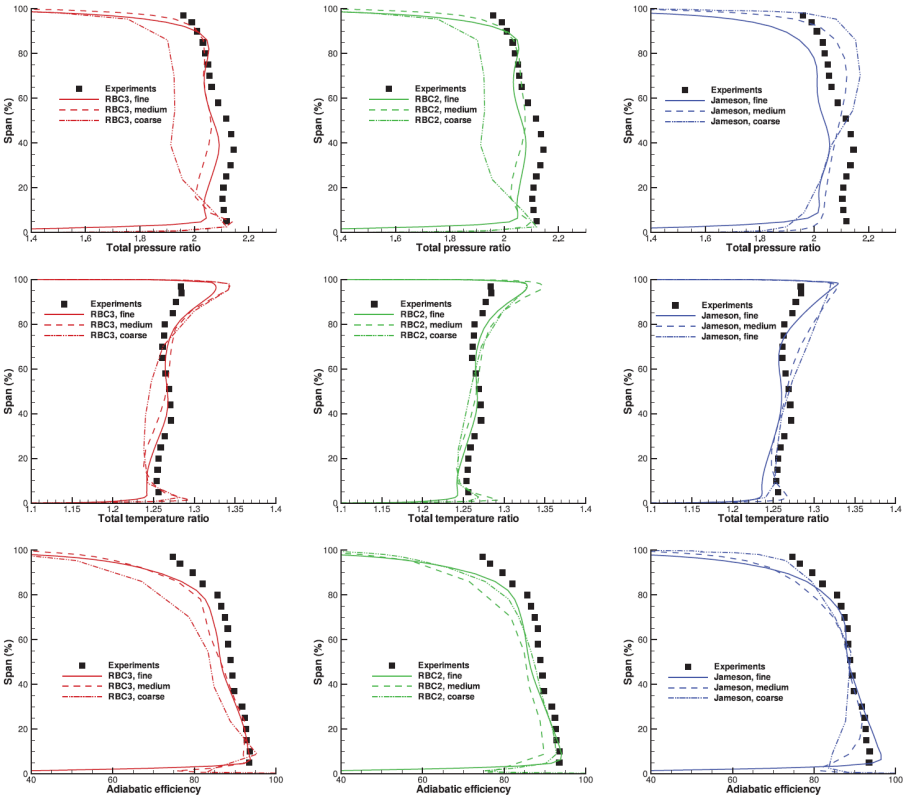


Fig. 13. Radial performance distributions for several schemes and grid resolutions. Mesh convergence study.

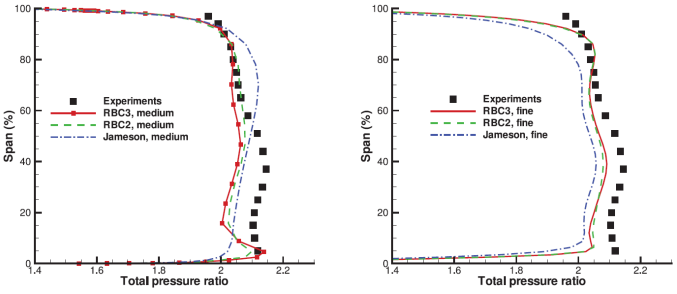


Fig. 14. Radial pressure ratio distributions for several schemes and grid resolutions. Comparisons among different schemes on the medium (left) and finer (right) grids.

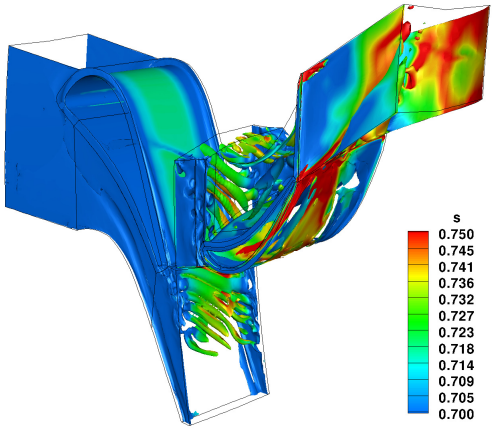
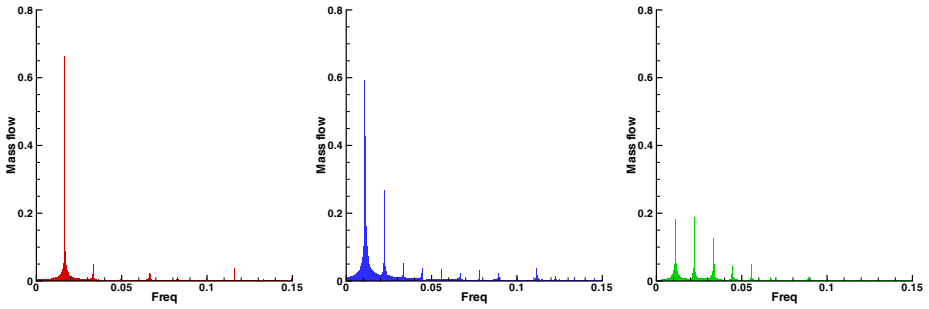


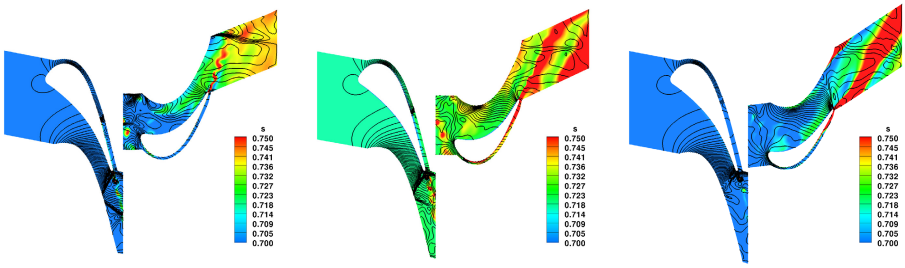
Fig. 15. VKI BRITE HP turbine: snapshot of Q criterion isosurface, Q=0.001, colored by the entropy



(a) Stator outlet. (b) Rotor inlet. (c) Rotor outlet.

Fig. 16. VKI BRITE HP turbine:: mass flow frenquency spectrum

The complex unsteadiness of the flow with rotor/stator interaction is also captured. Fig. 16 provides the Fourier spectra of of the normalized mass flows at stator outlet, rotor inlet and rotor outlet. The main frenquency at the stator outlet corresponds to the adimensional passage frequency of rotor blades with regard to the stator, $f_{rotor} = 0.016$, with its first harmonic confirming that the rotor/stator interaction is taken into account. Similarly the main frequency observed on rotor mass flows at inlet and outlet is the adimensioned frequency of passage of stator blades with regard to the rotor, $f_{stator} = 0.011$, with its first harmonics. The high frequency peak of low magnitude at $f_{vort} = 0.116$ in the stator spectrum is likely to correspond to the vortex shedding at the trailing edge of the stator blades since it does not match the frequency of an harmonic and its magnitude is too high. This frequency is seen by the rotor at a slightly different frequency probably because of the chorochronic periodicity conditions applied between the rotor and the stator, which filter the signal.



(a) Slice near the hub. (b) Slice at mid span of the blades. (c) Slice near the rotor housing.

Fig. 17. VKI BRITE HP turbine: slices of the solution at different spanwise locations; isobar lines and entropy contours

Fig. 17 provides a snapshot of three slices of the configuration located near the hub, at mid span of the blades and near the rotor housing. The slices represent entropy contours along with isobars. Strong shocks are created at the trailing edge of the stator and rotor blades near the hub. Morover the differences between these three slices confirm that this case is highly three-dimensional with a major impact of the tip clearance on the wake of the rotor blades.

6 Conclusions

A family of high-order Residual-Based compact schemes has been extended to unsteady flows and complex geometries. Unsteady RBC schemes of highest order (5 and 7) exhibit excellent resolvability properties, since their genuinely multi-dimensional numerical dissipation acts as a sharp cutoff filter close to the grid resolvability limit. In practice, very encouraging results were obtained for a very coarse implicit large eddy simulation of the separated flow over a periodic 2D hill by using an RBC scheme of 3rd-order accuracy. RBC schemes, developed in a finite difference framework, were extended to geometrically complex configurations by means of a robust and accurate finite volume formulation and an overset grid strategy. Applications to severe transonic turbomachinery flows prove the robustness of the proposed approach and show that significant gains, both in terms of degrees of freedom and workunits required to achieve a given numerical error level, can be obtained with respect to traditional finite volume schemes in use in industry.

References

1. Lerat, A., Corre, C.: A residual-based compact scheme for the compressible Navier-Stokes equations. Journal of Computational Physics 170, 642–675 (2001)
2. Lerat, A., Corre, C.: Residual-based compact schemes for multidimensional hyperbolic systems of conservation laws. Computers & Fluids 31, 639–661 (2002)

3. Deconinck, H., Ricchiuto, M.: Residual Distribution schemes: foundation and analysis. In: Encyclopedia of Computational Mechanics. John Wiley and Sons, Ltd. (2007)
4. Abgrall, R.: Residual distribution schemes: Current status and future trends. Computers & Fluids 35, 641–669 (2006)
5. Lerat, A., Corre, C.: Higher order residual-based compact schemes on structured grids. In: 34th Comput. Fluid Dyn. Course, von Karman Institute for Fluid Dynamics, VKI LS 2006-1, pp. 1–111 (2006)
6. Corre, C., Hanss, G., Lerat, A.: A residual-based compact scheme for the unsteady compressible Navier-Stokes equations. Computers & Fluids 34, 561–580 (2005)
7. Corre, C., Falissard, F., Lerat, A.: High-order residual-based compact schemes for compressible inviscid flows. Computers & Fluids 36, 1567–1582 (2007)
8. Lerat, A., Grimich, K., Cinnella, P.: On the design of high order residual-based dissipation for unsteady compressible flows. Journal of Computational Physics 235, 32–51 (2013)
9. Grimich, K., Cinnella, P., Lerat, A.: Spectral properties of high-order residual-based compact schemes for unsteady compressible flows. Journal of Computational Physics 252, 142–162 (2013)
10. Grimich, K., Michel, B., Cinnella, P., Lerat, A.: An accurate finite-volume formulation of a Residual-Based Compact scheme for unsteady compressible flows. Computers & Fluids 92, 93–112 (2014)
11. Rezgui, A., Cinnella, P., Lerat, A.: Third-order finite volume schemes for Euler computations on curvilinear meshes. Computers & Fluids 30, 875–901 (2001)
12. Outtier, P.Y., Content, C., Cinnella, P.: High-order residual-based compact schemes for compressible flows on overset grids. In: 32nd AIAA Applied Aerodynamics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014, Atlanta (to appear, June 2014)
13. Benek, J.A., Steger, J.L., Dougherty, F.C., Buning, P.G.: Chimera: A Grid-Embedding Technique. Report AEDC-TR-85-64, Arnold Engineering Development Center (April 1986)
14. Sherer, S.E., Scott, J.N.: High-order compact finite-difference methods on general overset grids. Journal of Computational Physics 210(2), 459–496 (2005)
15. Chicheportiche, J., Gloerfelt, X.: Study of interpolation methods for high-accuracy computations on overlapping grids. Computers & Fluids 68, 112–133 (2012)
16. Outtier, P.-Y., Content, C., Cinnella, P., Michel, B.: The high-order dynamic computational laboratory for CFD research and applications. In: 21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics (June 2013)
17. elsA, `http://elsa.onera.fr/`
18. Python web site, `http://www.python.org/`
19. CGNS website, `http://cgns.sourceforge.net/`
20. Website of the 1st workshop on High-Order CFD methods, `http://zjwang.com/hiocfd.html`
21. Website of the 2nd workshop on High-Order CFD methods, `http://www.dlr.de/as/desktopdefault.aspx/tabid-8170/13999_read-35550/`
22. Brachet, M., Meiron, D.I., Orszag, S.A., Nickel, B.G., Morf, R.H., Frisch, U.: Small-scale structure of the Taylor-Green vortex. Journal of Fluid Mechanics 130, 411–452 (1983)
23. Breuer, M., Peller, N., Rapp, C., Manhart, M.: Flow over periodic hills - numerical and experimental study over a wide range of Reynolds numbers. Computers and Fluids 38, 433–457 (2009)

24. Suder, K.L.: Experimental Investigation of the Flow Field in a Transonic, an Axial Flow Compressor With Respect to the Development of Blockage and Loss. NASA TM 107310 (1996)
25. Dunham, J. (ed.): CFDValidation for Propulsion System Components. AGARD Advisory Report AR-355 (1998)
26. Chima, R.K., Liou, M.-S.: Comparison of the AUSM+ and H-CUSP Schemes for Turbomachinery Applications. AIAA Paper 2003-4120 and NASA TM-2003-212457 (2003)
27. Denos, R., Arts, T., Paniagua, G., Michelassi, V., Martelli, F.: Investigation of the unsteady rotor aerodymics in a transonic turbine stage. Journal of Turbomachinery 123 (2001)
28. Gerolymos, G.A., Michon, G., Neubauer, J.: Analysis and application of chorochronic periodicity in turbomachinery rotor/stator interaction computations. Journal of Propulsion Power 18, 1139–1152 (2002)
29. Donzis, D.A., Yeung, P.K., Sreenivasan, K.R.: Dissipation and enstrophy in isotropic turbulence: Resolution effects and scalling in direct numerical simulations. Physics of Fluids 20, 045108 (2008)

A High-Order Discontinuous Galerkin Chimera Method for the Euler and Navier-Stokes Equations

M. Wurst, M. Kessler, and E. Krämer

Unversität Stuttgart, Institute for Aerodynamics and Gas dynamics, D-70569, Stuttgart, Germany
michael.wurst@iag.uni-stuttgart.de

Abstract. In this article a non-conservative Chimera method for the Euler and Navier-Stokes equations is introduced. The CFD solver for the Chimera method is based on a high-order Discontinuous Galerkin formulation and employs modal basis functions. As the method features at least two different grids, interpolation operators have to be defined between the two grids which is achieved by a discrete projection. A detailed description of the adaption of the temporal integration schemes is given and their implementation is validated for the explicit and implicit schemes against results using only a single grid.

Keywords: Discontinuous Galerkin, Chimera method.

1 Introduction

The Chimera method was first introduced for the Euler equations by Benek et al. [1]. Since then the method has been applied to many problems in fluid mechanics. It has basically two advantages compared to a single grid approach. The separate creation of a body-fitted grid and the background grid allows a meshing process, which would be more complicated with a single mesh or even impossible for some geometries, e.g. when using a structured solver. Additionally the Chimera method allows the modeling of structures with motion or relative motion between two structures as e.g. the varying pitch angles during the rotation of the blades of a helicopter. The Discontinuous Galerkin (DG) method was first used by Reed & Hill [2] for neutron transport and it was first applied for conservation laws by Cockburn and Shu [3]. In recent years the method was extended by Bassi et al. for the Navier-Stokes equations [4] and RANS equations [5]. A first implementation of the Chimera method for the DG method was done by Galbraith et al. for inviscid [6] and viscous flow [7].

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_19

2 Governing Equations

The governing equations in this study are the compressible Navier-Stokes equations which can be written in the following compact form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathcal{F}_i(\mathbf{u}) - \mathcal{F}_v(\mathbf{u}, \nabla \mathbf{u})) = 0. \tag{1}$$

where U is the vector of the conservative variables $(\rho, \rho u, \rho v, \rho w, \rho E)$. The inviscid fluxes $\mathcal{F}_i$ are given then by

$$\mathcal{F}_i^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)\, u \end{pmatrix}, \quad \mathcal{F}_i^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)\, v \end{pmatrix}, \quad \mathcal{F}_i^z = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)\, w \end{pmatrix}. \tag{2}$$

and the viscous fluxes $\mathcal{F}_v$ are given by

$$\mathcal{F}_v^x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xi} u_i + q_x \end{pmatrix}, \quad \mathcal{F}_v^y = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yi} u_i + q_y \end{pmatrix}, \quad \mathcal{F}_v^z = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zi} u_i + q_z \end{pmatrix}. \tag{3}$$

with τ as the viscous stress tensor and q as the heat flux vector.

3 Discontinuous Galerkin Discretisation

3.1 Spatial Discretisation

The Navier-Stokes equations (1) are discretised with a standard Discontinuous Galerkin method. The weak form of the Navier-Stokes equation can be obtained by multiplying the equation with an arbitrary test function $\mathbf{v}$, integrating over the domain Ω and integrating by parts of the divergence term:

$$\int_\Omega \mathbf{v}_h \cdot \frac{\partial \mathbf{u}_h}{\partial t}\, d\Omega + \oint_{\partial\Omega} \mathbf{v}_h \left(\mathcal{F}_i(\mathbf{u}_h) - \mathcal{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h)\right) \cdot \mathbf{n}\, d\sigma - \int_\Omega \nabla \mathbf{v}_h \cdot \left(\mathcal{F}_i(\mathbf{u}_h) - \mathcal{F}_v(\mathbf{u}_h, \nabla \mathbf{u}_h)\right) d\Omega = 0. \tag{4}$$

As the solution is double valued at faces, the Riemann problem in the face integral is solved with a HLL Riemann solver [8] or a Roe Riemann solver [9] for the inviscid fluxes. For the gradients in the viscous flux terms the BR2 scheme is applied [10].

3.2 Temporal Discretisation

The temporal discretisation can be done explicitly or implicitly. In this paper we choose standard explicit Runge-Kutta schemes up to 4th-order for the unsteady cases. For steady solutions we use an implicit Euler Backward scheme. The DG discretisation in equation (4) can be written in a compact form

$$\mathbf{M}\frac{d\mathbf{U}}{dt} = \mathbf{R}(\mathbf{U}), \tag{5}$$

with $\mathbf{U}$ as the global solution vector and $\mathbf{M}$ as the global mass matrix. Discretising this with an Euler backward scheme leads to:

$$\mathbf{M}\frac{\mathbf{U}^{n+1} - \mathbf{U}^{n+1}}{\Delta t} = \mathbf{R}(\mathbf{U}^{n+1}). \tag{6}$$

For the solution of the nonlinear system we linearise the system with a Newton method with only one Newton iteration per time-step:

$$\left(\frac{\mathbf{M}}{\Delta t} - \frac{\partial \mathbf{R}(\mathbf{U}^n)}{\partial \mathbf{U}}\right)\Delta\mathbf{U} = \mathbf{A}\Delta\mathbf{U} = \mathbf{R}(\mathbf{U}^n). \tag{7}$$

The linearisation of the residual $\mathbf{R}$ is done analytically and the resulting linear system is solved with a GMRES solver preconditioned with an ILU(0) method. For steady computations the time step is adapted in each step according to

$$t^{n+1} = t^0\frac{\mathbf{R}(\mathbf{U}^{n+1})}{\mathbf{R}(\mathbf{U}^0)}. \tag{8}$$

4 Chimera Method for the DG Discretisation

4.1 Basic Scheme

In the following the basic Chimera scheme is explained on the example of a two-dimensional cylinder in figure 1. In general, there are three different kind of cells in the background grid and the Chimera grid. They are distinguished with an additional vector I_{Blank} which has the size of the number of cells. First of all, there are these cells which do not lie in the overlapping region of the two grids. They are called active cells and the vector I_{Blank} will have a value of one for these cells (green cells in figure 1). The second kind of cells are in the background mesh in the area where the body lies (blue cells). These cells are identified within the process of hole cutting. The process can be very critical for a successful Chimera method. For simulations with a movement of the body the hole cutting has to be done in every time step and should be obviously implemented efficiently. For the stationary, static case which is treated here, we choose simple geometric forms as circles and rectangles for the form of the hole. The cells inside the hole are inactive cells and the vector I_{Blank} equals zero. The third kind of cells are the interpolation cells (red cells). They are important for

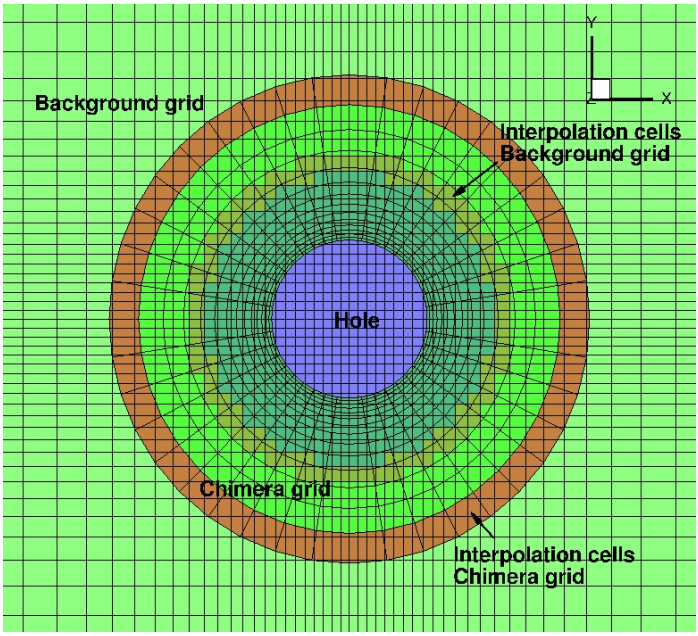


Fig. 1. Example of a Chimera grid for the Gauss pulse simulations

the coupling of both grids and can be found in the background as well as the Chimera grid. In the background grid they are the cells which are positioned next to the hole cells, and in the Chimera grid they are the cells at the outer boundary of the cells. It is important that both lines of interpolation cells form a closed loop without gaps, which e.g. for the background grid would mean that it is possible that fluid flows into the hole or vice versa. As the BR2 scheme needs one neighbour cell for the calculation of gradients, one row of interpolation cells is enough, leading also to a compact Chimera stencil compared to high-order finite difference or finite volume schemes. As interpolation cells are excluded from the actual temporal integration they will have an I_{Blank} value of zero. The cells between the interpolations cells in the background and the Chimera grid are active cells. In the small overlap region of active cells the flow solution will be double valued. However, this area is commonly well enough resolved in simulations so the differences should be small.

4.2 Definition of Interpolation Operators

For the coupling of the two grids we use a discrete L_2 projection to determine the modal coefficients of the basis functions. The L_2 has the advantage that by definition the error at the approximation becomes minimal in the L_2 sense.

The modal coefficients $\mathbf{a}$ can be obtained with the following definition of the L_2 projection

$$\mathbf{M}\,\mathbf{a} = \mathbf{b} \tag{9}$$

with $\mathbf{M}$ as the mass matrix of the interpolated cell. The rows of the vector $\mathbf{b}$ are defined by

$$b_i = \int_\Omega \phi_i \, u \, d\Omega = \sum_{k=1}^{n_{GP}} \phi_i(x_k) u(x_k) \omega_k. \tag{10}$$

That means, that the solution u has to be evaluated at all Gauss points x_k

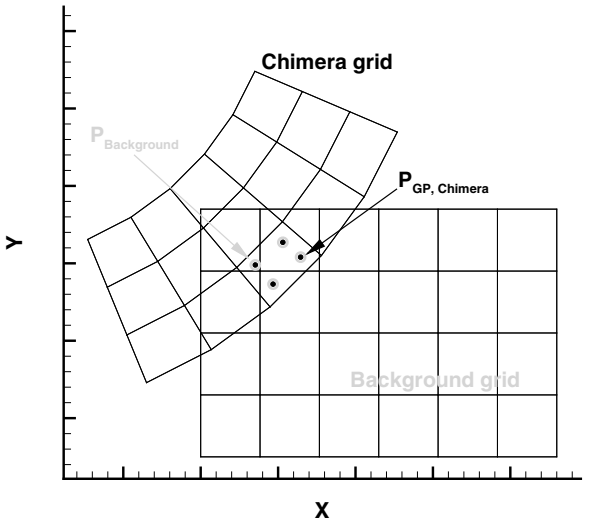


Fig. 2. Example of a Chimera grid for the Gauss pulse simulations

for the basis function ϕ_i. However, in the present Chimera method the solution u is not evaluated at the Gauss points of the interpolated cell: the solution is obtained from identical points from the corresponding grid. This can be seen in figure 2 for a generic case: For the Gauss points (black points) in the Chimera grid exist points in the background grid (grey points). In general, these points do not lie in one cell in the background grid but are spread over several cells. Additionally, it is not sufficient to find the correct partner cell, we also have to find the local coordinates (ξ, η) in the unit element to evaluate the solution in the background grid as our basis functions are defined in a unit element. The transformation is usually given as $\mathbf{x} = \mathbf{x}(\xi, \eta)$. However, we need the inverse transformation. As this is sometimes difficult to get analytically (e.g. for curved elements), we use a Newton method to find the unit coordinates after we have

identified the right cell for the interpolation. As long as the Jacobian of the transformation stays positive, which is neccessary anyway, the Newton iteration will converge, usually in very few steps.

4.3 Modification to the Time Integration

For explicit time integration the modification is quite straightforward: in active cells the update of the solution should be prevented. Basically for each time step or Runge-Kutta step the solution update in each cell has to be multiplied with the value of I_{Blank} in the cell:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + I_{Blank} \cdot \Delta\mathbf{U}. \tag{11}$$

For the implicit time integration (see equation (7)) the right hand side is also multiplied with I_{Blank}:

$$\mathbf{A}_{mod}\Delta\mathbf{U} = I_{Blank}\mathbf{R}(\mathbf{U}^n). \tag{12}$$

Additionally the matrix $\mathbf{A}$ has to be modified for each cell with $I_{Blank} = 0$: the modified matrix $\mathbf{A}_{mod}$ is the unit matrix in these cells to get no update to the solution inside the cell and no interference with regular active cells. The projection is done for both the Chimera and the background grid with the solution vector $\mathbf{U}^n$. However, also for those cells a solution at t_{n+1} has to be determined which means that additional iterations over the Newton algorithm have to be performed until the solution at the interpolation cells does not change anymore.

5 Results

5.1 Gauss Pulse in Density

To validate the presented Chimera method and to show that the high-order accuracy is still achieved, a convected Gauss pulse in density is calculated on a rectangular domain with the Euler equations. The size of the domain is 1x0.5 with the pulse given by:

$$\begin{aligned} \rho &= \rho_0 + a\ exp\left[2\frac{(\mathbf{x}-\mathbf{x}_0)^2}{\sigma^2}\right], \\ \rho u &= \rho u_0, \\ \rho v &= 0, \\ \rho E &= \frac{p_0}{\kappa - 1} + \frac{(\rho u_0)^2}{2\rho} \end{aligned} \tag{13}$$

where $\rho_0 = 1$, $u_0 = 0.25$, $p_0 = 1$, $\mathbf{x}_0 = (0.25, 0.25$, $a = 1$ and $\sigma = 0.05$. The final position of the vortex is $\mathbf{x}_f = (0.75, 0.25)$. The simulation is carried out on a sequence of grids for different orders for the Roe Riemann solver and the HLL Riemann solver and the error is measured with the L_2 norm. In all simulations

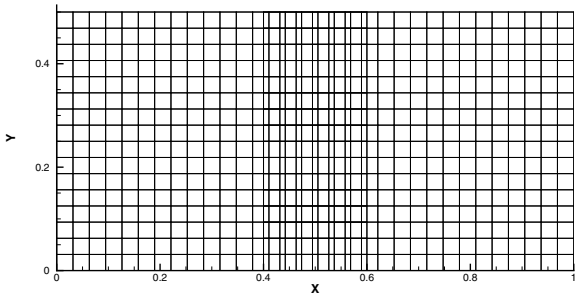
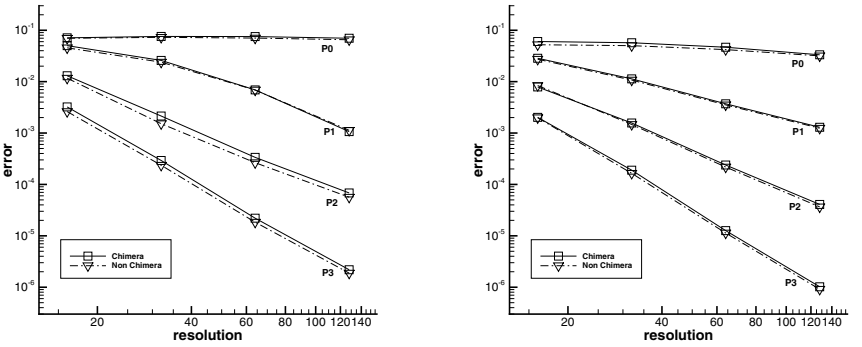


Fig. 3. Example of a Chimera grid for the Gauss pulse simulations

the temporal discretisation is carried out with an explicit Runge-Kutta method whose order equals the spatial order. Both single grids as well as Chimera grids are used. An example of the employed Chimera grids is found in figure 3. It can be seen that the Chimera grid has an overlap of two grids from $x = 0.4$ to $x = 0.6$. As the pulse is convected from the left side to the right side of the domain two Chimera interpolations will take place. The resulting errors are plotted in figure 4a for the HLL flux and in figure 4b for the Roe flux. Using the Chimera grid leads to almost the same errors as the single grid approach. The small visible gap between the two lines is partly a consequence of the evaluation of the error. It is not evaluated in the inactive interpolation cells, of course, but in the overlap region between the two grids it is evaluated twice, leading to a higher



(a) HLL Flux (b) Roe Flux

Fig. 4. Error of the Gauss pulse for a) the HLL numerical flux and b) the Roe numerical flux for different order P0,..,P3 using a single grid or a Chimera approach

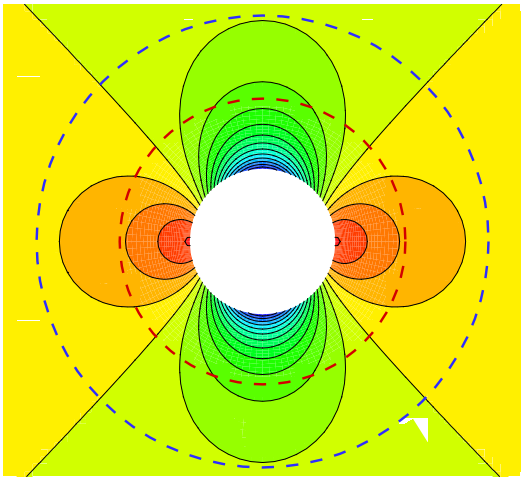
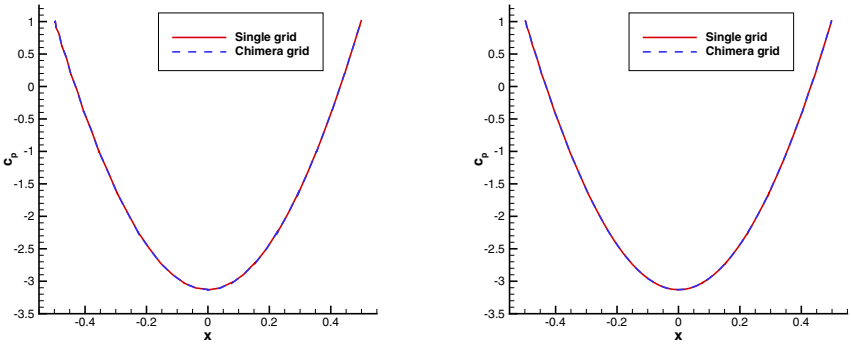


Fig. 5. Density distribution for the Euler simulation for P4, position of chimera interpolation cells marked in blue, hole interpolation cells in red

error. However, the difference should be small compared to the overall error as the pulse should be entirely in the right grid at the end of the simulations. While evaluating the error it was noticed that the error in the left grid is about 10% of the error in the right grid, almost independently from the polynomial order or grid resolution.

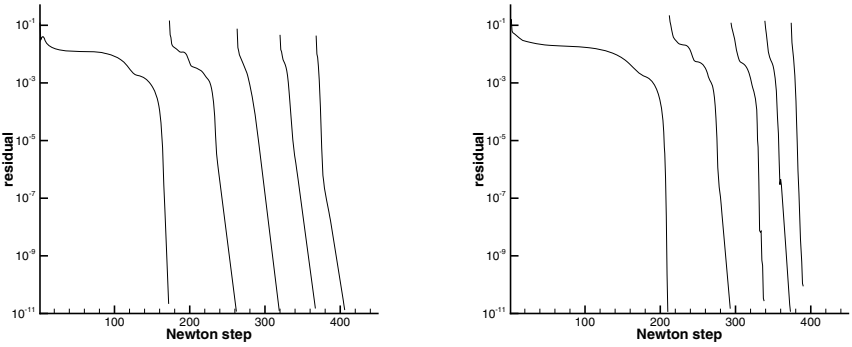
5.2 Two-Dimensional Cylinder

Euler Simulation. The inviscid simulation is carried out at Ma=0.2 assuring that the flow is subsonic in the whole computational domain with a size of 20x20 cylinder diameters d. Both a single grid and a Chimera grid simulation were run. The mesh for the Chimera simulation consisted of 560 cells in the Chimera grid and 3600 cells in the background grid, while the single grid consisted of 1520 cells. The more cells in the Chimera simulation were necessary to achieve a similar same grid resolution in the interpolation region. However, the near wall region has the same resolution, as the Chimera grid was used as a starting point in the meshing of the single grid version. The elements close to the wall of the cylinder were curved with a third-order polynomial obtained from the face normals [11]. In figure 5 the density distribution is plotted. The contour lines have no jumps between the two grids, leading to a smooth solution. Additionally, the position of both interpolation cells is marked showing that they are positioned in close to the wall of the cylinder. A comparison between the single grid and the Chimera grid is seen in the distribution of the pressure coeffiecient in figure 6a for P2 and in figure 6b for P4. The differences here are negligible.



(a) P2 (b) P4

Fig. 6. Pressure coefficient on the surface of the cylinder for a) P2 and b) P4



(a) Chimera (b) Single grid

Fig. 7. Convergence for P0,..,P4 for a) Chimera grid simulation and b) single grid simulation

Laminar Simulation. The laminar simulation of the two-dimensional cylinder was carried out with a Reynolds number Re=40 (based on the cylinder diameter d) at Ma=0.1. The flow at this Reynolds number is steady and does not develop a von Karman vortex street. The computational domain has a size of 200x200 d. The grid for the Chimera simulation consists of 800 cells in the Chimera grid and 3600 cells in the background grid, while the single grid consists of 1480. Both simulation were carried out with an order sequencing, that means that we started with a P0 computation and used each low-order solution as a starting point for the next higher order. The convergence history of the simulations can be seen in figure 7a for the Chimera simulation and in figure 7b for the single grid. It shows that using the Chimera method in this case does not lead to a higher-amount of Newton steps. In figure 8 the separation bubble behind the cylinder is visualized with streamlines. The separation bubble is symmetric as

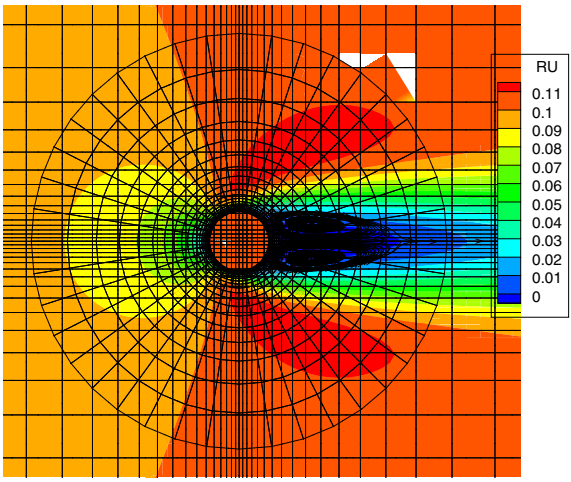


Fig. 8. Separation bubble behind the cylinder

expected. Again the contour plot has no jumps between the two grids, leading to a smooth solution. A quantitative comparison can be seen in table 1 and 2 where the drag coefficient and the separation bubble length for the two grids and at different orders is compared to the numerical results of Sen et al. [12]. By increasing the order of the simulation both values approach each other for the different grids and converge to the numerical reference values.

Table 1. Results for the drag coefficient c_d for different orders for the single grid and the Chimera grid compared to the reference value of Sen et al. [12]

	P0	P1	P2	P3	P4
single grid	10.8605	1.58886	1.51231	1.50946	1.50932
Chimera grid	10.2909	1.56467	1.50851	1.50955	1.50950
Sen et al.					1.5093

Table 2. Results for the separation bubble length for different orders for the single grid and the Chimera grid compared to the reference value of Sen et al. [12]

	P0	P1	P2	P3	P4
single grid	-	1.76937	2.30983	2.25654	2.25443
Chimera grid	-	1.78099	2.27476	2.25497	2.25421
Sen et al.					2.247

6 Conclusions

The feasibility of the Chimera method for inviscid and laminar two-dimensional flows has been shown. The paper explains the process of the Chimera technique and explains the necessary adjustments to the temporal integration schemes. The accuracy of the method was validated with the simulation of a Gauss pulse in density asserting almost the same errors as the single grid version. The simulation of an inviscid and a laminar cylinder revealed no differences concerning the amount of Newton steps or integral values. In the future the method will be extended for RANS simulations and three-dimensional flows.

References

1. Benek, J., Steger, J., Dougherty, F.: A flexible grid embedding technique with application to the Euler equations. In: Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics (July 1983)
2. Reed, W., Hill, T.: Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Laboratory (1973)
3. Cockburn, B., Shu, C.: TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation law II: General framework. Mathematics of Computation 52(186), 411–435 (1989)
4. Bassi, F., Rebay, S.: A high-order accurate Discontinuous Finite Element Method for the numerical solution of the compressible Navier-Stokes equations. Journal of Computational Physics 131, 267–279 (1997)
5. Bassi, F., Crivellini, A., Rebay, S., Savini, M.: Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k - \omega$ turbulence model equations. Computers & Fluids 34, 507–540 (2005)
6. Galbraith, M., Orkwis, P., Benek, J.: Extending the Discontinuous Galerkin scheme to the chimera overset method. In: Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics (June 2011)
7. Galbraith, M., Orkwis, P., Benek, J.: Discontinuous Galerkin scheme applied to chimera overset viscous meshes on curved geometries. In: Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics (June 2012)
8. Harten, A., Lax, P., Leer, B.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Review 25(1), 35–61 (1983)
9. Roe, P.: Approximate Riemann solvers, parameter vectors, and difference schemes. Journal of Computational Physics 43(2), 357–372 (1981)
10. Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., Savini, M.: A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere, R., Dibelius, G. (eds.) 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, pp. 99–108. Technologisch Instituut, Antwerpen (1997)
11. Landmann, B., Keßler, M., Wagner, S., Krämer, E.: A parallel, high-order Discontinuous Galerkin code for laminar and turbulent flows. Computers & Fluids 37(4), 427–438 (2008)
12. Sen, S., Mittal, S., Biswas, G.: Steady separated flow past a circular cylinder at low Reynolds numbers. Journal of Fluid Mechanics 620, 89–119 (2009)

High-Order Discontinuous Galerkin Schemes for Large-Eddy Simulations of Moderate Reynolds Number Flows

T. Bolemann, A. Beck, D. Flad, H. Frank, V. Mayer, and C.-D. Munz

Institute for Aerodynamics and Gasdynamics, University of Stuttgart
Pfaffenwaldring 21, 70569 Stuttgart, Germany
{bolemann,beck,flad,frank,munz}@iag.uni-stuttgart.de,
valentin.mayer@daimler.com

Abstract. In this article, we describe the capabilities of high order discontinuous Galerkin methods at the Institute for Aerodynamics and Gasdynamics for the Large-Eddy Simulation of wall-bounded flows at moderate Reynolds numbers. In these scenarios, the prediction of laminar regions, flow transition and developed turbulence poses a great challenge to the numerical scheme, as overprediction of numerical dissipation can significantly influence the accuracy of the integral quantities. While this increases the burden on the numerical scheme and the LES subgrid model, the moderate Reynolds numbers prevent the occurrence of thin wall boundary layers and allows the resolution of the boundary layer without the need for wall modelling strategies. We take full advantage of the low numerical errors and associated superior scale resolving capabilities of high order spectral approximations by using high order ansatz functions up to $12th$ order, which allows us to resolve the significant features of these flows at a very low number of degrees of freedom. Without the need for any additional filtering, explicit or implicit modelling or artificial dissipation, the high order scheme capture the turbulent flow at the considered Reynolds number range very well.

We apply our approach to standard benchmark test cases for transitional and turbulent flows in internal and external aerodynamics: A well investigated square duct channel at $Re_\tau = 395$, a closed channel configuration with streamwise periodic hills at $Re_h = 10,595$, a circular cylinder flow at $Re_D = 3900$ and a transitional airfoil test case at $Re = 60,000$. We focus on a comparison with established schemes of lower order with explicitly or implicitly added subgrid scale models, while using fewer or approximately the same number of degrees of freedom. We demonstrate that for all computations, we achieve an equal or better match to Direct Numerical Simulation and experimental results, while retaining perfect parallel scaling and achieving very low computing times.

Keywords: Discontinuous Galerkin, Large-Eddy Simulation, transitional and detached flows, polynomial dealiasing.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_20

1 Introduction

The initial introduction of the discontinuous Galerkin (DG) method in the 1970s by Reed and Hill [21] and Lesaint and Raviart [33] was followed by extensive efforts to extend its theoretical and practical aspects for the simulation of evolution equations of fluid mechanics by a number of researchers, e.g. [1, 3, 6, 10, 13–15]. Beyond applications in inviscid or laminar problems (e.g. [4, 36]), there is a strong interest in extending DG to time-averaged turbulent flows in the form of the Reynolds Averaged Navier Stokes equations (RANS). A number of successful high Reynolds number RANS simulations with DG have been reported in literature, e.g. [9, 19, 39, 45, 51].

In contrast to these efforts, few Large-Eddy Simulation (LES) or Direct Numerical Simulation (DNS) results have been reported. This is somewhat counterintuitive, as the low dispersion and dissipation errors of high order methods combined with the excellent parallel scaling and geometric flexibility make it an attractive candidate for baseline LES and DNS schemes. In 2002, Collis [17] was among the first to use high order DG methods for the DNS of compressible flows, and he applied a Smagorinsky eddy viscosity model to the small scales via the Variational Multiscale method in a LES simulation [18]. More recently, Uranga et al. [49] reported a DG simulation of a transitional flow over an airfoil, taking advantage of the numerical dissipation to account for the damping action of the unresolved scales, yielding an implicit LES approach. A similar investigation was carried out by Carton de Wiart and Hillewaert [11]. The accuracy of discontinuous Galerkin methods for high Reynolds number vortical flows for $N = 3$ was investigated in [12]. For higher orders ($N = 7...15$), Gassner and Beck [8] showed that de-aliased discontinuous Galerkin schemes applied to under-resolved simulations of isotropic turbulence compare very well with low order finite volume schemes on fine grids with explicit or implicit subgrid scale models. In terms of computational efficiency, it was shown in [50] that high order DG schemes compete well with finite difference and finite volume formulations for turbulent flows in a LES setting.

In this paper, we present a systematic study of high order discontinuous Galerkin schemes applied to transitional and turbulent flows a moderate Reynolds numbers. Of particular interest will be the evaluation of our "no model" simulations against established explicit and implicit LES modelling techniques for the same number of degrees of freedom.

This paper is organized as follows: We start by describing our discontinuous Galerkin Spectral Element Method (DGSEM) in Sec. 2. In Sec. 3, we report our results for external flows: The circular cylinder flow at $Re_D = 3900$, where the transition occurs in the detached shear layer in Sec. 3.1, followed by a transitional flow past and airfoil at $Re = 60,000$ with a laminar separation bubble, re-attachment and transition in Sec. 3.2. We then continue with a discussion internal flows in Sec. 4. In Sec. 4.1, we investigate the square channel at $Re_\tau = 395$, where the high order method allows us to resolve the essential features in the turbulent boundary layer. The final, most complex test case is presented in Sec. 4.2, where we show the results for the IDIHOM periodic hill $Re_h = 10,595$

computations. At the end of the paper we give a brief overview and a conclusive assessment of our results in Sec. 5.

2 Discontinuous Galerkin Spectral Element Method

In this section we derive the Discontinuous Galerkin Spectral Element (DGSEM) method. We start with the three-dimensional compressible Navier-Stokes equations expressed in conservation form

$$U_t + \boldsymbol{\nabla}_x \cdot \boldsymbol{F}(U, \boldsymbol{\nabla}_x U) = 0, \tag{1}$$

where U denotes the vector of conserved quantities $U = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho e)^T$, with ρ, v_1, v_2, v_3, e being the density, velocity components and specific total energy respectively. The subscript t denotes the time derivative and $\boldsymbol{\nabla}_x$ the gradient operator in physical space. The flux is given by $\boldsymbol{F} = \boldsymbol{F}^a(U) - \boldsymbol{F}^v(U, \boldsymbol{\nabla}_x U)$, where $\boldsymbol{F}^a$ denotes the advective and $\boldsymbol{F}^v$ the viscous part.

In order to solve the system of equations, the computational domain is subdivided into non-overlapping hexahedral elements, which we allow to be connected in a fully unstructured, but conforming way. In a first step, the Navier-Stokes equations are transformed to a reference element $E \in [-1, 1]^3$. The associated mapping function $\boldsymbol{x}(\boldsymbol{\xi})$ from physical to reference space $(\boldsymbol{x^1}, \boldsymbol{x^2}, \boldsymbol{x^3})^T \rightarrow (\boldsymbol{\xi^1}, \boldsymbol{\xi^2}, \boldsymbol{\xi^3})^T$ is used to calculate the Jacobian $J(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$ and the metric terms $J\boldsymbol{a}^i$, which are constructed following Kopriva [29] to ensure the free-stream preserving property:

$$J\boldsymbol{a}^i = \frac{\partial \boldsymbol{x}}{\partial \xi^j} \times \frac{\partial \boldsymbol{x}}{\partial \xi^k}, \quad \text{with } (i, j, k) \text{ cyclic} \tag{2}$$

Applying the transformation to Eq. (1) leads to

$$J(\boldsymbol{\xi})\, U_t + \boldsymbol{\nabla}_\xi \cdot \boldsymbol{\mathcal{F}}(U, \boldsymbol{\nabla}_x U) = J(\boldsymbol{\xi})\, U_t + \boldsymbol{\nabla}_\xi \cdot (\boldsymbol{\mathcal{F}^a}(U) - \boldsymbol{\mathcal{F}^v}(U, \boldsymbol{\nabla}_x U)) = 0, \tag{3}$$

with the transformed fluxes $\boldsymbol{\mathcal{F}}^i = J\boldsymbol{a}^i \cdot \boldsymbol{F}$.

For our scheme we employ a tensor product of 1-D Lagrange polynomials ℓ^N of degree N to express the discrete approximate solution vector as

$$U(\boldsymbol{\xi}, t) = \sum_{i,j,k=0}^{N} \hat{U}_{ijk}(t) \psi^N_{ijk}(\boldsymbol{\xi}), \qquad \psi^N_{ijk}(\boldsymbol{\xi}) = \ell^N_i(\xi^1) \ell^N_j(\xi^2) \ell^N_k(\xi^3), \tag{4}$$

where $\hat{U}_{ijk}(t)$ are time dependent nodal degrees of freedom. Following Kopriva [30], $N+1$ Gauss points $\{\xi_i\}_{i=0}^N$ are chosen as interpolation nodes. The discrete transformed flux $\boldsymbol{\mathcal{F}}$ is non-linearly dependent on the solution and is at most a polynomial of degree $3N$ for compressible flows. It is thus evaluated on $M+1$ Gauss-Legendre quadrature points, with $M \geq N$ to allow for polynomial dealiasing of the non linear fluxes [28], which reads as

$$\mathcal{F}^l(U(\boldsymbol{\xi}), \boldsymbol{\nabla}_x U(\boldsymbol{\xi})) \approx \sum_{i,j,k=0}^{M} \hat{\mathcal{F}}^l_{ijk} \psi^M_{ijk}(\boldsymbol{\xi}), \quad l = 1, 2, 3 \tag{5}$$

with $\psi^M_{ijk}(\boldsymbol{\xi}) = \ell^M_i(\xi^1)\ell^M_j(\xi^2)\ell^M_k(\xi^3)$. Multiplying Eq. (3) by the test function $\phi(\boldsymbol{\xi})$ and integrating over the reference element leads to

$$\int_E \left(JU_t + \boldsymbol{\nabla}_\xi \cdot \boldsymbol{\mathcal{F}}(U, \boldsymbol{\nabla}_x U)\right) \phi(\boldsymbol{\xi})\, d\boldsymbol{\xi} = 0. \tag{6}$$

We apply the Galerkin approach and thus choose the test functions to be identical to the basis functions in Eq. (4). With a spatial integration by parts we obtain the weak formulation

$$\frac{\partial}{\partial t}\int_E JU\phi\, d\boldsymbol{\xi} + \oint_{\partial E} \left(\mathcal{F}^{a*}_n - \mathcal{F}^{v*}_n\right)\phi\, ds - \int_E \boldsymbol{\mathcal{F}}(U, \boldsymbol{\nabla}_x U)\cdot\boldsymbol{\nabla}_\xi\,\phi\, d\boldsymbol{\xi} = 0, \tag{7}$$

where $\mathcal{F}^{a*}_n$ denotes the surface normal numerical flux function for the inviscid terms, given by $\mathcal{F}^{a*}_n := \mathcal{F}^*_n(U^+, U^-)$. The superscripts $\pm$ denote the values at the grid cell interface from the neighbor and the local grid cell, respectively. For the computation of the inviscid numerical flux a variety of state-of-the art flux functions are available. In this work we apply the local Lax-Friedrichs, a HLL-type and Roe's approximate Riemann solver. For a detailed description we refer to the textbook by Toro [48]. $\mathcal{F}^{v*}_n$ denotes the numerical flux function for the viscous term, resulting from the viscous flux $\boldsymbol{\mathcal{F}^v}$ in Eq. (3). The solution gradient $\boldsymbol{\nabla}_x U$ required to compute the viscous fluxes for both the surface and volume part is obtained by applying the BR1 method by Bassi and Rebay [1]

We approximate the integral containing the volume flux contribution from Eq. (7) using a tensor product of one-dimensional Gauss quadratures

$$\begin{aligned}\int_E \mathcal{F}^l(U, \boldsymbol{\nabla}_x U)\frac{\partial\phi(\boldsymbol{\xi})}{\partial\xi^l}d\boldsymbol{\xi} &\approx \sum_{p,q,r=0}^{M}\sum_{i,j,k=0}^{M} \hat{\mathcal{F}}^l_{ijk}\psi^M_{ijk}(\boldsymbol{\xi}^M_{pqr})\frac{\partial\phi(\boldsymbol{\xi})}{\partial\xi^l}\bigg|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{pqr}} \omega^M_p\omega^M_q\omega^M_r \\ &= \sum_{p,q,r=0}^{M} \hat{\mathcal{F}}^l_{pqr}\frac{\partial}{\partial\xi^l}\phi(\boldsymbol{\xi})\bigg|_{\boldsymbol{\xi}=\boldsymbol{\xi}_{pqr}} \omega^M_p\omega^M_q\omega^M_r, \quad l = 1,2,3\end{aligned} \tag{8}$$

where we apply $(M+1)^3$ integration points $(\boldsymbol{\xi}^M_{pqr})$ and weights $(\omega^M_p, \omega^M_q, \omega^M_r)$. As already mentioned before the discrete flux is at most a polynomial of degree $3N$ for compressible and $2N$ for incompressible flows. Thus, a collocation of the flux $\mathcal{F}^l$ on $(N+1)^3$ points $\boldsymbol{\xi}^N_{ijk}$ introduces an error in the interpolant (Eq. (5)) and subsequently in the associated integration. However, the integral can be solved exactly using a sufficient number of quadrature points M. Kirby and Karniadakis showed in [28] that de-aliasing for polynomial spectral methods can be achieved by choosing sufficient integration precision. From the exactness of the quadrature rule, M is found to be $2N$ for cubic and $\frac{3}{2}N$ for quadratic non-linear integrands. Consequently, an inexact integration of these terms, i.e. an insufficient choice of M, results in an approximation error often termed "aliasing".

After evaluating the integrals from Eq. (7) numerically, we obtain a semi-discrete form. The time integration is then finally performed using the explicit low-storage Runge-Kutta scheme proposed by Carpenter and Kennedy [10].

While aliasing is one of the effects of under-resolution, another is the lack of subgrid scales, which would provide physical dissipation. In a classical LES strategy, the dissipation is provided by explicit or implicit subgrid models. Unlike this traditional approach, we do not add any type of model, filter or adapted numerics. Instead we rely on the combination of two properties of the high order schemes: The wide bandwidth of resolved scales, which carry a non-negligible physical dissipation, and a spectral cut-off like dissipation character of the DG operator with increasing polynomial degree. Gassner and Kopriva [9] showed that with increasing polynomial degree the onset of dissipation errors due to insufficient resolution is pushed towards higher wavenumbers. The potential of high order schemes for under-resolved flows has been studied for the example of the Taylor-Green-Vortex flow by Gassner and Beck in [8], showing that the solution accuracy is strongly enhanced by applying high order methods, while leaving the total number of degrees of freedom (DOF) constant. It was pointed out that this effect is caused by the discussed low dissipation and dispersion error for a wide range of resolved scales.

3 External Aerodynamics

As a showcase for applications in external aerodynamics we use the flow past a cylinder at $Re_D = 3900$, which is extensively described in literature, and the flow around the SD7003 airfoil at $Re_c = 60,000$.

3.1 Cylinder Flow

The flow past a cylinder with circular cross section at $Re_D = 3900$ based on the diameter D, freestream velocity U_∞ and density ρ_∞ is a well-established test case for the evaluation of implicit and explicit LES modelling approaches and has been extensively investigated by a large number of researchers [6, 20, 22, 31, 35, 38, 41]. At the given flow conditions, the boundary layer remains laminar until the separation point, where the shear layer is shed periodically. The ensuing transition to turbulence occurs very close to the geometry in the wake, and interacts with the resulting reverse flow region. Due to this complex flow situation, this test case proves challenging for the LES methods, which explains the considerable spread in reported results. Tab. 1 collects the integral quantities of LES and DNS simulations of this flow from literature.

In our computations, the flow domain in a plane perpendicular to the cylinder axis ($x-y$ plane) is circular, with the origin located at the center of the geometry and a radial extension of $r = 20D$. We set a spanwise extension Δz of $4D$, following [38], which is discretized by an equidistant mesh. In the axial and tangential direction ($x - y$ plane), we choose an exponential stretching in the radial and a Gaussian stretching in the circumferential direction to cluster the grid points towards the geometry and in the wake, as shown in Fig. 1. Using again the reported data in Tab. 1 as a reference, we limit the number of degrees of freedom per conservative variable of our computations to a maximum of 1.2

Table 1. Integral quantities and simulation parameters for $Re_D = 3900$ cylinder flow

Author	$C_{p_{Base}}$	Str	C_D	L_r/D	Scheme	LES	DOF
Kravchenko et al. [31]	-0.94	0.210	1.04	1.35	B-Spline SEM	Smag.	1-2 M
Blackburn et al. [6]	-0.93	0.218	1.01	1.63	GL-SEM	Smag.	1.5 M
Meyer et al. [38]	-0.92	0.210	1.05	1.38	FV	ALDM	6.0 M
Fröhlich et al. [22]	-1.03	0.216	1.08	1.09	FV	Smag.	1.4 M
Ouvrard et al. [41]	-0.81	0.226	0.93	1.68	FV/FE-VMS	Smag	1.5 M
Franke et al. [20]	-0.85	0.209	0.98	1.64	FV	Yoshizawa	1.2 M
Ma et al. (I) [35]	-0.96	0.203	0.96	1.12	h/p-FEM	DNS	24 M
Ma et al. (II) [35]	-0.84	0.219	0.96	1.59	h/p-FEM	DNS	12 M
Current: $N = 11$	-1.00	0.212	1.09	1.26	DGSEM	-	0.5 M
Current: $N = 7$	-0.90	0.208	1.02	1.50	DGSEM	-	1.2 M

$C_{p_{Base}}$: pressure coefficient at the downstream position $x = D/2, y = 0$, Str: Strouhal number of the lift coefficient $f_{Lift}\, D/u_\infty$, C_D: drag coefficient, L_r: length of separation bubble

million. We choose two high order discretizations: Firstly, an $N = 7$ setup, leading to $(N+1)^3 = 512$ degrees of freedom per element and a total of 1.2 M DOF on a grid of $16 \times 12 \times 12$ elements in circumferential, radial and spanwise direction, respectively. The second discretization consists of an ansatz of $N = 11$ polynomials, yielding a total of only 0.5 M on a $8 \times 6 \times 6$ grid. A close-up view of both grids and the resulting time- and spanwise-averaged streamwise velocity distribution is shown in Fig. 1. We use Roe's approximate Riemann solver for the convective fluxes, and Bassi and Rebays first method BR1 for the computation of gradients for the viscous contributions.

For comparison with incompressible data, we set the flow Mach number to $Ma = 0.1$ and $Re_D = (u_\infty \rho_\infty D)/\mu = 3900$. In the spanwise direction, periodic boundary conditions are chosen, while the remaining domain boundaries are set to Dirichlet free stream conditions, except for the isothermal wall conditions on the cylinder surface.

The collection of the time-averaged data was started after the establishment of stable vortex shedding and continued for 40 shedding cycles. In a post-processing step, the gathered time-averaged fields were averaged in the spanwise direction.

Along with the data reported for LES with implicit and explicit modelling and DNS results, Tab. 1 lists the results for the integral flow quantities of our high order computations. We compare the base pressure $C_{p_{Base}}$, computed as $(p - p_\infty)/(0.5\rho_\infty u_\infty^2)$, the Strouhal number of the lift force from $\frac{f_{Lift} D}{u_\infty}$, the drag

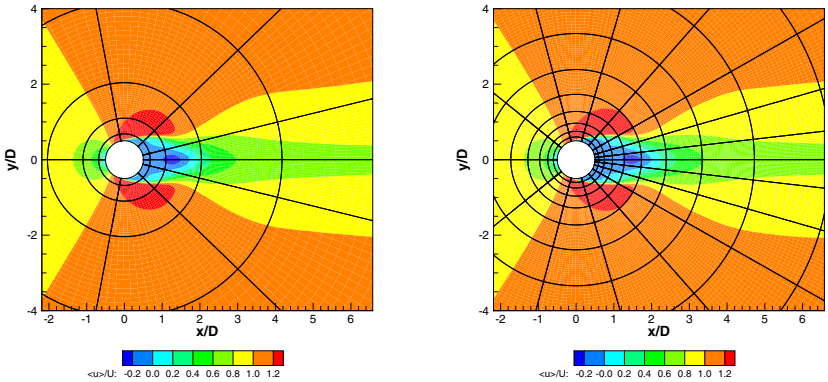


Fig. 1. Time- and spanwise-averaged streamwise velocity component $\langle u \rangle / U$ normalized by the freestream velocity and grid. *left:* $N = 11$ grid with $8 \times 6 \times 6$ elements (circumferential, radial and spanwise), *right:* $N = 7$ grid with $16 \times 12 \times 12$ elements.

coefficient C_D (computed from the streamwise force and with the projected cylinder surface in that direction) and the normalized length of the recirculation bubble L_r/D. In general, our integral quantities agree well with the published data, while the match for $N = 11$ case is the weaker of the two. This can be attributed to the very low total number of degrees of freedom and the insufficient wall resolution. Still, even for this severely under-resolved situation, the results are in reasonable agreement.

In Fig. 2, we show the time- and spanwise-averaged streamwise velocities $\langle u \rangle$, normalized by the freestream velocity, at different locations downstream of the geometry. We compare our profiles with experimentally determined values from Parnaudeau et al. [42], and two LES computations: An implicit approach through an approximate deconvolution method used in conjunction with a low order finite volume formulation presented in [38] and a high order B-Spline based Galerkin projection method with explicitly added Smagorinsky model [31]. The evolution of the wake profile is of particular interest, as the mixing and recovery of the velocity deficit is governed by viscous action, and therefore by the combination of resolved and numerical dissipation. A high viscosity generally leads to a fast interchange of momentum through shear and thus to a less pronounced, but longer wake.

The agreement of the current results for the streamwise velocity $\langle u \rangle$ (Fig. 2) is very good for all locations in the wake and for both the $N = 11$ and $N = 7$ case. In particular, for the farthest location $x/D = 2.02$, the $N = 7$ results continue to match the experimental data very well, while the published LES results show a slight upward drift on the centerline, suggesting a higher dissipation and mixing in that region. For the $N = 11$ case, the match with the experimental data at this location is worse (but very good for the LES results), suggesting the presence of insufficient dissipation due to the notably reduced spatial resolution.

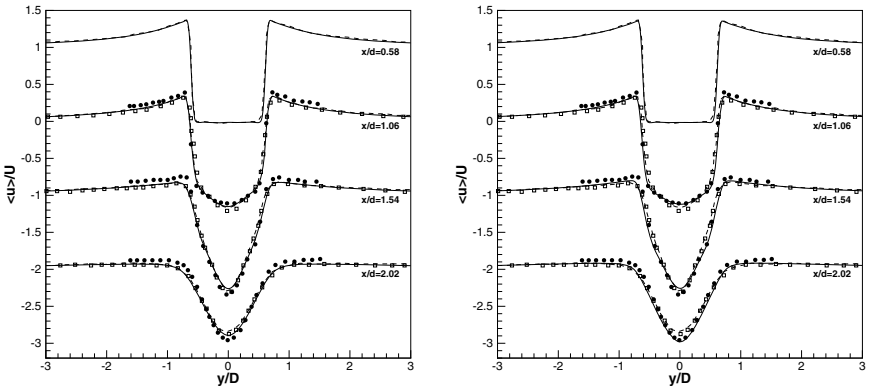


Fig. 2. Mean streamwise velocity at different downstream locations in the wake of the circular cylinder flow at $Re_D = 3900$. *left:* $N = 11$ computation, *right:* $N = 7$ computation. Solid lines: present DG result, dashed lines: computational LES results from [38], open squares: computational LES results from [31], filled circles: experimental results from [42].

Focusing now on the efficiency of the solver, Tab. 2 lists the computational details for our runs. All computations were conducted on the largest number of cores possible, given by the number of elements in the mesh and the MPI parallelization of our scheme. To facilitate comparisons between different computing architectures and between different runs of the same system, we compute the performance index PID as

$$\text{PID} := \frac{T_{CPU}}{n_{DOF} \times n_{\Delta t}} = \frac{T_{Run} \cdot n_{cores}}{n_{DOF} \times n_{\Delta t}}[s], \tag{9}$$

where $n_{\Delta t}$ denotes the number of explicit time steps and the total CPU time T_{CPU} is computed as the product of the number of computing cores n_{cores} and the simulation wall clock time T_{Run}. The number of spatial degrees of freedom per state vector component n_{DOF} is given by the product of the number of grid cells n_{cells} and the collocation points per element $(N+1)^3$. While the absolute value of the PID is strongly dependent on the computing systems, its relative changes within a single system allow us to find the optimal configuration in terms of processor load for a given system.

The total CPU time per convective time unit T_{CPU}/T^* and the associated wall clock time per convective unit T_{Run}/T^* in minutes are also listed. Due to the efficient parallelization of our scheme, we are able to compute a LES and gather meaningful statistics for this case in a matter of hours.

Summarizing the results, our discontinuous Galerkin computations show a very good agreement with published data. We employ high order schemes with near spectral cut-off dissipation behavior without LES modelling to compute these flows with a low number of degrees of freedom. The $N = 7$ computations at 1.2 M DOF show excellent matches with experimental results and are able to

Table 2. Computational cost for the cylinder flow computations

Case	DOF	cells	cores	$\overline{\Delta t}/T^*$	PID [μs]	T_{CPU}/T^*[h]	T_{Run}/T^*[min]
$N = 7$	1.180 M	2304	2304	$1.3E-3$	13	102	2.7
$N = 11$	0.498 M	288	288	$1.7E-3$	19	80	16.7

recover the shape of the near wake almost perfectly. The results of the $N = 11$ computations are also in very good agreement with reference data, but shows signs of underresolution, attributed to the very low number of 0.5 M DOF combined with the high polynomial degree and associated low inherent numerical dissipation.

3.2 Airfoil Flow

The prediction of flows at low to medium Reynolds numbers (10^4 to 10^6) gains rising attention in applications such as micro-aerial and unmanned vehicles. As these flows exhibit substantial laminar regions, a laminar separation bubble (LSB) is likely to form even at modest angles of attack, followed by transition in the separated shear layer and subsequent reattachment. The concurrent, coupled occurance of those three different flow regimes makes this type of flow very hard to predict when solving the Reynolds Averaged Navier Stokes equations. On the other hand, the computational cost of Direct Numerical Simulations in this Reynolds number regime is still very high. This leaves LES as the main candidate for the prediction of this type of flow.

We present the simulation around a Selig-Donavan (SD) 7003 airfoil at $Re_c = 60,000$ and angle of attack of $\alpha = 8°$. This airfoil has been subject of experimental [40, 46] and some computational investigations in the last years [7, 14, 23], most of them using implicit or explicit LES approaches. This setup features a LSB at the leading edge, a rapidly growing Kelvin-Helmholtz instability leading to transition to turbulence and subsequent reattachment at approximately the first third of the chord.

Following the instructions from the 2nd International Workshop on High-Order CFD Methods (2013), we run this case at $Ma = 0.1$ and employ isothermal walls with $T_{wall}/T_\infty = 1.002$. We discretize this domain with unstructured hexahedra, which allows a very efficient and coarse resolution of the farfield. To account for the airfoil curvature, the geometry is represented through a mapping with polynomial degree $N_{Geo} = 4$ within the wall-adjacent cells. The spanwise extent is 20% of the chord lengths C and periodic boundary conditions are employed at the spanwise boundaries. Two different simulations with polynomial degrees $N = 3$ and $N = 7$ are performed on two different meshes, keeping the number of degrees of freedom approximately constant. The resulting mesh for the $N = 7$ case is significantly coarser. Details on the two calculations are listed in Tab. 3.

Table 3. Degrees of freedom and resolution details of the SD7003 airfoil flow simulations

Case	DOF	cells	cells spanwise	$y^+_{x=0.8}$	N	M
$N = 3$	4.26 M	66,500	12	1.65	3	4
$N = 7$	4.55 M	8,900	6	1.10	7	9

To give an impression of the general flow behavior as well as the used mesh, vortex structures are visualized in Fig. 3 along with a cut of the computational grid. The build-up of spanwise Kelvin-Helmholtz and subsequent secondary instabilities are visible, as well as the immediate breakdown to turbulence.

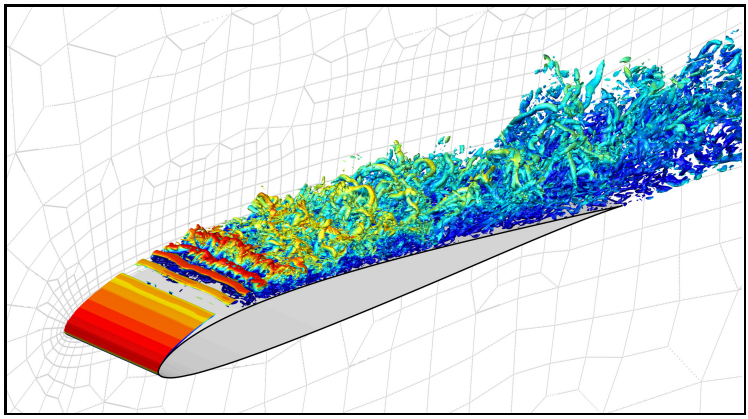


Fig. 3. Isosurfaces of the Q-criterion ($Q = 500$) colored with contours of instantaneous velocity magnitude

The aerodynamic loads as well as separation and reattachment positions predicted by our computations are compared to other published computational results and the experimental data of Selig et al. [46] in Tab 4. The current results, in particular the $N = 3$ case, agree well with the results of other groups. Compared to the results of Galbraith and Visbal [23] and Boom and Zingg [7], separation occurs earlier and reattachment later, a trend which intensifies for the higher order case $N = 7$. While the experimental lift coefficient is predicted reasonably, the drag is underpredicted by all computations. The pressure coefficient distributions are shown in Fig. 4, demonstrating fair agreement with all computations but the one of Boom and Zingg, where the LSB is shifted towards the leading edge. For the remainder of the references, good agreement is also found for the skin friction distribution until $x \approx 0.2$ and from $x \approx 0.5$.

Details on the computational cost of the two cases are listed in Tab 5. Note that the PID for the $N = 7$ case is even lower than for the $N = 3$ case, showing

Table 4. Mean aerodynamic loads and separation (x_s) and reattachment (x_r) locations and computational details of the SD7003 airfoil case

Author	C_L	C_D	x_s	x_r	Scheme	DOF
Galbraith & Visbal [23]	≈ 0.91	≈ 0.043	0.040	0.280	$\mathcal{O}(6)$ Comp. FD	5.70 M
Catalano & Tognaccini [14]	≈ 0.94	≈ 0.044	0.030	0.290	$\mathcal{O}(2)$ FD & dyn. Smag.	8.63 M
Boom & Zingg [7]	0.968	0.034	0.037	0.200	$\mathcal{O}(4)$ SBP-SAT	4.48 M
Selig et al (exp.) [46]	≈ 0.92	≈ 0.029	–	– –		–
Current: $N = 3$	0.923	0.045	0.027	0.310	DGSEM	4.26 M
Current: $N = 7$	0.932	0.050	0.030	0.336	DGSEM	4.55 M

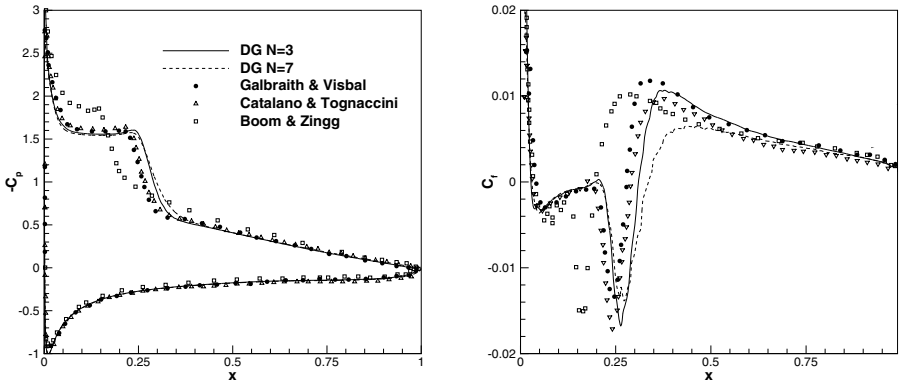


Fig. 4. Pressure coefficient for suction and pressure side (left) and skin friction coefficient for the suction side (right), SD7003 airfoil

that the high order operator is not necessarily more expensive. The increased CPU time stems mostly from a slightly lower time step.

In summary, the transitional SD7003 airfoil flow computations show mostly good agreement with published results in integral quantities and pressure and skin friction coefficient distributions. However, especially in the skin friction coefficient, spread among the available references and our results is present. For the high angle of attack of $\alpha = 8°$, the LSB is relatively short and the complex transition and reattachment processes occur on a very small portion of the airfoil. The distance to the farfield boundaries strongly affects the properties of the LSB, which presumably causes part of the spread. The complex physics renders this setup a good LES test case, but the creation of a DNS reference database to evaluate the quality of the LES results is highly desirable.

Table 5. Computational cost for the SD7003 airfoil computations

Case	DOF	cores	$\overline{\Delta t}/T^*$	cells/core	PID [μs]	T_{CPU}/T^* [h]	T_{Run}/T^* [min]
$N=3$	4.26 mio	5548	$9.7E-6$	12	13.4	3200	34.8
$N=7$	4.55 mio	4443	$7.3E-6$	2	12.5	4220	57

4 Internal Aerodynamics

Internal aerodynamics often centers on the accurate resolution of pressure-driven, wall-bounded flows, where wall-induced effects dominate the flow field. We have performed simulations of two well-established incompressible channel flow test cases and compare our results against reference DNS results from literature. The first case is a plain square duct channel at $Re_\tau = 395$, which is described extensively in literature [11, 32, 43]. Here we thoroughly investigate the behaviour of the numerical method on pure wall-bounded turbulence by the means of a direct numerical simulation. The second more complex case is a closed channel with streamwise periodic hills, a frequently used benchmark case for turbulence modelling [8,21,37,47]. Here we perform three LES computations for $Re_h = 10,595$ on the same mesh, which range from very coarse to DNS-like resolution by steadily increasing the polynomial degree. In contrast to the square duct flows, it covers physical phenomena like flow separation, reattachment and, depending on the Reynolds number, turbulent transition and relaminarization.

As the channels are chosen to be periodic in stream- and spanwise direction, we have to consider a streamwise pressure gradient $\frac{dp}{dx}$ for driving the flow. The pressure gradient is computed identically for both cases and is implemented as a spatially constant volume forcing term. We measure the mass flow rate $\dot{m}$ at some specified locations in the domain and continuously adjust the forcing to match it with a prescribed reference mass flow rate $\dot{m}_{ref}$ corresponding to the specific Reynolds number. The forcing itself is implemented by using a procedure which has been originally proposed by Benocci et al. [5]

$$\frac{dp}{dx}^{n+1} = \frac{dp}{dx}^{n} + \frac{\alpha}{A\Delta t}(\dot{m}_{ref} - 2\dot{m}^n + \dot{m}^{n-1}) \,, \tag{10}$$

where α is weighting factor, A_{ref} the reference area and $\Delta t, n$ the time step and time level. We found the above method to be sufficiently robust and accurate for a wide range of time steps. One remaining choice is the plane at which the mass flow rate $\dot{m}$ is evaluated. Although it is commonly measured at stream- or spanwise planes through the domain, we use the mass flux averaged over the whole domain, which to our experience reduces unwanted temporal oscillations of the forcing term.

4.1 Square Channel Flow

A Direct Numerical Simulation was performed for a fully developed two-dimensional turbulent channel flow at $Re_\tau = 395$. The Reynolds number (dependent

on the friction velocity u_τ which is linked to the averaged wall shear stress $u_\tau = \sqrt{T_w/\rho}$) and the Mach number of 0.1 are selected according to the reference of Moser et al. [43]. The two no-slip walls of the channel are separated by a distance of 2δ. The region of interest is periodic in streamwise and spanwise direction and the size of the computational domain is $2\pi\delta \times 2\delta \times \pi\delta$. Details of the grid and geometry parameters are listed in the Tab. 6. The flow is forced by a streamwise pressure gradient dp/dx to maintain a constant mass flux through channel as discussed above.

Table 6. Grid and geometry parameters for direct numerical simulation of a plane channel. Here Δy_c^+ gives the y-resolution at the center line.

Re_τ	L_x	L_y	L_z	$Cells: N_x \times N_y \times N_z$	Δx^+	Δy_c^+	Δz^+
395	$2\pi\delta$	2δ	$\pi\delta$	$86 \times 66 \times 64$	4.70	3.82	3.12

An impression of wall-bounded turbulence is given in Fig.5, where we visualize the coherent structures using the λ_2 vortex detection criterion.

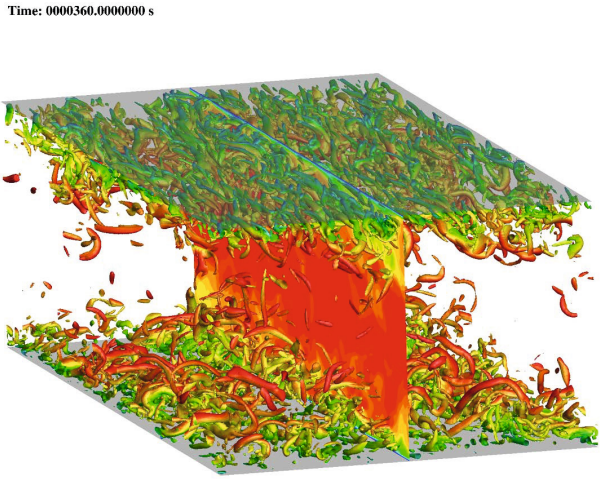


Fig. 5. Isosurfaces of the $\lambda_2 = -5$ criterion colored by contours of instantaneous streamwise velocity

Fig. 6 shows the averaged velocity profile normalized by the friction velocity in wall-coordinates $u^+ = \langle U \rangle / u_\tau$ and $y^+ = y\, u_\tau / \nu$. The results obtained by DGSEM are in very good agreement with the reference DNS. The velocity variance $\langle u_1 u_1 \rangle$ shows a slight difference above $y^+ = 75$, as presented in Fig. 7 (left). A possible cause for this deviation may be an insufficient time averaging period. On the other hand, the velocity variance $\langle u_2 u_2 \rangle$ and covariance

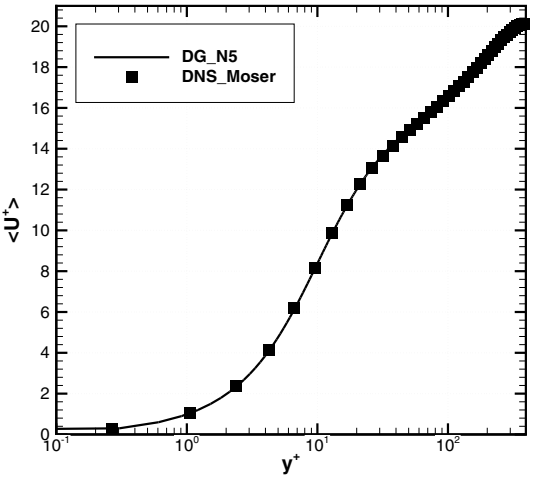


Fig. 6. Time- and spanwise-averaged velocity profile in wall units

Reynolds shear stress $\langle u_1 u_2 \rangle$ are almost perfectly superimposed on the reference data. Further expected characteristic of a wall bounded flow is that the production and dissipation of turbulent kinetic energy will be approximately in balance in the logarithmic region. However the results also confirm the observation by Moser et al. [43] that the ratio is slowly increasing over the log range, as shown this in Fig. 7 (right). Despite the small oscillations in the balance region, the profile is well captured. Details of computational cost of the DNS are summarized in Tab. 7. Here, the characteristic time $T^* = \delta / U_{Bulk}$ is related to the characteristic length scale δ and the streamwise bulk velocity U_{Bulk}.

Table 7. Computational cost for the plane channel computations

Case	DOFs	cores	cells/core	PID [μs]	$\Delta t/T^*$	T_{CPU}/T^*[h]	T_{run}/T^*[min]
$N = 5$	$83.1 \cdot 10^6$	8196	47	12.3	$3.1 \cdot 10^{-4}$	702	5.2

4.2 Periodic Hill Channel Flow

The setup of the periodic hill has been proposed by Mellen et al. [37], based on similar experimental considerations. The channel length is $L_x = 9h$ and corresponds to the distance between the two hills with h denoting the hill height. The channel height is $L_y = 3.035h$ and the spanwise extent is $L_z = 4.5h$. The Reynolds number of $Re_h = 10,595$ is based on the bulk velocity u_b at the hill crest. We are again using $Ma = 0.1$ to avoid compressibility effects. We apply

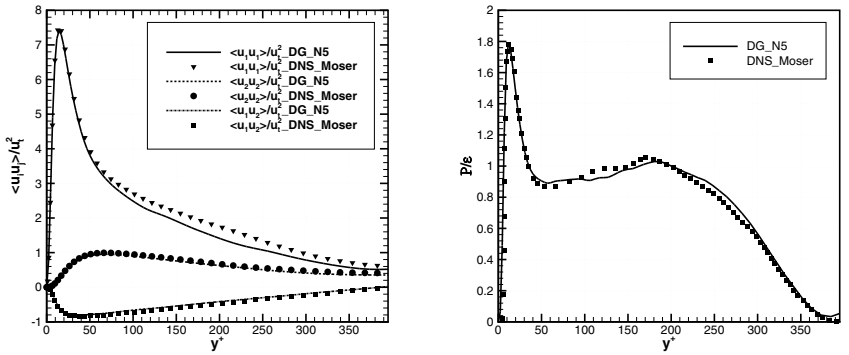


Fig. 7. Time- and spanwise-averaged velocities fluctuations $u_i u_j / u_\tau$ (left) and ratio of production (P) to dissipation (ϵ) of turbulent kinetic energy (right) in wall units

isothermal no-slip boundary conditions at the lower and upper walls with a non-dimensional temperature of 1, all other boundaries being periodic. For a more detailed description of the setup we refer to the ERCOFTAC database[1]. First thorough DNS and LES computations have been performed by Temmerman et al. [47] and Fröhlich et al. [21]. Numerical DNS and LES investigations comparing two FV codes have been carried out by Breuer et al. [8] also including experimental PIV-measurements. These DNS results cover Reynolds numbers from 100 to 10,595 and serve as an ERCOFTAC reference.

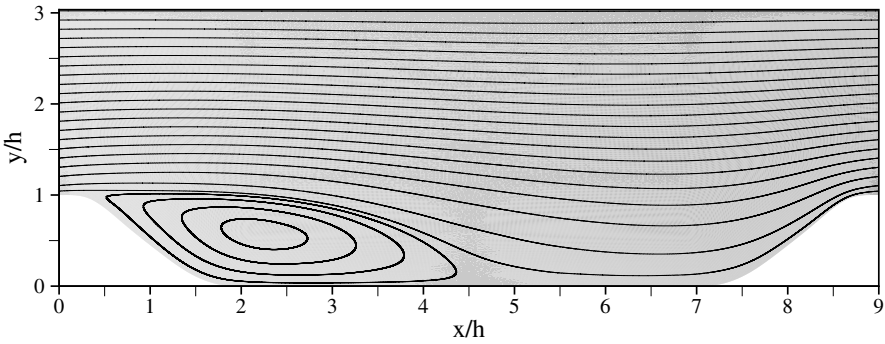


Fig. 8. Streamlines of the time- and spanwise-averaged flow

Our computational mesh is very coarse, consisting of only 8192 grid cells. To ensure an accurate representation of the geometry, we use a curved mesh with a polynomial degree of $N_{Geo} = 4$. We employ curved elements not only at the boundaries, but inside the whole domain, using an inner cell stretching

[1] `http://uriah.dedi.melbourne.co.uk/w/index.php/UFR_3-30`

Table 8. Simulation parameters for the $Re_h = 10,595$ periodic hill channel flow

Author	Scheme	Solver	DOF	$\Delta t/10^{-4}$	De-aliasing
Breuer et al. [8]	FV	Incompressible	13.1 M	18	-
Breuer et al. [8]	FV-IB	Incompressible	4.1 M	10	-
Current: $N = 4$	DGSEM	Compressible	1.02 M	6.1	$M = 5$
Current: $N = 6$	DGSEM	Compressible	2.81 M	3.2	$M = 8$
Current: $N = 9$	DGSEM	Compressible	8.19 M	1.9	$M = 12$

approach as described by Hindenlang et al. [27]. By distributing the scheme's sub-cell resolution inside the cell, we improve the boundary layer resolution.

We perform three computations, listed in Tab. 8, with an increasing the polynomial degree. We apply incomplete de-aliasing and keep the amount of de-aliased modes approximately the same for all computations with $M = 1.2-1.3N$.

The time-averaged streamlines in Fig. 8 give a general impression of the flow field at $Re_h = 10,595$. All time-averaged data was collected over 56 convective times $T^* = L_x/u_b$. The flow separates roughly at $x \approx 0.2$ on the lower wall, which is an effect of the high curvature of the descending hill, and reattaches at $x \approx 5$. The recirculation bubble inbetween is known to be strongly dependent on separation effects [47]. Breuer et al. [8] report a small recirculation region at the beginning of the second hill, which was confirmed by our computations.

The wall friction shown in Fig. 9 converges rapidly with increasing resolution near the walls, as the $N = 6$ and $N = 9$ nearly coincide. As expected, the skin friction rises significantly at the onset of the downstream hill, with its peak on the second plateau, where the biggest friction is expected to occur. The wall friction is negative in the region of the recirculation bubble and defines the size of the region. On the upper wall we see a constant decrease of the friction until the middle of the channel. This corresponds to the decrease of the velocity caused by the widening of the flow field as the normal extent of the recirculation bubble decreases.

For a detailed flow analysis, we pick the recirculation zone as a showcase, shown in Fig. 10. For the time-averaged velocities and Reynolds stresses there are no major discrepancies between our simulations and the references. The flow parameters above the crest meet the results better, as the resolution is higher at this point and there are less complex phenomena.

The results indicate that the $N = 4$ computation suffers from a lack of resolution; noticeable differences occur especially at $x = 2.0$. While the streamwise velocity $\langle u \rangle / u_b$ and Reynolds stresses $\langle u'u' \rangle / u_b$ are in good agreement with the ERCOFTAC references, there are major differences in the normal velocities and stresses, the latter being very sensitive to the amount of resolution. Oscillations near the peak can be observed for all stresses and turbulent kinetic energy. The time averaging intervall for $N = 4$ has been extended to 100 convective times

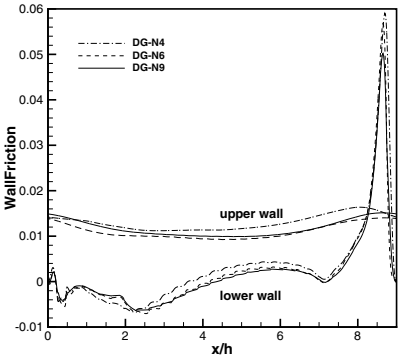


Fig. 9. Distribution of the time-averaged wall friction in streamwise direction on the lower and upper wall, $N = 4$, $N = 6$ and $N = 9$ DGSEM computations

to eliminate the risk of the oscillations stemming from a too short averaging period. We thus suspect that the oscillations indicate slight instabilities. Nevertheless the overall impression is still good, considering that barely 1.0 M degrees of freedom have been used.

Both the $N = 6$ and the $N = 9$ computations show a very good agreement with the references. While the $N = 6$ tends to have slightly lower peaks than the reference, e.g. in the normal components of the stresses $\langle u'u' \rangle / u_b^2$, the $N = 9$ slightly overshoots the published data. We note that while all three cases do well for the streamwise velocities, the improvements due to additional accuracy become apparent for the spanwise components and stresses.

For the periodic hill testcase we achieve a PID in the region of $12\mu s$. We observe a drop of the PID between $N = 4$ and $N = 6$, which is due to the caching effects of an optimal load per core. The $N = 9$ computation demonstrates that we are able to use only *one single element* per core for a high number of cores (8192) and still achieve a reasonable PID. All computations have been performed within wall-clock times ranging from a couple of hours to two days.

Table 9. Computational cost for the periodic hill computations

Case	DOFs	cores	cells/core	PID [μs]	$\Delta t/T^*$	T_{CPU}/T^* [h]	T_{run}/T^* [min]
$N = 4$	$1.02 \cdot 10^6$	640	8	13.9	$6.1 \cdot 10^{-4}$	121	11.4
$N = 6$	$2.81 \cdot 10^6$	2048	4	12.1	$3.2 \cdot 10^{-4}$	748	15.8
$N = 9$	$8.19 \cdot 10^6$	8192	1	17.1	$1.9 \cdot 10^{-4}$	5782	41.9

As a summary of the results obtained from the three computations, we see that they are in close accordance with the reference data. We demonstrate that high-order schemes with incomplete de-aliasing can achieve accurate results for

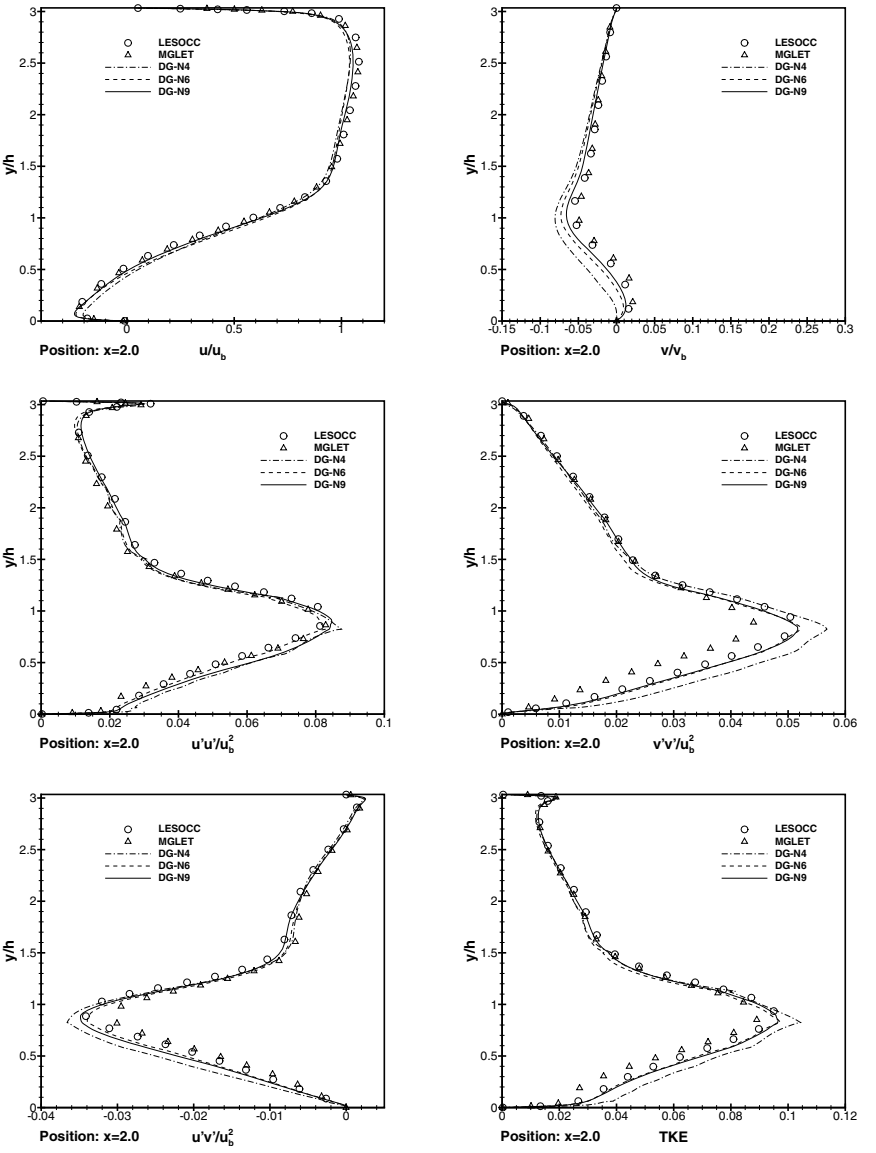


Fig. 10. Spatial- and time-averaged streamwise velocity $\langle u \rangle / u_b$, normal velocity $\langle v \rangle / u_b$, spanwise velocity, the Reynolds stresses $\langle u'u' \rangle / u_b^2$, $\langle v'v' \rangle / u_b^2$, $\langle u'v' \rangle / u_b^2$ and the turbulent kinetic energy in the separation region ($x = 2.0$)

this case with considerably less degrees of freedom than reported in literature. A good example is the $N = 4$ computation with 1.0 M degrees of freedom. Though it has a factor of 4, respectively a factor of 13 less degrees of freedom than the two reference FV solvers, the results still agree very well with the published data.

On the other hand, the $N = 6$ and $N = 9$ computations match very well with the reference results with negligible differences while still having less degrees of freedom.

5 Conclusion

Most of the recent work in the field of DG methods for fluid dynamics was centered around solving the RANS equations, also being the main motivation of the IDIHOM project. While efforts towards their industrial application on LES were not as pronounced, we demonstrate in this work that high order DG schemes have high potential in this field. We have performed DNS and LES computations for four well-established test cases from internal and external aerodynamics with low to moderate Reynolds numbers. We have employed a DGSEM scheme with high order to very high order approximations and used a higher quadrature rule for de-aliasing. All presented results are in good agreement with published reference data, though requiring considerably less degrees of freedom.

We point out that so far no additional explicit or implicit subgrid scale modelling terms or artificial viscosity have been applied for these results. For flows at higher Reynolds numbers, pure de-aliasing is no longer sufficient and sub-grid scale modeling becomes necessary. Due to their spectral cut-off character and the de-aliasing the DG schemes guarantee low dissipation and dispersion errors, which, combined with a high resolution per degree of freedom, overall minimizes the approximation errors. This is especially beneficial in an LES setup, where high order DG methods can be used as a clean numerical basis to build new LES models upon, as they reduce cross-influences between the numerics and the model.

While standard schemes featuring high accuracy are based on finite differences and thus support only structured meshes, the DG scheme is capable of handling complex geometries using unstructured meshes maintaining high order accuracy, which is relevant for industrial applications. However, in an industrial setup, not only accuracy but also the computation times play a major role. We have shown that large scale computations with a high number of degrees of freedom can be carried out within a few hours to days. This is due to excellent scalability of our codes down to a single element per core even on thousands of cores.

In summary we have demonstrated that transitional and turbulent flows can be treated accurately and efficiently using high order discontinuous Galerkin methods, without the need for additional modelling terms. Furthermore we showed that these methods fit well into an industrial setup, as they are highly scalable and are well fitted to LES applications, as they provide a clean numerical basis for future LES modelling.

References

1. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. 39(5), 1749–1779 (2002)

2. Bassi, F., Rebay, S.: A high order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. Journal of Computational Physics 131, 267–279 (1997)
3. Bassi, F., Rebay, S.: Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier–Stokes equations. International Journal for Numerical Methods in Fluids 40, 197–207 (2002)
4. Bassi, F., Rebay, S.: Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. International Journal for Numerical Methods in Fluids 40(1-2), 197–207 (2002)
5. Benocci, C., Pinelli, A.: The role of the forcing term in the large eddy simulation of equilibrium channel flow. In: Engineering Turbulence Modeling and Experiments, pp. 287–296. Elsevier (1990)
6. Blackburn, H.M., Schmidt, S.: Large eddy simulation of flow past a circular cylinder. In: Proceedings of 14th Australasian Fluid Mechanics Conference (2001)
7. Boom, P.D., Zingg, D.W.: Time-accurate flow simulations using an efficient Newton-Krylov-Schur approach with high-order temporal and spatial discretization
8. Breuer, M., Peller, N., Rapp, C., Manhart, M.: Flow over periodic hills numerical and experimental study in a wide range of Reynolds numbers. Computers and Fluids 38(2), 433–457 (2009)
9. Burgess, N., Mavriplis, D.J.: High-order discontinuous Galerkin methods for turbulent high-lift flows. In: Proceedings of Seventh International Conference on Computational Fluid Dynamics (ICCFD7) (2012)
10. Carpenter, M., Kennedy, C.: Fourth-order 2N-storage Runge-Kutta schemes. Tech. Rep. NASA TM 109111 (1994)
11. Carton de Wiart, C., Hillewaert, K.: DNS and ILES of transitional flows around a SD7003 using a high order discontinuous Galerkin method. In: Seventh International Conference on Computational Fluid Dynamics (ICCFD7) (2012)
12. Carton de Wiart, C., Hillewaert, K., Duponcheel, M., Winckelmans, G.: Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number. International Journal for Numerical Methods in Fluids (2013)
13. Casoni, E., Peraire, J., Huerta, A.: One-dimensional shock-capturing for high-order discontinuous Galerkin methods. International Journal for Numerical Methods in Fluids 71(6), 737–755 (2013)
14. Catalano, P., Tognaccini, R.: Large Eddy Simulations of the Flow around the SD7003 airfoil. In: AIMETA Conference (2011)
15. Cockburn, B., Hou, S., Shu, C.W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. Math. Comput. 54, 545–581 (1990)
16. Cockburn, B., Shu, C.-W.: The Runge-Kutta local projection p1-discontinuous Galerkin finite element method for scalar conservation laws. RAIRO Modél. Math. Anal. Numér 25(3), 337–361 (1991)
17. Collis, S.: Discontinuous Galerkin methods for turbulence simulation. In. In: Proceedings of the 2002 Center for Turbulence Research Summer Program, pp. 155–167 (2002)
18. Collis, S.: The DG/VMS method for unified turbulence simulation. In: 32nd AIAA Fluid Dynamics Conference and Exhibit (2002)
19. Crivellini, A., D'Alessandro, V., Bassi, F.: High-order discontinuous Galerkin rans solutions of the incompressible flow over a delta wing. Computers and Fluids 88(0), 663–677 (2013)

20. Franke, J., Frank, W.: Large eddy simulation of the flow past a circular cylinder at $Re_D = 3900$. Journal of Wind Engineering and Industrial Aerodynamics 90(10), 1191–1206 (2002)
21. Fröhlich, J., Mellen, C.P., Rodi, W., Temmerman, L., Leschziner, M.A.: Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. Journal of Fluid Mechanics 526, 19–66 (2005)
22. Fröhlich, J., Rodi, W., Kessler, P., Parpais, S., Bertoglio, J., Laurence, D.: Large eddy simulation of flow around circular cylinders on structured and unstructured grids. In: Hirschel, E.H. (ed.) Numerical Flow Simulation I. NNFM, vol. 66, pp. 319–338. Springer, Heidelberg (1998)
23. Galbraith, M., Visbal, M.: Implicit large eddy simulation of low Reynolds number flow past the SD7003 airfoil. AIAA paper 225 (2008)
24. Gassner, G., Beck, A.: On the accuracy of high-order discretizations for under-resolved turbulence simulations. Theoretical and Computational Fluid Dynamics 27(3-4), 221–237 (2013)
25. Gassner, G., Kopriva, D.: A comparison of the dispersion and dissipation errors of Gauss and Gauss-Lobatto discontinuous Galerkin spectral element methods. SIAM J. Scientific Computing 33(5), 2560–2579 (2011)
26. Gassner, G., Lörcher, F., Munz, C.-D.: A discontinuous Galerkin scheme based on a space-time expansion. II. Viscous flow equations in multi dimensions. J. Sci. Comp. 34(3), 260–286 (2008)
27. Hindenlang, F.J., Gassner, G.J., Munz, C.-D.: Improving the accuracy of discontinuous Galerkin schemes at boundary layers. submitted to International Journal on Numerical Methods for Fluid Dynamics
28. Kirby, R., Karniadakis, G.: De-aliasing on non-uniform grids: algorithms and applications. Journal of Computational Physics 191, 249–264 (2003)
29. Kopriva, D.: Metric identities and the discontinuous spectral element method on curvilinear meshes. Journal of Scientific Computing 26(3), 301–327 (2006)
30. Kopriva, D.: Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers, 1st edn. Springer Publishing Company, Incorporated (2009)
31. Kravchenko, A.G., Moin, P.: Numerical studies of flow over a circular cylinder at Re_D=3900. Physics of Fluids 12, 403–417 (2000)
32. Bricteux, L., Duponcheel, M., Winckelmans, G.: A multiscale subgrid model for both free vortex flows and wall-bounded flows. Physics of Fluids 21 (2009)
33. Lesaint, P., Raviart, P.: On a finite element method for solving the neutron transport equation. Academic Press (1974)
34. Lörcher, F., Gassner, G., Munz, C.-D.: An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations. Journal of Computational Physics 227(11), 5649–5670 (2008)
35. Ma, X., Karamanos, G.-S., Karniadakis, G.E.: Dynamics and low-dimensionality of a turbulent near wake. Journal of Fluid Mechanics 410, 29–65 (2000)
36. Mascarenhas, B.S., Helenbrook, B.T., Atkins, H.L.: Application of p-multigrid to discontinuous Galerkin formulations of the euler equations. AIAA Journal 47(5), 1200–1208 (2009)
37. Mellen, C.P., Fröhlich, J., Rodi, W.: Large eddy simulations of the flow over periodic hills. In: Deville, M., Owens, R. (eds.) IMACS World Congress (2000)
38. Meyer, M., Hickel, S., Adams, N.: Assessment of implicit large-eddy simulation with a conservative immersed interface method for turbulent cylinder flow. International Journal of Heat and Fluid Flow 31(3), 368–377 (2010)

39. Nguyen, N.C., Persson, P.-O., Peraire, J.: RANS solutions using high order discontinuous Galerkin methods. AIAA Paper 914, 2007 (2007)
40. Ol, M.V., McAuliffe, B.R., Hanff, E.S., Scholz, U., Kähler, C.: Comparison of laminar separation bubble measurements on a low Reynolds number airfoil in three facilities. AIAA paper 5149(1), 2005 (2005)
41. Ouvrard, H., Koobus, B., Dervieux, A., Salvetti, M.V.: Classical and variational multiscale LES of the flow around a circular cylinder on unstructured grids. Computers and Fluids 39(7), 1083–1094 (2010)
42. Parnaudeau, P., Carlier, J., Heitz, D., Lamballais, E.: Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. Physics of Fluids 20, 85–101 (2008)
43. Moser, R.D., Kim, J., Mansour, N.N.: Direct numerical simulation of turbulent channel flow upt o $re_\tau = 590$. Physics of Fluids 11(4), 943–945 (1999)
44. Reed, W., Hill, T.: Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory (1973)
45. Schoenawa, S., Hartmann, R.: Discontinuous Galerkin discretization of the Reynolds-averaged Navier-Stokes equations with the shear-stress transport model. Journal of Computational Physics 262, 194–216 (2014)
46. Selig, M.S., Guglielmo, J.J., Broeren, A.P., Giguere, P.: Summary of Low-Speed Airfoil Data, vol. 1. Soartech Publications (1995)
47. Temmerman, L., Leschziner, M.A., Mellen, C.P., Fröhlich, J.: Investigation of wall-function approximations and subgrid-scale models in large eddy simulation of separated flow in a channel with streamwise periodic constrictions. International Journal of Heat and Fluid Flow 24(2), 157–180 (2003)
48. Toro, E.: Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer (June 1999)
49. Uranga, A., Persson, P.-O., Drela, M., Peraire, J.: Implicit large eddy simulation of transition to turbulence at low Reynolds numbers using a discontinuous Galerkin method. International Journal for Numerical Methods in Engineering 87(1-5), 232–261 (2011)
50. Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., Visbal, M.: High-order cfd methods: current status and perspective. International Journal for Numerical Methods in Fluids 72(8), 811–845 (2013)
51. Wurst, M., Kessler, M., Kraemer, E.: Aerodynamic and acoustic analysis of an extruded airfoil with a trailing edge device using detached eddy simulation with a discontinuous galerkin method. In: 21st AIAA Computational Fluid Dynamics Conference (2013)

Aeroelastic System for Large Scale Computations with High Order Discontinuous Galerkin Flow Solver

K. Kotecki, H. Hausa, M. Nowak, W. Stankiewicz, R. Roszak, and M. Morzyński

Poznań University of Technology, Division of Virtual Engineering, 60-965, Poznań, Poland
kotecki.k@gmail.com

Abstract. Article presents the development process of aeroelastic system basing on finite volume CFD solver for higher order methods. The main aspect is interpolation tools which allows application of Discontinuous Galerkin solution of CFD solver. There is also described the elastic analogy deformation tool for curvilinear mesh. To summarize, the two testcases of wing and wing-body configuration aircraft are presented.

Keywords: Aeroelasticisty, high order, CFD.

1 Introduction

The concepts of CFD solutions, formulated already many years ago are becoming present in industrial practice. Finite Element Method, Discontinuous Galerkin Method, High Order Methods, Adjoint Mesh Refinement are confronted with traditional, second order Finite Volume Solvers. This transition in technology has to be accompanied in complete revision of CFD tools and methods, ranging from curvilinear grid generation to visualization of high order discontinuous results.

Significant participation of the CFD solver in aeroelastic simulations enforce the major changes in remaining parts of the system. Consequently, the development of interpolation tool between CFD and CSM domains, grid deformation tool is necessary. The replacement of CFD solver was possible because the AE System was primarily projected based on loosely coupling technique - two independent solvers are connected by external interface tools [1].

2 Aeroelastic System Development

Within IDIHOM project existing aeroelastic system, destined to full scale aircraft flutter simulations and based on second order CFD solver was adopted to the novel demands of high order, curvilinear grids and discontinuous Galerkin, Finite Element formulation. The new CFD solver - PADGE Code, delivered by

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_21

DLR, has replaced TAU Code in existing aeroelastic system. In the case of aeroelastic simulations [1], the use of high-order meshes leads to additional challenges in the AE coupling tools. One of them is mesh deformation, where existing deformation tools fail according to second-order grid tests, resulting in errors like concave elements or crossing edges. Another problem exists in interpolation tools where pressure distribution from CFD simulation is not continuous in case of Discontinuous Galerkin formulation.

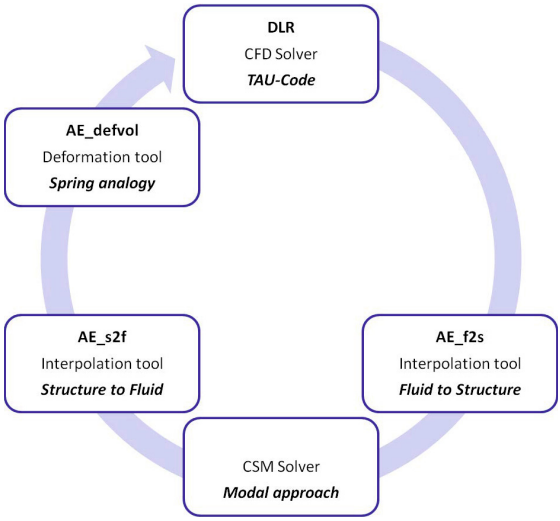


Fig. 1. Aeroelastic steady state loop developed in TAURUS project based on second order FV TAU Code solver

Simulations conducted on the aeroelastic system developed in TAURUS project (fig. 1) base on linear grids. The results from presented system (TAURUS AE System) are treated as a reference for IDIHOM AE System (fig. 2). The main aim of PUT in IDIHOM project was the implementation new CFD solver and adaptation other parts of the existing system. The CSM part of the system remained unchanged, while all the tools responsible for exchanging information between fluid and structure domains have been developed.

3 Interface for Discontinuous Solution

The first step in aeroelastic loop is to solve the problem of fluid flow. There are crucial differences between the solvers. TAU Code returns the force vectors in every node of the linear grid, while the result of PADGE Code-based simulation is discontinuous pressure distribution. Every node belonging to n elements has n values of pressure. That output from PADGE Code is not suitable for interpolation tools. Assuming the use of existing algorithm, PUT developed method to

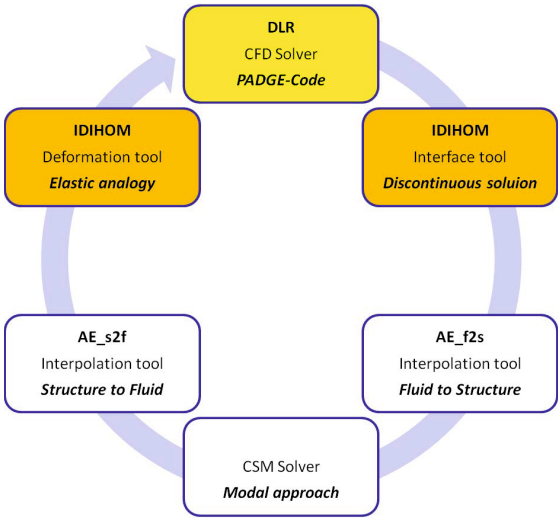


Fig. 2. Aeroelastic steady state loop developed in IDIHOM project based on high order Discontinuous Galerkin PADGE Code

calculate equivalent solution to the one from PADGE Code, in the format suitable for existing tool. It is required to preserve the conservation of the exchanged quantity.

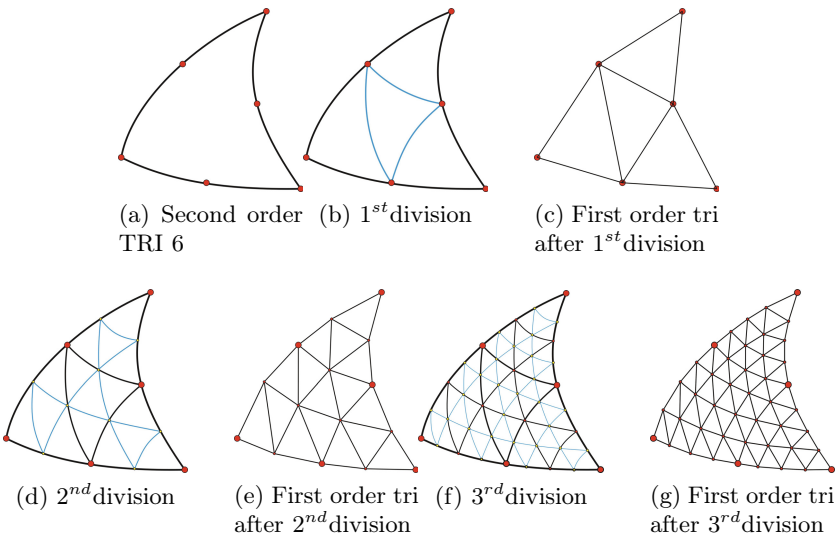
Another aspect, which should be taken into consideration is number of degrees of freedom of CFD grid. High-order meshes are characterized in that the same geometry is mapped by much smaller number of nodes and elements than linear ones. Application averaging force vectors only from CFD grid nodal points might be insufficient for proper calculation of the structure's deflection. Moreover, the average of all values in the node does not guarantee preservation of conservation. The idea which allows to preserve the conservation and ensure accurate transfer the continuous load (pressure) is based on the division of the elements to fine-enough pieces with constant load.

On the surface where the data is interpolated, there are only triangular and quadrilateral elements. Every second-order element contains tree nodes on each edge. These nodes allow the division of triangle onto four first-order triangles. By inserting one node on each edge of first-order triangle, the second-order triangle is created. This procedure can be repeated - after the next division the number of first-order triangles created from one second-order triangle will be 4^n, where n is number of divisions (tab.1). The idea of division is presented on (fig.3).

The second-order quad element is divided onto four first-order quads. Similarly to triangles, one node is inserted on each of the edges an additional one in the middle of first-order quad. As a result, the second-order quad is created. Again, this procedure can be repeated, leading to 4^n first-order quads created from single second-order quad (where n is number of divisions). At the end every

Table 1. Division of second order triangle elements

Division	No of first-order tri	No of extra nodes
1	4	0
2	16	9
3	64	39
4	256	147



(a) Second order TRI 6 (b) 1^{st}division (c) First order tri after 1^{st}division

(d) 2^{nd}division (e) First order tri after 2^{nd}division (f) 3^{rd}division (g) First order tri after 3^{rd}division

Fig. 3. Process of division of second-order triangle

first-order quad is divided into two triangles, hence the number of first-order triangles will be $2 \cdot 4^n$ (tab.2).

Coordinates of every extra added node, as well as the value of the pressure in this node, are calculated from shape functions of the primary element. Then the area and center of gravity are computed. The procedure integrating pressure over the surface of the triangle returns value of the force. The direction of the force vector is taken as the normal direction to the surface. The resulting pairs: coordinates of middle points from triangles and force vector correspond to the ones generated by TAU Code. Due to compatible format generated pairs, tools from existing AE System can be used. Thus, all the further steps: from providing the information from CFD solver to calculating deformations on the CFD surface nodes remain the same in both systems. Finally, CFD grid deformation is ran.

The number of applied division procedures is set by test procedure which calculates the convergence. Test procedure computes the sum of vector forces after first, second and next divisions, for the specific grid. Then, the convergence

Table 2. Division of second order quadrilateral elements

Division	No of first-order tri	No of extra nodes
1	8	0
2	32	16
3	128	72
4	512	280

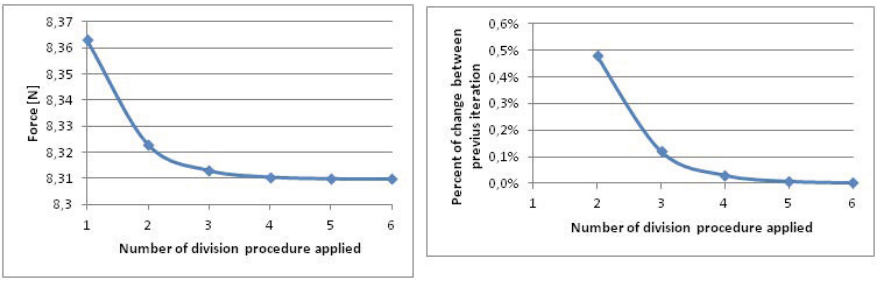


Fig. 4. Sum of force vectors (left) and percentual change between two consecutive divisions (right) for selected element depending on the number of divisions

of the results is calculated and sufficiently accurate division is chosen for the grid. For the rest of aeroelastic simulation this division is applied. The chart of example convergence for number of divisions is presented in (fig.4). The first chart shows the sum of vector forces on one element depending on from number of applied division procedures. Next one presents the percent of the change in force between actual and previous division.

4 Deformation of CFD Grid

The deformation of the CFD grid is the last stage in aeroelastic loop. The nodal values of displacement, calculated by CSM solver, are interpolated on the surface of CFD grid. The deformation is a process of moving the nodes inside CFD grid, basing on the displacement of surface nodes. The examples of such deformation, based on spring analogy, might be found in [2] and [3].

In the present study, novel approach of the mesh deformation for high-order methods and meshes is proposed. It is based on elastic analogy where CFD grid is treated as elastic body and displacement on surface is treated as kinematic load.

In iterative process, the material is associated for each cell of the grid. After solving steady state simulation of the structure, the mesh quality is checked.

Grid quality is crucial for aeroelastic simulation and especially for the CFD solution and its estimation is the basis for the decision of further iterations. In case of insufficient grid quality, the fictitious materials of grid are changed until the established quality is obtained.

As the metrics of the mesh quality, Jacobian values have been used so far, but the use of curvilinear meshes led to the need of new mesh quality metrics. These metrics, developed by Université catholique de Louvain (UCL) [5] [6], base on Bezier functions and allow assessment of the Jacobian in whole element, not only in nodal points. Cooperation with the partners from UCL allowed the implementation of these metrics in the PUT's deformation tool.

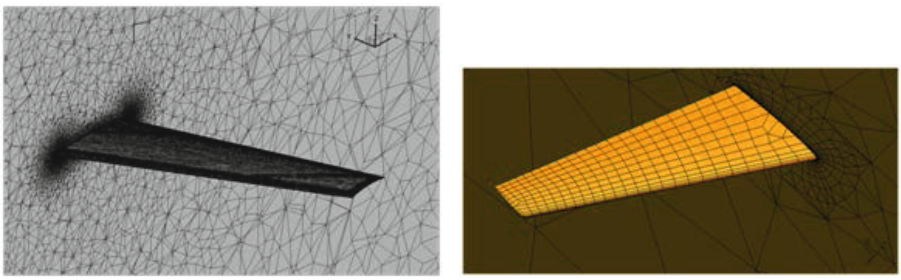
The details of deformation process are described in article: "Deformation of curvilinear meshes far aeroelastic analysis".

5 Testcases

The developed AE system has been tested on two testcases: LANN wing [3] and DLR-F6 wing-body configuration [4]. The LANN wing was underling testcase while the DLR-F6 was a challenge one. The simulation conducted in TAURUS AE System is treated as a reference solution and it was compared with IDIHOM AE System results.

5.1 LANN Wing

LANN wing is underlying testcase proposed for testing developed AE System. CAD Geometry has been delivered by Aircraft Research Association (ARA). The model has been used as a coupling surface for exchanging information between CFD and CSM models.



(a) linear (b) curvilinear

Fig. 5. LANN grids

The wing has a semispan 1m in length, a planform area of 0.25m^2, a root chord of 0.361m, a tip chord of 0,144m, has a 1/4-chord sweep angle of 25° and a linear twist from root to tip of 4.8°. The CFD simulation has been prepared with flow parameters: Ma = 0.4, Re = $4.9 \cdot 10^6$ and angle of attack equal 2.56°.

The linear mesh (fig.5a) is unstructured hybrid grid with prism elements in boundary layer. The curvilinear grid (fig.5b) for high-order computations consists of prism, pyramid, hexahedral and tetrahedral elements. The detaild of both meshes are given in tab.3.

Table 3. Grids details

Grid	linear	curvilinear
No of nodes	2.6M	98 469
No of elements	10.5M	19 981
- prism	2.5M	432
- pyramid	17k	673
- tetrahedral	7.5M	5 762
- hexahedral	0	12 114

The convergence of aeroelastic steady state computations is obtained after five iterations (fig. 6, 7).

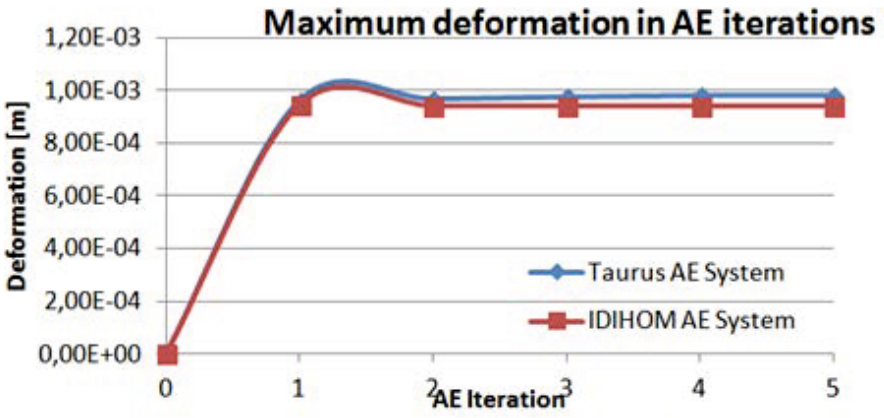
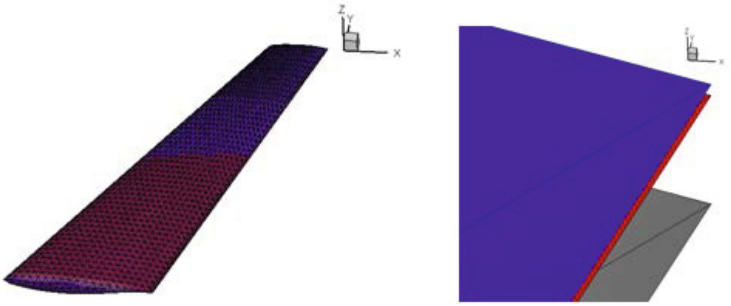


Fig. 6. Aeroelastic steady state LANN wing deformation computed using first order (TAURUS) AE System and high order (IDIHOM) AE System

The deformation computed using High-Order methods is 0.942mm. Comparing to the reference solution, where the deformation is 0.980mm, the difference is 3.96%.

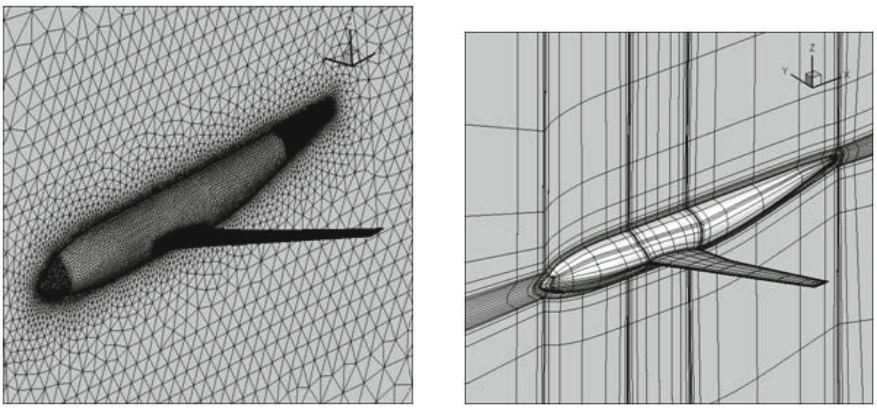


(a) (b)

Fig. 7. Deformed wing: grey - non-deformed structural model, blue - deformed model calculated in TAURUS AE System, red - deformed model calculated in IDIHOM AE System

5.2 DLR-F6 Wing-Body Configuration

DLR-F6 wing-body configuration has been the second testcase. In this case, for high order computations structured grid has been was used. In the reference computation, similarly as LANN wing, hybrid grid has been used. Both the grids are depicted in the fig. 8 and described in tab. 4.



(a) linear (b) curvilinear

Fig. 8. DLR-F6 grids

Again, the aeroelastic steady state computations converge after five iterations (fig. 9).

This time, the difference between reference solution (deformation: 1.733mm) and High-Order solution (deformation: 1.619mm) is 6.58% (fig. 10).

Table 4. Grids details

Grid	linear	curvilinear
No of nodes	4.0M	3.3M
No of elements	10.2M	50 618
- prism	6.1M	0
- pyramid	11k	0
- tetrahedral	5.1M	0
- hexahedral	0	50 618

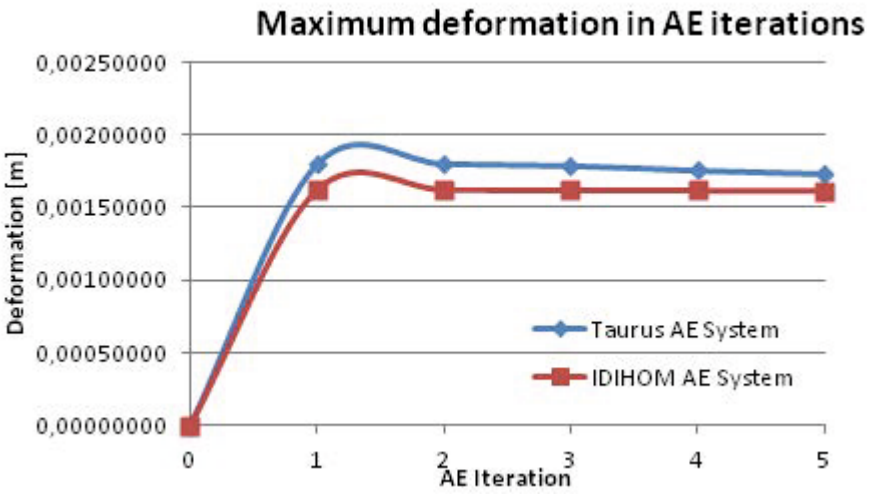
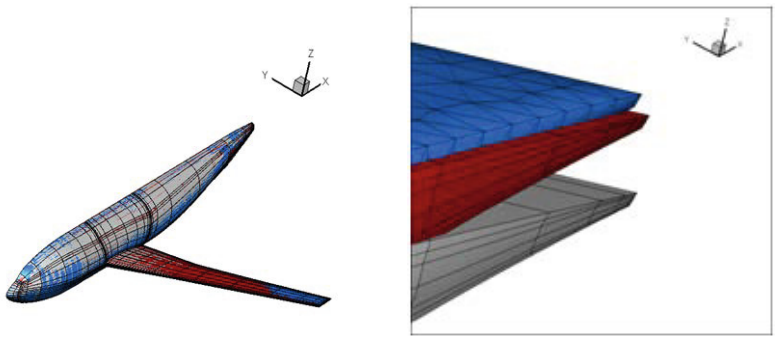


Fig. 9. Aeroelastic steady stare DLR-F6 aircraft deformation



(a) (b)

Fig. 10. Deformed aircraft: grey - non-deformed structural model; blue - deformed model calculated in TAURUS AE System; red - deformed model calculated in IDIHOM AE System

More details about testcases are presented in article: "Aeroelastic testcases".

6 Summary

In the present paper, aeroelastic system using high-order methods and high-order, curvilinear meshes has been described. The details of the necessary changes have been given. These changes include the interpolation of pressures/displacements between two problem domains, handling the data from discontinuous-Galerkin CFD solver and the deformation of curvilinear meshes preserving the quality. The last task required the implementation of new quality metrics. The new AE System has been compared with the reference AE System basing on FV CFD Solver.

References

1. Kamakoti, R.: Computational Aeroelasticity Using a Pressure-Based Solver. PhD thesis, University of Florida (2004)
2. Degand, C., Farhat, C.: A three-dimensional torsional spring analogy method for unstructured dynamic meshes. Comput. Struct. 80, 305 (2002)
3. Bloom, F.J.: Consideration on the spring analogy. Int. J. Numer. Methods Fluids 32, 647–668 (2000)
4. Roszak, R., Posadzy, P., Stankiewicz, W., Morzyski, M.: Fluid-structure interaction for large scale complex geometry and non-linear properties of structure. Archives of Mechanics 61(1), 3–27 (2009)
5. Geuzaine, C., Remacle, J.-F.: Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International Journal for umerical Methods in Engineering 79(11), 1309–1331 (2009)
6. Toulorge, T., Geuzaine, C., Remacle, J.-F., Lambrechts, J.: Robust untangling of curvilinear meshes. Journal of Computational Physics 254, 8–26 (2013)
7. Ruo, S.Y., Malone, J.B., Horsten, J.J., Houwink, R.: The LANN programan experimental and theoretical study of steady and unsteady transonic airloads on a supercritical wing. AIAA Paper 1983-1686
8. Brodersen, O., Strmer, A.: Drag Prediction of EngineAirframe Interference Effects using Unstructured Navier-Stokes Calculations. In: 19th AIAA Applied Aerodynamics Conference, Anaheim, CA, USA, June 11-14. AIAA Paper 2001-2414 (2001)

New Developments for Increased Performance of the SBP-SAT Finite Difference Technique

J. Nordström[1] and P. Eliasson[2]

[1] Department of Mathematics, Computational Mathematics, University of Linköping, SE-581 83 Linköping, Sweden
jan.nordstrom@liu.se
[2] Department of Aeronautics and Autonomous Systems, FOI, The Swedish Defense Research Agency, SE-164 90 Stockholm, Sweden
peter.eliasson@foi.se

Abstract. In this article, recent developments for increased performance of the high order and stable SBP-SAT finite difference technique is described. In particular we discuss the use of weak boundary conditions and dual consistent formulations. The use of weak boundary conditions focus on increased convergence to steady state, and hence efficiency. Dual consistent schemes produces superconvergent functionals and increases accuracy.

Keywords: High order accuracy, stability, finite difference, summation-by-parts, weak boundary conditions, convergence to steady state, dual consistency, super-convergence.

1 Introduction

We briefly describe two different new contributions to the theory for high order stable SBP-SAT finite difference schemes. The complete description of the weak boundary procedures for convergence to steady state is given in [1] while the development of dual consistence schemes is presented in [2], [3], [4]. The reader is referred to articles mentioned above for potentially missing details.

2 Weak Solid Wall Boundary Conditions and Steady State

The formulations of solid wall boundary conditions for the Navier-Stokes equations and the related slip condition for the Euler equations are well known. Less well known is the relation between these two formulations with a weak implementation. One of the more striking features is the fact that with a weak implementation of the boundary condition, the velocity at the wall becomes zero only for very fine meshes.

The main reason to use weak boundary procedures stems from the fact that together with summation-by-parts operators they lead to provable stable schemes.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_22

For application of this technique to finite difference methods, node-centered finite volume methods, spectral domain methods and various hybrid methods see the references in the original article [1] In this section we will consider a new effect of using weak boundary procedures, namely that it in many cases (all that we tried) speeds up the convergence to steady-state.

Most of the work on convergence acceleration to steady-state is done for the discrete problem. Techniques such as local time-stepping, multigrid, residual smoothing, etc are used to enhance the convergence to steady-state. In the majority of the investigations, the discrete problem including boundary conditions, is formulated first, and the numerical convergence acceleration technique is more or less independently added on afterwards. We will not consider this type of numerical convergence acceleration techniques but rather focus on more fundamental aspects related directly to the governing equations and the numerical scheme.

We will consider the basic requirement for convergence to steady-state, namely the position of the eigenvalues (the spectrum) of the Initial Boundary Value Problem (IBVP). The eigenvalues of the corresponding semi-discrete Initial Value Problem (IVP) problem obtained by using the method of lines are equally important and will also be discussed. We start by formulating the relevant IBVP and the data requirements for the existence of a steady-state solution. Next we make sure that we have a well posed procedure (such that it is possible to reach the steady-state) and finally quantify the speed of convergence using the utmost right lying eigenvalue in the spectrum. That is repeated with minor technical modifications (such as replacing the concept well-posedness by stability, investigating numerical eigenvalues, etc.) for the IVP.

Finally we apply the general theory to the specific case of solid wall boundary conditions. Our basic theoretical tools will be the classical ones, namely the energy method, the Laplace transform technique and the matrix exponential. Our basic computational tool is a node vertex edge based flow solver for unstructured grids, the Edge code developed by FOI.

2.1 The Formulation of the IBVP for Steady-State Calculations

Consider the following time-dependent one-dimensional model problem.

$$\begin{aligned} u_t + \mathcal{A}u &= F(x,t), \quad && x \geq 0, \quad && t \geq 0 \\ Lu &= g(t), && x = 0, && t \geq 0 \\ u &= f(x), && x \geq 0, && t = 0, \end{aligned} \tag{1}$$

where $u(x,t) = (u^1, u^2, .., u^m)^T$ is the solution vector with m components, and $\mathcal{A} = \mathcal{A}(\partial/\partial x)$ is the differential operator. The forcing function F, the boundary data g and the initial function f are the data of the problem. For simplicity we disregard the influence of the right boundary.

Assume now that we want to use (1) and compute a steady solution v that satisfies

$$\begin{aligned} \mathcal{A}v &= \tilde{F}(x), \quad x \geq 0 \\ Lv &= \tilde{g}, \quad x = 0. \end{aligned} \tag{2}$$

The difference problem that describes the possible convergence to steady-state is obtained by subtracting (2) from (1). The deviation $e = u - v$ from steady-state satisfies

$$\begin{aligned} e_t + \mathcal{A}e &= dF(x,t), \quad x \geq 0, \quad t \geq 0 \\ Le &= dg(t), \quad x = 0, \quad t \geq 0 \\ e &= df(x), \quad x \geq 0, \quad t = 0, \end{aligned} \tag{3}$$

where $dF(x,t) = F - \tilde{F}$, $dg(t) = g - \tilde{g}$ and $df(x) = f - v$. Note that we have used the fact that $v_t = 0$ to arrive at (3).

For a steady-state solution to exist, no time-dependent data are allowed. This means that dF and dg must vanish as $t \to \infty$. Furthermore, we cannot expect to be able to guess the initial condition that makes $df = 0$. The task is therefore to develop techniques for the IBVP (3) with zero forcing function, zero boundary data and non-zero initial function such that $e \to 0$ as $t \to \infty$.

For later reference, the final version of the steady-state problem under investigation is

$$\begin{aligned} e_t + \mathcal{A}e &= 0, \quad x \geq 0, \quad t \geq 0 \\ Le &= 0, \quad x = 0, \quad t \geq 0 \\ e &= f, \quad x \geq 0, \quad t = 0. \end{aligned} \tag{4}$$

Our ambition is to reduce e from its initial value f and reach zero fast.

2.2 Convergence to Steady-State of Solutions to the IBVP

We discuss the requirements for obtaining steady-state solutions ($e = 0$) in (4). In almost all practical cases, the spatial operator $\mathcal{A}$ is given. The speed at which $e \to 0$, the convergence rate to steady-state, can therefore only be manipulated by the boundary condition $Le = 0$. Different choices of the boundary operator L leads to different convergence rates.

The Laplace-Transform Method for the Convergence Rate. The convergence rate will be obtained by using the so called Laplace-transform technique, see the references in [1]. Assume that a suitable boundary operator L has been determined. The Laplace transformed version of (4) is

$$\begin{aligned} (sI + \mathcal{A})\hat{e} &= f, \quad x \geq 0 \\ L\hat{e} &= 0, \quad x = 0, \end{aligned} \tag{5}$$

where $s = \eta + i\xi$ is the dual variable to time and

$$\hat{e}(x,s) = \mathcal{L}e = \int_0^\infty e(x,t) \exp(-st) dt. \tag{6}$$

The integral in (6) is well defined if the time-growth of the solution e in (4) is bounded, i.e. if (4) is well posed and η is sufficiently large and positive.

To solve (5) we make the ansatz $\hat{e}_h = \psi \exp(\kappa x)$ for the homogeneous solution. The particular solution $\hat{e}_p$ which depends on the initial data f is assumed known. That leads to a generalized eigenvalue problem for $\kappa(s)$ of the form

$$(sI + \mathcal{A}(\kappa))\psi = 0, \quad |sI + \mathcal{A}(\kappa)| = 0. \tag{7}$$

The first equation in (7) has a non-trivial solution $\psi \neq 0$ if and only if there are κ such that the second relation is satisfied. $\mathcal{A}(\kappa)$ is a polynomial in κ with matrix coefficients. As an example, $\mathcal{A} = A\partial/\partial x + B\partial^2/\partial x^2$ leads to $\mathcal{A}(\kappa) = A\kappa + B\kappa^2$ where A and B are matrices. Note that $\psi = \psi(\kappa(s), s)$.

The homogeneous solution in the absence of multiple generalized eigenvalues κ, is

$$\hat{e}_h = \sum_i \sigma_i \psi_i \exp(\kappa_i x). = \Psi \bar{X}(x)\sigma, \tag{8}$$

where $\bar{X}(x) = \mathrm{diag}[e^{\kappa_1 x}, e^{\kappa_2 x}, ...]$, $\bar{X}(x) = \mathrm{diag}[e^{\kappa_1 x}, e^{\kappa_2 x}, ...]$, $\Psi = [\psi_1, \psi_2, ...]$ and $\sigma = [\sigma_1, \sigma_2, ...]^T$. The coefficients σ_i will be determined by the boundary conditions. The total solution is given by $\hat{e} = \hat{e}_h + \hat{e}_p$. The boundary conditions in (5) lead to $L\hat{e}_h = \hat{g}$ where $\hat{g} = -L\hat{e}_p$. By using (8), the final equation for the coefficients σ_i becomes

$$E(s)\sigma = \hat{g}, \qquad E(s) = L\Psi\bar{X}(0). \tag{9}$$

$E(s)$ is a matrix with the structure given by the boundary operator L, the eigenvectors ψ_i and the generalized eigenvalues κ_i. The right-hand side $\hat{g}$ is known and depend on the particular solution and it's gradients on the boundary.

A unique solution σ is obtained if E in (9) is non-singular. With s such that $\eta > \eta^*$ where all the possible singularities (or eigenvalues) in E lies to the left in the complex plane, we can solve for σ and formally transform back to time domain by

$$e(x,t) = \mathcal{L}^{-1}\hat{e} = \exp(\eta^* t)\left(\frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{e}(x, \eta^* + i\xi)\exp(i\xi t)d\xi\right). \tag{10}$$

Convergence to steady-state is obtained if $\eta^* < 0$. The way to increase the convergence rate (η^*) to steady-state is to choose the boundary operator L such that η^* lies as far as possible to the left in the complex plane. For later reference we denote $\eta^* =$ the continuous decay rate.

Remark: With multiple roots κ to the first equation in (7), the ansatz $\hat{e}_h = \psi \exp(\kappa x)$ must be reformulated. Instead of a constant vector, ψ is now of the form $\psi = \psi_0 + x\psi_1 ... + x^n\psi_n$, where $n+1$ is the multiplicity of the root κ. That complicates the derivation above technically, but in principle it remains the same. Also the conclusion that we have convergence to steady-state for $\eta^* < 0$ remains the same.

The Energy Method for Deriving Boundary Conditions. In the discussion above, the boundary operator was assumed given. To derive suitable boundary operators L such that $e \to 0$ fast as $t \to \infty$ we use the energy method. By choosing dissipative boundary operators L we hope to push the spectrum (the utmost right lying eigenvalues) as far left as possible in the left half plane. Multiply (4) with e^T, add the transpose of the equation, and integrate over the domain. That leads to

$$\|e\|_t^2 = -\int_0^\infty e^T(\mathcal{A}+\mathcal{A}^T)edx = BT(e,e_x)_{x=0} - \int_0^\infty R(e,e_x)dx. \tag{11}$$

For (11) to be well posed, the right-hand side (RHS) in (11) must be bounded by $const.\|e\|^2$.

Remark: In (11) we have assumed that the operator $\mathcal{A}$ may include second derivatives (integration-by-parts yield boundary and volume terms with first derivatives). All important hyperbolic (Euler, Maxwells, wave equations) and parabolic (heat, stress equations) as well as incompletely parabolic problems (Navier-Stokes equations) are thereby included.

For fast energy decay, the RHS in (11) should be made as negative as possible. In almost all practical cases, the spatial operator $\mathcal{A}$ is given and hence also the original form of the RHS. There is no possibility to modify the volume term R (which must be negative semi-definite). The rate at which the norm is decreasing (or increasing) can only be manipulated by the boundary condition $Le = 0$. Different choices of L leads to different sizes and signs of $BT(e,e_x)_{x=0}$ and different convergence rates.

Remark: No direct method for constructing boundary conditions that lead to fast convergence to steady-state exist. In this paper we assume that forcing the energy to decay fast, will lead to fast (a negative η^* with large magnitude) convergence to steady-state.

2.3 The Formulation of the IVP for Steady-State Calculations

We discretize the IBVP (4) in space and leave time continuous, i.e. use the method of lines. The focus is on the matrix properties of the resulting system of ordinary differential equations. The semi-discrete version of (4) can formally be written

$$\begin{aligned} \mathbf{e}_t + \bar{A}\mathbf{e} &= 0, \quad t \geq 0 \\ \mathbf{e} &= f, \quad t = 0. \end{aligned} \tag{12}$$

where $\mathbf{e} = (e_0, e_1, .., e_N)^T, e_j = (e^1, e^2, .., e^m)_j^T$ is the discrete version of the deviation e from steady-state. $\bar{A} = A - \Sigma$ is a modified version of A, where A is the $(N+1)m \times (N+1)m$ discretization matrix approximating the operator $\mathcal{A}$ and $N+1$ is the number of grid points. When the scheme is adjusted to include the numerical treatment of the boundary conditions (Σ) we obtain $\bar{A}$.

There are two distinctly different ways to prescribe boundary conditions for node vertex solvers where the unknowns are located on the boundary. One can

use a weak or a strong boundary procedure. In a weak boundary procedure, the quantities at the boundaries, even though they are known, are updated in time. The boundary value typically deviates slightly from the prescribed value but the deviation is reduced as the grid is refined. With a strong boundary procedure, on the other hand, the specified boundary value is injected into the dependent variable on the boundary. The boundary quantity is no longer an unknown, and there is hence no need for an update.

In the case of a weak boundary procedure we have $\bar{A} = A - \Sigma$ where the matrix A correspond to the internal discretization (often on so called summation-by-parts (SBP) form). The incorporation of the continuous boundary conditions $Le = 0$ is done weakly by using the penalty term $\Sigma \mathbf{e}$. The form of the penalty matrix Σ depend on the boundary operator L, i.e. $\Sigma = \Sigma(L)$. Roughly speaking, the penalty term must be accurate and lead to stability.

A strong implementation of the continuous boundary conditions is not clearly separable from the internal discretization (as is the weak boundary procedure). It essentially amounts to modifying the internal operator A directly in the boundary region and turn it into $\bar{A}$. To facilitate the comparison with weak boundary conditions we reformulate the strong implementation such that it directly mimics the weak one. That means that we add and subtract terms to obtain the matrix relation $\bar{A} = A - \Sigma$ where A is the same for both weak and strong boundary procedures.

2.4 Convergence to Steady-State of Solutions to the IVP

We discuss the requirements for obtaining steady-state solutions $\mathbf{e} = 0$ in (12). The sign and size of the real part of the eigenvalues to $\bar{A}$ is the crucial factor and they depend on the discretization of the spatial operator as well as on the boundary procedure. We will focus on the influence of the boundary procedure and keep the discretization of the spatial operator fixed.

The Matrix Exponential Method for Obtaining the Convergence Rate. Once we have the IVP on the form (12) we can write down the solution as

$$\mathbf{e} = \exp(-\bar{A}t) f, \quad t \geq 0, \tag{13}$$

using the definition of the matrix exponential. Next we assume that $\bar{A}$ is diagonalizable (the algebraic multiplicity of the eigenvalues λ_i is equal to the geometric multiplicity) such that $\bar{A} = X \Lambda X^{-1}$ where $\Lambda = diag(\lambda_i)$. That leads to

$$\mathbf{e} = X \exp(-\Lambda t) X^{-1} f, \quad t \geq 0, \tag{14}$$

and convergence to steady-state if all $\Re(\lambda_i) > 0$. The way to increase the convergence rate ($\min_i \Re(\lambda_i)$) to steady-state is to choose a boundary procedure such that $\min_i \Re(\lambda_i)$ lies as far as possible to the right in the complex plane. The sign convention is such that $\min_i \Re(\lambda_i) \to -\eta^*$ as the mesh is refined.

Remark: If there are multiple roots λ_i such that $\bar{A}$ is *not* diagonalizable (the algebraic multiplicity of the eigenvalues λ_i is larger than the geometric multiplicity), then we can use the Jordan decomposition of $\bar{A}$. The derivation above

remains almost the same and the conclusion that we have convergence to steady-state for $\min_i \Re(\lambda_i) > 0$ remains exactly the same.

The Energy Method for Deriving Boundary Procedures. To derive a suitable discretization, i.e. a matrix $\bar{A}$ with eigenvalues such that $\mathbf{e} \to 0$ as $t \to \infty$ as was discussed above, we use the energy method. Multiply (12) with $\mathbf{e}^T \bar{P}$ from the left, where $\bar{P} \equiv (P \otimes I_m)$. I_m is the $m \times m$ identity matrix and the $(N+1) \times (N+1)$ matrix P is symmetric and positive definite. The same holds for $\bar{P}$, which thus is a valid norm $\|\mathbf{e}\|_{\bar{P}}^2 = \mathbf{e}^T \bar{P} \mathbf{e}$. This leads to

$$(\|\mathbf{e}\|_{\bar{P}}^2)_t = -\mathbf{e}^T (\bar{P}\bar{A} + (\bar{P}\bar{A})^T)\mathbf{e}. \tag{15}$$

Recall that $\bar{A}$ includes both the interior approximation and the boundary procedure. For (15) to be energy stable, the right-hand side (RHS) must be bounded by $const.(\|\mathbf{e}\|_{\bar{P}}^2)$.

For a steady state to exist, we know from the previous section that the eigenvalues of $\bar{A}$ must have strictly positive real parts. We also know that if $\min_i \Re(\lambda_i)$ is large, the convergence will be fast. To manipulate the eigenvalues of $\bar{A}$ directly is almost impossible. However, by using the energy method, see (15), we can modify the matrix $(\bar{P}\bar{A})^S = (\bar{P}\bar{A} + (\bar{P}\bar{A})^T)/2$ in such a way that it modifies $\bar{A}$ and most likely enhances the convergence rate.

In most cases, the internal part of the discretization (A) is given by the class of method chosen. The choice of boundary approximation (Σ), on the other hand is crucial and will be considered in detail. As was mentioned above ($\Sigma = \Sigma(L)$) and it is therefore highly dependent on the boundary conditions in (4) but also on the specific numerical implementation technique. In this paper we focus on the latter.

Let us assume that we have chosen a particular boundary approximation such that $(\bar{P}\bar{A})^S$ is positive definite and leads to an energy decay in (15). Let λ and x be an eigenvalue and eigenvector to $\bar{A}$. By multiplying the relation $\bar{A}x = \lambda x$ from the left with $x^* \bar{P}$ and rearranging we get

$$\lambda = \Re(\lambda) + i\Im(\lambda), \quad \Re(\lambda) = \frac{x^*(\bar{P}\bar{A})^S x}{x^* \bar{P} x}, \quad \Im(\lambda) = \frac{x^*(\bar{P}\bar{A})^{AS} x}{x^* \bar{P} x}, \tag{16}$$

where $(\bar{P}\bar{A})^{AS} = (\bar{P}\bar{A} - (\bar{P}\bar{A})^T)/2$. Consequently, we can guarantee convergence to steady-state if we have an energy decay in (15). We can also see from (16) that by making $(\bar{P}\bar{A})^S = (\bar{P}\bar{A} + (\bar{P}\bar{A})^T)/2$ more positive definite, there is a possibility that we can increase the convergence rate $\min_i \Re(\lambda_i)$. For later reference we denote $\min_i \Re(\lambda_i) =$ the discrete decay rate.

The strategy in this paper for studying the influence of weak and strong solid wall boundary procedures can be summarized as follows. First we make sure that the continuous solid wall boundary conditions are well posed and lead to convergence to steady-state (otherwise further investigation is meaningless). Next we discretize and derive the different forms of $\bar{A} = A - \Sigma$. The inner scheme A is the same for all cases but Σ varies depending on the form of boundary

implementation we use. We will study three types, the weak, the strong and an intermediate form of implementation. After that we use the energy method and try to determine if the schemes are stable. Finally we compute the eigenvalues of $\bar{A}$ and see which procedure yields the largest value of $\min_i \Re(\lambda_i)$.

2.5 Numerical Results Using a Fully Nonlinear Finite Volume Solver

The analysis of our model problem, see details in the full article [1], indicate that the weak scheme lead to faster convergence to steady-state. To investigate if this result carries over to the fully nonlinear Navier-Stokes equations, we use the finite volume solver Edge, which is applicable on both structured and unstructured grids. The governing equations are integrated explicitly with a multistage local time-stepping Runge-Kutta scheme to steady-state and acceleration by FAS agglomeration multigrid can be used. There are numerous boundary conditions in Edge for walls, external boundaries and periodic boundaries. In all calculations we have used the same boundary conditions on all boundaries that are not solid walls. These boundary conditions are imposed weakly.

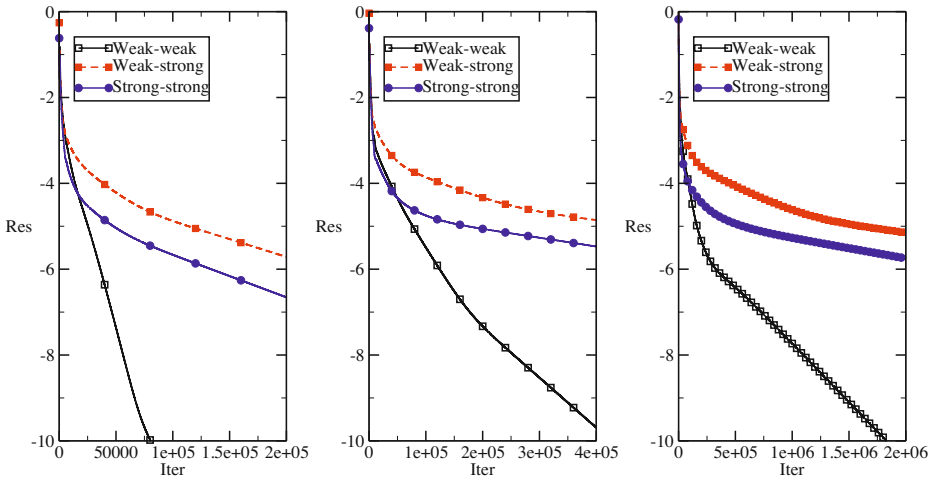
Remark: The eigenvalues must be located in the right-half plane also for local time-stepping schemes. Consider an IVP on the form $V_t + \bar{A}V = F$. Diagonalizing the IVP yields a system of scalar equations of the form $(w_j)_t + \lambda_j w_j = \tilde{f}_j$. The scalar equations all require eigenvalues λ_j with positive real part in order not to grow. This requirement is independent of whether one uses a time-accurate or local time-stepping scheme.

The flow conditions for the NACA0012 airfoil are as follows: Mach number $M = 0.8$, Reynolds number $Re = 500$ and $\alpha = 0^\circ$ as the angle of attack. These parameters yield a subsonic, low Reynolds flow which is assumed laminar.

Figure 1 displays the convergence to steady-state using the density residual. A 3-stage first order accurate Runge-Kutta time integrator with CFL=1.25 is used in combination with local time steps. No artificial dissipation is used. The weak solution converges faster than the solutions obtained by the other two schemes. The difference is more pronounced for the coarsest grid, which is what is expected from the linear results. Next we investigate the flow over a flat plate. For $x < 0$, where no plate exist, a symmetry condition is used, and for $x \geq 0$ the no-slip solid wall condition is imposed. The flat plate is 7 m long and has a Reynolds number of 10.5×10^6 based on that length. The flow is modelled as fully turbulent with an Explicit Algebraic Reynolds Stress Model (EARSM) formulation based on a two-equation $\kappa - \omega$ model.

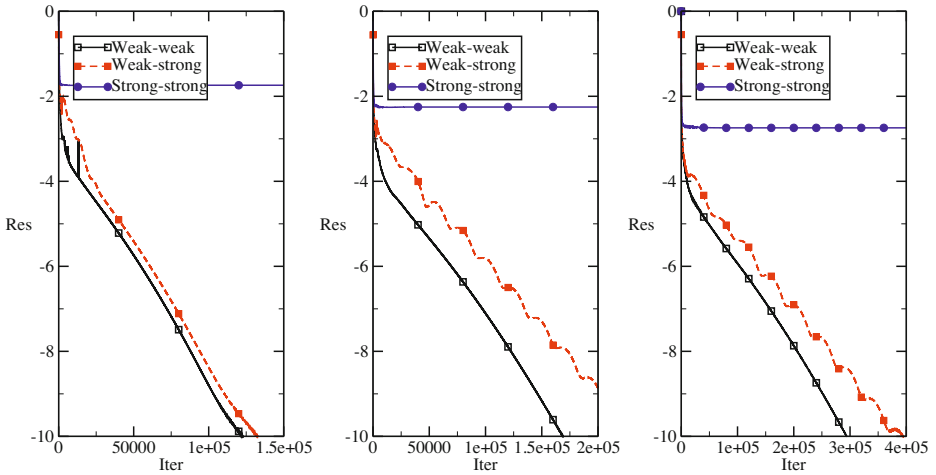
The convergence of the density residual is shown in Figure 2. The weak scheme converges slightly faster than the mixed scheme, while the strong scheme does not converge at all. This behavior is also consistent with the linear analysis which shoved that the strong scheme had an eigenvalue passing zero for the coarsest mesh.

Next we investigate the influence of solid wall boundary procedures on the multigrid acceleration technique. We consider the same flow field (the flat plate) again. Now neither the mixed nor the strong boundary procedure converges, see



(a) Coarsest grid (b) Medium grid (c) Finest grid

Fig. 1. The convergence of the density residual. NACA0012 at Mach number $M = 0.8$, Reynolds number $Re = 500$ and $\alpha = 0°$ as the angle of attack.

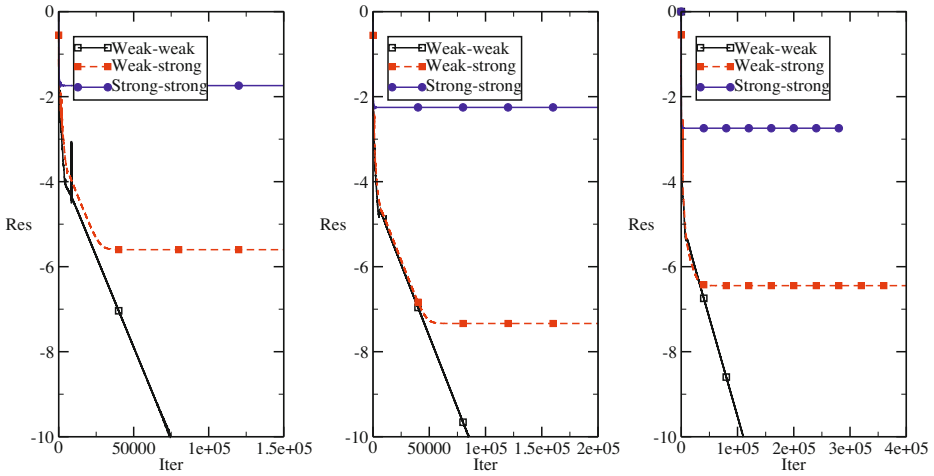


(a) Coarsest grid (b) Medium grid (c) Finest grid

Fig. 2. The convergence of the density residual. Flat plate at Mach number $M = 0.07$ and Reynolds number $Re = 10.5 \cdot 10^6$. Single grid calculation.

Figure 3. This is probably due to the fact that eigenvalues close to or equal to zero (for our analogous linear problem in the full article) gives a non-decaying energy.

Remark: The differences in steady state convergence between the boundary conditions are most evident without numerical dissipation. In other calculations the differences are smaller but still in favour of the weak boundary conditions.



(a) Coarsest grid (b) Medium grid (c) Finest grid

Fig. 3. The convergence of the density residual. Flat plate, $M = 0.07$, Reynolds number $Re = 10.5 \cdot 10^6$. Multigrid with three grid levels.

3 Functionals and Dual Problems

The solution of the governing equations might not be the output of primary interest in many CFD applications. Of equal, or even greater, importance is the computation of functionals from the solution. In general, a functional is defined as any map from a vector space V into the underlying scalar field $\mathcal{K}$. Every vector space has an associated vector space called its dual (or adjoint) space. The dual space is denoted by V^* and is defined as the space of all linear functionals $V \to \mathcal{K}$.

The adjoint, or dual, operator L^* of a linear operator L is the (unique) operator satisfying

$$(v, Lu)_V = (L^* v, u)_V, \tag{17}$$

where $(.,.)_V$ denotes the inner product on the space V. The study of linear functionals and dual spaces is the topic of functional analysis and additional preliminaries can be found in any functional analysis textbook.

In this section, we consider initial boundary value problems of the form

$$\begin{aligned} u_t + \mathcal{L}(u) &= F, \ x \in \Omega, \\ \mathcal{B}(u) &= g_\Gamma, \ x \in \Gamma \subseteq \partial\Omega, \\ u &= f, \ t = 0. \end{aligned} \tag{18}$$

For applications in CFD, a linear functional of interest usually represents the lift or drag on a solid body in a fluid, which is computed in terms of an integral

of the solution of (18). The functional can be represented in terms of an integral inner product as

$$J(u) = (g, u) = \int_{\Omega} g^T u d\Omega, \tag{19}$$

where g is a weight function. A main complication in CFD is that no physically relevant solutions have compact support in the computational domain. The dual operator is obtained through integration by parts which will introduce boundary terms that must be removed. The dual PDE has thus to be supplied with dual boundary conditions to close the system.

The associated dual problem has been extensively studied and used in the context of error control and adaptive mesh refinement as well as within optimization and control problems. In error control and mesh adaptation, the dual problem is derived and treated as a variational problem. In optimization and control problems, the dual problem is derived and treated as a sensitivity problem with respect to design parameters. In the end, the two different formulations yield the same dual problem. A similarity for the different areas of applications is that most of them are based on unstructured methods.

3.1 Quadrature Accuracy

Only recently was the study of duality introduced to structured methods, such as the SBP-SAT technique. Recall that the SBP operator was constructed to satisfy

$$(v_h, D_1 u_h)_h = u_h^T (E_N - E_0) v_h - (D_1 v_h, u_h)_h, \tag{20}$$

which mimics an integration property, rather than a differentiation property. While the differentiation properties of the SBP operator has been extensively studied and used, the integration properties of the matrix P have been much less explored. The integration properties of P was thoroughly investigated by Hicken and Zingg [5]. It was shown that the requirements on P to obtain an accurate SBP operator include, and extend, the Gregory formulas for quadrature rules using equidistant points. Two main results were proven and are restated here for convenience. The first theorem establishes the accuracy of P as an integration operator;

Theorem 1. *Let P be a full, restricted-full, or diagonal mass matrix from an SBP first-derivative operator $D_1 = P^{-1}Q$, which is a $2p$-order accurate approximation to the first derivative in the interior. Then the mass matrix P constitutes a $2p$-order accurate quadrature for integrands $u \in C^{2p}(\Omega)$.*

The second theorem extends the results to include discrete integrands computed from an SBP differentiation;

Theorem 2. *Let $D_1 = P^{-1}Q$ be a an SBP first derivative operator with a diagonal mass matrix P and $2p$-order interior accuracy. Then $(v_h, D_1 u_h)_h$ is a $2p$-order accurate approximation of (v, u_x).*

These theorems proved in summary that it is possible to retain the full order of accuracy when computing integrals from an SBP discretization, even with a diagonal P.

3.2 Dual Consistency

For initial boundary value problems (IBVPs), it is not sufficient to integrate the solution obtained by an SBP-SAT discretization using P to obtain a functional of $2p$-order accuracy. It was shown in [6] that an additional property of the discretization was required—the so called dual consistency property. The main result in [6] extends the results in [5] to include SBP-SAT solutions to IBVPs. Even though the solution u_h to an IBVP using SBP-SAT is accurate of order $p+1$ when using a diagonal P, any linear functional of u_h is accurate of order $2p$ when integrated using P, if the discretization is dual consistent.

As suggested by the name, dual consistency requires that the discretization of the primal problem is also a consistent approximation of the dual problem. In order to construct a dual consistent discretization, one first have to derive the dual problem and work with both the primal and dual problems simultaneously. To obtain the dual differential operator we consider the linear, or linearized, Cauchy problem,

$$\begin{aligned} u_t + Lu &= f, \qquad x \in \Omega, \\ u &= 0, \qquad t = 0, \\ J(u) &= (g, u) \end{aligned} \tag{21}$$

where $J(u)$ is a linear functional of interest. We seek a function θ, in some appropriate function space, such that

$$\int_0^T J(u)dt = \int_0^T (\theta, f)dt. \tag{22}$$

Using integration by parts, we can write

$$\begin{aligned} \int_0^T J(u)dt &= \int_0^T J(u)dt - \int_0^T (\theta, u_t + Lu - f)dt \\ &= \int_0^T (\theta_t - L^*\theta + g, u)dt - [(\theta, u)]_{t=T} + \int_0^T (\theta, f)dt \end{aligned} \tag{23}$$

and it is clear that $\theta = 0$ at $t = T$ is needed, and that θ has to satisfy the dual equation $-\theta_t + L^*\theta = g$. The time transform $\tau = T - t$ is usually introduced, and the dual Cauchy problem becomes

$$\begin{aligned} \theta_\tau + L^*\theta &= g, \, x \in \Omega, \\ \theta &= 0, \, \tau = 0. \end{aligned} \tag{24}$$

The situation is more complicated for IBVPs. Since the primal equation does not have compact support in general, the boundary terms resulting from the integration by parts procedure has to be properly taken care of by the homogeneous primal boundary conditions. The dual boundary conditions are defined as the minimal set of homogeneous conditions such that the boundary terms vanish after the homogeneous primal boundary conditions have been applied. Still, one needs to investigate the well-posedness of the dual equation with the

resulting dual boundary conditions. A well-posed set of boundary conditions for the primal problem does not necessary lead to a well-posed dual problem.

A discretization of a problem with a functional of interest can be written as

$$\begin{aligned} \tfrac{d}{dt}u_h + L_h u_h &= f, \\ J_h(u_h) &= (g, u_h)_h, \end{aligned} \tag{25}$$

where the entire spatial discretization, including the boundary conditions, has been collected into the discrete operator L_h. Recall that the inner product is defined as

$$(v_h, u_h)_h = v_h^T P u_h \tag{26}$$

in an SBP-SAT framework. The discrete adjoint operator L_h^* is defined, analogously to (17), as the unique operator satisfying

$$(v_h, L_h u_h)_h = (L_h^* v_h, u_h)_h. \tag{27}$$

The discrete adjoint operator can hence be explicitly computed, using (26) and (27), as

$$L_h^* = P^{-1} L_h^T P. \tag{28}$$

The discrete dual problem is obtained analogously to the continuous case by finding θ_h such that $\int_0^T J_h(u_h)dt = \int_0^T (\theta_h, f)dt$. Integration by parts and (28) gives

$$\begin{aligned} &\int_0^T J_h(u_h)dt = \int_0^T (g, u_h)_h dt - \int_0^T (\theta_h, \tfrac{d}{dt}u_h + L_h u_h - f)_h dt \\ &= \int_0^T (\tfrac{d}{dt}\theta_h - L_h^*\theta_h + g, u_h)_h dt - [(\theta_h, u_h)_h]_{t=T} + \int_0^T (\theta_h, f)_h dt \end{aligned} \tag{29}$$

and hence the θ_h has to satisfy the discrete dual problem

$$\begin{aligned} \tfrac{d}{d\tau}\theta_h + L_h^*\theta_h &= g, \\ \theta_h &= 0, \ \tau = 0, \end{aligned} \tag{30}$$

where $\tau = T - t$. Dual consistency can now be defined in terms of L_h^* and L^*;

Definition 1. *A discretization is called dual consistent if L_h^* is a consistent approximation of L^* and the continuous dual boundary conditions.*

The above definition is not specific for SBP-SAT discretizations. Any discretization which can be written in the form (25) is applicable. The SBP-SAT technique is particularly well-suited for this framework because of the well-defined inner product and operator form.

It is common, in optimization for example, that continuous and discrete adjoint methods are distinguished. This is because the discrete adjoint operator does not approximate the continuous adjoint operator and boundary conditions in general. In the SBP-SAT framework, the dual consistency property can allow for very efficient use of adjoint based techniques due to the unification of the continuous and discrete adjoints. SBP-SAT is not the only method which

offers consistency with the dual equations. It was shown that, for example, the discontinuous Galerkin method can also exhibit this property.

The dual consistency property can be easily exemplified using the model problem (31). Dual consistency does not depend on any data of the problem but only the differential operator and the form of the boundary conditions. We hence consider the inhomogeneous problem with homogeneous boundary and initial conditions,

$$\begin{aligned} u_t + \bar{u}u_x &= f, \qquad 0 \le x \le 1 \\ u(0,t) &= 0, \\ u(x,0) &= 0, \\ J(u) &= (g,u) \end{aligned} \tag{31}$$

where $J(u)$ is a linear functional of interest. We seek a function θ so that $\int_0^T J(u)dt = \int_0^T (\theta, f)dt$ and integration by parts gives

$$\begin{aligned} &\int_0^T J(u)dt = \int_0^T J(u)dt - \int_0^T (\theta, u_t + \bar{u}u_x - f)dt \\ &= \int_0^T (\theta_t + \bar{u}\theta_x + g, u)dt - \int_0^1 [\theta u]_{t=T}dx - \int_0^T [\bar{u}\theta u]_{x=1}dt + \int_0^T (\theta, f)dt. \end{aligned} \tag{32}$$

It is clear that θ has to satisfy the dual problem

$$\begin{aligned} \theta_\tau - \bar{u}\theta_x &= g, \quad 0 \le x \le 1, \\ \theta(1,\tau) &= 0, \\ \theta(x,0) &= 0, \end{aligned} \tag{33}$$

where we have introduced the time transform $\tau = T - t$.

The model problem (31) can be discretized as

$$\begin{aligned} \tfrac{d}{dt}u_h + \bar{u}D_1 u_h &= \sigma P^{-1}(e_0^T u_h - 0)e_0 + f, \\ J_h(u_h) &= (g, u_h)_h, \end{aligned} \tag{34}$$

and the parameter σ has to be determined so that the scheme is not only stable, but also a consistent approximation of the dual problem (33). It is convenient to rewrite (34) in operator form as

$$\frac{d}{dt}u_h + L_h u_h = f, \tag{35}$$

where the spatial discretization, including the boundary condition, is included in the operator

$$L_h = \bar{u}D_1 - \sigma P^{-1}E_0. \tag{36}$$

The discrete dual operator can be directly computed as

$$L_h^* = P^{-1}L_h^T P = -\bar{u}D_1 + \bar{u}P^{-1}E_N - (\sigma + \bar{u})P^{-1}E_0, \tag{37}$$

and it is seen that L_h^* imposes a boundary condition at $x = 0$, due to the last term in (37), unless $\sigma = -\bar{u}$. With $\sigma = -\bar{u}$, the discrete dual problem becomes

$$\frac{d}{d\tau}\theta_h - \bar{u}D_1\theta_h = -\bar{u}P^{-1}E_N\theta_h + g, \tag{38}$$

Table 1. Convergence rates q, and functional errors for the dual inconsistent and consistent schemes

5th-order ($2p = 8$)						
	$\sigma = -1/2$			$\sigma = -1$		
N	$q(u_h)$	$q(J_h(u_h))$	Error	$q(u_h)$	$q(J_h(u_h))$	Error
96	4.58	4.51	1.87e-05	5.14	8.20	7.54e-09
128	4.87	4.80	3.02e-06	5.34	7.96	2.71e-10
160	4.97	4.91	7.58e-07	5.41	8.02	2.74e-11
192	5.02	4.97	2.53e-07	5.44	8.06	4.58e-12
224	5.05	5.01	1.02e-07	5.46	8.21	1.05e-12
256	5.06	5.04	4.72e-08	5.46	8.62	2.97e-13

which is a consistent approximation of the dual problem (33). Since $\sigma = -\bar{u}$ does not contradict the stability condition ($\sigma \leq -\bar{u}/2$), the scheme is both stable and dual consistent. In Table 1 we show the convergence rates q for the solution and the functionals, together with the functional error, using the dual inconsistent and consistent schemes. As we can see from Table 1, the convergence rate for the linear functional increases from $p + 1$ to $2p$ when using the dual consistent discretization. Also notice that dual consistency is merely a choice of parameters. The solution of the dual problem is never required and hence the increased rate of convergence for linear functionals comes at no extra computational cost.

3.3 Boundary Conditions

The theory of dual consistency is not only useful for deriving schemes with superconvergent integral functionals. By simultaneously considering the primal and dual equals, new boundary conditions can be derived. As an example, we will consider the linearized and symmetrized compressible Navier–Stokes and Euler equals in two space dimensions:

$$U_t + AU_x + BU_y = \varepsilon((C_{11}U_x + C_{12}U_y)_x + (C_{21}U_x + C_{22}U_y)_y). \tag{39}$$

The Euler equations are obtained by setting $\varepsilon = 0$. This seemingly small change, to have $\varepsilon = 0$ or not, have a huge impact on the boundary conditions. Lets consider the unit square for simplicity.

A commonly used far-field boundary condition is of the form

$$\begin{aligned} H_W U - \varepsilon(C_{11}U_x + C_{12}U_y) = 0,\ H_E U + \varepsilon(C_{11}U_x + C_{12}U_y) = 0, \\ H_S U - \varepsilon(C_{21}U_x + C_{22}U_y) = 0,\ H_N U + \varepsilon(C_{21}U_x + C_{22}U_y) = 0, \end{aligned} \tag{40}$$

where the matrices $H_{W,E,S,N}$ (W,E,S,N refers to the west, east, south, and north boundaries, respectively) have to be construed for well-posedness. A well-known problem is that for subsonic outflow boundaries, matrices that give a well-posed problem for one of the equations, give an ill-posed problem for the other. One

can hence not switch between the Navier–Stokes and Euler equations by simply putting $\varepsilon = 0$ or not. When attempting to remedy this problem, one has to put up general matrices with general coefficients and try to determine them to get the properties one wants. In two dimensions, the matrices are 4×4, each with 16 undetermined coefficients, and the parameter space simply becomes too large.

More equations are needed to deal with the amount of undetermined parameters. By considering not only well-posedness of the primal equations, but also of the dual equations, we get exactly what we need. To derive the dual Navier–Stokes equations we consider (39) in the form

$$\begin{aligned} U_t + LU &= F, & (x,y) &\in \Omega, \\ BU &= 0, & (x,y) &\in \Gamma \subseteq \partial\Omega, \\ U &= 0, & t &= 0, \\ J(U) &= (G,U). \end{aligned} \tag{41}$$

In (41), $J(U)$ is a linear integral functional with a weight function G and B implements the boundary conditions in (40). The right-hand side F may be identically zero, but a symbol is needed to perform integration by parts when deriving the dual problem. The differential operator L is given by

$$L = A\frac{\partial}{\partial x} + B\frac{\partial}{\partial y} - \varepsilon\left(\left(C_{11}\frac{\partial}{\partial x} + C_{12}\frac{\partial}{\partial y}\right)\frac{\partial}{\partial x} + \left(C_{21}\frac{\partial}{\partial x} + C_{22}\frac{\partial}{\partial y}\right)\frac{\partial}{\partial y}\right). \tag{42}$$

We seek a function $\Theta = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ such that $\int_0^T J(U)dt = \int_0^T (\Theta, F)$, and we get by the Gauss–Green formula

$$\begin{aligned} &\int_0^T J(U)dt = \int_0^T (G,U) - \int_0^T (\Theta, U_t + LU - F)dt \\ &= \int_0^T (\Theta, F)dt + \int_0^T (\Theta_t - L^*\Theta + G, U)dt \\ &+ \int_0^T \int_W \Theta^T(AU - \varepsilon(C_{11}U_x + C_{12}U_y)dydt + \varepsilon\int_0^T \int_W (\Theta_x^T C_{11} + \Theta_y^T C_{12})U dydt \\ &- \int_0^T \int_E \Theta^T(AU - \varepsilon(C_{11}U_x + C_{12}U_y)dydt - \varepsilon\int_0^T \int_E (\Theta_x^T C_{11} + \Theta_y^T C_{12})U dydt \\ &+ \int_0^T \int_S \Theta^T(BU - \varepsilon(C_{21}U_x + C_{22}U_y)dxdt + \varepsilon\int_0^T \int_S (\Theta_x^T C_{21} + \Theta_y^T C_{22})U dxdt \\ &- \int_0^T \int_N \Theta^T(BU - \varepsilon(C_{21}U_x + C_{22}U_y)dxdt - \varepsilon\int_0^T \int_N (\Theta_x^T C_{21} + \Theta_y^T C_{22})U dxdt \\ &- \int_\Omega [\Theta^T U]_{t=T} d\Omega, \end{aligned} \tag{43}$$

where $\int_{W,E,S,N}$ denotes integration over the west, east, south, and north boundary, respectively. The dual operator, L^*, is given by

$$L^* = -A\frac{\partial}{\partial x} - B\frac{\partial}{\partial y} - \varepsilon\left(\left(C_{11}\frac{\partial}{\partial x} + C_{12}\frac{\partial}{\partial y}\right)\frac{\partial}{\partial x} + \left(C_{21}\frac{\partial}{\partial x} + C_{22}\frac{\partial}{\partial y}\right)\frac{\partial}{\partial y}\right), \tag{44}$$

and we obtain the dual boundary conditions by applying the homogeneous primal boundary conditions to the boundary integral terms.

By using (40), we can write (43) as

$$\begin{aligned}\int_0^T J(U)dt = &\int_0^T (\Theta, F)dt + \int_0^T (\Theta_t - L^*\Theta + G, U)dt \\ &+ \int_0^T \int_W U^T((A - H_W^T)\Theta + \varepsilon(C_{11}\Theta_x + C_{12}\Theta_y))dydt \\ &- \int_0^T \int_E U^T((A + H_E^T)\Theta + \varepsilon(C_{11}\Theta_x + C_{12}\Theta_y))dydt \\ &+ \int_0^T \int_S U^T((B - H_S^T)\Theta + \varepsilon(C_{21}\Theta_x + C_{22}\Theta_y))dxdt \\ &- \int_0^T \int_N U^T((B + H_N^T)\Theta + \varepsilon(C_{21}\Theta_x + C_{22}\Theta_y))dxdt.\end{aligned} \tag{45}$$

We introduce the dual time variable, $\tau = T - t$, and the function Θ has to satisfy the dual Navier–Stokes equations

$$\Theta_\tau - A\Theta_x - B\Theta_y = \varepsilon((C_{11}\Theta_x + C_{12}\Theta_y)_x + (C_{21}\Theta_x + C_{22}\Theta_y)_y) + G \tag{46}$$

with the dual boundary conditions

$$\begin{aligned}(A - H_W^T)\Theta + \varepsilon(C_{11}\Theta_x + C_{12}\Theta_y) = 0,\ (A + H_E^T)\Theta + \varepsilon(C_{11}\Theta_x + C_{12}\Theta_y) = 0, \\ (B - H_S^T)\Theta + \varepsilon(C_{21}\Theta_x + C_{22}\Theta_y) = 0,\ (B + H_N^T)\Theta + \varepsilon(C_{21}\Theta_x + C_{22}\Theta_y) = 0,\end{aligned} \tag{47}$$

together with a homogeneous initial condition at $\tau = 0$.

An interesting property is that the primal equations (39) with the primal boundary conditions (40), and the dual equations (46) with the dual boundary conditions (47), share the same energy estimate. Let Φ be either the primal variable U or the dual variable Θ. Then

$$\begin{aligned}&||\Phi||_{t,\tau}^2 + 2\varepsilon\int_\Omega (\nabla\Phi^T)C(\nabla\Phi)^T d\Omega = \\ &-\int_W \Phi^T(-A + H_W + H_W^T)\Phi dy - \int_E \Phi^T(A + H_E + H_E^T)\Phi dy \\ &-\int_S \Phi^T(-B + H_S + H_S^T)\Phi dx - \int_N \Phi^T(B + H_N + H_N^T)\Phi dx,\end{aligned} \tag{48}$$

where C is a positive semi-definite matrix. It is clear that $H_{W,E,S,N}$ must be chosen such that

$$\begin{aligned}-A + H_W + H_W^T \geq 0,\ A + H_E + H_E^T \geq 0, \\ -B + H_S + H_S^T \geq 0,\ \ B + H_N + H_N^T \geq 0,\end{aligned} \tag{49}$$

in order to obtain a bounded energy growth and hence an energy estimate for both the primal and dual Navier–Stokes and Euler equations. Energy estimates are, however, not sufficient. It is also required that the correct number of boundary conditions are imposed to get well-posed problems, see Table 2. An operator which have an energy estimate with a minimal number of boundary conditions, such that existence is guaranteed, is called maximally semi-bounded and leads directly to well-posedness.

Table 2. Number of boundary conditions required for the primal and dual Navier–Stokes and Euler equations under subsonic conditions with positive velocity components

	Number of b.c.			
Boundary	West	East	South	North
Primal Navier–Stokes	4	3	4	3
Dual Navier–Stokes	3	4	3	4
Primal Euler	3	1	3	1
Dual Euler	1	3	1	3

For subsonic outflow boundaries, we chose to construct H_E such that the pressure is specified for the primal Euler equations. In this case, H_E has to satisfy:

1. The top row of H_E is zero
2. The top row of $A + H_E^T$ is non-zero
3. $\text{rank}(H_E) = 1$
4. $\text{rank}(A + H_E^T) = 3$
5. $A + H_E + H_E^T \geq 0$

Requirements (1) and (2) set the correct number of boundary conditions for the primal and dual Navier-Stokes equations, requirements (3) and (4) set the correct number of boundary conditions for the primal and dual Euler equations, and requirement (5) gives energy estimates of both the primal and dual Navier-Stokes and Euler equations, respectively. The above requirements make the matrix H_E uniquely determined. From potentially 16 undetermined parameters, the requirements of primal and dual well-posedness gives unique solutions to all parameters. The result is summarized in

Theorem 3. *Let the matrix H_E be given by*

$$H_E = \frac{\bar{u}^2 - \bar{c}^2}{\bar{c}\sqrt{\gamma}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \sqrt{\gamma - 1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{50}$$

Then the boundary conditions

$$H_E\Phi + \varepsilon(C_{11}\Phi_x + C_{12}\Phi_y) = 0 \tag{51}$$

are well-posed subsonic outflow conditions for the primal and dual Navier–Stokes and, for $\varepsilon = 0$, the primal and dual Euler equations.

The details of the proof, and the remaining boundaries together with the dual consistent SBP-SAT discretizations, can be found in [4].

4 Multigrid for Higher Order Accurate Finite Difference Schemes

Multigrid is well established for industrial second order accurate CFD codes like Edge and other codes for structured grids [7]. For higher order accurate finite difference CFD codes, however, multigrid has so far not been used for convergence acceleration. We have taken the first steps towards using multigrid in a higher order accurate finite difference solver. We have analyzed the eigenvalues to the iteration matrix using multigrid for a model problem of the linearized one-dimensional Navier-Stokes equations. The spatial operators considered are exactly the operators based on the SBP-SAT technique.

The Navier-Stokes equations in one dimension can be written as

$$u_t + F_x = 0; F = F^I - \epsilon F^V \tag{52}$$

where superscripts I, V denotes the inviscid and viscous fluxes, respectively and u is the vector of the three conservative variables. In order to study the eigenvalues of the spectra we consider the non-dimensional and linearized Navier-Stokes equations

$$w_t + A_w w_x - B w_{xx} = 0 \tag{53}$$

where w contains the primitive variables. Matrices A_w, B are constant matrices and contain averaged values from which the linearization is made.

Multigrid is applied to accelerate the convergence to steady state of a linear problem formulated as

$$w_t + Lw = 0 \tag{54}$$

where the spatial operator L contains the entire discretization in one space dimension x including boundary conditions,

$$L(w) = (\xi_x \otimes I_3)\left((D_\xi \otimes A_w)w + (D_\xi \otimes I_3)(\xi_x \otimes I_3)(D_\xi \otimes B)w - Pen\right), \tag{55}$$

where ξ_x is the metric operator, I_3 is a 3×3 identify matrix, $D_\xi = P^{-1}Q$ is the first difference SBP operator and Pen contains the boundary penalty terms.

In non-linear multigrid the same equations are solved and iterated on coarser grids but with a forcing term on the right hand side. We assume $N_{grid} \geq 1$ multigrid levels and introduce index l to denote the current grid level and denote

$l = 1$ the finest level with the original grid and $l = N_{grid}$ the coarsest grid. Then the following equations are solved on the different grid levels:

$$\begin{array}{ll} \textit{Level } 1 & : w_{lt} + L_l w_l = 0 \\ \textit{Level } l, 1 < l \leq N_{grid} & : w_{lt} + L_l w_l = F_l = L_l r_{l-1}^l w_{l-1} - r_{l-1}^l L_{l-1} w_{l-1} \end{array} \tag{56}$$

where F_l is the forcing function and is constant when iterating on the coarse grids. r_{l-1}^l is the restriction operator used to transfer information from fine to coarse grid between levels $l-1$, l. The coarse grid solution is iterated one time level after which the solution difference is interpolated, or prolongated, back to the finer level using a prolongation operator p_l^{l-1}.

The solution is iterated in time using a time integrator which is denoted a smoother. Here we consider an explicit m-stage Runge-Kutta time integrator which integrates (54) as

$$\begin{array}{ll} w_l^0 & = w_l^n \\ w_l^1 & = w_l^0 - \alpha_1 \Delta t (L_l w_l^0 - F_l) \\ & \vdots \\ w_l^m & = w_l^0 - \alpha_m \Delta t (L_l w_l^{m-1} - F_l) \\ w_l^{n+1} & = w_l^m \end{array} \tag{57}$$

where $F_1 = 0$ and Δt the time step. The smoother S_l can then be formulated as

$$w_l^{n+1} = S_l w_l^n + (I - S_l) L_l^{-1} F_l \tag{58}$$

where w_l^n, w_l^{n+1} are the solution vectors at two consecutive time levels $n, n+1$ and where

$$S_l = I + \beta_1 \Delta t L_l + \beta_2 \Delta t^2 L_l^2 + + \beta_3 \Delta t^3 L_l^3 + \dots \tag{59}$$

and $(\beta_1, \beta_2, \beta_3, \dots) = (\alpha_m, \alpha_m \alpha_{m-1}, \alpha_m \alpha_{m-1} \alpha_{m-2}, \dots)$.

We want to compare the eigenvalues of the fine grid iteration matrix M_1, sometimes denoted amplification matrix, where

$$w_1^{n+1} = M_1 w_1^n. \tag{60}$$

Stability requires that all eigenvalues satisfy $\mid M_1 \mid \leq 1$. The iteration matrix for multigrid can be written as

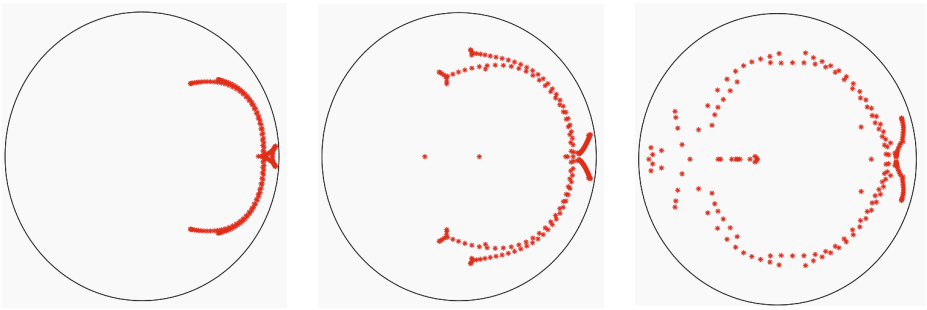
$$\begin{array}{ll} \textit{Level } l, 1 \leq l < N_{grid} & : M_l = S_l^{\nu_1} \left(I - p_{l+1}^l (I - M_{l+1}) L_{l+1}^{-1} r_l^{l+1} \right) S_l^{\nu_2} \\ \textit{Level } N_{grid} & : M_{N_{grid}} = S_{N_{grid}} \end{array} \tag{61}$$

where ν_1, ν_2 are the number of pre- and post smoothing steps. A single smoothing step is performed on the coarsest grid.

In the example below we apply wall boundary conditions at one end ($x = 0$) and characteristic outflow conditions at the other end ($x = 1$) with corresponding penalty terms with stable values of penalty parameters. We use a first order

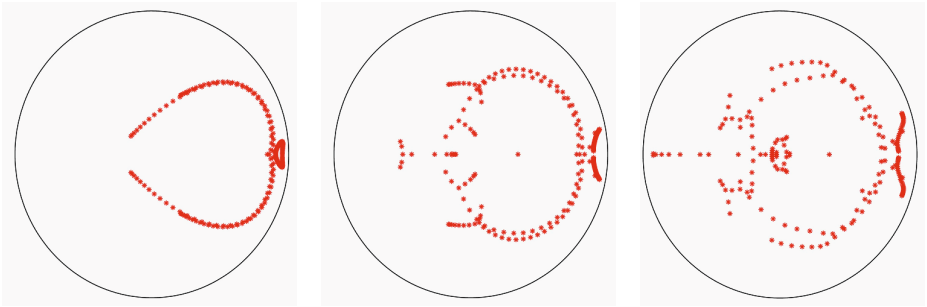
accurate 3-stage Runge-Kutta time integrator with $(\alpha_1, \alpha_2, \alpha_3) = (2/3, 2/3, 1)$, and we choose inviscid flow conditions, $\epsilon = 0$, at a low Mach number $M_\infty = 0.1$. The grid is stretched towards the wall reflected in the metric operator ξ_x in (55).

In Figures 4 and 5 the eigenvalues corresponding to a second and third order (fourth order in the interior and second order on the boundaries) accurate discretization are shown using $1 - 3$ grid levels. The eigenvalues are within the stability limits with both discretizations. The stability depends on the choice of grid transfer operators (restriction and prolongation) though which requires further analysis investigations.



(a) $N_{grid} = 1$ (b) $N_{grid} = 2$ (c) $N_{grid} = 3$

Fig. 4. Eigenvalues of iteration matrix M_1 using a second order SBP discretization



(a) $N_{grid} = 1$ (b) $N_{grid} = 2$ (c) $N_{grid} = 3$

Fig. 5. Eigenvalues of iteration matrix M_1 using a third order SBP discretization

5 Conclusions

Two completely new developments for increased performance of the high order and stable SBP-SAT finite difference technique has been described. The use of weak boundary conditions leads to increased convergence rate to steady state both for single and multigrid calculations. Dual consistent schemes produces

superconvergent functionals such as lift and drag and increases accuracy. The dual consistency comes at no extra cost.

The first steps have been taken to introduce multigrid in a higher order accurate finite difference scheme by analyzing eigenvalues of a one-dimensional model problem of the linearized Euler equations. The results look promising and there is no indication that multigrid should not work as convergence accelerator in a higher order accurate finite difference code, further investigations are required.

Finally we mention that the obvious benefits shown in this chapter of using weak boundary procedures and dual consistent schemes are also valid for other computational techniques techniques such discontinuous Galerkin, finite continuous finite elements, spectral elements etc. Only technical implementation issues differ, but not the fundamental gains that can be obtained.

References

1. Nordström, J., Eriksson, S., Eliasson, P.: Weak and Strong Wall Boundary Procedures and Convergence to Steady-State of the Navier–Stokes Equations. Journal of Computational Physics 231, 4867–4884 (2012)
2. Berg, J., Nordström, J.: Superconvergent functional output for time-dependent problems using finite differences on summation-by-parts form. Journal of Computational Physics 231, 6846–6860 (2012)
3. Berg, J., Nordström, J.: On the impact of boundary conditions on dual consistent finite difference discretizations. Journal of Computational Physics 236, 41–55 (2013)
4. Berg, J., Nordström, J.: Duality based boundary conditions and dual consistent finite difference discretizations of the Navier–Stokes and Euler equations. Journal of Computational Physics 259, 135–153 (2014)
5. Hicken, J.E., Zingg, D.W.: Summation-by-parts operators and high-order quadrature. Journal of Computational and Applied Mathematics 237, 111–125 (2013)
6. Hicken, J.E., Zingg, D.W.: Superconvergent Functional Estimates from Summation-By-Parts Finite-Difference Discretizations. SIAM Journal on Scientific Computing 33, 893–922 (2011)
7. Wesseling, P., Oosterlee, C.W.: Geometric multigrid with applications to computational fluid dynamics. Journal of Computational and Applied Mathematics 128, 311–334 (2001)

Higher-Order RANS and DES in an Industrial Stabilized Finite Element Code

F. Chalot, F. Dagrau, M. Mallet, P.E. Normand, and P. Yser

DASSAULT AVIATION, DGT/DTIAE,
78, quai Marcel Dassault, Cedex 300,
92552 Saint-Cloud Cedex, France
frederic.chalot@dassault-aviation.com

Abstract. This chapter describes the contribution of Dassault Aviation to the IDIHOM Project. It focuses on the extension of its stabilized finite element Navier-Stokes code to higher-order elements and more specifically on industrial RANS and DES applications.

Keywords: higher-order stabilized finite elements, SUPG, GLS, compressible Navier-Stokes, RANS turbulence modeling, Large Eddy Simulation, Detached Eddy Simulation, industrial applications

1 Introduction

Although Dassault Aviation (also referred to as DASSAV) started from the beginning with unstructured meshes and a Navier-Stokes code based on a finite element formulation, the claim that finite elements can fairly effortlessly and in a straightforward manner go high in order was never fully exploited. Linear elements which yield second-order accuracy are still currently used for all industrial Navier-Stokes calculations [4], [8], [9].

The ADIGMA Project was a first step towards the use of higher-order methods [11]. The participation of Dassault Aviation concentrated on the assessment of higher-order stabilized finite elements with 2-D test cases which covered a broad range of flow regimes.

The European project **IDIHOM** (Industrialisation of High-Order Methods – A Top-Down Approach) aimed at bringing the higher-order capability to more industrial applications. Contrary to most partners in the Project, Dassault Aviation uses a stabilized finite element formulation based on *continuous* shape functions. It is more related to residual distribution schemes, although the method has a pure finite element foundation. We focused on its extension to higher-order elements and more specifically on industrial RANS and DES applications.

2 Higher-Order Stabilized Finite Element Schemes for the RANS Equations

We present our numerical method in the following sections and highlight the adjustments required by higher-order elements.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_23

2.1 General Description of Our Flow Solver

Dassault Aviation's Navier-Stokes code, called `AeTher`, uses a finite element approach, based on a symmetric form of the equations written in terms of entropy variables. The advantages of this change of variables are numerous: in addition to the strong mathematical and numerical coherence they provide (dimensionally correct dot product, symmetric operators with positivity properties, efficient preconditioning), entropy variables yield further improvements over the usual conservation variables, in particular in the context of chemically reacting flows [3], [4].

The code can handle the unstructured mixture of numerous types of elements (triangles and quadrilaterals in 2-D; tetrahedra, bricks, and prisms in 3-D). In practice mostly linear triangular and tetrahedral meshes are used.

Different one- and two-equation Reynolds-averaged turbulence models are available: Spalart-Allmaras, K-ε, K-ω, K-ℓ, K-KL... These models are either integrated down to the wall, use a two-layer approach with a low-Reynolds modeling of the near wall region, or adopt a wall function treatment of the boundary layer. More advanced RANS models, such as EARSM and RSM, and extensions to LES and DES are also available (see [6], [7], [9], and [10]).

Convergence to steady state of the compressible Navier-Stokes equations is achieved through a fully-implicit iterative time-marching procedure based on the GMRES algorithm with nodal block-diagonal or incomplete LDU preconditioning [21].

The code has been successfully ported on many computer architectures. It is fully vectorized and parallelized using the MPI message passing library [5].

2.2 The Symmetric Navier-Stokes Equations

As a starting point, we consider the compressible Navier-Stokes equations written in conservative form:

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i}^{\text{adv}} = \boldsymbol{F}_{i,i}^{\text{diff}} \tag{1}$$

where $\boldsymbol{U}$ is the vector of conservative variables; $\boldsymbol{F}_i^{\text{adv}}$ and $\boldsymbol{F}_i^{\text{diff}}$ are, respectively, the advective and the diffusive fluxes in the i^{th}-direction. Inferior commas denote partial differentiation and repeated indices indicate summation.

Equation (1) can be rewritten in quasi-linear form:

$$\boldsymbol{U}_{,t} + \boldsymbol{A}_i \boldsymbol{U}_{,i} = (\boldsymbol{K}_{ij} \boldsymbol{U}_{,j})_{,i} \tag{2}$$

where $\boldsymbol{A}_i = \boldsymbol{F}_{i,\boldsymbol{U}}^{\text{adv}}$ is the i^{th} advective Jacobian matrix, and $\boldsymbol{K} = [\boldsymbol{K}_{ij}]$ is the diffusivity matrix, defined by $\boldsymbol{F}_i^{\text{diff}} = \boldsymbol{K}_{ij} \boldsymbol{U}_{,j}$. The $\boldsymbol{A}_i$'s and $\boldsymbol{K}$ do not possess any particular property of symmetry or positiveness.

We now introduce a new set of variables,

$$\boldsymbol{V}^T = \frac{\partial \mathcal{H}}{\partial \boldsymbol{U}}$$

where $\mathcal{H}$ is the generalized entropy function given by

$$\mathcal{H} = \mathcal{H}(\boldsymbol{U}) = -\rho s$$

and s is the thermodynamic entropy per unit mass. Under the change of variables $\boldsymbol{U} \mapsto \boldsymbol{V}$, (2) becomes:

$$\widetilde{\boldsymbol{A}}_0 \boldsymbol{V}_{,t} + \widetilde{\boldsymbol{A}}_i \boldsymbol{V}_{,i} = (\widetilde{\boldsymbol{K}}_{ij} \boldsymbol{V}_{,j})_{,i} \tag{3}$$

where

$$\begin{aligned} \widetilde{\boldsymbol{A}}_0 &= \boldsymbol{U}_{,\boldsymbol{V}} \\ \widetilde{\boldsymbol{A}}_i &= \boldsymbol{A}_i \widetilde{\boldsymbol{A}}_0 \\ \widetilde{\boldsymbol{K}}_{ij} &= \boldsymbol{K}_{ij} \widetilde{\boldsymbol{A}}_0. \end{aligned}$$

The Riemannian metric tensor $\widetilde{\boldsymbol{A}}_0$ is symmetric positive-definite; the $\widetilde{\boldsymbol{A}}_i$'s are symmetric; and $\widetilde{\boldsymbol{K}} = [\widetilde{\boldsymbol{K}}_{ij}]$ is symmetric positive-semidefinite. In view of these properties, (3) is referred to as a symmetric advective-diffusive system.

For a general divariant gas, the vector of so-called (physical) entropy variables, $\boldsymbol{V}$, reads

$$\boldsymbol{V} = \frac{1}{T} \begin{Bmatrix} \mu - |\boldsymbol{u}|^2/2 \\ \boldsymbol{u} \\ -1 \end{Bmatrix}$$

where $\mu = e + pv - Ts$ is the chemical potential per unit mass; $v = 1/\rho$ is the specific volume. More complex equations of state are treated in [2]. We would like to stress the formal similarity between the conservation variables $\boldsymbol{U}$ and the entropy variables $\boldsymbol{V}$, which can be made more apparent if we write the conservation variables in the following form:

$$\boldsymbol{U} = \frac{1}{v} \begin{Bmatrix} 1 \\ \boldsymbol{u} \\ e + |\boldsymbol{u}|^2/2 \end{Bmatrix}$$

where $v = 1/\rho$ is the specific volume.

Taking the dot product of (3) with the vector $\boldsymbol{V}$ yields the Clausius-Duhem inequality, which constitutes the basic nonlinear stability condition for the solutions of (3). This fundamental property is inherited by appropriately defined finite element methods, such as the one described in the next section.

2.3 The Galerkin/Least-Squares Formulation

Originally introduced by Hughes and Johnson, the Galerkin/least-squares (GLS) formulation is a full space-time finite element technique employing the discontinuous Galerkin method in time [1], [22]. The least-squares operator ensures good stability characteristics while retaining a high level of accuracy. The local

control of the solution in the vicinity of sharp gradients is further enhanced by the use of a nonlinear discontinuity-capturing operator.

Let Ω be the spatial domain of interest and Γ its boundary. The semi-discrete Galerkin/least-squares variational problem can be stated as:

Find $\boldsymbol{V}^h \in \mathcal{S}^h$ (trial function space), such that for all $\boldsymbol{W}^h \in \mathcal{V}^h$ (weighting function space), the following equation holds:

$$\begin{aligned}\int_\Omega \Big(W^h \cdot \boldsymbol{U}_{,t}(\boldsymbol{V}^h) - \boldsymbol{W}^h_{,i} \cdot \boldsymbol{F}^{\mathrm{adv}}_i(\boldsymbol{V}^h) + \boldsymbol{W}^h_{,i} \cdot \widetilde{\boldsymbol{K}}_{ij} \boldsymbol{V}^h_{,j}\Big)\, d\Omega \\ + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \Big(\mathcal{L}\boldsymbol{W}^h\Big) \cdot \boldsymbol{\tau}\Big(\mathcal{L}\boldsymbol{V}^h\Big)\, d\Omega \\ + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \nu^h g^{ij} \boldsymbol{W}^h_{,i} \cdot \widetilde{\boldsymbol{A}}_0 \boldsymbol{V}^h_{,j}\, d\Omega \\ = \int_\Gamma \boldsymbol{W}^h \cdot \Big(- \boldsymbol{F}^{\mathrm{adv}}_i(\boldsymbol{V}^h) + \boldsymbol{F}^{\mathrm{diff}}_i(\boldsymbol{V}^h)\Big) n_i\, d\Gamma. \end{aligned} \tag{4}$$

The first and last integrals of (4) represent the Galerkin formulation written in integrated-by-parts form to ensure conservation under reduced quadrature integration.

The second integral constitutes the least-squares operator where $\mathcal{L}$ is defined as

$$\mathcal{L} = \widetilde{\boldsymbol{A}}_0 \frac{\partial}{\partial t} + \widetilde{\boldsymbol{A}}_i \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i}\Big(\widetilde{\boldsymbol{K}}_{ij} \frac{\partial}{\partial x_j}\Big). \tag{5}$$

$\boldsymbol{\tau}$ is a symmetric time-scale matrix for which definitions can be found in [22].

The third integral is the nonlinear discontinuity-capturing operator, which is designed to control oscillations about discontinuities, without upsetting higher-order accuracy in smooth regions. g^{ij} is the contravariant metric tensor defined by

$$[g^{ij}] = [\boldsymbol{\xi}_{,i} \cdot \boldsymbol{\xi}_{,j}]^{-1}$$

where $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})$ is the inverse isoparametric element mapping and ν^h is a scalar-valued homogeneous function of the residual $\mathcal{L}\boldsymbol{V}^h$. The discontinuity capturing factor ν^h used for linear elements is an extension of that introduced by Hughes, Mallet, and Shakib [17], [22].

A key ingredient to the formulation is its consistency: the exact solution of (1) satisfies the variational formulation (4). This constitutes an essential property in order to attain higher-order spatial convergence.

2.4 Extension to Higher-Order Elements

In principle everything is contained in the weighted residual given by Eq. (4). There is no new term to code, no interpolation technique specific to higher order to derive: everything is already there. We just have to compute the integrals of (4), taking into account the new higher-order shape functions.

The volume and surface integrals are numerically evaluated with quadrature rules. All is needed is the values of the shape functions (and their gradients) at the integration points. Higher-order functions only require more precise integration rules. In 3-D, we use 4-, 10-, and 24-point rules, respectively for linear, quadratic, and cubic tets.

For a given number of degrees of freedom, higher-order meshes contain much fewer elements than P1 meshes. In 3-D the ratio is 1/8th for quadratic triangles, and 1/27th for cubic. Although more integration points are required, the higher-order computation of (4) is actually cheaper. The extra cost comes from the implicit linear system which possesses a much larger bandwidth.

In fact, one single term in the weighted residual must be specially treated in the context of higher-order elements for the Navier-Stokes equations. The last term in (5) vanishes to zero for linear elements. It appears in the second integral of (4). This term must be computed with higher-degree shape and test functions in order to preserve consistency. In practice, it is evaluated using an L_2-projection.

One-dimensional studies showed that there was no significant differences between SUPG and Galerkin/least-squares. We have chosen to concentrate solely on SUPG which is easier to implement.

As a final remark, we want to stress the fact that whatever the order of the elements, all operations remain local (viz. at the element level). Consequently higher-order elements engender no implicitation nor parallelization issue [5].

2.5 Isoparametric Meshes with Curved Boundaries

Since ADIGMA, we have made the seemingly obvious choice of higher-order **isoparametric** elements [11]. One of the advantages of these elements, besides the higher-order shape functions, is the use of higher-order polynomials to represent curved boundaries. They only ensure C^0 continuity across elements, but locate all the nodes on the actual surface.

All our higher-order meshes are obtained by adding nodes to a coarse initial P1 mesh. Higher-order degrees of freedom are added regularly on the faces, edges and possibly inside each P1 coarse element. This is illustrated for 2-D P1 and P2 elements in Figure 1. The coarse P1 "skeleton" mesh is at the top; the P2 and the corresponding nested P1 meshes are respectively in the right and left columns. The "natural" option is presented in the second line. All additional degrees of freedom are located in the middle of the edges. This configuration does not preserve the geometric progression of points in the boundary layer, but all edges remain straight lines. Three alternatives are presented next: they all preserve the progression in the layers and straight edges, but display uneven distributions of points that any experienced CFD user would reject. A fourth choice is discussed in Section 4.

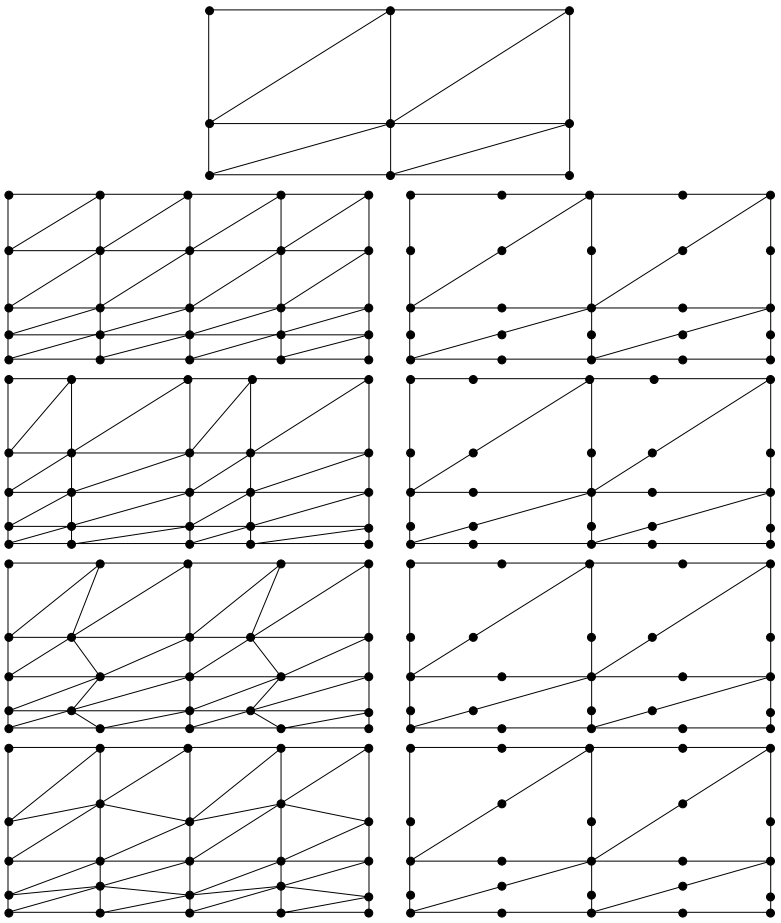


Fig. 1. Different options to add higher-order degrees of freedom to a coarse P1 mesh with a boundary layer stretching

3 Overview of Main ADIGMA Two-Dimensional Results

During the ADIGMA Project, Dassault Aviation computed four of the two-dimensional Mandatory Test Cases which covered a wide range of flow conditions: inviscid subsonic and transonic for MTC's 1 and 2, laminar Navier-Stokes for MTC 3, and transonic turbulent for MTC 5. All four test cases were run with the baseline second-order version of Dassault Aviation's industrial Navier-Stokes code `AeTher` and with its third and fourth order extension. We highlight here the most spectacular and significant results. The reader is referred to [11] for the complete description of these test cases.

In the inviscid subsonic MTC 1 test case (see Figure 2), the spurious entropy layer generated at the stagnation point is much reduced with quadratic

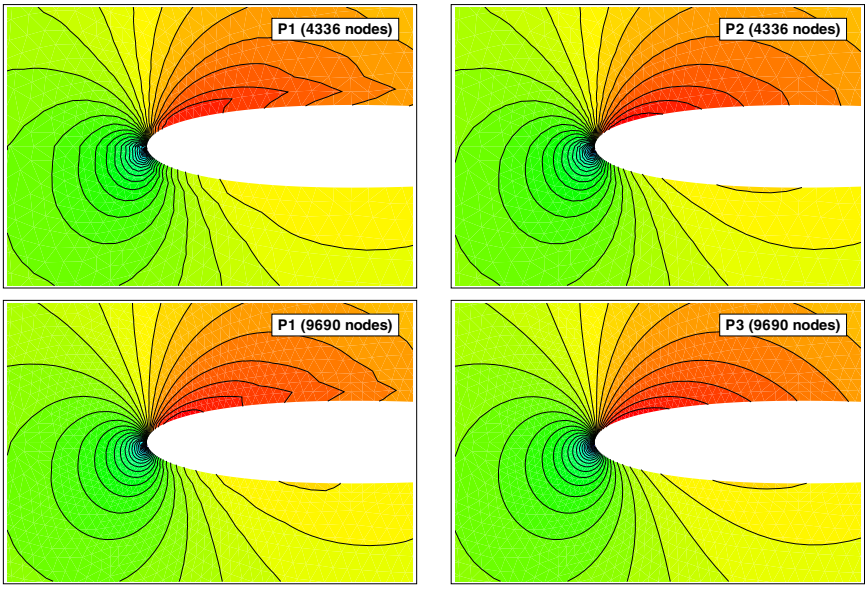


Fig. 2. ADIGMA MTC 1: NACA0012, $M = 0.50$, $\alpha = 2^{\circ}$, inviscid. Mach number contours on matching P1 iso-P2 and P2 grids, and P1 iso-P3 and P3 grids.

elements and virtually disappears with cubic elements. This directly impacts the Mach number contours which traditionally present kinks near the wall on coarse P1 meshes. These kinks are removed from higher-order calculations, which also present much cleaner contours for the same number of degrees of freedom. This test case is iconic in the sense that the benefit from higher-order elements is intriguingly visual and graphic. We wish the assessment of higher order were always so blatant; unfortunately, this is not the case most of the time and the added value is often extremely subtle, especially if we look at more complex and industrial test cases.

The inviscid transonic MTC 2 test case (not shown) came somehow as a surprise, revealing that shock waves are in fact not an issue even with *continuous* higher-order shape functions. In many instances, shocks can actually be sharper than in the corresponding second order solutions.

For the laminar MTC 3 test case, we have to look at the convergence of force coefficients presented in Figure 3. Pressure drag and lift converge faster with quadratic elements; cubic elements yield values close to the asymptotic limit for every computed grid, even the coarser ones. Unexpectedly viscous fluxes appear as a real challenge for this laminar test case. Second order viscous drag is still not converged for the finest mesh which contains over 1.5 million nodes: the asymptotic value is provided by the quadratic results. Linear elements have a hard time getting within one drag count of the asymptotic value of the friction drag, whereas all higher-order results are within half of the same margin. Heat

flux convergence (plotted in log scale) shows the substantial advantage of higher-order elements. The error in heat flux (which should be zero for an adiabatic wall condition) can be reduced by several orders of magnitude.

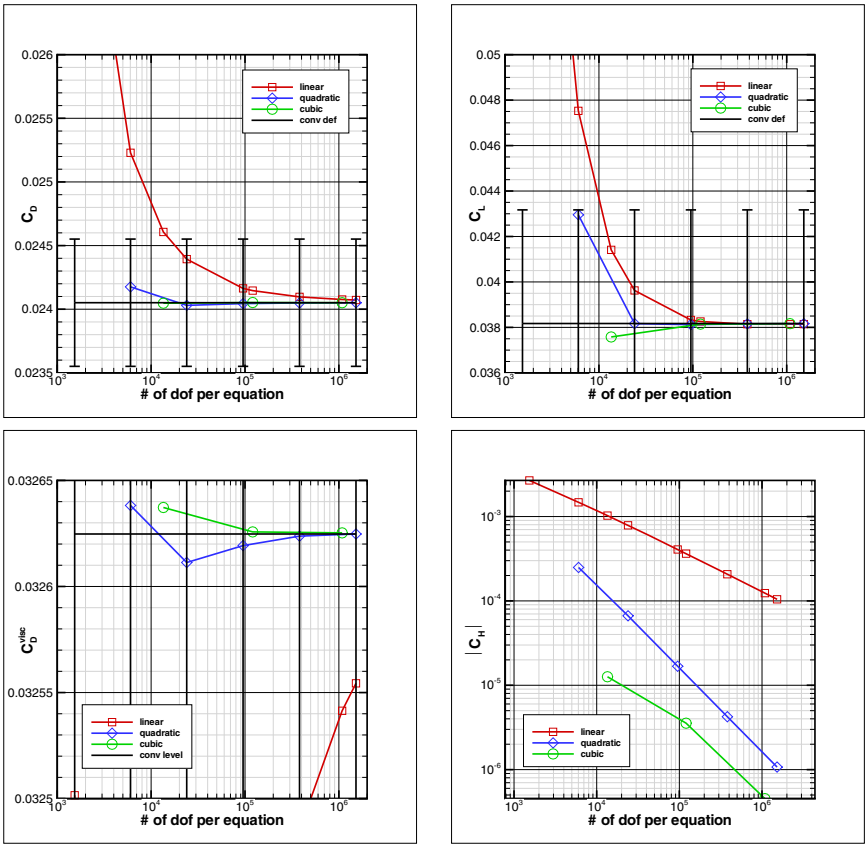


Fig. 3. ADIGMA MTC 3: NACA0012, $M = 0.50$, $\alpha = 2°$, $Re = 5,000$. Convergence of force and heat flux coefficients for P1, P2, and P3 elements.

Although a Navier-Stokes test case, MTC 3 is still far from a concrete industrial application. It exemplifies the difficulty of getting converged Navier-Stokes solutions. One can anticipate an even greater challenge with complex 3-D RANS computations.

Finally, MTC 5 deals with a transonic high Reynolds number RANS problem. The convergence plot of the viscous drag coefficient is presented in Figure 4 together with typical Mach number contours. There is no real distinction between the P1, P2, and P3 schemes. They converge at the same rate towards the same asymptotic values.

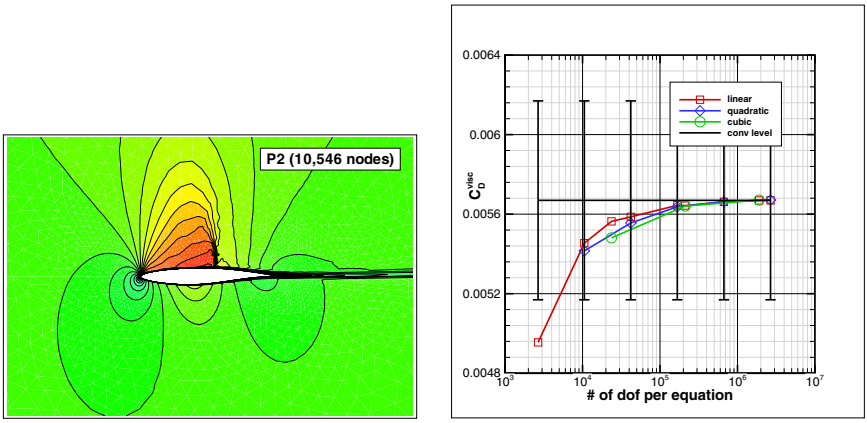


Fig. 4. ADIGMA MTC 5: RAE2822, $M = 0.734$, $\alpha = 2.79°$, $Re = 6,500,000$. Mach number contours on P2 grid and convergence of viscous drag coefficient.

4 Beyond ADIGMA: Towards Industrial Higher-Order RANS Computations

At the beginning of IDIHOM, Dassault Aviation brought higher-order capability to the latest version of its Navier-Stokes code `AeTher`: up to P4 elements in 2-D and up to P3 in 3-D, with corresponding richer integration rules (up to 12 points for triangles and 24 points for tetrahedra).

In the numerical method described in Section 2.1, the turbulence equations are solved in a staggered manner, with a second-order residual distribution scheme, and are weakly coupled to the Navier-Stokes field through the turbulent viscosity μ_t. As a first step, for higher-order calculations, RANS turbulent equations were solved on an underlying P1 mesh, and thus remained second-order accurate.

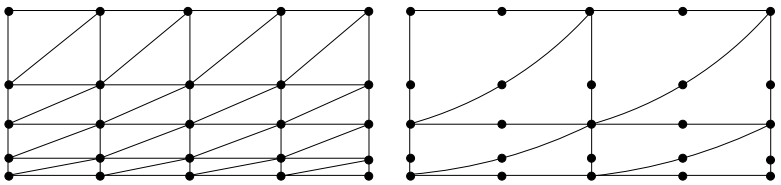


Fig. 5. Best option to add higher-order degrees of freedom to a coarse P1 mesh with a boundary layer stretching

MTC 5 of ADIGMA (RAE2822 airfoil) was revisited with the higher-order residual distribution schemes of [19]. Results were disappointingly similar to those obtained during ADIGMA and showed no particular advantage of higher-order elements (in this case quadratic) over standard linear 2nd order elements. This may be due to our weakly coupled approach. We also computed the case on a P2 grid which was curved and optimized by UCL with a similar outcome.

Following the discussion in Section 2.5, a new set of meshes with additional constraints on the growth rate of layers close to the airfoil was designed. They are sketched in Figure 5. Interestingly, these meshes exhibit curved edges even for flat surfaces. Moreover they demonstrate a different behavior, with higher-order elements converging faster to the asymptotic solution, especially with the coarser meshes (see Figure 6). This once again draws attention to the importance of mesh generation to extract all the benefits from higher-order elements.

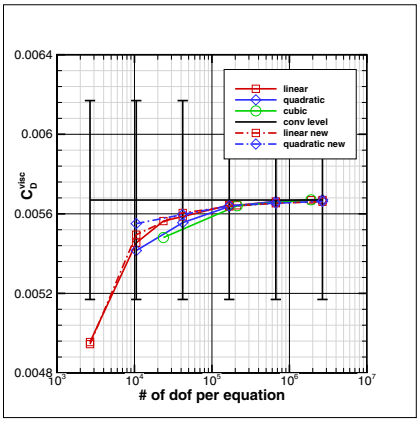


Fig. 6. ADIGMA MTC 5: RAE2822, $M = 0.734$, $\alpha = 2.79°$, $Re = 6,500,000$. Convergence of viscous drag coefficient for P1, P2, and P3 elements.

5 Test Case A.04: Falcon Aircraft

The selected test case to assess higher-order RANS capability on industrial configurations is a Falcon business jet including all its geometrical details. Two versions of this test case were considered labeled A.04a and A.04b.

Test Case A.04a is a first test of higher-order 3-D computation on an industrial configuration. It was performed at the very end of the ADIGMA Project and serves as a *baseline* higher-order solution for the IDIHOM Project. This test case has many limitations: the mesh is too fine, not curved, and most of all, as a real Falcon geometry, it could not be made open to other partners.

Consequently, the Generic Falcon Test Case A.04b was designed in order to give access to a complex aircraft configuration to IDIHOM partners in addition to Dassault Aviation. It includes all geometrical details of a real aircraft, except for the engines which are represented by flow-through nacelles.

5.1 Reference and Baseline Results for Test Case A.04a: Falcon 900EX

At the end of ADIGMA, since all the ingredients were there (we had checked the higher-order three-dimensional capability of the code on simple test cases such as an inviscid subsonic sphere and the ONERA M6 wing), it felt very tempting to test a real 3-D industrial geometry. We were only missing a higher-order 3-D mesh. We looked for the coarsest full aircraft mesh available. We found a Falcon 900EX design mesh dating from a few years, complete with vertical tail and empennage, pylons, nacelles and S-duct. It contains "only" 2,512,073 nodes. A cut through this mesh downstream of the wings, at the level of the engines is presented in Figure 8 in blue. One must note that this mesh is not adjusted for drag prediction with no specific refinement in the wake regions.

We built two grids based on this reference mesh: one linear P1 iso-P2 grid, and one quadratic P2 grid both containing 19,905,887 nodes. A cut through the linear refined tetrahedral mesh can be see in red in Figure 8.

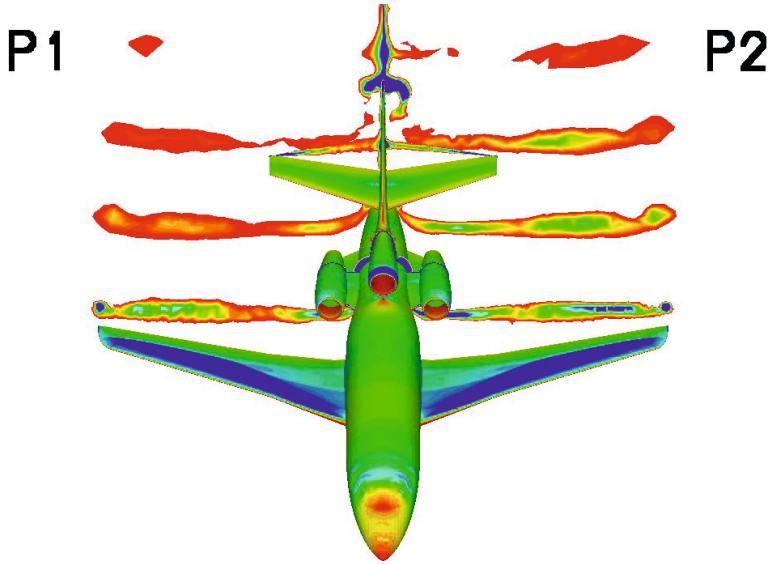


Fig. 7. A.04a: Falcon 900EX, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Pressure contours on the aircraft surface, entropy in the wake; linear (left) and quadratic elements (right) on the same 19,905,887-node mesh.

The exercise has a few limitations. First the additional body nodes were not projected on the actual surface of the aircraft: all the elements have straight edges and sub-parametric coordinate transformations are used (although everything is coded with isoparametric transformations). As described in section 4, all computations were performed using a second-order scheme for the turbulence equations. Only quadratic elements were tested and on a single mesh size.

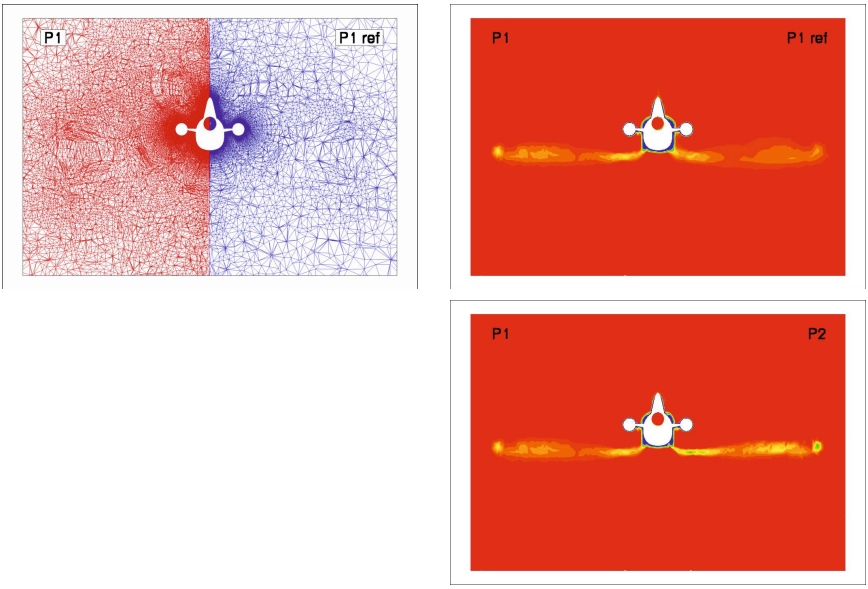


Fig. 8. A.04a: Falcon 900EX, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Clockwise from top left: cut through original 2,512,073-node P1 mesh (right) and corresponding uniformly refined 19,905,887-node P1 iso-P2 mesh (left); entropy contours for the original P1 mesh (right) and the uniformly refined P1 iso-P2 mesh (left); entropy contours for the P2 mesh (right) and the uniformly refined P1 iso-P2 mesh (left).

We used respectively a 1-point integration rule for linear tetrahedra and an 8-point rule for quadratic ones. The ratio between the number of elements of a P1 and a P2 tetrahedron mesh with the same number of nodes is 8. Consequently in principle the cost of a residual evaluation should be similar on the P1 and P2 meshes. Due to the higher-order coupling between degrees of freedom, the implicit Jacobian is roughly twice as big for P2 elements. This yields an extra cost when generating the operator and during the actual linear solve. Globally, the third-order computation is 68% more expensive than the second-order one, using the same CFL setting. This could be reduced for instance if the implicit operator is not updated at each time step or if the original P1 matrix is used (possibly at the cost of reduced robustness). The memory requirement is increased by 61%. Convergence is similar between P1 and P2 calculations with the same CFL

settings. Computations were done in parallel on an IBM Blue Gene/P using 1024 tasks.

Figure 8 presents entropy contours in vertical cuts through the aft part of the aircraft, in the engine region, downstream of the wing. The wake of the wing is slightly more defined in the 20-million-node P1 mesh in comparison with the original 2.5-million-node mesh. The third order result displays a much more detailed wake and a stronger tip vortex. This is also exemplified in Figure 7 which shows a more persistent and stronger wake in the quadratic element result.

It is common to see discrepancies between near-field and far-field drag analyses [14], [24]. The difference between these two drag evaluations is known as "spurious drag." In the reference 2.5-million-node P1 computation, the spurious drag amounts to 33 drag counts (10^{-4}). With the uniformly refined 20-million-node P1 mesh, it drops to 8 counts which is pretty good. Drag analysis performed on the third-order solution (with standard linear tools) indicates that the spurious drag is further reduced to just 1 count! This preliminary drag analysis suggests that spurious drag virtually disappears with increased order of accuracy.

5.2 Reference and Higher-Order Results for Test Case A.04b: Generic Falcon

The configuration of the Generic Falcon Test Case A.04b is presented in Figure 9. It is a twin-engine Falcon-like aircraft with flow-through nacelles. A CAD file was produced for this test case and released to participating Partners (namely INRIA, NUMECA, and VKI), instead of a triangular surface mesh, as initially planned. The flow conditions are: $M_\infty = 0.80$, $\alpha = 2.0°$, alt $= 40,000$ ft, which corresponds to $Re = 14,512,000$ based on the mean aerodynamic chord (2.888 m). The reference area is 49 m^2.

Reference Computation. A state-of-the-art Reference P1 mesh, according to current industrial standards, was generated for Test Case 1.04b. It is presented in Figure 10. It contains 8.4 million grid points.

Reference 2nd-order computations were performed by DASSAV with its industrial NS code `AeTher`. Different turbulence models were used. Figure 16 presents pressure coefficient plots along the wing at eight stations every meter from 2 to 9 m along the wing span. Reference results appear in black for the $K - \varepsilon$ turbulence model. Results obtained with the Spalart-Allmaras model are presented and further discussed in [5]; they are compared with the solutions obtained by other partners in the Project.

Higher-Order Mesh Generation and Curving. In order to build on the experience of Test Case A.04a, the objective was to construct higher-order meshes that would contain fewer degrees of freedom than the Reference P1 mesh and take advantage of higher-degree polynomials to represent curved regions of the model with fewer elements, typically the leading edges of the aerodynamic surfaces and the nacelle lips. A first attempt at building such a P2 mesh is presented in Figure 11.

We recall that the present higher-order meshing strategy is to add higher-order degrees of freedom to a coarse P1 "skeleton" mesh. The extra degrees of freedom are added on the edges, sides, and in the interior of the P1 linear elements. The resulting higher-order mesh requires "curving". This is performed in two steps: first the surface mesh is projected onto the CAD definition of the geometry; then the displacement is pushed into the volume mesh with some deformation technique to make the higher-order mesh valid.

The "skeleton" surface mesh consists of 17,460 P1 triangular elements and 8880 nodes. The leading edges were purposely undermeshed in order to evaluate the ability of higher-order elements to accurately represent the flow field in curved regions. For instance, only 4 elements are placed along the leading edge of the main wing and the lips of the nacelle are described with an uneven distribution of fairly large elements (see Figure 11 top line).

A first set of "skeleton" volume meshes were built from this surface mesh using the same parameters as the Reference mesh (10 μm for the height of the first element and a growing ratio of 1.15^2 in the boundary layer). The set contains meshes with first element height corresponding to y^+ of 2, 4, and 10. The finest mesh ($y^+ = 2$) contains 310,000 grid points. A set of P2 meshes was built from this set of P1 "skeleton" meshes corresponding to y^+'s of the first element of 1, 2, and 5. The finest P2 mesh ($y^+ = 1$) contains 2,500,000 grid points. Both coarser grids ($y^+ = 2$ and 5) do not save many degrees of freedom, since only a couple of element layers next to the aircraft surface are removed. In the light of the

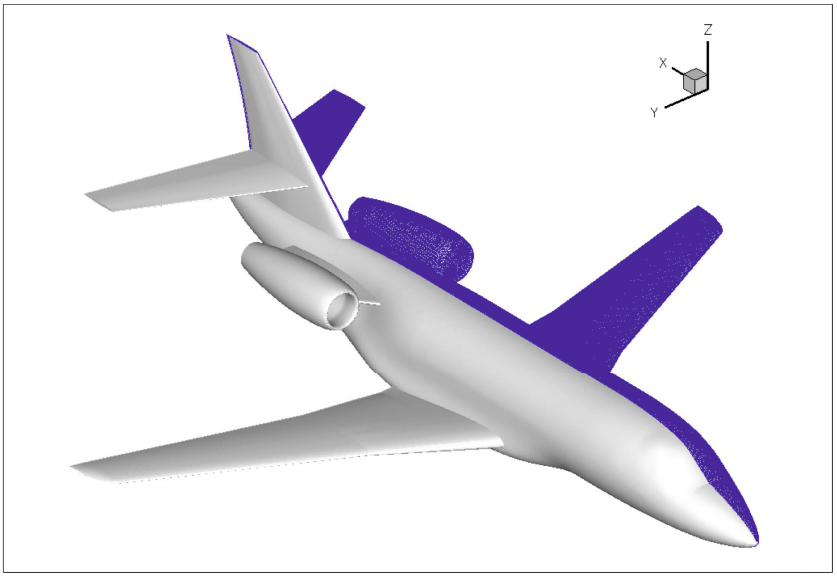


Fig. 9. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Test case definition.

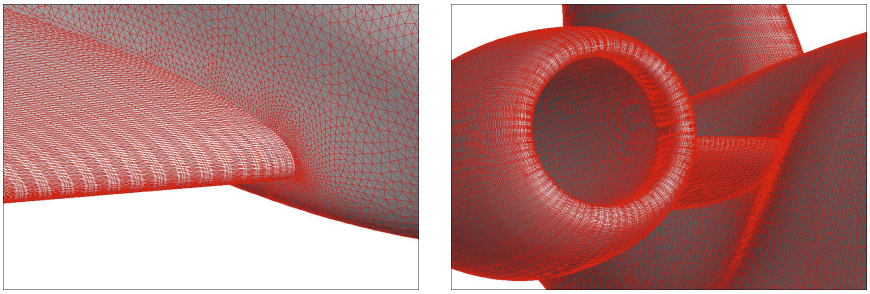


Fig. 10. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Views of the Reference P1 surface mesh.

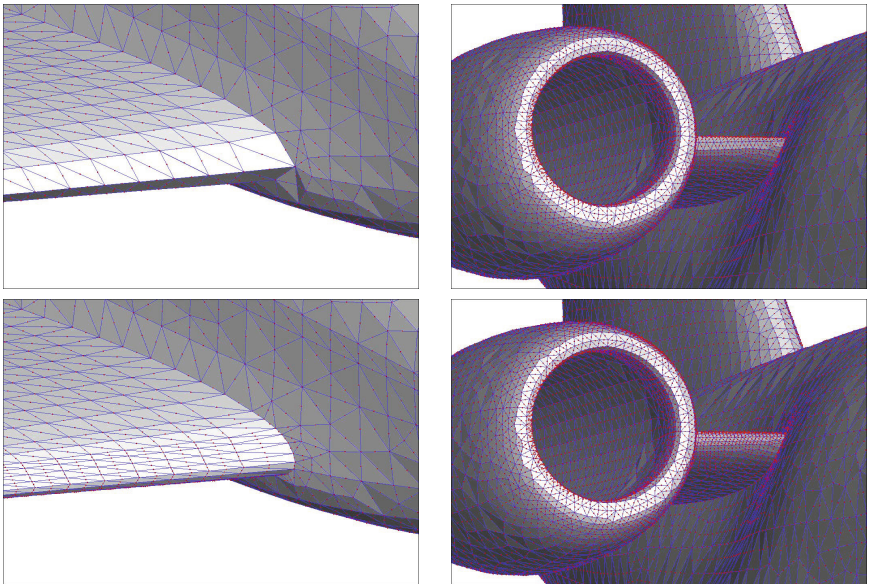


Fig. 11. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Views of the uncurved P2 meshes: initial P2 mesh (top); P2 mesh refined along the leading edges.

2-D study performed on the RAE2822 test case (see section 4), it was concluded that these first P2 meshes were too fine to make a valid assessment.

A new coarser "skeleton" mesh was generated with a $y^+ = 2$ for the first element layer, but a much more aggressive progression in the boundary layer, resulting in only 250,000 grid points. P2 and P3 grids (and the corresponding nested P1 grids) were buit based on this "skeleton" mesh. The P2 mesh contains just under 2,000,000 degrees of freedom (the P3, 6,750,000).

Computations were performed on the P2 and the nested P1 meshes, before any attempt at curving them. Surface pressures are compared with the Reference computation in Figure 12. The 3rd-order P2 computation shows qualitatively a better agreement with the Reference computation, especially on the leading edge of the wing, with 4 times fewer degrees of freedom. These computations are further discussed in the next section.

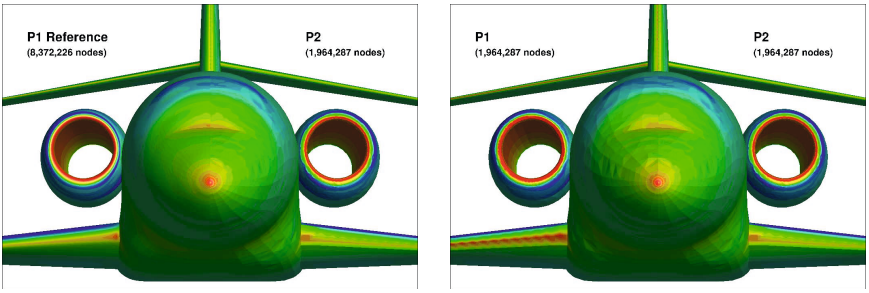


Fig. 12. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Pressure contours: comparison of Reference P1 computation with P1 and P2 results on nested uncurved meshes.

After projection of the higher-degrees of freedom onto the CAD, the curving technique used so far was based on a Laplacian mesh deformation applied to the P1 nested mesh. It worked very well on ADIGMA on 2-D test cases. Unfortunately it was shown in [19] that a priori splitting of 3-D higher-order elements into P1 elements could induce mesh locking and prevent mesh deformation. Going from a Laplacian operator to full elasticity does not help.

Several alternate curving methods were investigated: 1) analytical, which tries to reproduce the effect of a direct curved mesh generation in propagating the surface displacement into the volume (this works fairly well in the quasi-structured mesh in the boundary layer, but it is more difficult to apply the idea in very unstructured regions of the mesh. The idea is to stop the propagation once it has reached an element which is big enough to handle the deformation on a single face without the need of any further propagation. The algorithm, although promising, reveals extremely difficult to implement in practice in real unstructured industrial meshes); 2) Radial Basis Functions (RBF), which only use the displacement of the surface nodes to compute the displacement in the volume; 3) higher-order elasticity, which intrinsically does not require the a priori splitting of elements into P1 elements and focuses on the global validity of higher-order

elements, not just their individual P1 components; 4) and the higher-order mesh optimization tools available in Gmsh (see [16], [23], and [20] in Chapter II.1). All failed to a certain degree.

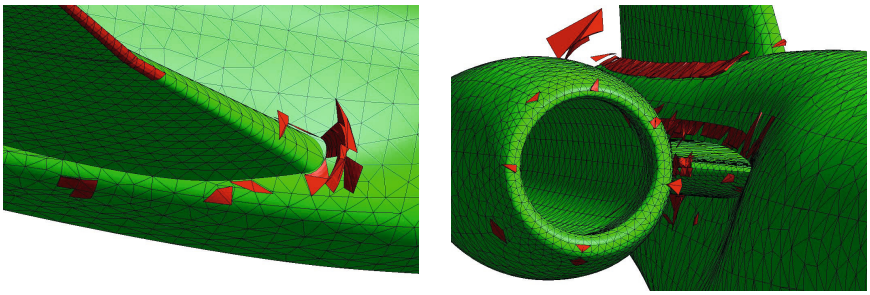


Fig. 13. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Visualization of P2 invalid tets with Gmsh.

We found out though that Gmsh could be of a great help in analyzing 3-D higher-order grids. First of all, it can display curved elements. Areas where higher-order degrees of freedom are not exactly projected onto the CAD definition can easily be detected. It was also discovered that invalid (or very close to invalid) elements could be generated on the surface, for instance at the junction between the wing and the fuselage (see Figures 11 and 13). Surface elements with too sharp angles on the fuselage could produce curved elements with angles going to zero (and consequently Jacobians) when the leading edge is curved. In addition, Gmsh can visualize invalid 3-D elements in the volume: they appear in red in Figure 13. Most are localized in highly-curved regions, e.g. the leading edges, in corners where condensation of grid points occur during the mesh generation process, and in rare occasions in the volume.

Analytical curving yields a mesh with 1600 invalid elements (they may have a positive volume, but their Jacobian can go to zero or negative somewhere inside the element) over a total of 1,450,000 P2 elements. Two passes of optimization with Gmsh can reduce this number to 381. Direct optimization with Gmsh produces 1773 invalid elements, starting with 17,500 invalid elements at the surface of the aircraft. It appears that Gmsh has a better time with invalid elements somewhere inside the volume, rather than next to the wall with extremely high aspect ratios.

With the P2 elastic deformation, the number of invalid elements is reduced to 807, although all with a globally positive volume. Further optimization with Gmsh can reach 444 invalid elements. It seems that higher-order elasticity in combination with Gmsh works best at producing valid curved higher-order meshes.

A new version of Gmsh, tuned and parametrized by UCL, was able to reduce the number of invalid elements to just 11, starting with the mesh obtained after

an analytical deformation and two passes of optimization with the standard version of Gmsh.

It is quite possible that the right combination of curving techniques could render this P2 mesh valid. Nevertheless, it was decided to refine the surface mesh in highly curved regions, viz. along the leading edges of the wing (4×), the empennage (2×), and the nacelle pylons (2×). The leading edge of the vertical tail and the lips of the nacelle were not modified. Elements connected to regions were curving occur were also modified in order to avoid the small angle issue described earlier. The new surface "skeleton" P1 mesh can be seen in Figure 11 (lower part). It contains 9503 nodes and 18,708 elements. The refined P2 3-D mesh is only slightly bigger than the original one with 75,000 additional degrees of freedom.

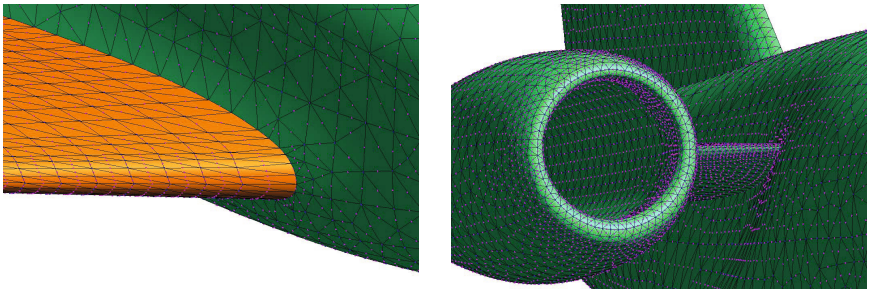


Fig. 14. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Views of P2 mesh, curved and optimized with Gmsh.

The best combination of curving techniques was applied (P2 elastic deformation followed with an optimization with the tuned version of Gmsh): only 250 invalid elements were obtained after elastic deformation and Gmsh was able to produce a valid mesh in less than a minute. The minimum Jacobian is 0.09. The P2 surface mesh after curving with Gmsh can be seen in Figure 57. Note that the nacelle lips, although not refined, are particularly well represented with P2 curved elements. Numerical results are presented in the next section.

Higher-Order Computations. The valid curved P2 mesh produced in interaction with UCL was computed with the K-ε model. The solution is compared with the Reference solution in Figure 15. The acceleration of the flow over the leading edge of the wing is much better represented than in the solution obtained on the uncurved mesh (see Figure 12). We anticipate that this is due to the curving, rather than to the mesh refinement of the leading edge.

Pressure plots at different stations along the wing span obtained by different computations during the IDIHOM Project are gathered in Figure 16. The solutions computed with the K-ε turbulence model appear with a solid line, those

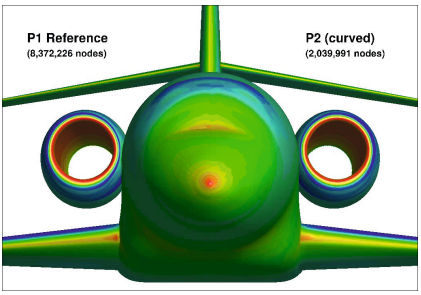


Fig. 15. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Pressure contours: comparison of Reference P1 computation with P2 result on **curved** mesh.

computed with the Spalart-Allmaras model with a dashed line. The Reference solution is represented with a solid black line. It must be compared with the solutions obtained on the uncurved meshes described in the preceding section: 2nd order P1 in red, 3rd order P2 in blue, and 4th order P3 in green. Except for the P2 all were computed with the K-ε turbulence model. Even on uncurved grids, P2 solutions are much closer in terms of pressure distribution to the Reference computation than the P1 solution with the same number of degrees of freedom. It is even more so with the Spalart-Allmaras turbulence model. On the other hand, the 4th order P3 solution is at best as good as the P2 solution in terms of shock location and present many pressure oscillations in smooth regions of the flow. This is probably due to the increased accuracy of the scheme and the greater sensitivity to curvature discontinuities of the surface which are magnified with uncurved P3 elements.

Table 1. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Spurious drag for the different computed grids.

mesh	**element type**	**grid size**	**spurious drag** ($\times 10^{-4}$)
Reference	P1	8.37×10^6	7.81
"skeleton"	P1	0.25×10^6	149.61
uncurved linear	P1	1.96×10^6	89.79
uncurved quadratic	P2	1.96×10^6	18.54
curved quadratic	P2	2.04×10^6	5.96

The P2 solution obtained on the curved mesh is plotted in pink. In order to better compare this last solution with the Reference one, both are shown again in Figure 17. The agreement between the two solutions is remarkable although the 3rd-order P2 solution was obtained with 4 times fewer degrees of freedom (2 vs 8.4 million), at a similar CPU cost and with a reduced memory imprint (-62%).

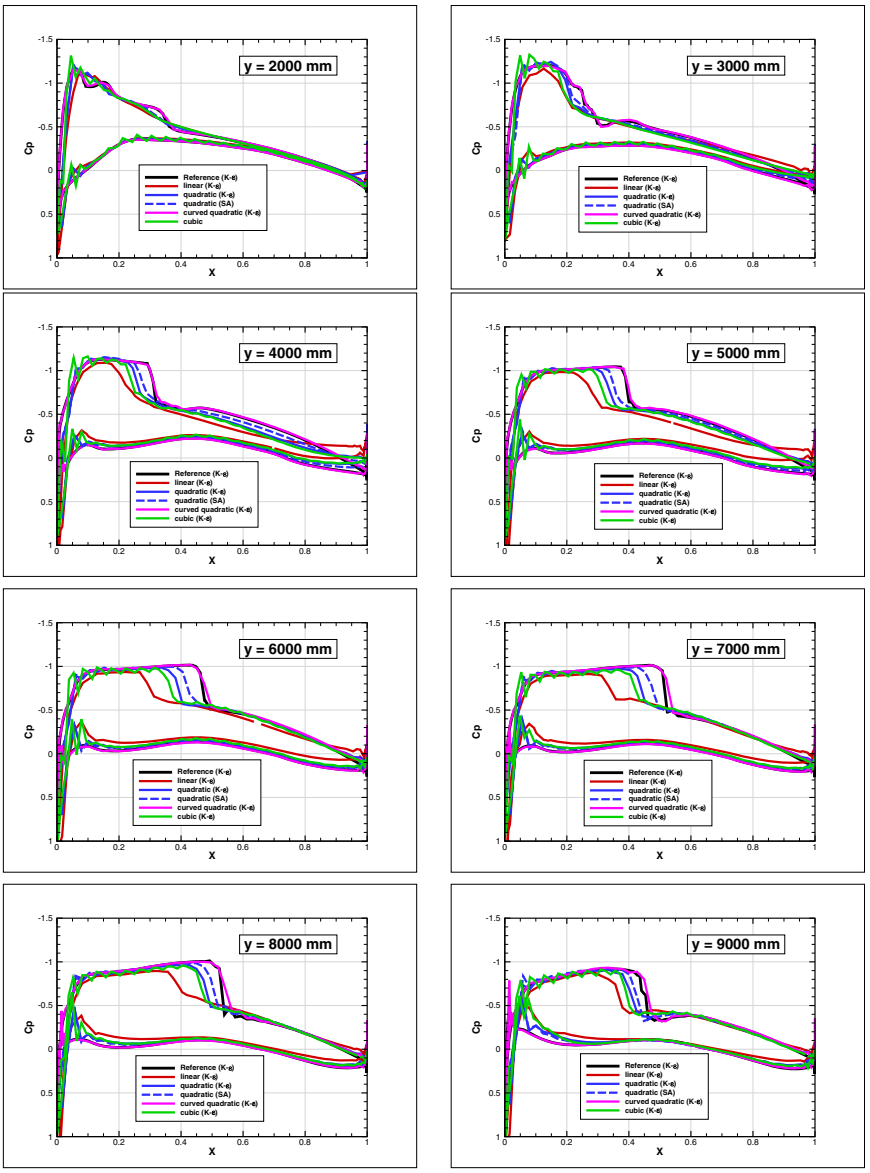


Fig. 16. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Comparison of C_p plots at different stations along the wing span.

Field drag integration was used for Test Case A.04a. Although the quadratic mesh was too fine and not curved, it revealed as a discriminating tool to assess the rewards of higher-order elements. The evaluations of spurious drag for the different grids used for Test Case A.04b are gathered in Table 1.

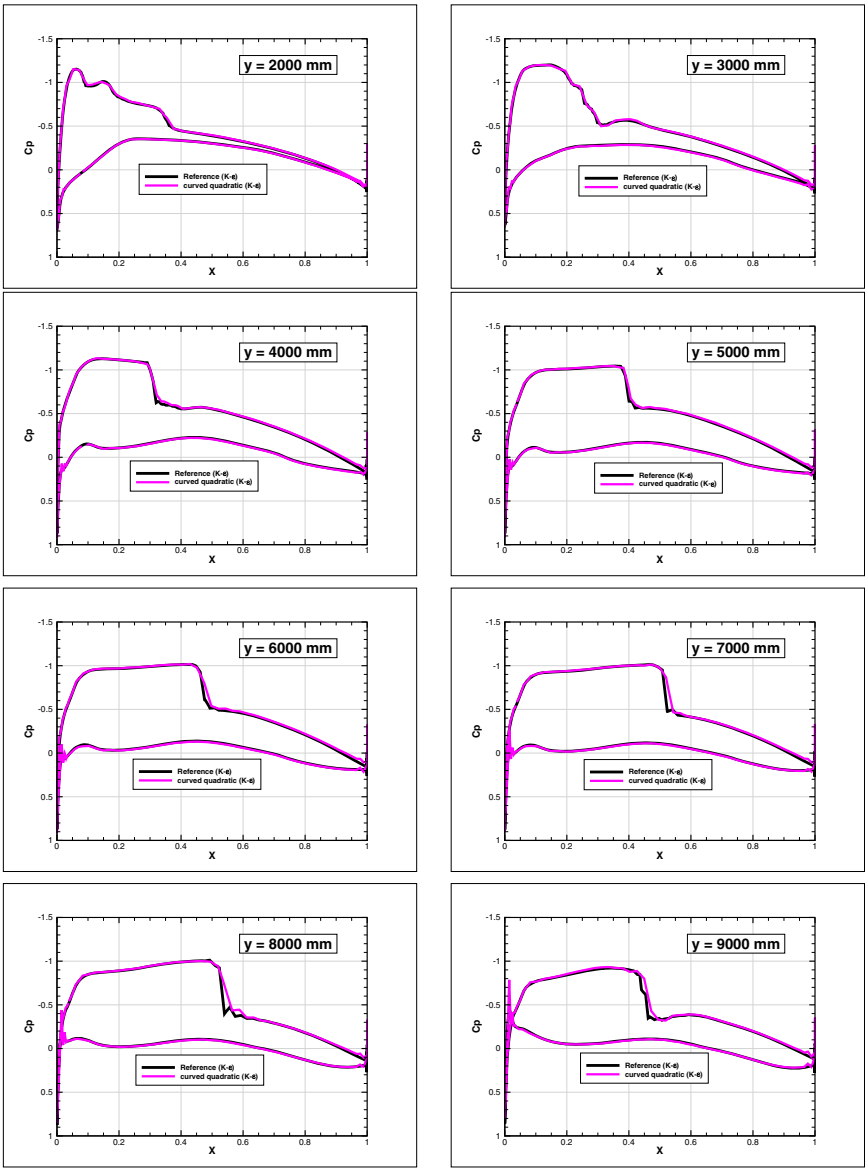


Fig. 17. A.04b: Generic Falcon, $M = 0.80$, $\alpha = 2.0°$, $Re = 14,500,000$. Comparison of C_p plots at different stations along the wing span between the Reference P1 and the curved P2 computations.

Going from 2nd to 3rd order with the uncurved grids reduces the spurious drag by a factor of 5 (from 89.79 to 18.54), to a value which is still higher than the Reference of 7.81 drag counts. With the curved P2 mesh with refined leading edges, the level of spurious drag drops to less than 6 drag counts which is 25% lower than the Reference 2nd order computation with 4 times fewer degrees of freedom.

6 Extension to Unsteady Flows and LES/DES

DASSAV worked on the extension to higher order of the DES/LES models within its stabilized finite element code `AeTher`: accurate time integration for unsteady flows; adapt the element size evaluation to higher-order unstructured meshes for the DES switch calculation; extend the "probe nodes" (were pressure signal is recorded at every time step) to higher-order meshes.

For unsteady calculations fully implicit time integration with dual time stepping is performed based on the standard second-order Gear's scheme. Third order calculation may suffer from a time integration which is only second order accurate. Nevertheless, keeping everything equal, including the time integration scheme, enables a fair comparison between 2nd and 3rd order spatial accuracy.

6.1 Assessment of Space Accuracy for Unsteady Flows

In order to assess the increase in space resolution brought by higher-order elements for LES/DES calculations and adjust the mesh requirements accordingly, a simple unsteady wave propagation problem was set up: a vibrating membrane in a 2-D nacelle-like geometry would propagate a plane wave into an open domain at different frequencies. The unsteady problem was solved with a deforming mesh in Arbitrary Lagragian/Eulerian formulation (ALE) with polynomial orders up to P4. The influence of the integration rule and the interaction between the time accuracy and the space stabilization were studied. The outcome was that 9 points per wave length are required with the reference 2nd order code. This figure can be reduced to respectively 6 or even 5 points with quadratic (3rd order) or quartic (5th order) elements.

The impact of the limited second order time accuracy was evaluated in comparing the unsteady results with those obtained with the linearized in the frequency domain option of the code, which can be brought to higher-order fairly transparently once the higher-order capability exists in the nonlinear code. Unsteady and linearized results compare favorably. The more complex 3-D test case of a generic nozzle provided by Rolls-Royce Deutschland is presented in Figure 18.

At 700 Hz on the same mesh which corresponds to 16 degrees of freedom per wave length, P1 and P2 results are very similar. On the contrary, at 2000 Hz the number of grid points per wave length drops to 5.5 and the increased accuracy of the P2 solution becomes more obvious. Higher-order elements open up a frequency range for aeroacoustic applications that was unattainable before. They should bring a similar advantage to LES and DES.

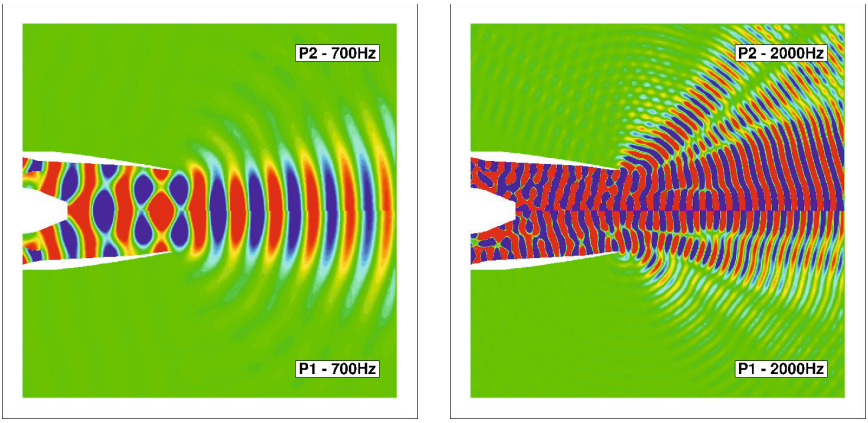


Fig. 18. Comparisons of P1 and P2 linearized Navier-Stokes results on a RRD generic nozzle test case for two different plane mode frequencies 700 and 2000 Hz

7 Test Case A.15: M219 Cavity

This test case was selected for two main reasons: first of all to try and assess the advantage of higher-order element in the context of unsteady flows and more specifically LES/DES; second, to conduct this assessment serenely, without any second thoughts about higher-order mesh generation. This is enabled by the simplicity of the geometry which only contains flat surfaces and thus does not require any mesh curving. The details about the geometry and the experimental set-up can be found in [15]. The flow conditions are $M_\infty = 0.85$ and $Re = 6.8 \times 10^6$, based on the cavity length (20 in).

7.1 Computational Domain and Grids

DASSAV has computed the M219 cavity in a configuration close to its original experimental set-up, as can be seen in Figure 19.

The Reference linear mesh is an unstructured tetrahedral mesh build on top of a triangular surface mesh with a characteristic edge size of 2 mm. In the boundary layer, the mesh is constructed with a first layer of elements of 10 μm height and a growing ratio of 1.25. In the mixing layer, an unstructured block of tetrahedra with edge lengths of 2 mm in the x and y directions and 2.5 mm in the z direction is inserted. The grid contains slightly less than 3.4 million grid points. Higher order grids are generated by adding degrees of freedom to a coarse P1 grid. The difficulty is to build a "skeleton" grid which is coarse enough to yield higher-order grids with a controlled number of grid points and still preserve the qualities in geometric representation, point density, and element distortion of the original reference grid. Fortunately for this test case, all surfaces of the model are flat which alleviates the issue of curving the elements in the volume. With

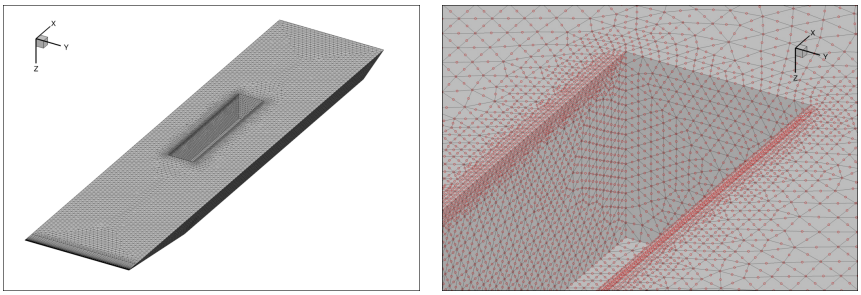


Fig. 19. A.15: M219 Cavity. Configuration and surface view of the P2 mesh.

these constraints, the skeleton P1 mesh built on a different surface grid with a characteristic edge size of 4 mm could not be made coarser than about 450,000 grid points; consequently the corresponding P2 grid (and the matching P1 grid) contains 3.5 million grid points. Coarsening techniques developed later on for the A.04 Falcon test case could probably have been similarly applied to the cavity to produce yet coarser higher-order grids (see section 5.2).

7.2 Reference and Higher-Order Results

For all simulations, the flow field is initialized with a steady RANS computations using the Spalart-Allmaras turbulence model. The computation is pursued in a unsteady mode with a Zonal DES approach [11]. The same time step of 1.5×10^{-5} s (that is 121 time steps per through flow over the cavity length) is used for all computations whatever the order of the space integration. After a settling time of 195 ms, unsteady data is acquired for another 225 ms, which accounts for a total simulation time of 420 ms (i.e., 230 through flows). 186 ms are post-processed to produce the energy spectra and the OASPL curves. For a better comparison, especially at low frequencies, the experimental pressure history is post-processed over the same time interval. All calculations were performed on 1024 cores of an IBM BlueGene/P.

In Figure 20, isovalues of the Q-criterion colored by the Mach number are presented for the 2nd-order linear P1 and the 3rd-order quadratic P2 solutions computed on nested grids and thus containing the same number of degrees of freedom. The higher-order simulation transitions sooner to full 3-D turbulence and contains much finer turbulent structures.

A typical SPL frequency spectrum corresponding to Kulite #21 is displayed in Figure 21a. It shows a reasonable agreement between the experiment (in red) and the different simulations both in terms of Rossiter peak locations and amplitude. One must note that the higher-order solution (in blue) exhibits more energy in the higher part of the spectrum.

In Figure 21b, OASPL plots corresponding to different realizations of the simulation are presented. The red and green curves represent respectively the Reference solution and the P1 solution computed with the same degrees of freedom

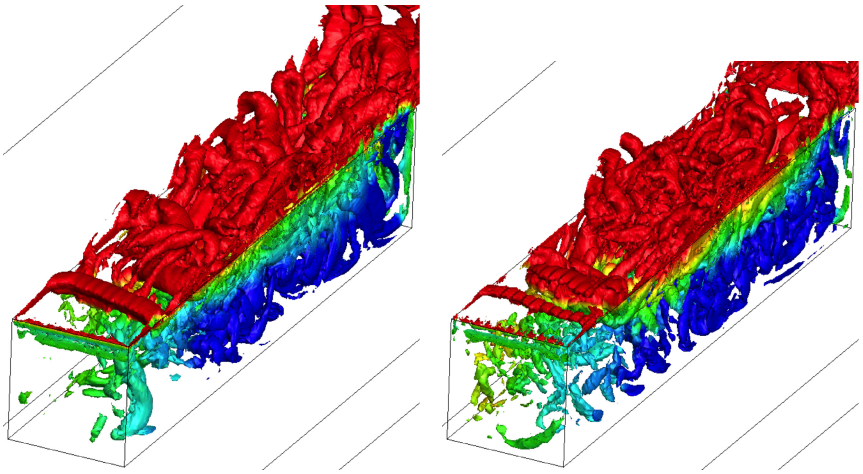
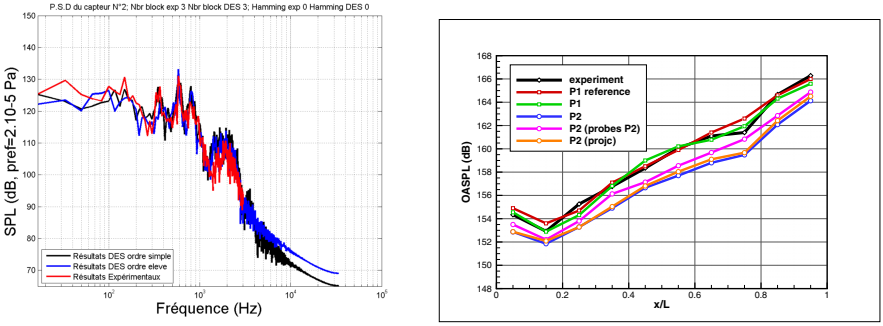


Fig. 20. A.15: M219 Cavity. Turbulent structures in the QinetiQ M219 cavity: standard 2nd-order linear P1 elements (left) vs. 3rd-order quadratic P2 elements (right).

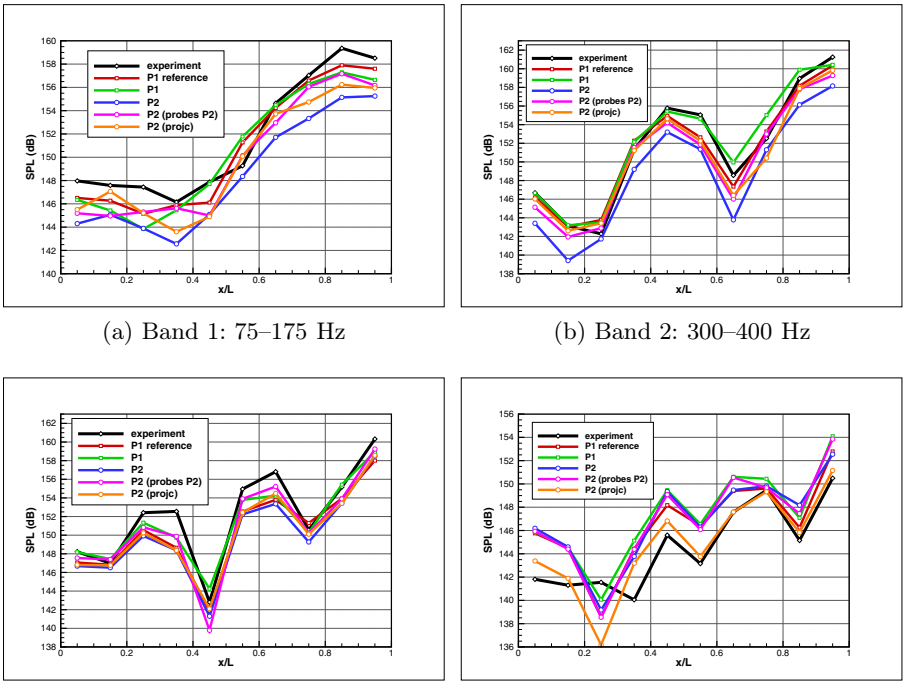
as the higher-order solutions. Both solutions are quite similar and are in good agreement with the experiment shown in black (a few tenth of a dB on the average). Note that the Reference solution corresponds to a longer simulation time of 350 ms. This confirms that the statistics are converged for the newer computations which account for simulations times of 225 ms. The P2 third-order solution is plotted in blue. It shows a drop in OASPL 1 to 2 dB below the experiment.

Two additional curves are presented in Figure 21b. For the sake of simplicity, earlier higher-order computations were carried out with the same sampling routine as P1 computations, that is the pressure signal was linearly interpolated at the location of the experimental Kulites. The effect of the actual higher-order interpolation is shown in the pink curve. Depending on the pressure tap, an



(a) SPL at Kulite K21, $x/L = 15\%$ (b) OASPL

Fig. 21. A.15: M219 Cavity. Local and Overall Sound Pressure Levels.



(a) Band 1: 75–175 Hz (b) Band 2: 300–400 Hz

(c) Band 3: 550–650 Hz (d) Band 4: 750–850 Hz

Fig. 22. A.15: M219 Cavity. Band-integrated Sound Pressure Levels.

increase of up to 1 dB is observed in OASPL, bringing the higher-order P2 results very close to the refined mesh P1 results. This stresses the importance of post-processing higher-order solutions with adapted higher-order techniques. This remark is also valid for line and contour plots. The orange curve in Figure 21b represents the effect of the second order Navier-Stokes derivatives in the SUPG/GLS stabilization term of DASSAV residual-based stabilized finite element code. The last term in eq. (5) is zero for P1 elements and was dropped so-far from the stabilization operator (second line in (4)) for higher-order computations. The effect is not so sensitive on OASPL's but is more significant at certain frequencies as can be seen in the band-integrated SPL's in Figure 22. Frequency bands are defined for this test case according to the analysis of Larchevêque [12]. Again, the difference in SPL with respect to the original P2 solution is significant, and the experimental level can even be retrieved for the higher frequency band. Unfortunately, the combination of both higher-order interpolation for pressure probes and of the higher-order term in the stabilization could not be tested during the duration of the project.

Regarding computational cost, DASSAV 3rd-order P2 simulations last about 2.5 times as long as a standard 2nd-order P1 simulation (6 days compared to 2.5 days on 1024 cores of an IBM BlueGene/P). Higher-order computations use

10 integration points per tetrahedron, whereas a single point is used for 2nd-order runs. It is quite possible that on this configuration where all elements have uncurved edges, a four-point integration rule would suffice, further reducing the cost of the higher-order simulation.

7.3 Effect of Mesh Refinement, Subgrid-Scale Model and Time-Step

We believe that the more precise P2 computation reveals some of the limitations of the underlying Smagorinsky subgrid scale model in the DES. A finer P1 simulation, shown in a green dashed line in Figure 23, exhibits the same tendency of decreased OASPL's. The corresponding mesh is similar to the P1 and P2 meshes, except in the cavity region where the typical grid size has been reduced from 2 to 1 mm; this mesh contains close to 3.9 million grid points.

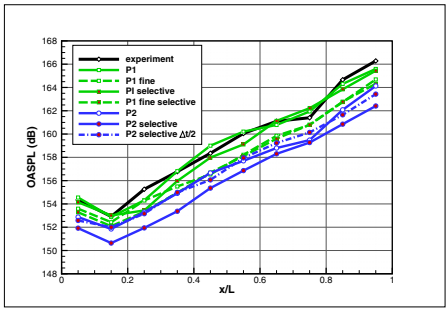
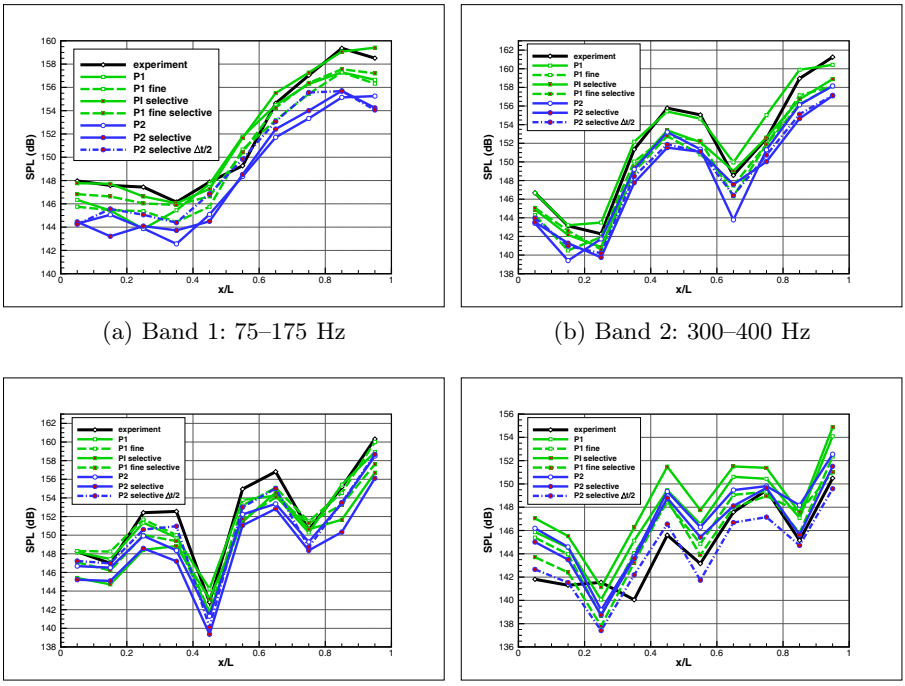


Fig. 23. A.15: M219 Cavity. Overall Sound Pressure Levels.

The effect of the subgrid scale model is studied next. The P1, P2, and refined P1 meshes were recomputed using a selective model, where the subgrid scale model is only active in regions of the flow where 3-D turbulent structures are present [6] [7]. The selective DES results are shown with red symbols in Figures 23 and 24. As could have been anticipated, the 2nd order solutions do not show much sensitivity to the subgrid scale model, especially for OASPL's. This is unexpectedly also the case on the fine grid. On the other hand, 3rd order results display a greater response to the model, but unfortunately the OASPL's go further away from the experimental values.

A time step reduction ($\times 0.5$) was also applied to the P2 selective simulation. It appears that the combination of a more precise spatial discretization (P2) with a less dissipative model (selective) could beneficiate from a more precise time integration scheme.

The band-integrated sound pressure levels (SPL's) in Figures 22 and 24, plotted in the same colors as the OASPL's, show that the main physics is captured in all simulations over the complete range of frequencies. As could be anticipated, higher frequencies are better captured on P1 refined meshes and higher-order solutions with the best numerical ingredients. It would be of interest to run a P2



(a) Band 1: 75–175 Hz (b) Band 2: 300–400 Hz

(c) Band 3: 550–650 Hz (d) Band 4: 750–850 Hz

Fig. 24. A.15: M219 Cavity. Band-integrated Sound Pressure Levels.

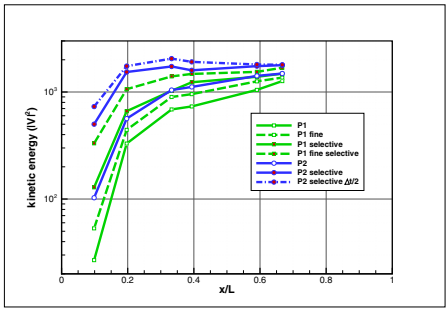


Fig. 25. A.15: M219 Cavity. Integration of kinetic energy over 2000–6000 Hz band.

simulation with the higher-order term in the stabilization and a reduced time step possibly in combination with the selective model, as this could not be done during the Project.

Although not very conclusive in view of the SPL's, the effect of the selective model is more obvious if we have a look at the kinetic energy spectrum (instead of pressure) sampled along a line in the mixing layer. It is presented integrated over the 2000-6000 Hz band in Figure 25. All the different ingredients (mesh

refinement/order, subgrid scale model and time step size) add up in a natural way to bring more energy in the fine structures of the flow. Unfortunately, there are no experimental values to validate the computed levels.

8 Conclusions

The work of Dassault Aviation during the IDIHOM Project covered two aspects of the industrialization of higher-order methods for CFD. It applied its stabilized finite element code to a generic Falcon test case and to the aeroacoustic test case of an open cavity, respectively computed with a RANS turbulence model and a DES approach.

DASSAV obtained 3rd (K-ε and SA) and 4th order (K-ε) solutions on uncurved higher-order meshes for the A.04b Generic Falcon test case. A 3rd order solution on a curved mesh (which was produced in interaction with UCL) was also obtained for this test case. The 3rd order solution on the curved mesh shows a remarkable agreement with the reference solution for pressure distribution along the wing span with 4x fewer degrees of freedom. It also confirms the important reduction in spurious drag observed on much finer higher-order grids at the end of ADIGMA.

A lot of effort was put into the higher-order mesh generation activity, especially to obtain a valid higher-order curved mesh for Test Case A.04b with a moderate number of degrees of freedom. Higher-order meshes are currently built by adding degrees of freedom to a coarse "skeleton" P1 grid. They need to be curved thereafter. The best option so far is a combination of higher-order elastic deformation and mesh optimization with Gmsh in interaction with UCL. In the long run, we should look at a dedicated way to generate directly curved higher-order meshes. This seems the best option to make it industrially viable.

In the mean time, higher-order elements might show a unique potential for Large Eddy Simulations. DASSAV obtained 3rd order DES solutions for the A.15 M219 Cavity test case. These solutions show reasonable SPL frequency spectra, but are not as good as the reference P1 solution in terms of OASPL (1-2dB below experiment). Higher-order solutions go into the direction of mesh refinement for linear solutions. Sensitivities to subgrid scale model (ZDES and selective), time step size, higher-order probes and higher-order viscous term in the stabilization operator were also studied. Band integration shows that the main physics is captured. In addition to higher-order linearized Navier-Stokes in the frequency domain, higher-order stabilized finite elements show a tremendous potential for industrial aeroacoustic applications in the very near future.

References

1. Brooks, A.N., Hughes, T.T.R.: Streamline Upwind Petrov Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Computer Methods in Applied Mechanics and Engineering 32, 199–259 (1982)

2. Chalot, F., Hughes, T.J.R., Shakib, F.: Symmetrization of conservation laws with entropy for high-temperature hypersonic computations. Computing Systems in Engineering 1, 465–521 (1990)
3. Chalot, F., Hughes, T.J.R.: A consistent equilibrium chemistry algorithm for hypersonic flows. Computer Methods in Applied Mechanics and Engineering 112, 25–40 (1994)
4. Chalot, F., Mallet, M., Ravachol, M.: A comprehensive finite element Navier-Stokes solver for low- and high-speed aircraft design. Paper #94-0814, AIAA 32nd Aerospace Sciences Meeting, Reno, NV, January 10-13 (1994)
5. Chalot, F., Dinh, Q.V., Mallet, M., Naïm, A., Ravachol, M.: A multi-platform shared- or distributed-memory Navier-Stokes code. In: Parallel CFD 1997, Manchester, UK, May 19-21 (1997)
6. Chalot, F., Marquez, B., Ravachol, M., Ducros, F., Nicoud, F., Poinsot, T.: A consistent Finite Element approach to Large Eddy Simulation. Paper #98-2652, AIAA 29th Fluid Dynamics Conference, Albuquerque, NM, June 15-18 (1998)
7. Chalot, F., Marquez, B., Ravachol, M., Ducros, F., Poinsot, T.: Large Eddy Simulation of a compressible mixing layer: study of the mixing enhancement. Paper #99-3358, AIAA 14th Computational Fluid Dynamics Conference, Norfolk, VA, June 28-July 1 (1999)
8. Chalot, F.: Industrial aerodynamics. In: Stein, E., de Borst, R., Hughes, T.J.R. (eds.) Encyclopedia of Computational Mechanics, ch. 12. Computational Fluid Dynamics, vol. 3. Wiley (2004)
9. Chalot, F., Levasseur, V., Mallet, M., Petit, G., Réau, N.: LES and DES simulations for aircraft design. Paper #2007-0723 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8-11 (2007)
10. Chalot, F., Daumas, L., Levasseur, V., Mallet, M., Rau, N.: LES and DES aeroacoustic simulations for in-flight opened weapon bay. paper #2009-099, International Forum on Aeroelasticity and Structural Dynamics, Seattle, WA, June 21-25 (2009)
11. Chalot, F., Normand, P.-E.: Higher-order Stabilized Finite Elements in an Industrial Navier-Stokes Code. In: Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.) ADIGMA. NNFM, vol. 113, pp. 145–165. Springer, Heidelberg (2010)
12. Sørensen, K.A., Johnston, C., Leicht, T., Chalot, F., Eliasson, P., Bassi, F., Couaillier, V., Kessler, M.: External Aerodynamic Test Cases. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach. NNFM, vol. 128, pp. 559–614. Springer, Heidelberg (2015)
13. Deck, S.: Recent improvements in the Zonal Detached Eddy Simulation (ZDES) formulation. Theor. Comput. Fluid. Dyn. 26, 523–550 (2012)
14. Destarac, D.: Far-field/near-field drag balance and applications of drag extraction in CFD. In: CFD-based Aircraft Drag Prediction and Reduction, November 3-7. VKI Lecture Series. National Institute of Aerospace, Hampton (2003)
15. Koloszar, L., Villedieu, N., Deconinck, H., Bosnyakov, I.S., Mikhaylov, S.V., Morozov, A.N., Podaruev, V.Y., Troshin, A.I., Vlasenko, V.V., Wolkov, A.V., Eliasson, P., Bolemann, T., Chalot, F.C.: Aeroacoustic Test Cases. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach. NNFM, vol. 128, pp. 671–694. Springer, Heidelberg (2015)
16. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)

17. Hughes, T.J.R., Mallet, M.: A new finite element formulation for computational fluid dynamics: IV A discontinuity-capturing operator for multidimensional advective - diffusive systems. Comp. Meth. in Applied Mech. and Eng. 58, 329–336 (1986)
18. Larchevêque, L., Sagaut, P., Lê, T.-H., Comte, P.: Large-Eddy Simulation of a compressible flow in a three-dimensional open cavity at high Reynolds number. Journal of Fluid Mechanics 516, 265–301 (2004)
19. Normand, P.-E.: Application de Méthodes d'Ordre Elevé en Eléments Finis pour l'Aérodynamique, Thèse de Doctorat de l'Université de Bordeaux (2011)
20. Geuzaine, C., Johnen, A., Lambrechts, J., Remacle, J.-F., Toulorge, T.: The generation of valid curvilinear meshes. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach. NNFM, vol. 128, pp. 15–39. Springer, Heidelberg (2015)
21. Shakib, F., Hughes, T.J.R., Johan, Z.: A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis. Computer Methods in Applied Mechanics and Engineering 75, 415–456 (1989)
22. Shakib, F., Hughes, T.J.R., Johan, Z.: A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. Computer Methods in Applied Mechanics and Engineering 89, 141–219 (1991)
23. Toulorge, T., Geuzaine, C., Remacle, J.-F., Lambrechts, J.: Robust untangling of curvilinear meshes. Journal of Computational Physics 254, 8–26 (2013)
24. van der Vooren, J., Destarac, D.: Drag/thrust analysis of jet-propelled transonic transport aircraft; definition of physical drag components. Aerospace Science & Technology 8, 545–556 (2004)

High-Order Visualization with ElVis

J. Peiro[1], D. Moxey[1], B. Jordi[1], S.J. Sherwin[1], B.W. Nelson[2], R.M. Kirby[2], and R. Haimes[3]

[1] Department of Aeronautics, Imperial College London, UK
{j.peiro,d.moxey,b.jordi11,s.sherwin}@imperial.ac.uk
[2] School of Computing, University of Utah, USA
{bnelson,kirby}@cs.utah.edu
[3] Department of Aeronautics and Astronautics, MIT, USA
haimes@mit.edu

Abstract. Accurate visualization of high-order meshes and flow fields is a fundamental tool for the verification, validation, analysis and interpretation of high-order flow simulations. Standard visualization tools based on piecewise linear approximations can be used for the display of high-order fields but their accuracy is restricted by computer memory and processing time. More often than not, the accurate visualization of complex flows using this strategy requires computational resources beyond the reach of most users. This chapter describes ElVis, a truly high-order and interactive visualization system created for the accurate and interactive visualization of scalar fields produced by high-order spectral/*hp* finite element simulations. We show some examples that motivate the need for such a visualization system and illustrate some of its features for the display and analysis of simulation data.

1 Introduction

High-order simulations are now becoming increasingly common and codes for producing high-order solutions are advancing at a rapid rate but the tools which scientists and engineers use to visualise both their solution data and the underlying high-order geometry on which the problem is solved are still firmly embedded in the world of traditional linear finite elements. Whereas high-order methods typically utilise non-linear polynomials (or other similar non-linear functions) to represent the solution which has been obtained on a given element, the vast majority of visualization software assumes that the data are provided in a linear format, and therefore a conversion is necessary before the data can be viewed.

At worst this poses a major challenge for the adoption of high-order methods and at best does not show the full potential of such methods, for both the developers and end-users of high-order software. From the perspective of a user, whilst it is certainly possible to obtain a visualization of a given simulation, the error that is present when the solution is interpolated to a set of linear functions may yield visualization artifacts which may or may not be present in the actual solution field. For developers this may result in an additional level of debugging which must be undertaken, since it is not always clear whether these errors occur

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_24

from the numerical properties of a given scheme or indeed from a mistake within the code itself.

Additionally, from the perspective of mesh generation, it is clear that high-order meshes are not just desirable but necessary in order to obtain accurate and meaningful solutions. However, without the means of visualising the 'true' surface of a mesh and not its linear interpolation, it is either very difficult or impossible to predict where simulations may encounter issues when considering complex geometries.

Solutions to this problem do already exist. A naive solution is to simply oversample a mesh, using a large number of linear functions to represent the high-order data. This approach may also be considered in combination with an adaptive technique where the mesh is subdivided until a tolerance between the linear interpolation and high-order data is achieved. However, such approaches are expensive in terms of computational time meaning that they may not be interactive with the user, and additionally require large amounts of storage and memory in which to operate.

The purpose of this chapter is to give a brief overview of the Element Visualiser (ElVis) [18], which aims to address these issues through a novel approach that provides interactive and accurate visualization of high-order simulations[1]. The simulations are represented by spectral/hp finite element fields that are produced by the continuous or discontinuous Galerkin methods. ElVis was originally created by B.W. Nelson, as part of his PhD at the University of Utah [3]. Currently ElVis is actively maintained by R.M. Kirby at the University of Utah and R. Haimes at MIT, and has been the focus of our development efforts on high-order visualization throughout the course of IDIHOM. We outline the features of ElVis and give examples which demonstrate its advantages over traditional linear visualization software.

2 The Importance of Visualization Accuracy

The main reason for the development of an accurate high-order visualizer is that linear approximations where high-order data is sampled onto linear primitives for visualization does not represent high-order data well and results in visualization errors. We illustrate this with two examples where we compare high-order and linear visualizations[2].

2.1 Visualization Errors or Artifacts?

We consider the simulation of the compressible viscous flow past a half delta-wing geometry with a symmetry plane running down the centre chord-line of the

[1] The program is accessible at `www.sci.utah.edu/download/elvis`

[2] All the linear visualizations in this chapter have been performed with the code Visual 3 which is available at `raphael.mit.edu/visual3`

wing. The delta-wing geometry is linear and the computational mesh consists of 19410 quadratic tetrahedral elements with straight edges. The numerical solution has been obtained via a discontinuous Galerkin solver[3]. Figure 1 depicts a linear visualization of the solution on a plane cut located at the trailing edge of the wing. The two insets in the figure highlight areas where visualization errors or artifacts might be present. To determine the nature of these visual features requires an increase of the accuracy of the visualization.

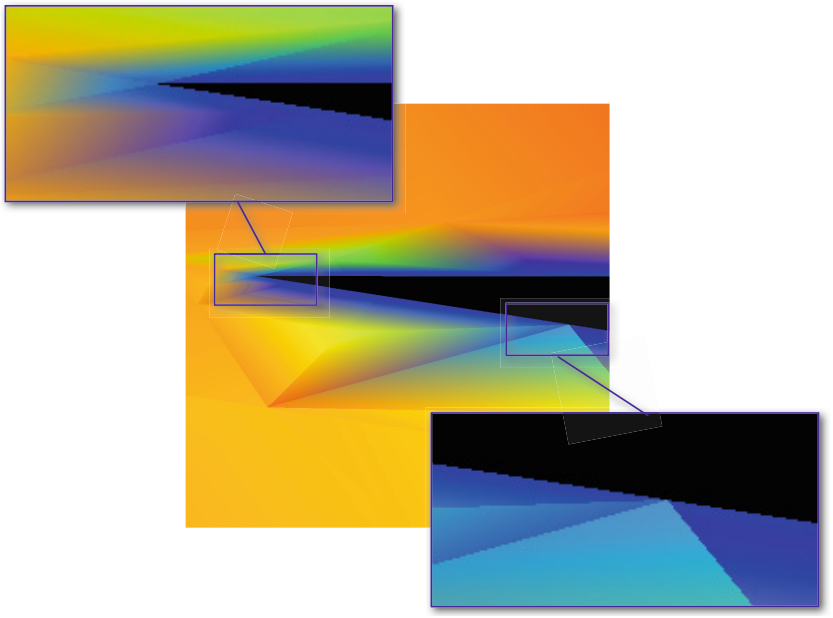


Fig. 1. Cut-surface at the trailing edge of a delta wing simulation displayed using a piecewise linear approximation. The two inset figures show enlargements of the areas where visualization errors or artifacts might have occurred.

Focusing in the area near the edge of the wing, Figure 2 depicts a comparison of the linear visualization with a pixel exact display of the solution using ElVis which shows that the visual effect of the linear visualization is due to a lack of resolution of the boundary layer.

On the other hand, the comparison of linear and high-order visualizations in Figure 3 illustrate that the sharp changes observed in both are due to the discontinuous Galerkin approximation employed by the solver.

[3] Numerical solution computed by the Project X code: `raphael.mit.edu/projectx.html`

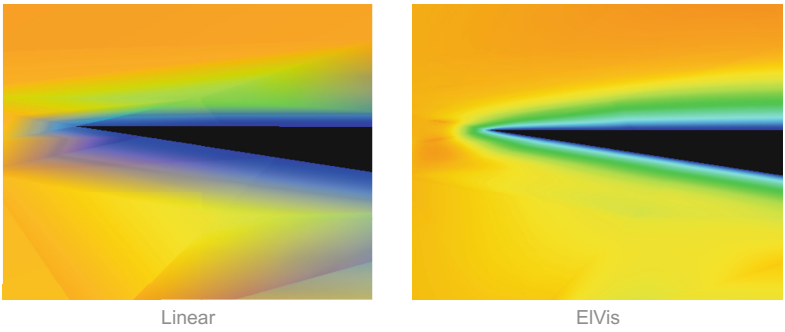


Fig. 2. Visualization errors: the linear visualization (left) shows apparent under-resolution of the boundary layer, whereas in reality the boundary layer is clearly present (right)

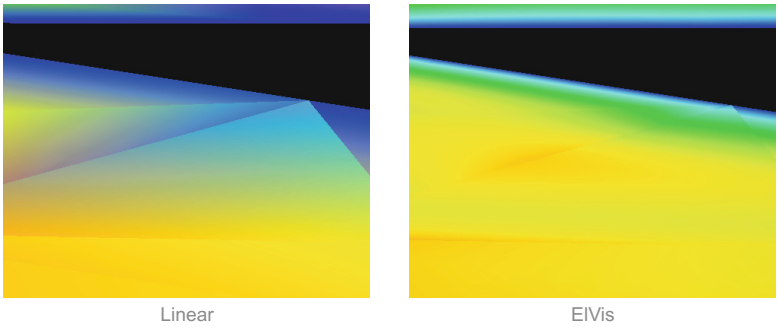


Fig. 3. The sharp changes observed in both the linear visualization (left) and the high-order visualization (right) are part of the piecewise discontinuous approximation of the solution

2.2 Visualization for Code Debugging

Visualization tools can play a role in detecting bugs and errors within the solver. Figure 4 shows several visualizations of isocontours of the distance function $d(\boldsymbol{x})$, a scalar function which defines the minimum distance between a point $\boldsymbol{x}$ and a surface S. This is an important function used in, for example, turbulence models which rely on this quantity to in order to estimate stresses on the flow due to viscous effects near walls. Errors in this function may, in the worst cases, substantially affect physical observations or indeed prevent the simulation from converging; in other cases however the effect can be minimal or non-existent.

Figure 4 shows how the use of linear visualization tools can prevent the detection of bugs in distance calculation routines. The top row shows results with a code which has a bug in its distance calculation routine, and the bottom where the offending bug has been fixed. On the left-hand side we see that the visual-

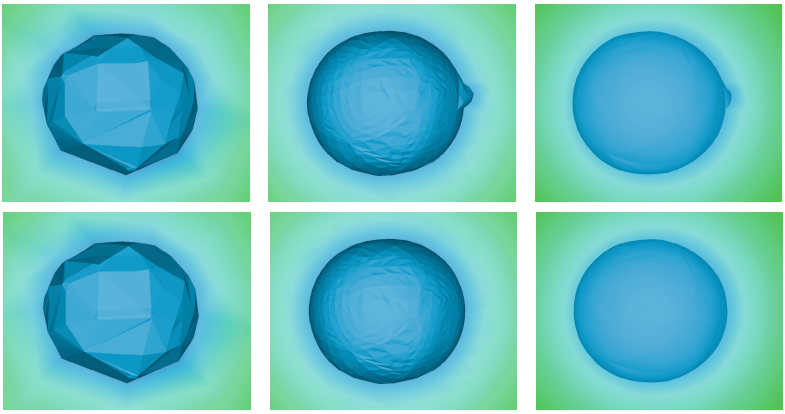


Fig. 4. Visualizations indicating a bug relating to the distance function. The top row shows a version of the code where the bug is present, and the bottom where the bug is fixed. From left-to-right: linear interpolation of elements, linear interpolation with one level of refinement, ElVis. Note that the linearly interpolated elements without refinement appear identical.

ization is sufficiently coarse that no problem is detected even after the bug has been fixed. When the linear mesh is refined (middle), shows a small protrusion – however, the shape can still be inferred to be reasonable given the original linear image. ElVis on the other hand shows an obvious protrusion from an otherwise smooth surface. This example clearly demonstrates how using only linear tools to visualise data can lead to issues not being detected.

3 ElVis and Its Main Features

The previous examples have shown that linear visualization tools operating on high-order data can introduce errors. The main motivation for the development of a truly high-order visualizer is the realization that these errors can be minimized if we take into account our knowledge of the data. In recent years, a series of tools have been developed in an attempt to address the lack of appropriate visualization software for high-order data. A key issue with these tools is that they have been developed with the presumption of a single type of basis function to be used in each element. Whilst a conversion of high-order data may be possible from one data format to another if each tool assumes a polynomial expansion, the wide variety of non-linear basis functions that can be employed in high-order methods means that this is not always possible, and indeed for users can be a time-consuming task.

A further issue with existing visualization software is that if often comes with the drawback of high computational cost. Whilst it is clear that high-order visualization demands additional processing when compared to linear interpolation,

existing software can require the use of expensive custom-made hardware in order to produce visualizations in reasonable amounts of time for users, particularly for large three-dimensional problems. Since interactivity is a key part of the visualization process, this is a significant problem which must be overcome.

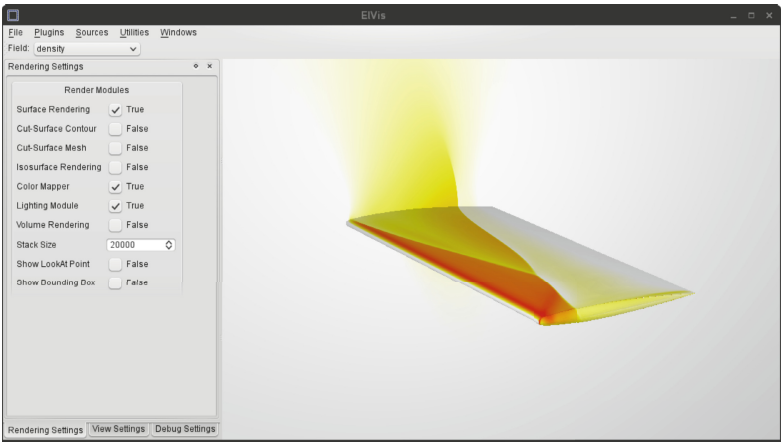


Fig. 5. ElVis' graphical user interface, with a visualization of density on an ONERA M6 wing

ElVis has been designed to address each of these issues. Firstly, its design is deliberately modular, with the intention that the code responsible for visualization should be separated from the implementation and definition of the high-order basis functions. The way that this is achieved is through the use of an extensible plugin architecture that provides a simplified interface to the high-order data which exposes a minimal amount of functionality required to generate a visualization. This makes ElVis broadly applicable to a wide range of tasks in high-order visualization. The main modules of ElVis are: a *Plugin Module* that handles communication between ElVis and the simulation package, a *Visualization Module* that performs the visualization of the high-order data and a *Data Exploration Module* that gives users the ability to interactively explore the high-order data. The graphical user interface for the visualization and data exploration modules of ElVis is shown in Figure 5.

In order to realise the goal of performance and interactivity, ElVis utilises NVIDIA GPU hardware in combination with CUDA and the OptiX ray-tracing engine in order to provide real-time visualizations. Once data is available within ElVis, either through a data conversion or runtime plugin, selected GPU algorithms for the most common tasks of cut-surfaces/contour lines, volume rendering and isosurface generation have been generated in such a way as to obtain the required visualization accuracy from the underlying high-order data.

3.1 High-Order Elements

The numerical solution in ElVis is represented by high-order spectral/*hp* finite element approximations via continuous or discontinuous Galerkin methods. For these methods, the computational domain is tessellated into elements. The four basic element types are the hexahedron, prism, tetrahedron, and pyramid. These elements are defined through a mapping, $T : R^3 \rightarrow R^3$, between a standard reference element, the cube $\Omega_{st} : [-1, +1]^3$, in a *reference space* and the high-order element, Ω^e, in a *physical space* as illustrated in Figure 6.

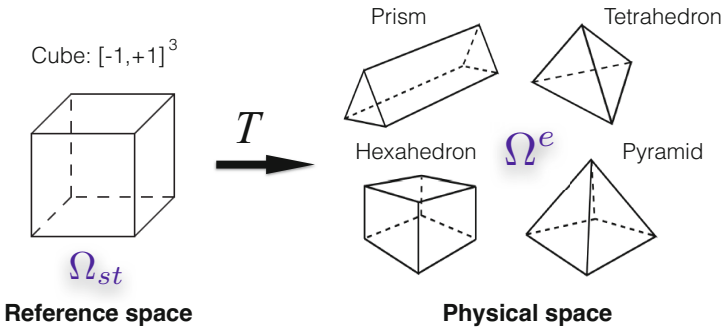


Fig. 6. Notation used for spectral/*hp* finite elements in ElVis

The high-order representation of the solution within the finite element is a polynomial function evaluated in tensor-product fashion in the reference element. The expressions for the nodal and modal expansions used to approximate the solution within the elements are given in [1][4]. If the high-order elements employed by the flow solver have different polynomial basis functions then a data conversion process can be performed to the set of spectral/*hp* standard basis functions. This minimises the amount of coding required. Otherwise, more complex functions can be implemented directly as a runtime plugin.

ElVis uses the representation of the flow field via the spectral/*hp* basis functions and applies ray-tracing ideas to achieve a pixel-exact images in which the visualization algorithm is applied to each pixel. An overview of how ElVis achieves pixel-exact images is given in the following sections.

3.2 Isosurfaces

Perhaps one of the most often used visualizations of a data set is the isosurfaces or contour lines which depict where a given field attains a particular value. ElVis adopts an elemental approach to isosurface rendering [2,5], whereby the field is

[4] These are implemented in the open-source code Nektar++ available at `www.nektar.info`

projected onto a polynomial as a ray passes through the volume so that finding the isosurface becomes a root finding problem. This is illustrated in Figure 7.

The solution along the ray, assumed to be smooth within the element in the physical space, is approximated by a Legendre polynomial of order M. Given the entry and exit points of the ray within the element, a set of Legendre-Gauss-Lobatto quadrature points is located along the ray. The number of quadrature points is a function of the approximating polynomial order and it is chosen so that inner products of polynomials of degree M are within round-off error. The high-order finite element function is evaluated at the quadrature points along the ray and these values are used to interpolate a polynomial. The roots of this polynomial which correspond to the sought values of the isosurface are evaluated using the QR root-finding algorithm. This type of projection introduces error, but it converges to the field as we increase the order, so we can increase the order until we achieve pixel-exact images. A more detail description of the algorithm and analysis of the associated errors in given in [5].

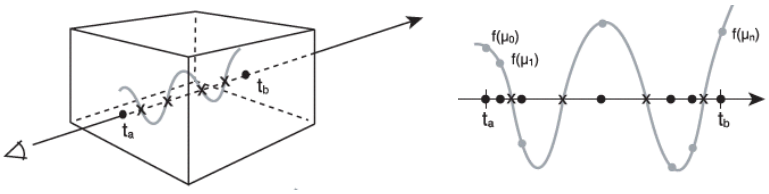


Fig. 7. Isosurface evaluation within an element in ElVis. A projected polynomial is formed along the ray, the scalar field within the element is sampled along the ray to interpolate a polynomial and its roots give the values of the isocontour sought.

One of the advantage of this method over more traditional isosurface rendering techniques such as the marching cubes algorithm is that it permits the visualization of discontinuities within the isosurfaces across element boundaries if this is present in the underlying formulation. An example of this can be seen in Figure 8 where the inset figure shows the discontinuities in the numerical solution.

ElVis also supports the rendering of contour lines, as shown in figure 9. We again see that linear elements, even when heavily subdivided, do not yield the same quality of visualization, and indeed do not accurately depict the true regularity of the solution field.

3.3 Surface and Cut-Surface Visualizations

Possibly the most widely used visualization tool is that of a surface rendering, where the solution field is restricted to a particular surface or cut-surface and a colour map is applied to show variation in the solution. Surfaces are simple to visualise and offer a clear indication of how the solution field is behaving at

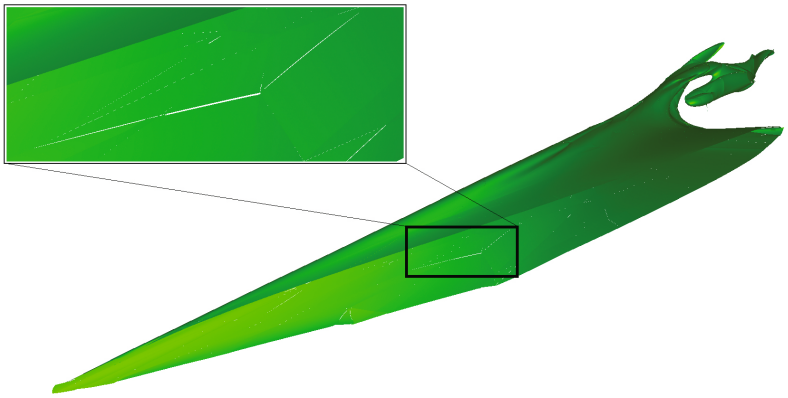


Fig. 8. Isosurfaces of the Mach number of a delta wing simulation. Cracks are visible within the surface of this isocontour due to the use of a discontinuous Galerkin formulation which are accurately rendered by ElVis.

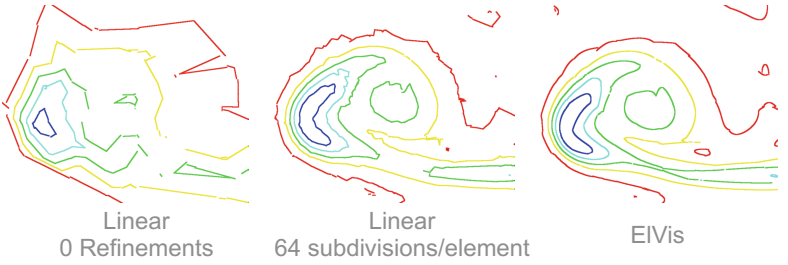


Fig. 9. Contour plots demonstrating the apparent discontinuity and lack of regularity that linearly interpolated visualization shows in the solution field

specific points in the domain. ElVis supports the use of multiple cut planes, as well as the visualization of solution fields on curvilinear mesh surfaces (as depicted later).

The algorithm that renders each surface is again based on ray-tracing. The first step is to calculate the intersection of the ray with the cut-surface which might be defined as an implicit or a parametric surface. Apart from simple cases such as a plane, finding the location of the intersection point is a non-linear problem that must be solved via iteration. Once an intersection point is located, the element containing the point is found and the solution field evaluated using the element basis functions at the corresponding coordinates in the reference space. It is worth noting that finding the coordinates in the reference space for curved elements is a non-linear problem that requires an iterative method. A more detailed description of the algorithms involved in this process is given in reference [4].

Figure 10 shows the effect that high-order rendering using ElVis has on the visualizations for a delta wing mesh. We see that the linearly interpolated solutions in Figures 10(a) and 10(b) present an unclear rendering of the flow characteristics, and in particular the boundary layer is nearly impossible to detect, which may lead one to believe that the simulation is under-resolved. In Figure 10(c) however, ElVis gives a clearer picture of the flow behaviour, and the boundary layer is clearly identifiable.

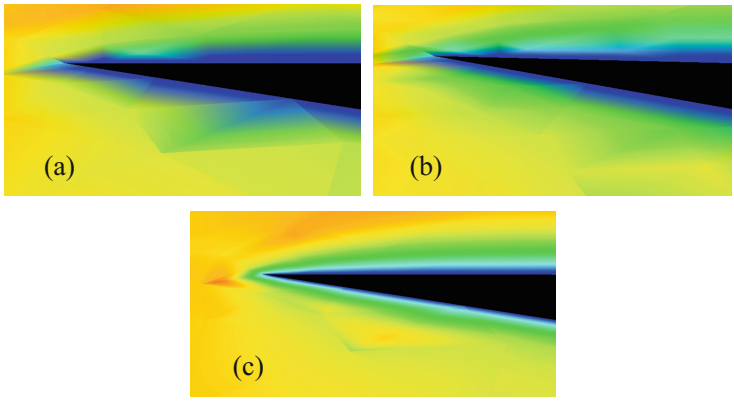


Fig. 10. Linear versus high-order visualization of planar cuts: (a) Linear visualization without subdivision; (b) linear visualization with subdivision; (c) high-order visualization

3.4 Volume Rendering

It is often the case that the visualization of noisy data, as arises in for example turbulence simulations, can be difficult. When this occurs, it can be useful to instead consider a volume rendering which can visualise the isocontours in a way that is easier to interpret. Volume rendering in ElVis is based on the emission-absorption optical model [6] in which the irradiance is evaluated along rays through the integral

$$\int_a^b \kappa(t)\,\tau(t) e^{-I(t)}\,dt\,; \quad I(t) = \int_a^t \tau(u) du \tag{1}$$

where $[a, b]$ is denotes a parametric interval along the ray and $\kappa(t)$ and $\tau(t)$ are the colour and density transfer functions, respectively.

The evaluation of the integrals in equation (1) is performed element by element. This requires a traversal of the mesh and the calculation of *all* the intersections of the ray with the boundary of the elements to determine the ray segments contained within the element. This operation is complicated by the presence of curved element since a ray might intersect a curved element face several times. For a ray segment, the colour and density transfer function falls

into one of three categories: empty, smooth and piecewise smooth. The evaluation of the volume rendering integral (1) is handled on a case-by-case basis and appropriate quadrature rules are use for each of these categories. One downside of integrating functions that are piece-wise smooth is that the best order of convergence that can be achieved is second order.

An example of a typical volume rendering produced by ElVis can be seen in figure 11. This figure also illustrates the importance of selecting an appropriate quadrature for the evaluation of both integrals in equation (1).

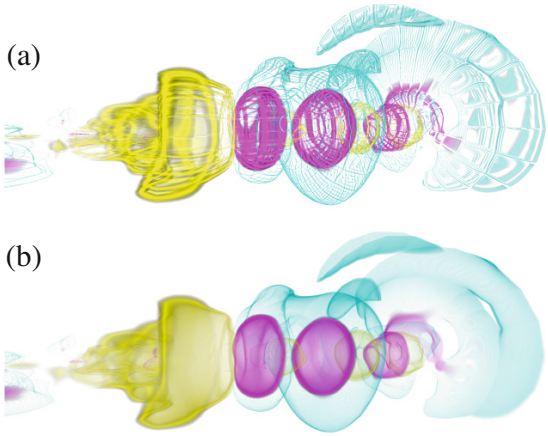


Fig. 11. Effect of quadrature on the accuracy of volume rendering: (a) Riemann integration and (b) high-order integration, both with 60 million samples

A more detail description of the algorithms and their performance is given in reference [6].

3.5 Mesh Visualization

The generation of curvilinear meshes for high-order simulations is a challenging and difficult issue. However, this problem is made more difficult by the fact that it is often impossible, extremely difficult, or computationally prohibitive to accurately visualise the surface of a curvilinear mesh. Whilst linearly interpolated visualizations may give an overall idea of the mesh surface, small oscillations may remain hidden, as the example of Section 2.2 (and figure 4) demonstrate.

One of the core capabilities of ElVis is to assist in the visualization of curvilinear meshes. Figure 12 shows a curvilinear hemisphere mesh in combination with a cut-plane, which also demonstrates the use of several simultaneous visualization techniques. We clearly see that the introduction of curvature yields two very different images. Additionally, we note that even when subdivision or refinement is used to more clearly visualise curvature using linear visualization

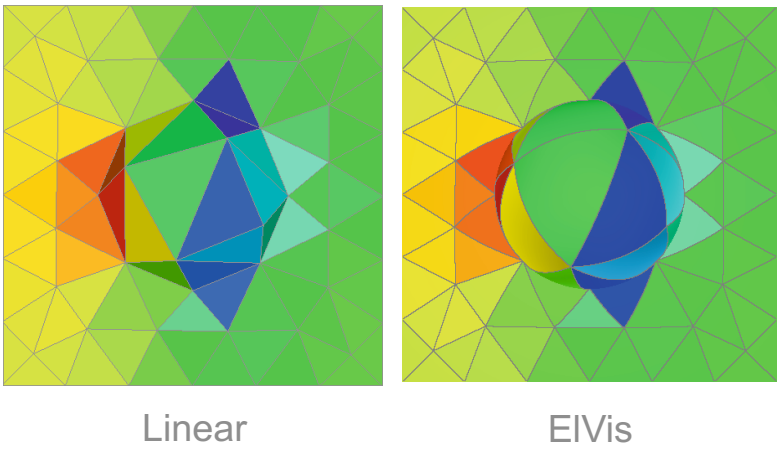


Fig. 12. A visualization of a curvilinear hemisphere using linearly interpolated elements (left) and ElVis (right)

software, this usually causes confusion in determining where the high-order elemental boundaries lie, since additional faces are drawn wherever refinement occurs. ElVis therefore gives an accurate representation of the features of the numerical discretisation.

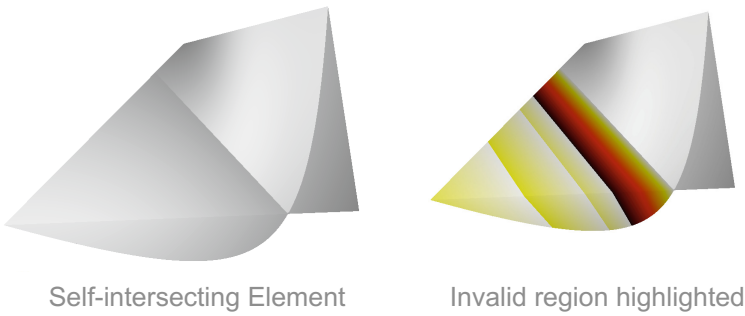


Fig. 13. Visualization of a self-intersecting tetrahedron, where on the right ElVis is used to highlight areas where the element is invalid

We conclude with a final observation regarding mesh generation and how visualization tools such as ElVis provide a valuable debugging tool. Curvature in an element Ω^e is usually imposed through the use of a mapping T from a reference element Ω_{st}, which is necessarily required to be invertible and preserve orientation so that the determinant of the Jacobian of T is positive at all points

within Ω_{st}. However, it is often the case that when curvature is imposed in the mesh generation procedure, self-intersection can occur, causing the resulting mesh to be invalid and be unsuitable for simulations. For linear finite elements, detecting invalid elements is an easy task – however at higher polynomial orders, the task turns into a multi-dimensional root finding problem which is extremely costly.

An alternative approach is to select a distribution of nodal points at which the Jacobian determinant is evaluated and is confirmed to be positive. However, this does not give a guarantee that the element is valid. High-order visualization tools such as ElVis are therefore a powerful tool if it is suspected that the mesh may contain negative Jacobians. Figure 13 shows how the Jacobian function can be visualised to highlight areas in elements which are invalid.

4 Conclusions

We have given an overview of the various capabilities and features of the Element Visualizer ElVis, an integrated visualization system designed specifically for high-order finite element solutions. ElVis has an extensible architecture that supports data originating from arbitrary simulation systems and its visualization algorithms are decoupled from the data representation. This provides accurate visualization and avoids introducing error into the final image by operating on the high-order data directly. ElVis also uses the parallel processing capabilities of recent Graphics Processing Units (GPU) to provide interactive visualizations on desktop computers.

We have shown that high-order visualization is not only desirable, but necessary in order to truly appreciate the solution data that high-order methods produce, and indeed to visualise the geometry on which such methods are used. Furthermore, these techniques are crucial not only to visualise solution data but can play a crucial role in diagnosing problems and also in the mesh generation process. It is clear that if high-order methods are to be more widely adopted amongst industry, the development of tools such as ElVis is essential.

References

1. Karniadakis, G.E., Sherwin, S.J.: Spectral/*hp* element methods for computational fluid dynamics, 2nd edn. Oxford University Press (2005)
2. Meyer, M., Nelson, B.W., Kirby, R.M., Whitaker, R.: Particle systems for efficient and accurate high-order finite element visualization. IEEE Transactions on Visualization and Computer Graphics 13(5), 1015–1026 (2007)
3. Nelson, B.W.: Accurate and interactive visualization of high-order finite element fields. PhD thesis, School of Computing, The University of Utah, Salt Lake City (August 2012)
4. Nelson, B.W., Haimes, R., Kirby, R.M.: GPU-based interactive cut-surface extraction from high-order finite element fields. IEEE Transactions on Visualization and Computer Graphics 17(12), 1803–1811 (2011)

5. Nelson, B.W., Kirby, R.M.: Ray-tracing polymorphic multi-domain spectral/hp elements for isosurface rendering. IEEE Transactions on Visualization and Computer Graphics 12(1), 114–125 (2006)
6. Nelson, B.W., Kirby, R.M., Haimes, R.: GPU-based volume visualization from high-order finite element fields. IEEE Transactions on Visualization and Computer Graphics 20(1), 70–83 (2014)
7. Nelson, B.W., Liu, E., Kirby, R.M., Haimes, R.: ElVis: A system for the accurate and interactive visualization of high-order finite element solutions. IEEE Transactions on Visualization and Computer Graphics 18(12), 2325–2334 (2012)

Direct Visualization of Piecewise Polynomial Data

T. Bolemann[1], M. Üffinger[2], F. Sadlo[2], T. Ertl[2], and C.-D. Munz[1]

[1] Institute for Aerodynamics and Gasdynamics, University of Stuttgart
Pfaffenwaldring 21, 70569 Stuttgart, Germany
{bolemann,munz}@iag.uni-stuttgart.de
[2] Visualization Research Center, University of Stuttgart
Allmandring 19, 70569 Stuttgart, Germany
{ueffinger,sadlo,ertl}@visus.uni-stuttgart.de

Abstract. High order methods are regarded as a primary means to significantly improve the efficiency of numerical techniques. While the particular high order methods can be very distinct, most have in common that their solution is represented by piecewise polynomials. However, since high order methods evolved only recently, most of the present visualization techniques are not suited for the needs of resulting simulation data; they are predominantly based on tensor product linear interpolation and applying them to high order data is nowadays accomplished by static resampling, involving prohibitive storage and computation costs, and providing at most sufficient results. In this paper, we describe two of our existing visualization approaches and discuss some of the involved problems and concepts, and exemplify them using different high order simulation results.

Keywords: High order methods, direct volume rendering, GPU-based visualization, piecewise polynomial data, unstructured grids, parallel rendering.

1 Introduction

In the development of the next generation of numerical methods for fluid dynamics, the main goals are to reduce computational cost significantly and to offer a higher accuracy—in short, to generally raise the efficiency of the method over today's second order finite volume schemes. Schemes with a high order of accuracy, amongst those the herein presented discontinuous Galerkin (DG) schemes, are commonly regarded as a promising way to achieve these goals. Initially introduced by Reed and Hill [21], a considerable amount of research has been carried out to lay the theoretical groundwork for the method and drive it towards industrial applicability [1, 3, 6, 14]. While current DG implementations are already very mature, major tasks remain in the field of mesh generation and post-processing. An issue, which has not yet been sufficiently addressed in the past, is the visualization of data generated by DG codes. There, the DG solution

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_25

is represented as a piecewise polynomial, continuous inside the cell, but discontinuous at its boundaries. Today's visualization tools, however, mostly rely on tensor product linear interpolation, and are thus not directly applicable to DG solutions. A common way to visualize these data is therefore to resample them and to apply those visualization tools to the resampled approximation. However, due to the high variation of high order data, a conservative resampling would involve prohibitive cost in storage and computation, and therefore in practice the overall approach leads to severe deterioration of the original data. Some existing approaches in the field of visualization of high order data are the Gmsh framework [12,22], relying on an error-adaptive resampling of the high order solution and the ElVis framework [18]. In this paper, we report on two of our techniques for high order visualization: direct volume rendering [24] and isosurface extraction [20]. Further examples for high order visualization techniques include the extraction of ridge lines and vortex core lines [19], and direct visualization of Particle-Partition of Unity data [25].

We demonstrate the capabilities and the flexibility of our visualization approaches on three typical benchmark cases from fluid dynamics, where we analyze high order datasets, obtained from two different discontinuous Galerkin CFD codes. The first benchmark case is a laminar flow around a sphere. This is clearly the most demanding dataset regarding the visualization, as it has been computed on an unstructured curved mesh with $30,000$ elements containing hanging nodes. Furthermore, p-adaptivity has been applied during the simulation, resulting in a polynomial degree varying throughout the domain in a range from $p = 2$ to $p = 6$.

In the second test case, a Large Eddy Simulation of a round jet is analyzed, which is a shear layer flow covering a broad range of turbulent scales [5]. The results have been computed at a Mach number of $\mathrm{Ma} = 0.9$ and $\mathrm{Re} = 5,000$ using a hexahedral mesh with $70,000$ elements and a polynomial degree of $p = 5$, which results in 13.0 mio degrees of freedom. Due to the high Mach number the simulation has been stabilized using a Smagorinsky model.

For the third example, we selected the Taylor-Green vortex, which is a commonly used benchmark problem for turbulence investigations [8]. It features a turbulent energy cascade, which is used to study the effects of turbulent transition and decay. While the mesh consists of only 512 elements, we employed a very high order of $\mathcal{O}(16)$, resulting in 2.1 mio degrees of freedom.

The computations have been performed with two distinct inhouse DG frameworks. The flow around a sphere has been computed using the HALO framework [10,11], the round jet and the Taylor-Green vortex using the Flexi framework [9,13].

2 Discontinuous Galerkin Schemes

In this section, we give a brief overview over the DG schemes, which were used to compute the results presented in this work, and describe the fundamental challenges they pose to the visualization. The first DG framework HALO is based

on a mixed nodal/modal formulation by Gassner et al. [11]. It supports fully unstructured hybrid meshes and utilizes acceleration techniques like adaptive mesh refinement (h) and polynomial adaptivity (p), both termed hp-adaptivity. Mesh adaptation is a very complex task, as modifications are required to be as local as possible, the code is thus capable of handling meshes with hanging nodes. It furthermore uses an explicit local time-stepping approach to reduce computation times.

The second framework Flexi is based on a discontinuous Galerkin Spectral Element (DGSEM) formulation by Kopriva and Gassner [9], which uses a collocation of solution points and integration points. Due to its formulation, it is limited to hexahedral elements, but can use unstructured meshes with hanging nodes. It has been designed for massively parallel computations and supports very high polynomial orders, e.g., as the aforementioned $\mathcal{O}(16)$ computations of the Taylor-Green vortex.

To illustrate the basic ideas, we derive a DG scheme for the mixed nodal/-modal approach for a scalar nonlinear conservation law

$$u_t + \nabla_x \cdot \mathbf{f}(u) = 0, \tag{1}$$

where u is the conserved variable, $\mathbf{f}(u)$ the flux function and the subscripts $\mathbf{x}$ and t denote the spatial and temporal derivatives. We approximate our solution within a grid cell Q by a polynomial

$$u(\mathbf{x}, t) = \sum_{i=1}^{N} \hat{u}_i(t)\varphi_i(\mathbf{x}), \tag{2}$$

where $\hat{u}_i(t)$ denote the degrees of freedom and N is the dimension of the polynomial space with $N = \frac{(p+d)!}{p!d!}$, where p is the polynomial degree and d the spatial dimension. In case of the HALO code, we use orthonormal modal basis functions $\varphi(\mathbf{x})$ to represent the solution, which possess the property

$$\int_Q \varphi_i(\mathbf{x})\varphi_j(\mathbf{x})\,\mathrm{d}\mathbf{x} = \delta_{ij}. \tag{3}$$

As a next step, we derive the weak formulation for a cell and therefore multiply the conservation law with a test function $\phi(\mathbf{x})$, integrate over Q, and perform integration by parts

$$\int_Q u_t\phi(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int_{\partial Q} \mathbf{f}^*(u^+, u^-)\,\phi(\mathbf{x})\,\mathrm{d}s - \int_Q \mathbf{f}(u)\cdot\nabla\phi(\mathbf{x})\,\mathrm{d}\mathbf{x} = 0. \tag{4}$$

$\mathbf{f}^*(u^+, u^-)$ denotes the numerical flux function normal to the cell boundary, depending on the two distinct states u^- inside the local grid cell and u^+ of the adjacent cell. Following the Galerkin approach, we choose the test function $\phi(\mathbf{x})$ identical to the basis function $\varphi(\mathbf{x})$. We evaluate the integrals using a Gauss quadrature rule and therefore have to evaluate the modal representation of the

solution at nodal points. In contrast to this approach, the DGSEM scheme relies on a purely nodal representation of the solution throughout the scheme and collocates solution points and quadrature points, thus being simpler and more efficient.

During the simulation, we benefit from the fact that all operations are performed on a static set of points and thus many operations can be saved by precomputing data at the beginning of the computation. For the visualization process however, the solution has to be evaluated at arbitrary points, depending on the viewpoint. It is therefore preferable to use another set of basis functions for interactive visualization. As numerical inaccuracies are negligible in the visualization process, the presented tools use monomial basis functions defined in the barycenter. Prior to the visualization process the solution data is converted into this form.

A further characteristic of DG methods in general is their requirement for a high order geometry representation, to retain their high order accuracy near curved boundaries [2]. This requirement effectively leads to grid cells that can be arbitrarily curved and can be present not only along curved boundaries, but throughout the computational domain, posing further challenges on the visualization side. We see, that aside from the pure polynomial representation of the solution, many other aspects have to be considered, making the visualization of data generated by advanced discontinuous Galerkin codes a nontrivial task.

3 Visualization

Traditional visualization techniques are based on tensor product linear interpolation, i.e., bilinear in 2D and trilinear in 3D. Although this approach has proven successful in a wide area of research and applications, direct visualization of high order simulation results is not amenable by these techniques. Hence, it has been practice to resample the high order data, typically on regular grids, and to subsequently apply those traditional techniques for visualization of the resampled data. This, however, introduces manifold drawbacks, especially with respect to accuracy, memory consumption, and speed. In particular, there is the problem that the fine structure is typically missed due to the difficulty of finding an appropriate sampling that captures all detail at sub-cell resolution.

The building blocks for high order visualization include direct evaluation of the high order data within the complete visualization pipeline. Because this is computationally demanding, graphics hardware is employed for evaluation. Hence, one needs to transfer the piecewise polynomial data to the graphics card in an appropriate representation, evaluate them, and construct the visualization representation and visualize these, all on graphics hardware.

Scientific visualization offers a multitude of different techniques for the analysis of simulation results. Most common is the extraction of a subset of the 3D domain as a geometric representation, e.g., cutting planes, and the subsequent rendering of the geometry using one or more attributes of the field solution for color mapping. Such traditional geometric methods are mostly limited to linear, or low order interpolation. However, there has been some effort to enable

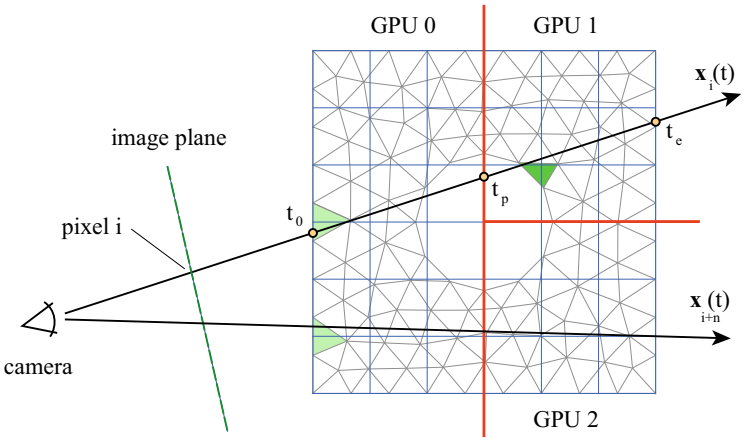


Fig. 1. Distributed raycasting of an unstructured grid with holes. Here, the domain is partitioned into three convex regions. The workload is distributed by assigning each region to one GPU, e.g., GPU 0 raycasts segment $[t_0, t_p]$ and GPU 1 raycasts segment $[t_p, t_e]$ of ray $\mathbf{x}_i(t)$. A spatial acceleration datastructure, depicted as the blue regular grid, enables efficient detection of the grid entry cells (light green) and the re-entry cells (dark green) after holes.

the visualization of high order field solutions on geometric representations [17]. In our work, we concentrate on high order visualization using other techniques: direct volume rendering with raycasting (Section 3.1) and feature extraction (Section 3.2).

3.1 Direct Volume Rendering

The rendering of field data given on a 3D domain inherently has to deal with occlusion issues due to the mapping of the 3D field to the 2D image representation displayed on the screen. A set of isocontours, for example, usually captures only a subset of the domain. This subset gets denser with an increasing number of isolevels, however, simultaneously, visual clutter and occlusion become a limiting factor. Additionally, extracting and storing the geometry of a large number of isocontours may be computationally too costly. Direct volume rendering (DVR), in contrast, enables a direct dense visualization of volumetric field data by visualizing the volumetric field in a semi-transparent manner. It enables the user to highlight interesting features of the data set by directly mapping their values to opaque color and still allows to display the spatial context of the features by using more transparency for their vicinity. The color mapping is established by the transfer function, which maps field values $f(\mathbf{x})$ at a sample location $\mathbf{x}$ to color $\mathbf{c}$ and opacity τ, offering the user an intuitive interface for defining the desired exploration task.

Splatting is one example for an object-based DVR technique. It employs the representation of the volume data by means of 3D interpolation kernels by

projecting these kernels to the 2D image plane on which the color and opacity contribution of the 3D data set is accumulated in a 2D convolution process. Volume rendering often employs a physically motivated light emission and absorption model, given as the rendering integral:

$$\mathbf{I} = \int_{t_0}^{t_e} \mathbf{c}(f(\mathbf{x}(t)))\tau(f(\mathbf{x}(t)))e^{-\int_{t_0}^{t} \tau(f(\mathbf{x}(t'))) \, \mathrm{d}t'} \, \mathrm{d}t, \tag{5}$$

with $\mathbf{x}(t)$ being a ray shot through the volume. With this volume rendering model the cells have to be projected in visibility order, requiring a costly sorting, e.g., in curvilinear or unstructured grids.

Raycasting is an image-based DVR technique, which typically approximates (5) with a Riemann sum. Viewing rays are shot from a virtual camera through the pixels of the image plane, entering the simulation domain at $\mathbf{x}(t_0)$ and leaving it at $\mathbf{x}(t_e)$, see Figure 1. Along those rays, the field solution f is sampled, the transfer function is applied to each field sample, and finally, the color contribution $\mathbf{I}$ is accumulated. This approach naturally ensures a front-to-back or back-to-front ordering of the samples.

Traditional volume raycasters for grid-based simulation data [7] employ linear interpolation techniques to reconstruct a dense representation of the field solution. This enables interactive display rates of large volume data sets with modern multi-core compute hardware (e.g., GPUs [23]). The development of our interactive raycaster for *hp*-adaptive simulation data with curved elements and a polynomial field solution of high order [24] involved multiple challenges:

1. The computational effort of evaluating the field polynomials increases with the order of the solution. Additionally, a very large number of samples, at locations depending on the camera setup, need to be evaluated along the rays. Thus, an appropriate *polynomial representation* has to be chosen, fitting the time-critical evaluation scheme of the raycaster.
2. Efficiency is also critical during the *traversal of the grid*. The lookup of the grid cell containing the location of a field sample has to be efficient.
3. The *placement of the field samples along the rays* is also critical. Due to performance reasons, a brute-force regular sampling with a very fine step size would be prohibitive.

Our raycaster addresses the performance challenge by employing parallelization on multiple levels. Firstly, the raycasting is implemented in CUDA, a language for programming the massively parallel GPU architecture. Our CUDA raycasting code performs the sampling of the volume along distinct rays in parallel, as rays x_i and x_{i+n} in Figure 1 can be computed independently from each other. Additionally, as illustrated in the figure, the raycaster is designed for the parallel execution on a GPU compute cluster. Therefore, the simulation domain is partitioned, with each compute node raycasting a small convex subset of the field. A dynamic load balancing strategy tries to ensure an even load when using a larger number of compute nodes. The convex partitions have to be sorted in

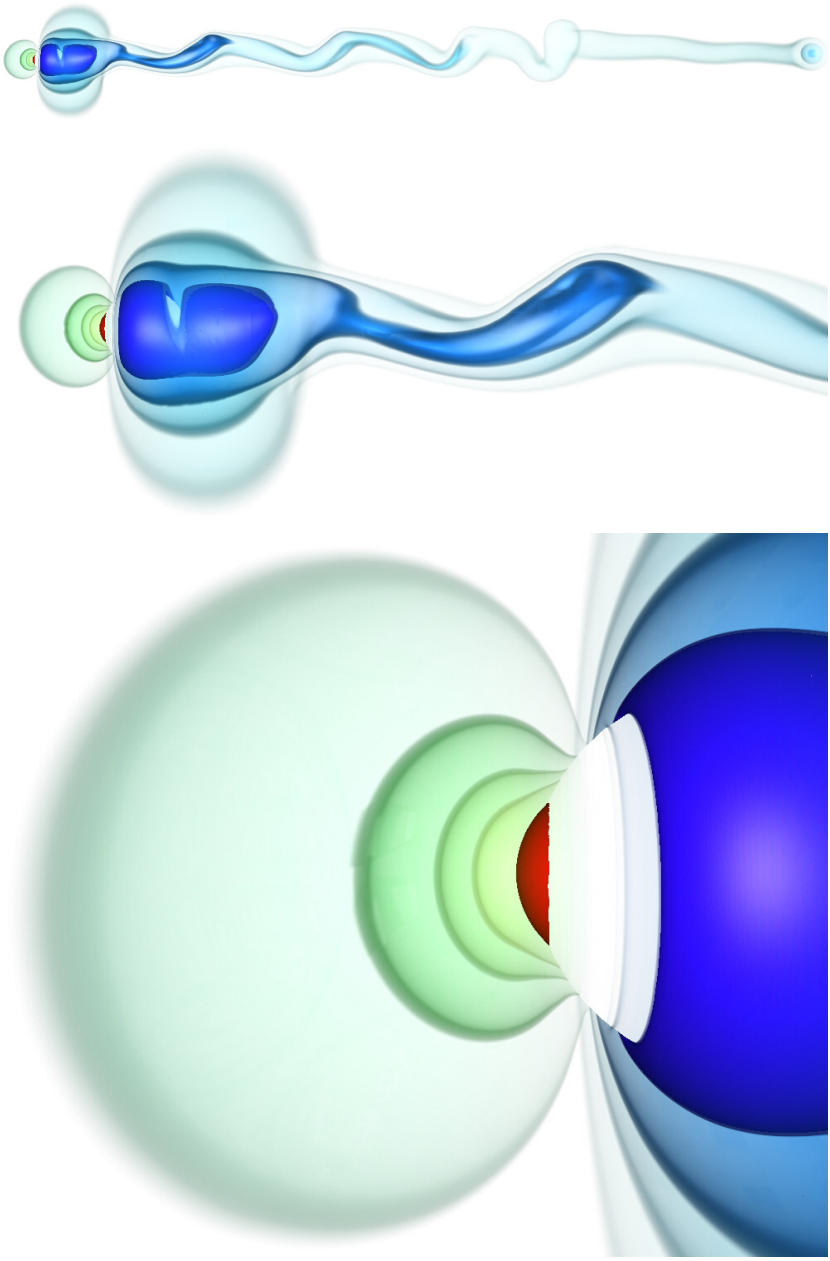


Fig. 2. Direct volume renderings of the density field of the *hp*-adaptive laminar flow around a sphere

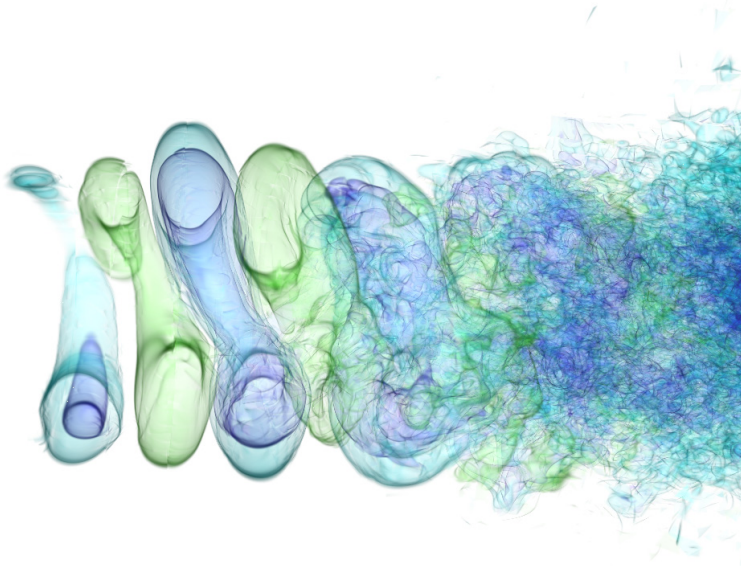


Fig. 3. Density levels of a computation of a turbulent jet with $p = 5$, visualized with high order volume rendering

visibility order and their image results can then be blended appropriately to obtain the final image.

With the monomial field representation, described in Section 2, an efficient iterative evaluation of the field solution along the rays becomes possible. Thereby, the basis functions do not need to be stored explicitly for the monomial representation. This is advantageous with respect to compact storage and the upload of the field solution to the GPU. Additionally, by representing the solution polynomials in physical space, the costly inversion of the geometry mapping is avoided during the time-critical raycasting stage.

Spatial acceleration data structures (Figure 1) are used to ensure efficient traversal and partitioning of the complex grids. The GPU grid data structure additionally contains grid topology information, to be able to efficiently detect the neighbor cells of a face when traversing the domain along the rays.

A very effective acceleration approach is the reduction of the number of samples taken along the rays. Our raycaster employs an adaptive sampling strategy which adjusts the sampling step size to the complexity of the underlying color function [4]. In a preprocessing step, for each cell a rough approximation of the bounds of the field values $[f_{\min}, f_{\max}]$ and the maximum gradient magnitude $\|\nabla f\|_{\max}$ is determined within its cover. During runtime, we compute the

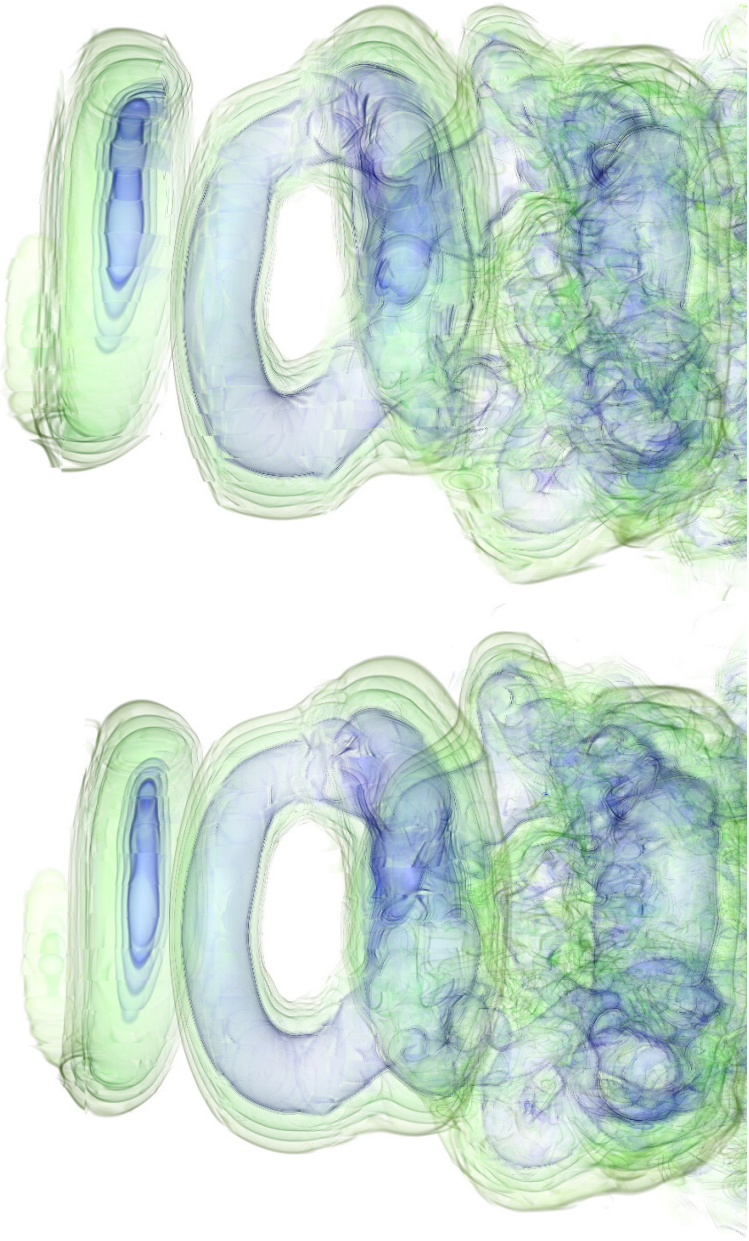


Fig. 4. Direct volume rendering of the density field of the turbulent jet flow, simulated with $p = 5$. Selective visualization up to specified polynomial degree $p_{vis} = 2$ (top) and $p_{vis} = 3$ (bottom) provides a qualitative notion of the impact of the polynomial order on the simulation result.

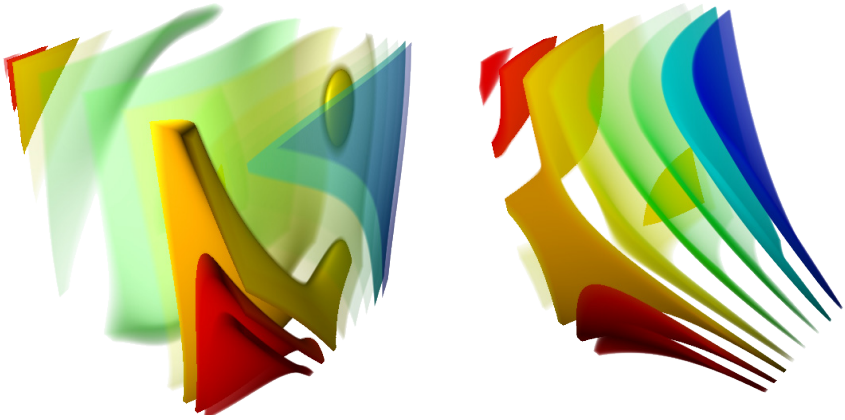


Fig. 5. Single hexahedral element with 5-th order scalar field. High order volume rendering (left) provides its true structure, in contrast to high order volume rendering of its approximation by trilinear interpolation (right). Both results were obtained without supersampling.

highest frequency ν_{TF} of the transfer function within the range $[f_{\min}, f_{\max}]$ for each cell. Then, a conservative approximation of the band-limiting frequency

$$\nu_c = \nu_{\mathrm{TF}} \|\nabla f\|_{\max} \tag{6}$$

of the cell's color function $\mathbf{c}\tau$ is computed, and, from this, an appropriate sampling step size is derived for the cell.

Figure 2 shows results for the laminar flow around a sphere. Our raycaster employs the Blinn-Phong shading model, using the field gradient $\nabla f(\mathbf{x})$ as the normal at sample location $\mathbf{x}$. This considerably improves the visualization of isosurface features contained within the data. Note, the monomial representation of f has the advantage that the computation of ∇f at the sampling locations introduces virtually no extra cost.

Volume renderings of the turbulent jet flow are given in Figures 3 and 4, while Figure 5 illustrates the superior quality of the high order rendering approach, compared to trilinearly supersampled data. It is apparent that the presented technique appropriately resolves fine detail of the high order data. Figure 6 shows two visualizations of a Taylor-Green vortex computation. One can observe fine details and at the same time the volume rendering provides context of the overall organization of the vortical flow.

In contrast to established visualization techniques, the presented approach also allows for a selective visualization of the single polynomial modes, to evaluate the influence of the polynomial degree on the solution. Visualization of two different polynomial degrees for the same dataset is provided at the example of the turbulent jet in Figure 4. One can clearly see that there is a quality jump between the two result images. The magnitude of the high modes is frequently used as a resolution indicator, especially for turbulent flows, thus selective visu-

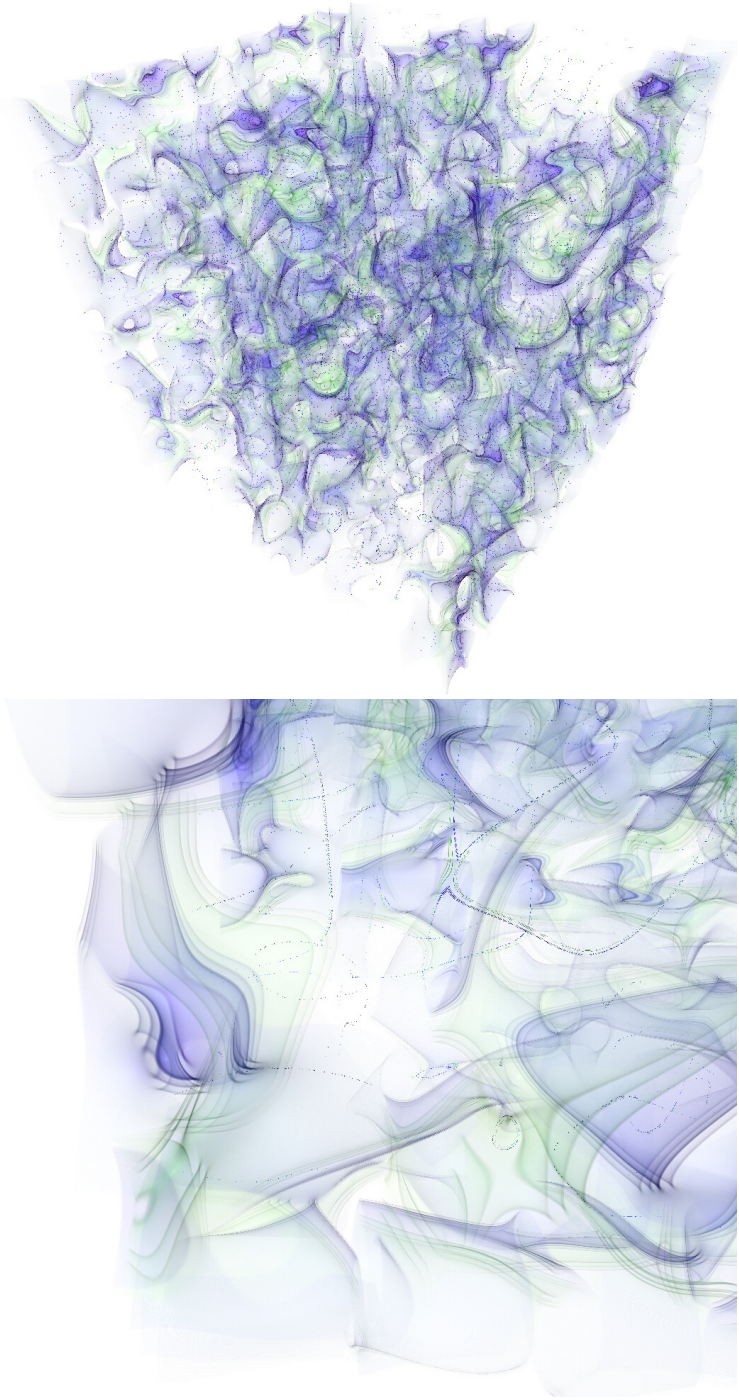


Fig. 6. Volume renderings of the Taylor-Green vortex computation with polynomial order $p = 9$

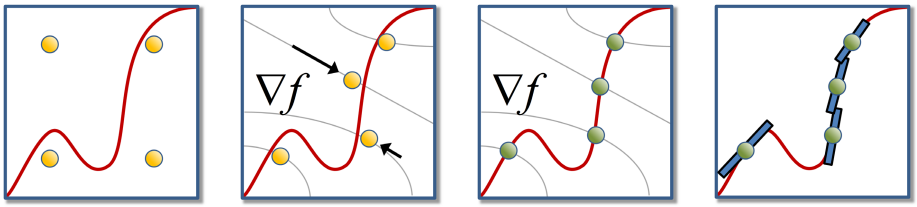


Fig. 7. First phase of isosurface extraction in high order fields (from [20]). (a) Distribute points uniformly within cell. (b) Move points toward aimed isosurface (red) by streamline integration in gradient field. (c) Fit points to isosurface by directional Newton-Raphson iteration. (d) Generate surface-tangent patch centered at each point.

alization allows one to easily identify underresolved regions in the domain. On the other hand, the ability to visualize lower polynomial degrees can be advantageous especially for very large datasets, if compuational resources are limited or if only a global overview is required.

3.2 Feature Extraction

Although many common problems are absent in volume rendering, its flexibility and generality involves other difficulties. On the one hand, transfer functions offer great flexibility for visualization but at the same time transfer function design is a demanding task. On the other hand, the integrative visualization effect of semitransparent rendering is a powerful approach to provide overall data context but at the cost of reduced depth perception and difficulty of perceiving individual features. Although it is possible to utilize volume rendering for extracting isosurfaces, there are approaches that are better suited for this task.

Isosurface extraction from discretized data, based on trilinear interpolation, is nowadays commonly achieved by the Marching Cubes (MC) family of algorithms [15]. The power of this approach relies on its assumption of tensor product linear variation of the field within a cell. Since these interpolations exhibit linear data variation along cell edges, the MC algorithms assume that a cell edge is intersected by an isosurface no more than once. This property allows the 3D algorithm to rely on only 256 possible configurations of isosurface patches within a cell. By constructing a lookup table for these cases and by marching, i.e., traversing, the data cell by cell, MC achieves superior performance at comparably low algorithmic complexity. Unfortunately, in the high order data domain, such assumptions cannot be made, necessitating alternative approaches.

Since construction of surface meshes from the typically highly variable isocontours in polynomial data represents a difficult undertaking, a possible approach is the extraction of point-based isosurface representations [16]. Point-based surfaces avoid the explicit representation of connectivity; the surfaces are defined by points scattered along the intended manifold, each of them supplied with a

(a) (b)

Fig. 8. Second phase of isosurface extraction in high order fields (from [20]). (a) Rasterize quads in image space. (b) Move fragments to isosurface by directional Newton-Raphson. Finally, fragments are shaded using ∇f as surface normal.

radius and a surface normal. It is the subsequent rendering step that employs appropriate blending to close the gaps, resulting in smoothly shaded surfaces similar to the results based on triangle meshes. A concern is thereby, however, the appropriate distribution of the sample points, e.g., with respect to surface curvature. Our technique [20] is located between these approaches: it employs a first phase where points (Figure 7a) are distributed onto the aimed isosurface by moving them along the field lines of the scalar field's gradient (Figure 7b), with subsequent Newton-Raphson refinement (Figure 7c), and generation of a patch centered at each point and aligned perpendicular to ∇f (Figure 7d). In a second phase, the surface patches are rasterized into fragments (Figure 8a), and each fragment is moved along the line of sight onto the isosurface with Newton-Raphson refinement (Figure 8b). The main reason for this two-phase approach is the difficulty of root finding in polynomial multivariate data. Along each viewing ray of the resulting image, we need to place the fragment onto the isosurface, i.e., at the intersection of the isosurface with the viewing ray. However, there are typically many intersections along the rays—and although only the one closest to the virtual camera is of interest (the others are occluded by it and hence one does not need to determine them), it is a nontrivial undertaking to find that first intersection along a ray. Due to the implicit formulation of an isosurface, numerical methods are typically employed. We apply Newton-Raphson iterations directed along the ray, involving the problem that these iterations tend to diverge from the aimed solution if the chosen starting point is not sufficiently close. This is the motivation for the first phase of our algorithm: it allows us to obtain initial estimates that are sufficiently close to the aimed solution. The second phase then necessitates only comparably few Newton-Raphson iterations to make the point converge toward the desired isosurface. Figure 9 provides a result for the density field of the laminar flow around a sphere. It is apparent that our technique correctly represents the discontinuities, i.e., cracks, at

the cell boundaries—in contrast to traditional resampling-based isosurface extraction. Although such cracks can involve perception issues, they represent the true shape of isosurfaces and can thus serve for validation and adjustment of *hp*-adaptivity.

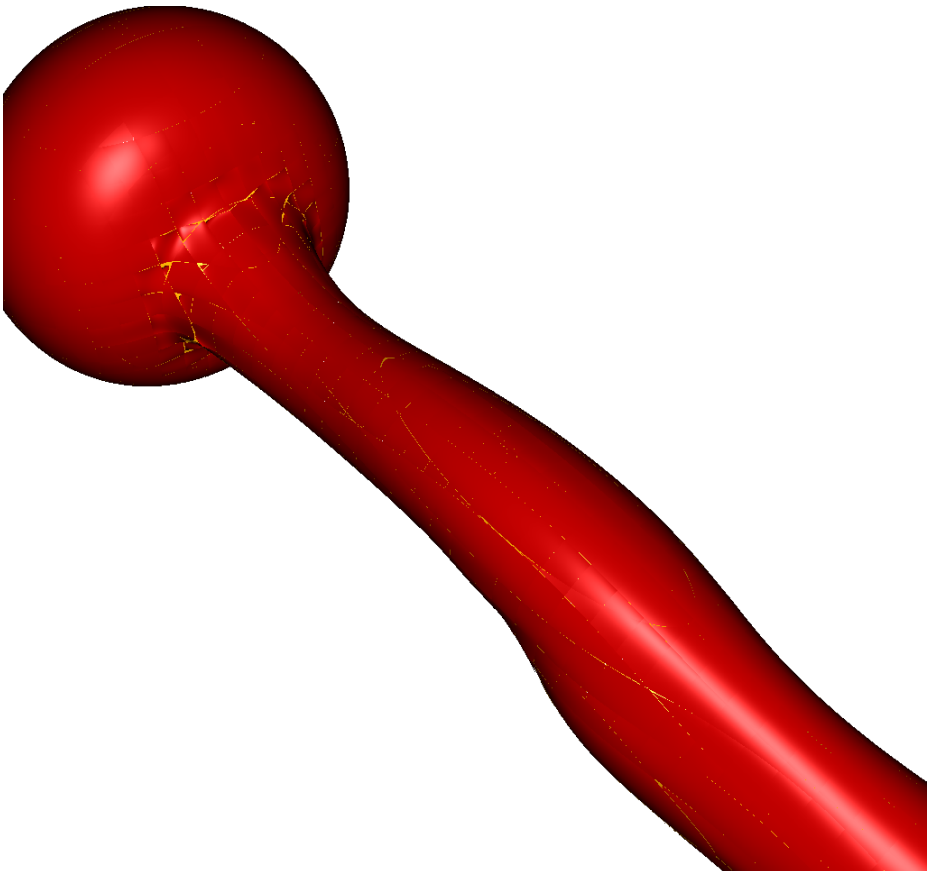


Fig. 9. Density isosurface extracted from laminar flow around sphere (from [20]). The discontinuities at the cell boundaries of the simulation are apparent and can serve for validation of simulation quality.

4 Conclusion

Discontinuous Galerkin codes have seen a considerable amount of new developments in the last years and can be regarded as mature from a numerical point of view. However, one remaining task is the efficient visualization of the piecewise polynomial data generated by these codes, as most widespread visualization tools can only handle tensor product linear data. Appropriate visualization of

high order data is a nontrivial task, as the difficulties are not only with respect to the polynomial representation. Beyond that, the techniques have to account for advanced features of modern DG codes, like curved elements, mesh and polynomial adaptivity, hanging nodes, very high polynomial degrees, and very large data sizes.

In this paper, we reported on two of our existing high order visualization techniques—direct volume rendering and isosurface extraction, and evaluated them using different simulation data. To this end, we conducted simulations on three well-investigated benchmark problems: the flow around a sphere, featuring a complex curved geometry, a round jet with a high number of degrees of freedom, and an isentropic vortex flow computed at $\mathcal{O}(16)$. We demonstrated the advantages and limitations of our approaches using these cases and discussed their application in the overall high order CFD process. As future work, we plan to integrate our approaches into a single framework to make them more widely available.

Acknowledgments. This work was partially funded and supported by the IDIHOM project of the FP7 framework of the European Union and by Deutsche Forschungsgemeinschaft (DFG) within the Cluster of Excellence in Simulation Technology.

References

1. Bassi, F., Rebay, S.: A high order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. Journal of Computational Physics 131, 267–279 (1997)
2. Bassi, F., Rebay, S.: A high-order discontinuous Galerkin finite element method solution of the 2D Euler equations. Journal of Computational Physics 138, 251–285 (1997)
3. Bassi, F., Rebay, S.: Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier–Stokes equations. International Journal for Numerical Methods in Fluids 40, 197–207 (2002)
4. Bergner, S., Möller, T., Weiskopf, D., Muraki, D.J.: A spectral analysis of function composition and its implications for sampling in direct volume visualization. IEEE Transactions on Visualization and Computer Graphics 12(5), 1353–1360 (2006)
5. Bogey, C., Bailly, C.: Large eddy simulations of transitional round jets: influence of the reynolds number on flow development and energy dissipation. Physics of Fluids 18(6), 65101 (2006)
6. Cockburn, B., Shu, C.-W.: The Runge–Kutta local projection p1-discontinuous Galerkin finite element method for scalar conservation laws. RAIRO Modél. Math. Anal. Numér 25(3), 337–361 (1991)
7. Garrity, M.P.: Raytracing irregular volume data. Symposium on Volume Visualization 24(5) (1990)
8. Gassner, G., Beck, A.: On the accuracy of high-order discretizations for underresolved turbulence simulations. Theoretical and Computational Fluid Dynamics 27(3-4), 221–237 (2013)

9. Gassner, G., Kopriva, D.: A comparison of the dispersion and dissipation errors of Gauss and Gauss-Lobatto discontinuous Galerkin spectral element methods. SIAM J. Scientific Computing 33(5), 2560–2579 (2011)
10. Gassner, G., Lörcher, F., Munz, C.-D.: A discontinuous Galerkin scheme based on a space-time expansion. II. Viscous flow equations in multi dimensions. J. Sci. Comp. 34(3), 260–286 (2008)
11. Gassner, G.J., Lörcher, F., Munz, C.-D., Hesthaven, J.S.: Polymorphic nodal elements and their application in discontinuous galerkin methods. Journal of Computational Physics 228(5), 1573–1590 (2009)
12. Geuzaine, C., Remacle, J.-F.: Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)
13. Hindenlang, F., Gassner, G., Altmann, C., Beck, A., Staudenmaier, M., Munz, C.-D.: Explicit discontinuous Galerkin methods for unsteady problems. Computers and Fluids 61, 86–93 (2012)
14. Lörcher, F., Gassner, G., Munz, C.-D.: An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations. Journal of Computational Physics 227(11), 5649–5670 (2008)
15. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: SIGGRAPH 1987: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, vol. 21, pp. 163–169 (1987)
16. Meyer, M., Nelson, B., Kirby, R., Whitaker, R.: Particle systems for efficient and accurate high-order finite element visualization. IEEE Transactions on Visualization and Computer Graphics 13(5), 1015–1026 (2007)
17. Nelson, B., Haimes, R., Kirby, R.M.: GPU-based interactive cut-surface extraction from high-order finite element fields. IEEE Transactions on Visualization and Computer Graphics 17(12), 1803–1811 (2011)
18. Nelson, B., Liu, E., Kirby, R., Haimes, R.: ElVis: A system for the accurate and interactive visualization of high-order finite element solutions. IEEE Transactions on Visualization and Computer Graphics 18(12), 2325–2334 (2012)
19. Pagot, C., Osmari, D., Sadlo, F., Weiskopf, D., Ertl, T., Comba, J.: Efficient parallel vectors feature extraction from higher-order data. Computer Graphics Forum 30(3), 751–760 (2011)
20. Pagot, C., Vollrath, J., Sadlo, F., Weiskopf, D., Ertl, T., ao Luiz Dihl Comba, J.: Interactive isocontouring of high-order surfaces. In: Scientific Visualization: Interactions, Features, Metaphors. Dagstuhl Follow-Ups, vol. 2, pp. 276–291 (2011)
21. Reed, W., Hill, T.: Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory (1973)
22. Remacle, J.-F., Chevaugeon, N., Marchandise, E., Geuzaine, C.: Efficient visualization of high-order finite elements. International Journal for Numerical Methods in Engineering 69(4), 750–771 (2007)
23. Stegmaier, S., Strengert, M., Klein, T., Ertl, T.: A simple and flexible volume rendering framework for graphics-hardware-based raycasting. In: Fourth International Workshop on Volume Graphics 2005, pp. 187–241 (2005)
24. Üffinger, M., Frey, S., Ertl, T.: Interactive high-quality visualization of higher-order finite elements. Computer Graphics Forum 29(2), 115–136 (2010)
25. Üffinger, M., Schweitzer, M.A., Sadlo, F., Ertl, T.: Direct visualization of particle-partition of unity data. In: Proceedings VMV 2011, pp. 255–262 (2011)

Part III

Assessment of High-Order Methods

External Aerodynamic Test Cases

K.A. Sørensen[1], C. Johnston[2], T. Leicht[3], F. Chalot[4],
P. Eliasson[5], F. Bassi[6], V. Couaillier[7],
and M. Kessler[8]

[1] Airbus Defence & Space, 85077 Manching, Germany
[2] Aircraft Research Association Ltd., MK41 7PF Bedford, England
[3] DLR, 38108 Braunschweig, Germany
[4] Dassault Aviation, 92214 Saint-Cloud, France
[5] FOI, 16490 Stockholm, Sweden
[6] Università degli studi di Bergamo, 24129 Bergamo, Italy
[7] ONERA, 92322 Chatillon Cedex, France
[8] Universität Stuttgart, 70550 Stuttgart, Germany

Abstract. In this chapter the testcases in the IDIHOM project treating external aerodynamic flow are subjected to an industrial assessment. The chapter serves to illustrate the level of complexity that can be presently handled using higher order discretization methods. In addition, a discussion on the efficiency and robustness of the codes compared with the current state of the art is included.

Keywords: CFD, industrial assessment, higher order discretization, external aerodynamics.

1 Introduction

The main bulk of testcases considered in IDIHOM are in the scope of external aerodynamics. Since a central goal of the project is the industrialization of higher order approaches, most of the cases considered are of a relatively high level of complexity. In addition, simpler testcases have been included as a testbed for the new approaches as well as allowing for more detailed, quantitative, analyses normally too expensive for industrial cases.

An important part of the project has been comparing the new approaches to a well–defined industrial reference. The approach used is similar to what was attempted in the ADIGMA project [1], incorporating asymptotic analyses on mesh cascades, resulting in machine–independent performance data. Also considered are other aspects of importance for industry, such as robustness, ease of use and memory usage. Some of the testcases are of a complexity not often attempted with higher order schemes. For such cases, the successful generation of plausible data in itself represents progress in the field of higher order solvers.

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_26

The asymptotic convergence analysis planned for the testcases is based on the assumption that the numerical error for an integral value can be expressed in the following form:

$$E = kN_{DOF}^{-\alpha} \tag{1}$$

where N_{DOF}, α and k are the number of degrees of freedom per equation, the (dimensionally dependent) order and a constant, respectively. Ideally, a mesh cascade of at least three nested meshes of sufficient resolution is available for the analysis, allowing for the determination of the order, the constant k and the asymptotic value of the integral coefficient. The creation of higher order mesh cascades, particularly for unstructured meshes, unfortunately turned out to be a major challenge for the geometries considered in the project. Simplified approaches were therefore frequently necessary to enable a quantitative statement to be made. One approach used was determining the coefficient asymptotic value through curve–fitting of results of different discretization order, then assuming the discretization order to equal the nominal scheme value and finally determining the constant k for the error for each computation. These additional assumptions are likely to significantly affect the accuracy of the analysis. It was however hoped that enough data would become available for the identification of general trends of the scheme performance. For platform–independent comparisons of wall–clock time required for convergence, a normalized time-unit was introduced, based on a benchmark code available to all partners [2]. For the majority of the cases considered, it was however discovered that any form of formal evaluation of the efficiency of the schemes would exceed the available resources. These cases are however still important in obtaining a picture of the applicability and robustness of higher order methods in their current form.

2 Testcases

Higher order computations of eight external aerodynamic test cases were performed within the project. Three of these testcases are termed underlying as they involve geometries and physics that are not considered of industrial complexity. The remaining testcases consist of aircraft geometries at design point and high angle of attack/high lift conditions as well as a train configuration and a rotor case. For each testcase, a document was created at the beginning of the project and continuously updated, in which the convergence definitions, the available meshes and current status of the case was described.

2.1 Case U.1: VFE2 Delta Wing

This underlying testcase was coordinated by Airbus D&S and is considered to be important since it is often stated that higher order methods are particularly advantageous for rotating flows. If this could be shown, the higher order discretizations could constitute a large improvement compared with the second order FV methods presently in use, known to require very fine meshes in order

to reduce vortex dissipation below an acceptable level. Three subcases were defined for the delta wing. The U.1a testcase consists of the VFE2 geometry with sharp leading edge, $M = 0.07$, $\alpha = 23°$ and $Re = 1 \cdot 10^6$. This case was to be computed using hybrid RANS/LES methods. For U.1b and U.1c, the geometry with the medium radius leading edge is considered. For U.1b, the flow conditions are $M = 0.4$, $\alpha = 13.3°$, $Re = 3.0 \cdot 10^6$, and for U.1c, $M = 0.8$, $\alpha = 20.5°$ and $Re = 2.0 \cdot 10^6$. In addition to integral forces and surface pressure measurements [3], DLR has conducted PSP and PIV measurements [4] and TUM has conducted hot wire anemometry measurements for U.1a, yielding high frequency velocity data [5].

The industrial reference was provided by Airbus D&S, employing the DLR Tau code [6] and NLR, using the ENFLOW [7] code. For U.1a, no mesh convergence study was attempted. Instead, the solution quality considered state of the art was indicated. For U.1b, an asymptotic convergence study was conducted, indicating the current cost in obtaining a solution of engineering accuracy, defined as a discretization error in the lift and pitching moment coefficients below 0.005, and below 0.001 for the drag coefficient. The industrial reference was found to be dominated by the drag convergence, requiring 1.1e7 time units for convergence. The lift and pitching moment convergence require $1.5 \cdot 10^5$ and $1.9 \cdot 10^5$ time units respectively. The mesh sizes required for convergence of the discrete systems are $5.8 \cdot 10^6$, $1.3 \cdot 10^8$ and $7.0 \cdot 10^6$ DOFs per equation (mesh nodes) for lift, drag and pitching moment respectively. The memory requirements for the drag-converged mesh is 190 Gb. The same analysis was attempted for the transonic case U.1c, unfortunately without conclusive results, showing irregular asymptotic behavior. From experience with second order finite volume solvers, it is however reasonable to assume that the transonic mesh convergence has similar characteristics to the subsonic case.

Higher order inputs for this testcase were supplied by DLR, UNIBG and NLR. Unfortunately, the higher order unstructured meshes that were planned to be supplied by ARA were, due to technical difficulties, not completed during the project. A workaround to this major problem was found by acquiring a higher order unstructured mesh from professor O. Hassan, University of Wales, Swansea, who, not being part of IDIHOM, graciously supplied the grid without remuneration, Figure 1. Two higher order structured meshes were generated by UNIBG through agglomeration of linear meshes for their computations, Figures 2, 3. DLR also used structured higher order meshes created by agglomeration, based on the same meshes as UNIBG. NLR employed a linear, structured, mesh for their XLES simulation. Since proper higher order mesh cascades were not available for this testcase, the simplified asymptotic analysis had to be used.

For the subsonic DLR computations of case U.1b, it is seen that the $p = 1$ results fit reasonably well with the (second order) industrial baseline data, Table 1, both for mesh size and convergence time. It is also observed that increasing p significantly reduces the mesh size required, thus showing improvement

compared with the current industrial state of the art. The computational cost is however similar to industrial solvers. The memory usage is on the other hand an order of magnitude higher than the industrial reference. Also observed is that the efficiency improvement resulting from increasing p from 1 to 2 is considerably larger than a subsequent increase in p from 2 to 3; in computational time this last increase does not seem to bring improvement for this case.

The subsonic UNIBG computations were originally performed on a very coarse, highly regular structured mesh, Figure 2. The, DOF based, drag convergence data for these computations are consistently two orders of magnitude better than the unstructured computations described above, Table 2. The computations thus show an impressive efficiency increase of around three orders of magnitude compared with the industrial baseline. The higher order solver seems to reach the industrial reference convergence rate for this very small discrete system, resulting in a three orders of magnitude improvement also in computational time. As seen for the unstructured computations, the increase to $p = 2$ shows the largest gains and this discretization order appears to be optimal for this case. The memory usage per degree of freedom is still very high, but due to the very small meshes needed for convergence, it is smaller than the industrial baseline. To confirm these results, a convergence analysis of the lift and drag coefficients was also attempted. For these integral values the p-convergence was however not monotone. By taking the $p = 4$ values as asymptotic values, the convergence data was approximated, Tables 3 and 4. It is observed that the analysis yields irregular results, except for $p = 3$, where the convergence is dominated by the drag values, as for the industrial reference. These truly excellent improvements for the higher order solver have not been observed for other testcases. Even though the analysis is based on a very coarse mesh, the asymptotic values found are almost identical to the unstructured higher order computations and the industrial reference, except for the asymptotic drag coefficient which is found to be around 4% higher than the DLR unstructured result but very close to the industrial reference. It is also unclear why for $p = 1$ the discrete system size is two orders smaller than the unstructured DLR computations and the industrial baseline. To clarify these results, the testcase was repeated by UNIBG for a finer structured mesh, Figure 3. These computations resulted in a completely different picture, showing irregular asymptotic behavior with increasing scheme order. Generally it can however be observed that the convergence of the integral coefficients is much slower on this mesh. The drag coefficient also appears to converge towards a value closer to the unstructured DLR results. The stellar performance resulting from the coarse mesh analysis could thus unfortunately not be confirmed and it seems reasonable to assume that the coarse mesh UNIBG behavior is not representative for the higher order codes on this problem.

For the transonic case, U.1c, results from UNIBG are available, Summarized in Table 3 as well as incomplete data from DLR, Table 4. For these computations, the finer structured mesh Figure 3, was used. Even though industrial baseline data are not available for this testcase, it is reasonable to assume that the 2. order FV behavior is of the same order of magnitude as for case U.1b. This would

thus indicate around a one order of magnitude improvement in discrete system size for the higher order approach. It is however observed that the computational time required for convergence does not improve with the scheme order, indicating a less efficient solver for higher polynomial representations for transonic cases.

Also demonstrated for case U.1b by DLR is the usage of adjoint adaptation for higher order methods. In Figures 4 and 5 the improvement in mesh convergence is shown. A reduction in discrete system size significantly higher than one order of magnitude compared with the industrial reference is achieved. In Figures 6 and 7, the reduction in computational time is observed to also reach more than an order of magnitude compared with the reference. The comparison with a similar adaptive scheme based on 2nd order finite volume solvers has not been performed in the project.

NLR demonstrated the usage of their 4th order finite volume solver for case U.1a, showing good correlation with the experimental data, Figures 8, 9. The accuracy obtained is stated to be similar to that found for the baseline computations, but with a reduction of around 40% of the grid points due to the, in IDIHOM implemented, hanging–node capability of the code. This testcase was also simulated by UNIBG using the X-LES approach, Figures 10,11. UNIBG have also achieved progress in the higher-order discretization of turbulence models, obtaining results for U.1b using the EARSM method, Figure 12. Similar problems regarding stability as can often be observed for industrial codes are also reported for the higher order implementation, there are however indications of improved solution quality for the VFE2.

In general is appears that progress has been made in the robustness of higher order codes and achieving well converged results has become routine for this case. This has also been confirmed by Airbus D&S using the DLR PADGE code for various angles of attack and (subsonic) Mach numbers showing, in general, stable properties for low and medium Mach numbers.

Table 1. Asymptotic convergence analysis for the U.1b test case based on DLR PADGE coefficients on an unstructured higher order mesh. For the lift and pitching moment coefficients the p–convergence displayed irregular behavior, the asymptotic coefficient was therefore approximated by the third order value.

p	N_{DOF}			Time $[t_s]$			Memory $[Gb]$		
	C_l	C_d	C_m	C_l	C_d	C_m	C_l	C_d	C_m
1	$2.6 \cdot 10^7$	$3.9 \cdot 10^8$	$2.4 \cdot 10^7$	$9.1 \cdot 10^5$	$1.4 \cdot 10^7$	$8.6 \cdot 10^5$	$1.1 \cdot 10^3$	$1.6 \cdot 10^4$	$9.8 \cdot 10^2$
2	$1.0 \cdot 10^6$	$6.7 \cdot 10^7$	$4.6 \cdot 10^6$	$7.3 \cdot 10^5$	$5.2 \cdot 10^6$	$3.4 \cdot 10^5$	$7.5 \cdot 10^1$	$5.0 \cdot 10^3$	$3.4 \cdot 10^2$
3	–	$4.3 \cdot 10^7$	–	–	$5.2 \cdot 10^6$	–	–	$4.6 \cdot 10^3$	–

Table 2. Asymptotic convergence analysis for the U.1b test case based on UNIBG MIGALE coefficients on a structured higher order mesh. For the lift and pitching moment coefficients the p–convergence was not monotone, the asymptotic coefficient was therefore approximated by the fourth order value. The excellent performance obtained for this mesh could not be reproduced when applying the same solver on finer meshes of similar quality and is thus not considered representative for the higher order solvers.

p	N_{DOF}			Time $[t_s]$			Memory $[Gb]$		
	C_l	C_d	C_m	C_l	C_d	C_m	C_l	C_d	C_m
1	$3.6 \cdot 10^4$	$1.4 \cdot 10^6$	$9.9 \cdot 10^4$	$1.3 \cdot 10^3$	$5.1 \cdot 10^4$	$3.7 \cdot 10^3$	$2.0 \cdot 10^0$	$9.0 \cdot 10^1$	$6.0 \cdot 10^0$
2	$2.0 \cdot 10^5$	$1.3 \cdot 10^5$	$2.2 \cdot 10^6$	$1.0 \cdot 10^4$	$7.0 \cdot 10^3$	$1.1 \cdot 10^5$	$1.9 \cdot 10^1$	$1.0 \cdot 10^1$	$2.0 \cdot 10^2$
3	$2.9 \cdot 10^4$	$2.1 \cdot 10^5$	$5.7 \cdot 10^4$	$1.7 \cdot 10^3$	$1.9 \cdot 10^4$	$1.1 \cdot 10^4$	$5.0 \cdot 10^0$	$3.0 \cdot 10^1$	$9.0 \cdot 10^0$
4	–	$2.4 \cdot 10^5$	–	–	$5.1 \cdot 10^4$	–	–	$6.0 \cdot 10^1$	–

Table 3. Asymptotic convergence analysis for the U.1c test case based on UNIBG MIGALE coefficients on a structured higher order mesh

p	N_{DOF}			Time $[t_s]$			Memory $[Gb]$		
	C_l	C_d	C_m	C_l	C_d	C_m	C_l	C_d	C_m
1	$4.5 \cdot 10^7$	$1.1 \cdot 10^8$	$5.8 \cdot 10^7$	$3.8 \cdot 10^6$	$9.6 \cdot 10^6$	$5.0 \cdot 10^6$	$2.9 \cdot 10^3$	$7.0 \cdot 10^3$	$3.7 \cdot 10^3$
2	$7.6 \cdot 10^6$	$1.5 \cdot 10^7$	$1.0 \cdot 10^7$	$5.7 \cdot 10^6$	$1.1 \cdot 10^7$	$7.6 \cdot 10^6$	$6.1 \cdot 10^2$	$1.2 \cdot 10^3$	$8.0 \cdot 10^2$
3	$4.9 \cdot 10^6$	$8.3 \cdot 10^6$	$6.4 \cdot 10^6$	$7.5 \cdot 10^6$	$1.3 \cdot 10^7$	$1.0 \cdot 10^7$	$7.1 \cdot 10^2$	$1.2 \cdot 10^3$	$9.3 \cdot 10^2$

Table 4. Asymptotic mesh convergence analysis for the U.1c test case based on DLR PADGE coefficients on a structured higher order mesh. The data is very similar to the UNBG results.

p	C_l	C_d
1	$3.8 \cdot 10^7$	$8.8 \cdot 10^7$
2	$8.5 \cdot 10^6$	$1.6 \cdot 10^7$
3	$6.0 \cdot 10^6$	$9.7 \cdot 10^6$

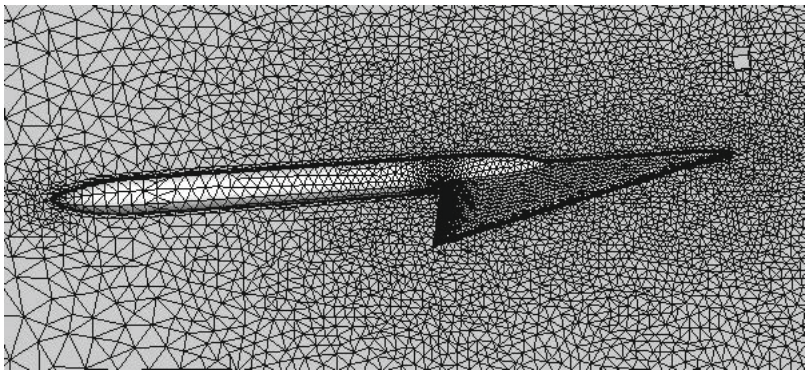


Fig. 1. Unstructured higher order mesh used for the U.1 case. The mesh consists of $1.1 \cdot 10^6$ tetrahedral elements and was supplied to the project by the University of Wales, Swansea.

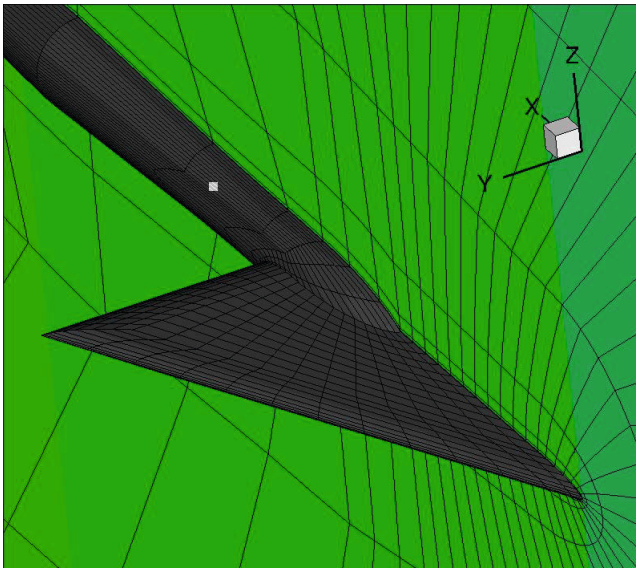


Fig. 2. Structured higher order mesh used for the U.1 case, consisting of $1.4 \cdot 10^4$ elements

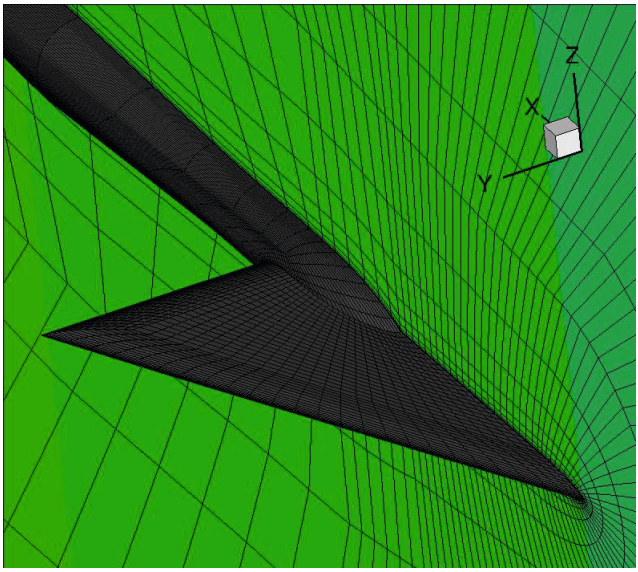


Fig. 3. Structured higher order mesh used for the U.1 case, consisting of $1.1 \cdot 10^5$ elements

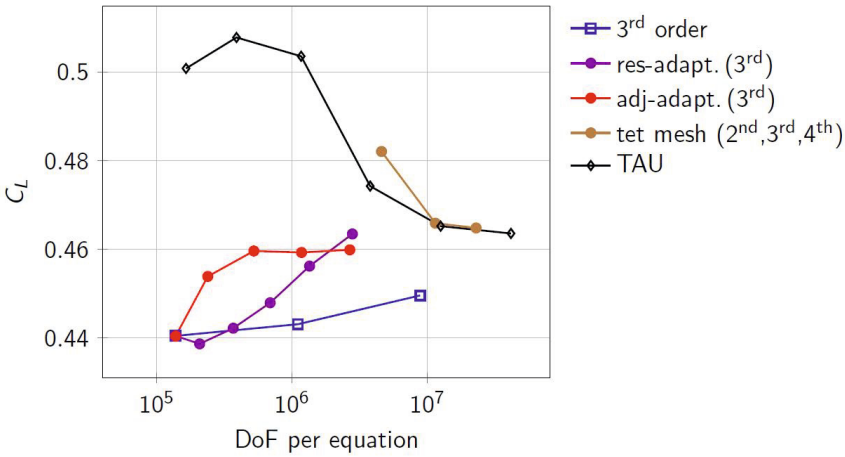


Fig. 4. Convergence plots of the computed lift coefficient vs. number of degrees of freedom for case U.1c as computed by DLR. Shown are the results for the higher order scheme on a cascade of structured meshes, the convergence of sequences of meshes generated by residual–based and adjoint–based adaptation, p–convergence on the unstructured higher order simplex mesh as well as the industrial baseline.

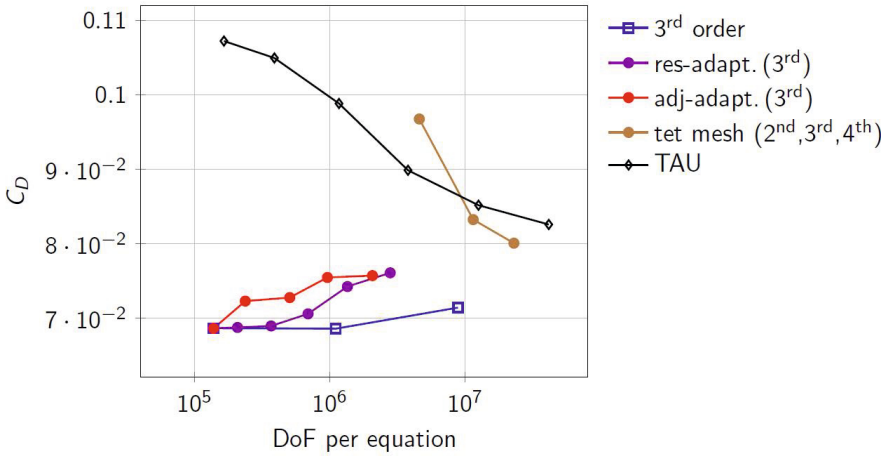


Fig. 5. Convergence plots of the computed drag coefficient vs. number of degrees of freedom for case U.1c as computed by DLR. Shown are the results for the higher order scheme on a cascade of structured meshes, the convergence of sequences of meshes generated by residual–based and adjoint–based adaptation, p–convergence on the unstructured higher order simplex mesh as well as the industrial baseline.

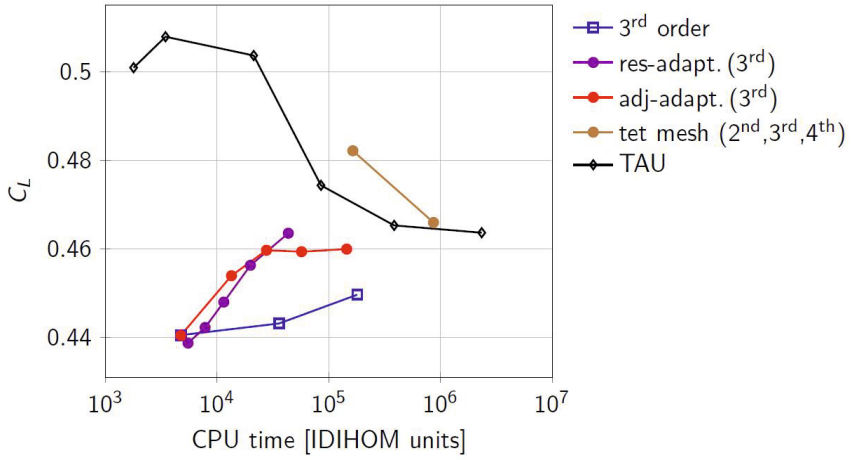


Fig. 6. Convergence plots of the computed lift coefficient vs. normalized CPU time for case U.1c as computed by DLR. Shown are the results for the higher order scheme on a cascade of structured meshes, the convergence of sequences of meshes generated by residual–based and adjoint–based adaptation, p–convergence on the unstructured higher order simplex mesh as well as the industrial baseline.

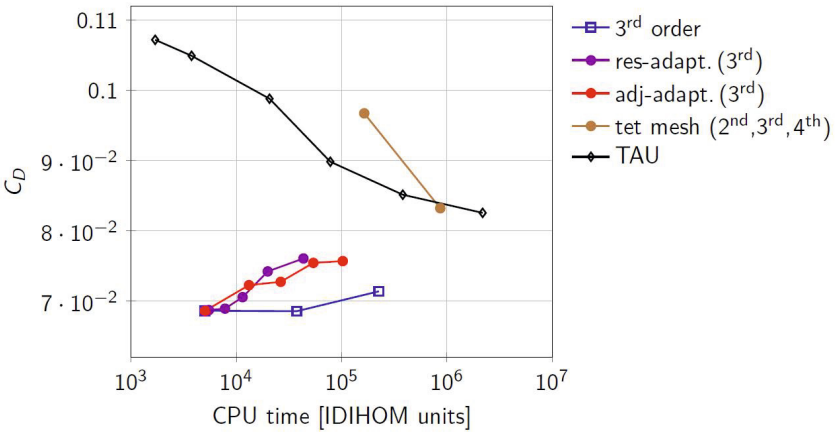


Fig. 7. Convergence plots of the computed drag coefficient vs. normalized CPU time for case U.1c as computed by DLR. Shown are the results for the higher order scheme on a cascade of structured meshes, the convergence of sequences of meshes generated by residual–based and adjoint–based adaptation, p–convergence on the unstructured higher order simplex mesh as well as the industrial baseline.

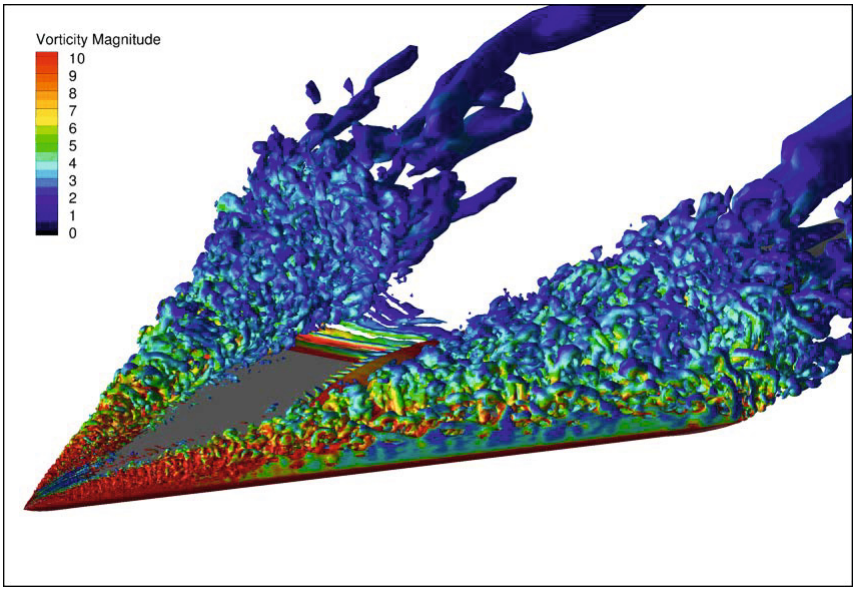


Fig. 8. Instantaneous iso–contour of Q–criterion colored with vorticity magnitude for the U.1a NLR computation

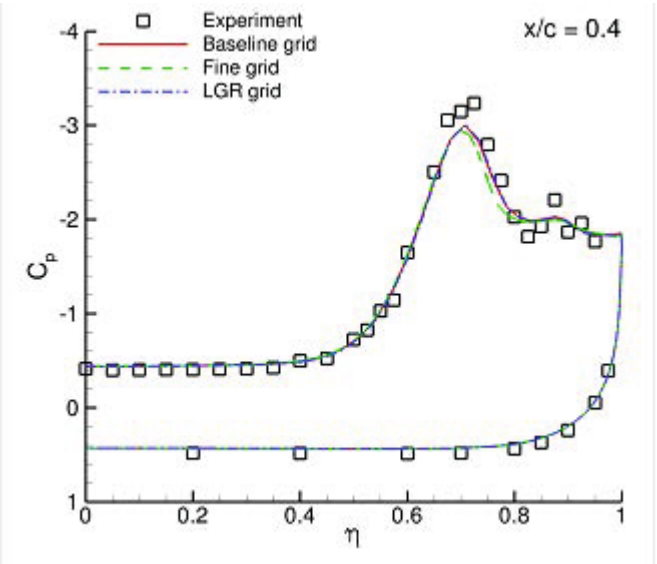


Fig. 9. Comparison between the computed and measured surface time–averaged pressure coefficients for the U.1a NLR computations on three different grids on station 0.4

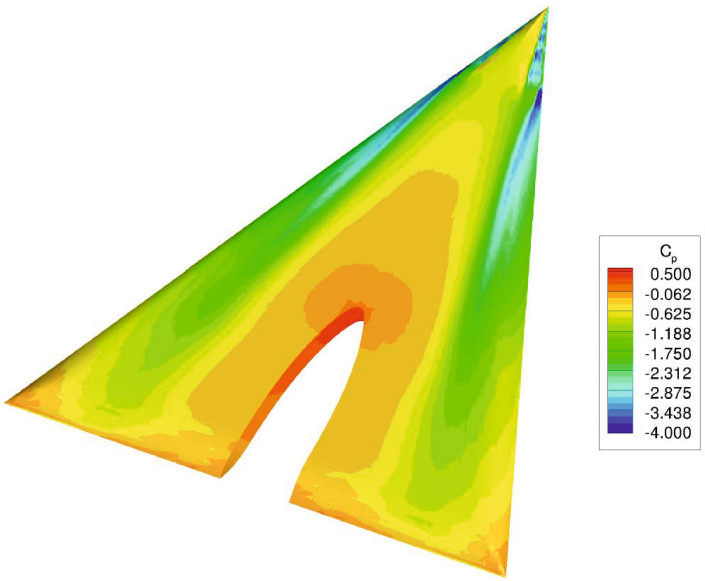


Fig. 10. Surface pressure plot of the U.1a UNIBG X–LES computation

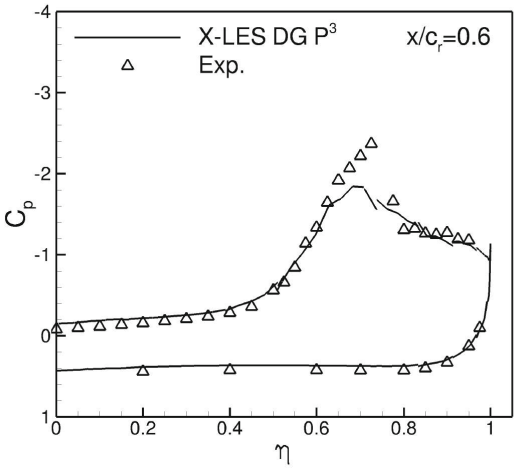


Fig. 11. Comparison between the computed and measured surface pressure coefficients for the U.1a UNIBG X–LES computation at station 0.6

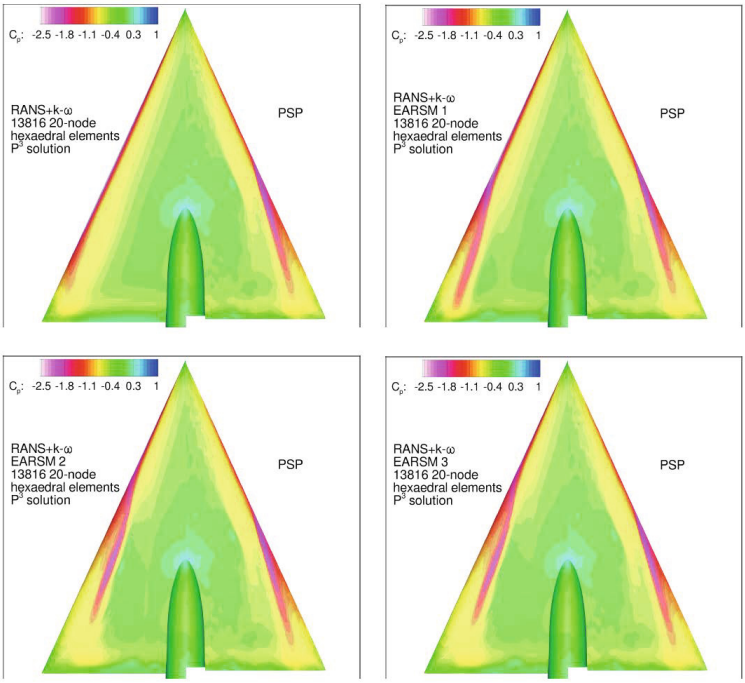


Fig. 12. Comparison of surface pressure plots between UNIBG EARSM computations and PSP measurements for the U.1b testcase

2.2 Case U2: M6 Wing

This underlying testcase was coordinated by ARA and was envisaged to serve as basis for typical transport configurations at cruise. Involved in the testcase were TsAGI, INRIA, WUT, ARTS, NUMECA, ICL and ARA. Available for this testcase are surface pressure profiles [8] which have been extensively used in the code comparisons. The flow conditions considered were $M = 0.8395$, $\alpha = 3.06°$ and $Re = 1.172 \cdot 10^7$.

Four separate grid families were generated for this testcase. NUMECA produced a set of unstructured P2 hexahedral meshes with hanging nodes at three resolution levels, Figure 13. ICL produced two unstructured tetrahedral meshes for the case, Figure 14. WUT created a series of grids through their h-adaptation code, Figure 15. ARTS used a series of structured linear meshes, generated by coarsening a C–topology grid acquired from the NASA Validation website [9], Figure 16.

Unfortunately, for this testcase very little data was produced on which a quantitative study can be based. The evaluation of the various schemes is therefore only based on comparisons with experimental data, for which it is known that industrial codes correlate strongly. In Figures 17 to 24 the results obtained by the various partners are given. The TsAGI results are shown to converge towards the experimental and second order FV results with increasing discrete system size. No significant reduction in DOFs for the same level of accuracy is however observed for the higher order scheme. The ARTS results also show convergence towards experiments, where only the finest mesh applied correlates well for the RBC schemes, as well as the industrial baseline code. Again, no improvement for the higher order methods with respect to DOF convergence is observed. The results from WUT's adaptive simulations are undoubtedly the best, displaying good capture of shock strength and location at all stations, these results were however obtained by a second order scheme and are thus not higher order according to the IDIHOM definition. Considering the mesh size resulting from the adaptation, the results are believed to be comparable in efficiency with the industrial codes as reported by ARTS.

Viewed simply as a validation testcase, all the solvers considered produce solutions aligned with experiment, provided a sufficiently high number of DOF is used. The underlying flow physics thus appear to be captured by the higher order schemes. However, carrying out comparisons between the higher order methods and those from the Finite–Volume reference simulations, it is difficult to ascertain any clear advantage of the new approaches for this particular case, there being little, if any, improvement in accuracy over the reference with a similar number of DOF. This may at least partially be caused by the shock system dominating this particular case.

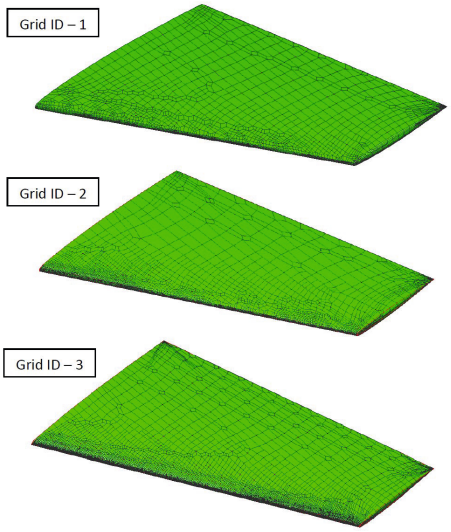


Fig. 13. NUMECA meshes for testcase U2. The three mesh levels consist of $3.8 \cdot 10^4$, $7.6 \cdot 10^4$ and $2.0 \cdot 10^5$ P2 hexahedral elements.

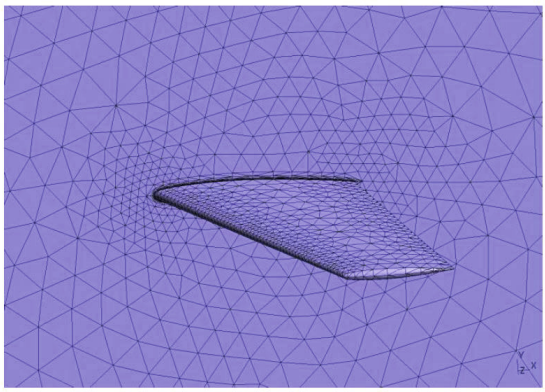


Fig. 14. ICL mesh for testcase U2. Two mesh levels were generated, consisting of $1.2 \cdot 10^5$ and $1.4 \cdot 10^5$ P2 tetrahedral elements. The first of these meshes were constructed with an initial layer height of $y+ \sim 6$, the second with $y+ \sim 1$.

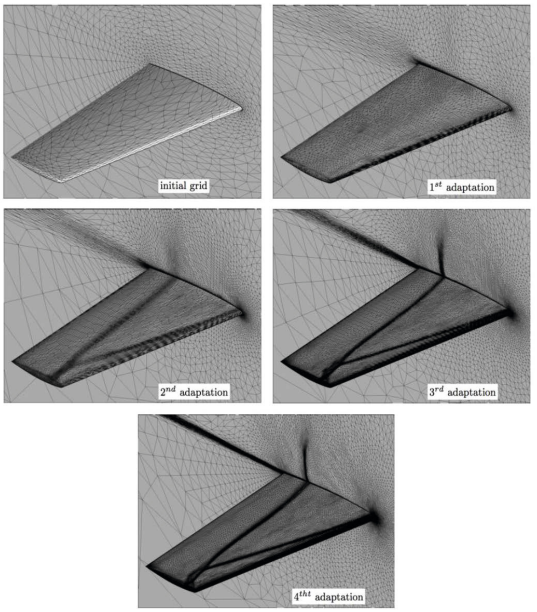


Fig. 15. WUT meshes for testcase U2. The meshes consist of linear tetrahedra and contain $1.1 \cdot 10^4$, $1.4 \cdot 10^5$, $2.8 \cdot 10^5$, $8.0 \cdot 10^5$ and $2.7 \cdot 10^6$ nodes.

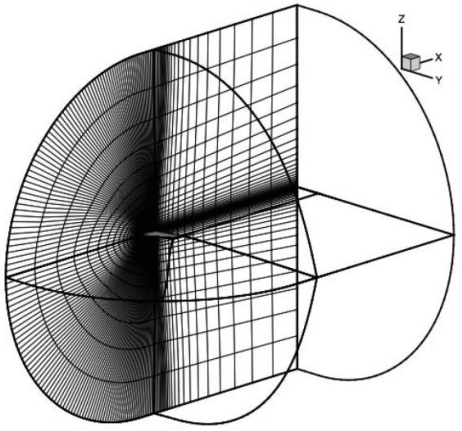


Fig. 16. Mesh used by ARTS for testcase U2. Three meshes were created, consisting of $4.6 \cdot 10^3$, $3.7 \cdot 10^4$, and $2.9 \cdot 10^5$ linear hexahedra.

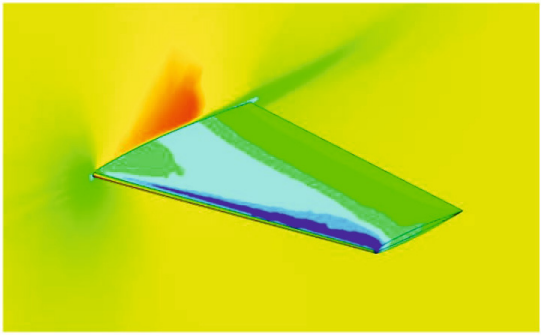


Fig. 17. TsAGI computation of case U2, showing pressure on the wing surface and Mach distribution on the symmetry plane

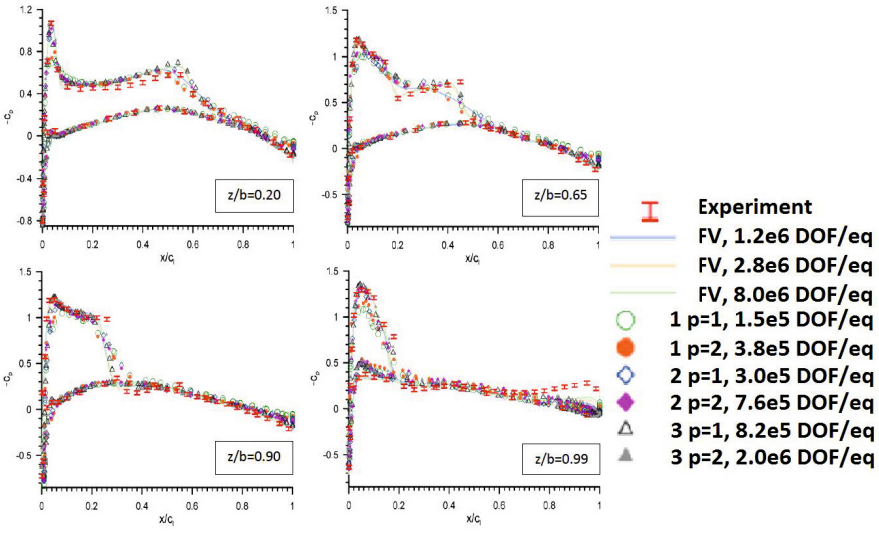


Fig. 18. Comparison between experiment, finite volume and different higher order solutions as reported by TsAGI for testcase U2

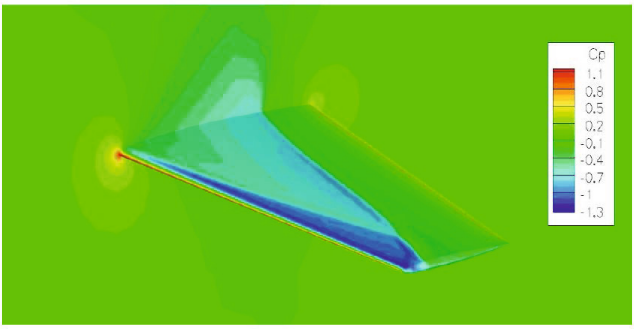


Fig. 19. INRIA pressure field computed for case U2

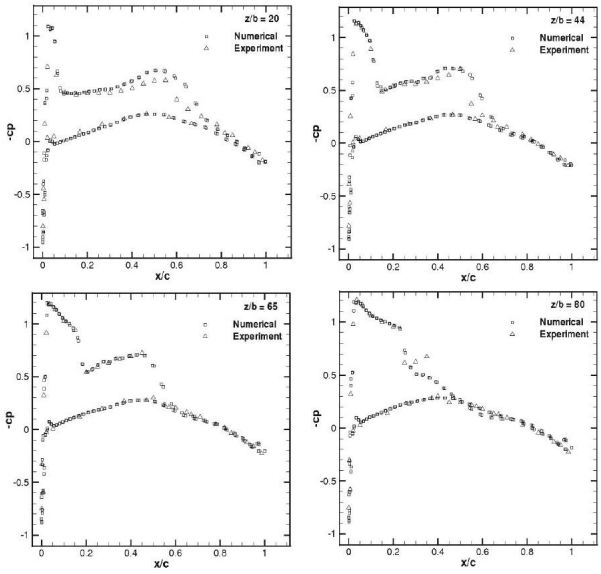


Fig. 20. Comparison between experiment and the P2 solution computed by INRIA for testcase U2

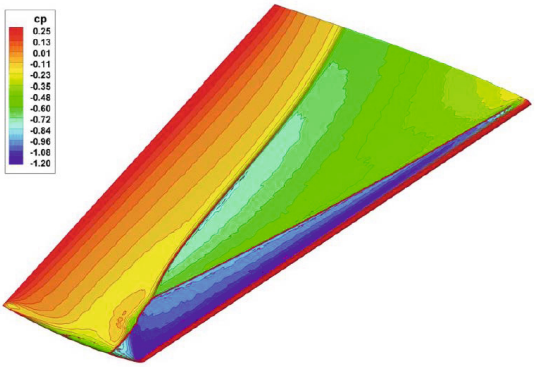


Fig. 21. WUT pressure distribution for Grid 5, testcase U2

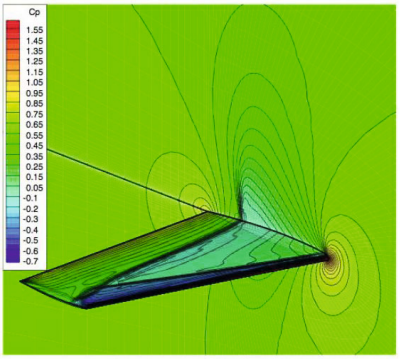


Fig. 22. ARTS pressure distribution for the RBC3 scheme, testcase U2

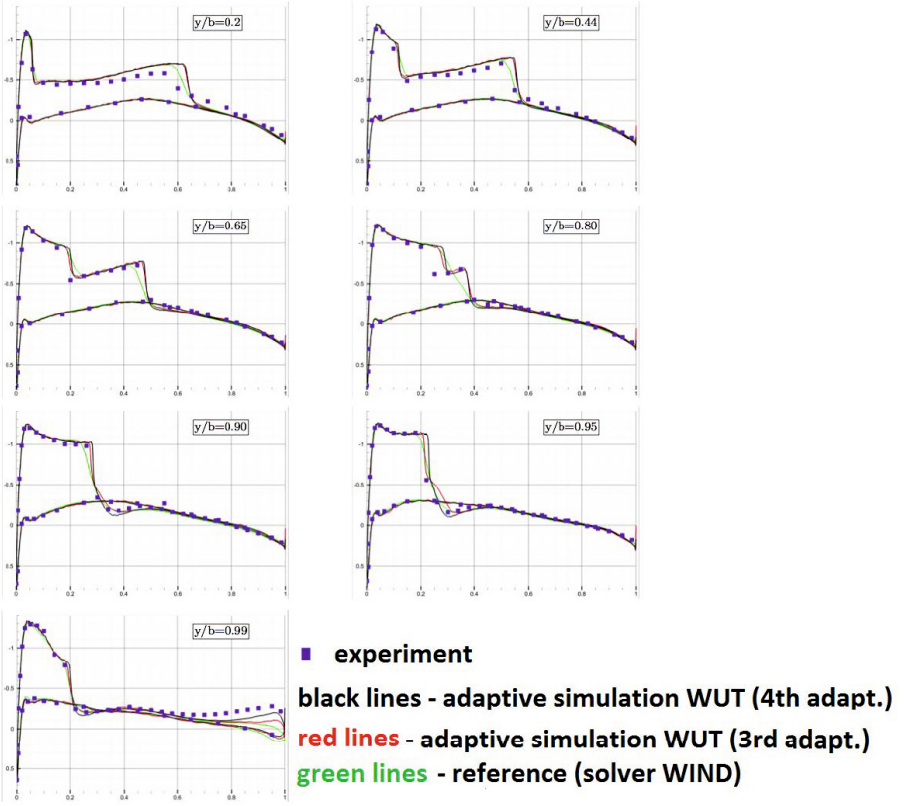


Fig. 23. Comparison between experiment, reference and the WUT results for testcase U2

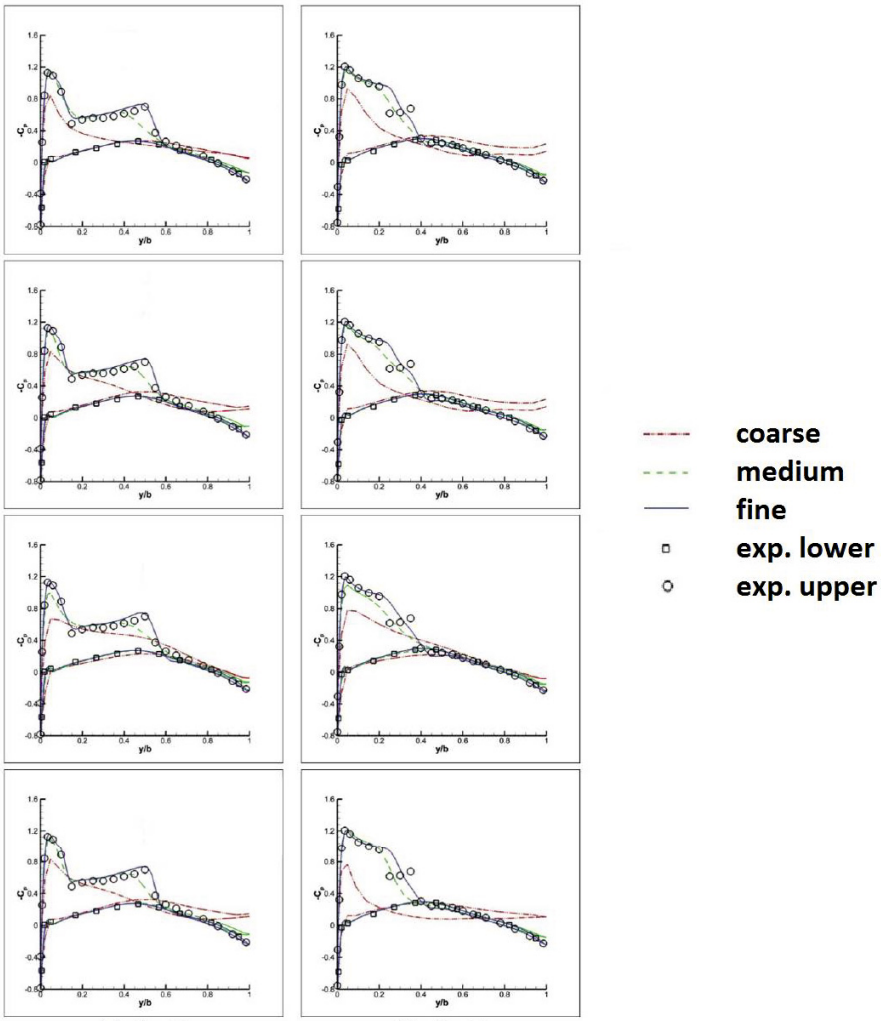


Fig. 24. ARTS results for testcase U2. Shown are surface pressure coefficients at stations 0.2 and 0.8. From top to bottom: RBS3, RBC2, Jameson and Roe. Jameson and Roe results are computed using the industrial code elsA.

2.3 Case U3: L1T2 Multi Component Airfoil

This testcase was coordinated by DLR. It features a simple three element high lift airfoil and thus serves as underlying case to the application challenge A.3, which can be considered a 3D version of this simpler 2D case. The L1T2 airfoil is a well documented case with experimental results and has often been used to verify and validate numerical results.

The flow conditions considered are $M = 0.197$, $\alpha = 20.18°$ and $Re = 3.52 \cdot 10^6$. Engineering accuracy is defined as reducing the error below 0.04 and 0.001 for the lift and drag coefficients respectively.

Two families of nested multi-block structured grids have been generated and provided by DLR for this testcase. The finer grid levels can be exploited to deliver interpolation points for high order meshes, such that the resulting meshes are high order curvilinear. DLR has created meshes with quartic mappings out of mesh family A and offered them to the consortium. The meshes are available with 4 268, 17 072, 68 288, and 273 152 elements. UNIBG has used a similar approach to convert meshes from family B to high order. Here, mesh resolutions of 1 185, 4 740, and 18 960 elements have been produced. TsAGI has used its own sequence of unstructured purely hexahedral hanging node meshes generated with NUMECA's mesh generator HEXPRESS. The meshes are treated as curvilinear based on a reconstruction within the solver. Figure 25 visualizes the three mesh levels with 2 836, 7 432, and 20 854 elements, respectively. While the refinement can be seen, it is also clear that these three meshes do not constitute a nested hierarchy of meshes and do not necessary belong to the same "family of meshes".

DLR has provided reference solutions with the unstructured node-centered Finite Volume code TAU [6] for the Spalart-Allmaras turbulence model on all four meshes from family A. Only the straight-sided meshes have been employed. The proposed evaluation procedure on the three finest meshes yields asymptotic values of 4.051 for the lift and 0.0608 for the drag coefficient. The observed orders of accuracy are 2.3 and 2.6 for lift and drag, respectively. While this is slightly larger than the expected order of two, the results seem plausible.

UNIBG and DLR have provided baseline computations with their respective DG codes on families A and B of the structured meshes using the Wilcox-$k\omega$ turbulence model at various solution orders. Asymptotic grid converged values have been obtained via extrapolation of high order results. For the lift coefficient, asymptotic values of 4.000 and 4.003 have been obtained by DLR and UNIBG, respectively, while the reference value for the drag coefficient is determined as 0.0670 and 0.0666, respectively. These values agree well with one another and their difference is well below the tolerance used for evaluating efficiency. Figure 26 illustrates the behavior of the error between the computed baseline results and the respective mesh converged values versus both degrees of freedom and CPU time (measured in IDIHOM work units). The reference results are plotted along with the baseline computations.

Some deviations from an ideal behavior can be observed, in particular the results from UNIBG do not follow a straight line in the log-log-plot for all design orders, which should be the case for any constant observed order of convergence. Probably the initial mesh is too coarse to be within the asymptotic range of convergence. However, in general the values converge well with both mesh and order refinement. The tolerance for lift is easily achieved by several of the computations, while the accuracy in drag is only sufficient for some of the computations. As expected, the advantages of very high order solutions are more pronounced when comparing degrees of freedom, while very high order is less important in terms of CPU time required to obtained values within the tolerance band. However, computations of order three and higher are clearly superior to second order results, when comparing only within the DG results.

The reference Finite Volume results are the fastest to achieve the tolerance in lift, and only a single third order computation of DLR is faster to achieve the tolerance in drag. This outcome might be due to several effects. The TAU code is a well optimized industrial code, whereas both DG codes should be considered as research codes with a lower amount of performance optimization. Furthermore, it is well known that the Spalart-Allmaras turbulence model is more likely to yield favorable solver behavior than two-equation turbulence models.

WUT has successfully applied the developed mesh adaptation strategy to this case. The obtained results based on the Spalart-Allmaras turbulence model are in good agreement with both experimental data and Finite Volume reference results. Figure 27 illustrates the transition from an initial mesh to the flow-adapted final mesh after seven adaptation cycles. Figure 28 plots the pressure coefficient over the airfoil computed on the final adapted mesh with VKI's second order solver Thor [10], which is also employed to drive the adaptation. The agreement with experimental data is excellent, the difference in pressure level on the slat is observed in all numerical computations. Finally, Figure 29 illustrates the error in lift and drag over the sequence of adapted meshes. While the overall error level is larger than for the reference solution at same degrees of freedom, the effectivity of the procedure can be seen in the considerable drop of error in both lift and drag between the last two adaptation cycles with a minimal increase in mesh size. Furthermore, the absolute level of error is also strongly influenced by the flow solver, not only by the mesh itself.

UNIBG and TsAGI have supplied results for their new implementations of an Explicit Algebraic Reynolds Stress Model (EARSM). An EARSM has been implemented in the k-log(ω) model of the UNIBG's DG code MIGALE, using the 1988 (and 2006) Wilcox's model constants. Completeness and correctness of EARSM analytic contributions to the residual Jacobian matrix have been thoroughly assessed. Despite the careful implementation, which resulted in a very small computational overhead, the results of the U.3 test case did not show any improvement over the standard k-log(ω) model. UNIBG experience using EARSM is that convergence to steady state solutions is often quite difficult, if possible at all. For the U.3 test case, instability of the shear layer enclosing the flow in the cove region of the main profile prevented convergence of higher-order

(orders six and seven) DG solutions. An accurate numerical investigation on basic flat plate and shear flow configurations revealed that EARSM, as other non-linear eddy viscosity models, under-predicts the spreading rate of shear layers. Numerical experiments, modifying the model constants and details of its implementation, showed that it is, in fact, possible to improve EARSM prediction of such basic flows. How these modifications affect force coefficients of the U.3 test could not be deeply investigated. Some of the results of UNIBG are illustrated in Figure 30, which demonstrates the low level of difference between the $k\omega$ and EARSM results.

In Figure 31, the results from all the computations performed for the testcase are presented. Under mesh refinement, the TsAGI computations converge towards values similar to the reference for both lift and drag, at least for orders three and four. For the second order results, the meshes are probably still too coarse to obtain good results in the integral coefficients. This effect can also be seen in the extrapolated mesh asymptotic values per order of the DG method. The lift and drag values behave monotonically (per order) for the results of TsAGI. The extrapolation to an asymptotic value was conducted in two different ways; by using three meshes for a given scheme order or by using the two finest mesh results and assuming the nominal scheme order. The results obtained for both methods are summarized in Table 5 for lift and Table 6 for drag. The observed orders of accuracy are unrealistically large for polynomial degrees two and three, this may be caused by the non–nested meshes used. However, the associated asymptotic values stemming from the two approaches used are very similar for a given order. Moreover, the obtained third and fourth order results are reasonably close to each other, and also compare well with the results from other partners. The remaining difference could be due to the use of a different turbulence model variant. In Figure 31, the TsAGI results are related to an optimal reference value found by averaging the asymptotic values obtained via the two extrapolation values. In general, two main effects can be observed. Firstly, the higher order solvers are rarely capable of improving the efficiency, measured in computational effort, of the of the reference computations. Secondly, the largest improvement for the DG schemes is obtained by increasing the nominal scheme order from second to third order, the fourth order results are less efficient for this testcase.

In previous work, UNIBG has presented a formulation of the high order DG method on agglomerated meshes with arbitrary element shapes. Based on this formulation, DLR has developed an h-multigrid formulation in which such agglomerated meshes are exploited in order to accelerate the solution on the original mesh. A similar procedure has later also been implemented by UNIBG.

The proposed procedures have been applied to this case. Figure 32 compares UNIBG results on the performance of the single–grid implicit solver with a variant of multigrid for a third order computation on the unstructured mesh generated by UNIBG. Both solvers use the same linearized Backward-Euler smoother with ILU(0) preconditioner and a maximum of 15 steps within the GMRes linear solver. The multigrid variant uses an FAS approach on two additional coarser

levels, with an agglomeration ratio of two. Due to the increased cost per cycle the advantages of multigrid are reduced when measured w. r. t. CPU time, but they are still considerable in this case. Figure 33 illustrates the original mesh and the two agglomerated coarse levels used by UNIBG.

DLR has computed the same case on the same mesh. While the underlying smoother and the number of GMRes iterations is the same, DLR has used other settings according to its best practice. Again, a total of three levels is used, but with an agglomeration ratio of four. Although a non-linear FAS multigrid formulation is used, the preconditioner in the GMRes linear solver on each level is a linear multigrid algorithm on the same mesh sequence, which employs an iterative line-based smoother. The convergence obtained is illustrated in Figure 34, which shows the behavior of all residual components on all levels of the full multigrid procedure. While the number of multigrid cycles (on the fine mesh) is considerably lower than for UNIBG, each of these is more expensive.

Overall, similar CPU times have been obtained by UNIBG and DLR, see Table 7. It should be noted that UNIBG has implemented the multigrid algorithm in its standard DG code MIGALE, whereas DLR's implementation has been done as prototype separate from the standard DG code PADGE.

DLR has computed a DG solution on the adapted high order mesh generated by WUT. As the adaptation was done based on a solution for the Spalart-Allmaras turbulence model, the same model has been applied in the DG computations. Figure 35 gives an impression of the mesh and the Mach number distribution computed with a third order DG method.

Figure 36 illustrates the error in lift and drag computed for a third and fourth order solution on this mesh in comparison with the Finite Volume reference data. The asymptotic value of the Finite Volume data has been used for the computation of errors in the DG case, as no mesh convergence study could be done on the adapted mesh. The actual asymptotic values might be slightly off due to subtle differences in the airfoil geometry approximation as well as the location and treatment of far-field boundaries.

The large variety of meshes generated for this testcase, in particular, unstructured mixed–element curvilinear meshes, represents progress in the higher–order mesh generation capabilities, it however remains unclear whether the progress is transferable to 3D. Already the baseline results, computed in the beginning of the project, demonstrated the applicability of the higher order methods to this high lift configuration. For the DG methods, order three is essential to reduce the required discrete system size, it is however not clear whether improvements can be made in the computational time for this testcase. The final higher order results from UNIBG, DLR and TsAGI are in good agreement and new developments in flow solver technologies have been successfully demonstrated for this testcase. This includes the usage of agglomerated multigrid iterative solver algorithms as well as the use of enhanced turbulence models.

Table 5. Extrapolation of asymptotic reference values for the lift coefficient for the results of TsAGI for testcase U.3

	using two meshes		*using three meshes*	
polynomial degree	theoretical order	asymptotic value	observed order	asymptotic value
1	2	3.956	2.2	3.939
2	3	4.000	6.7	3.995
3	4	3.990	5.0	3.988

Table 6. Extrapolation of asymptotic reference values for the drag coefficient for the results of TsAGI for testcase U.3

	using two meshes		*using three meshes*	
polynomial degree	theoretical order	asymptotic value	observed order	asymptotic value
1	2	0.1185	3.4	0.1215
2	3	0.0673	4.4	0.0678
3	4	0.0638	6.5	0.0640

Table 7. Comparison of single-grid and multigrid CPU times (in IDIHOM work units) for results of UNIBG and DLR at order three on the same mesh for testcase U.3

		time to converge	
partner	algorithm	C_L	C_D
UNIBG	single-grid	2 894	
UNIBG	multigrid	866	
DLR	multigrid	757	809

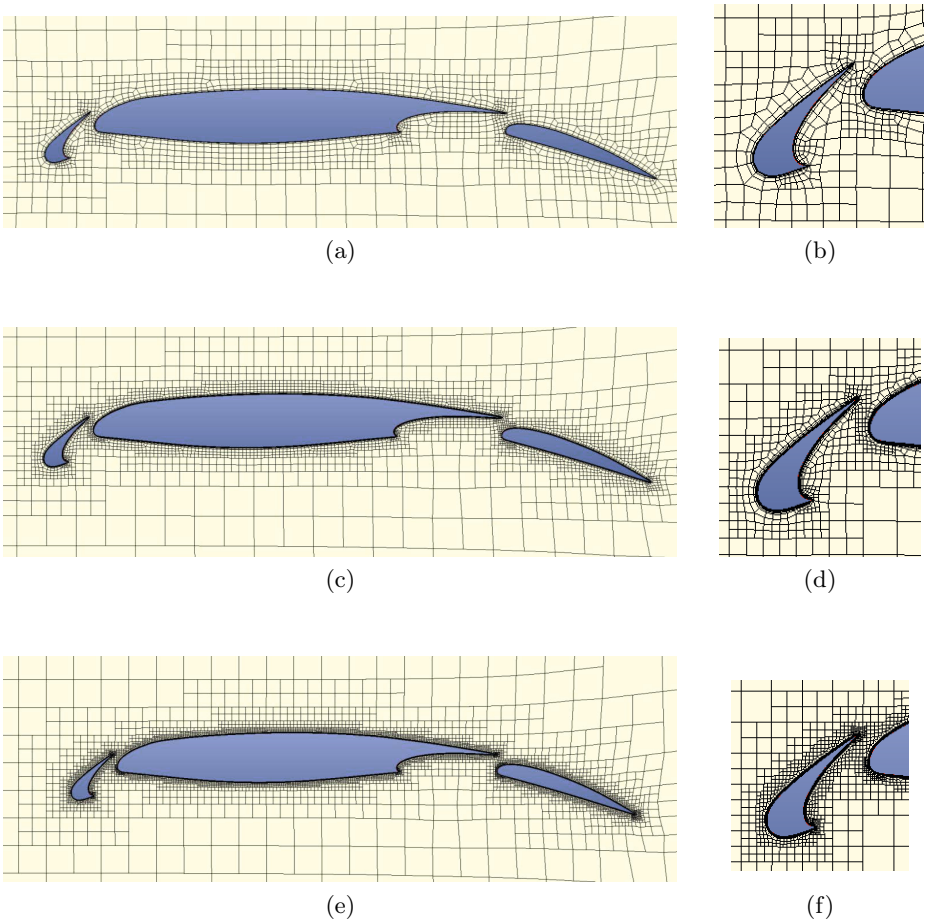


Fig. 25. Unstructured meshes generated by TsAGI from coarse (top) to fine (bottom): overall view of the airfoil (left) and details at the slat (right)

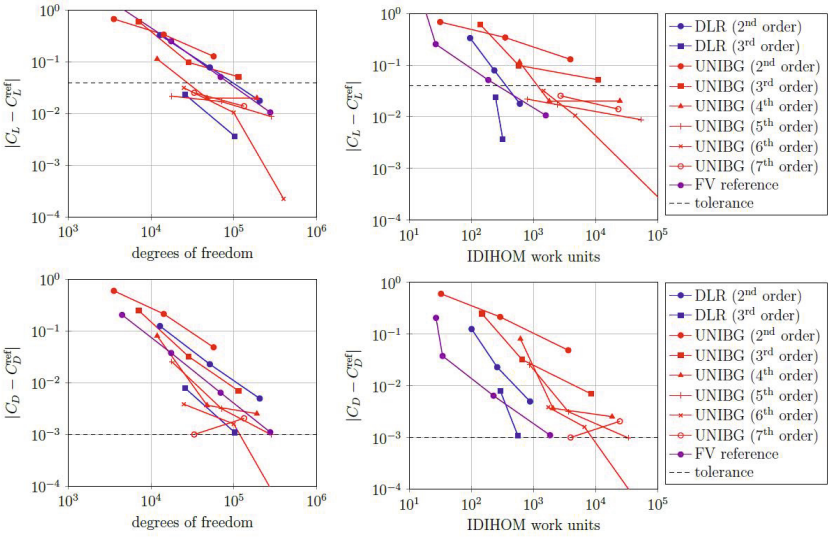
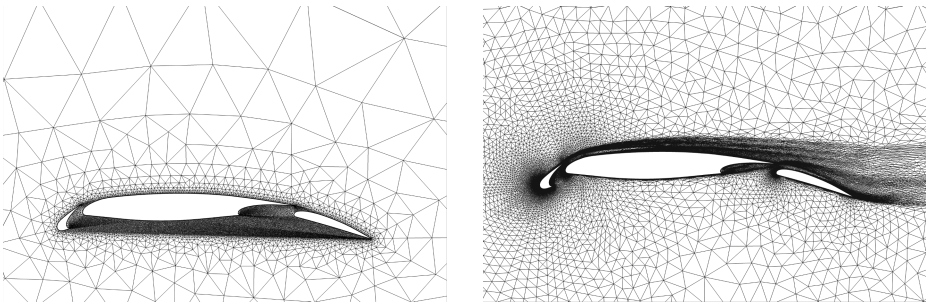


Fig. 26. Baseline asymptotic convergence plots for case U.3. Shown are higher order results from DLR and UNIBG as well as the industrial reference.



(a) (b)

Fig. 27. Initial mesh (left) used in the adaptive procedure of WUT and final flow adapted mesh after seven adaptation cycles (right) for testcase U.3

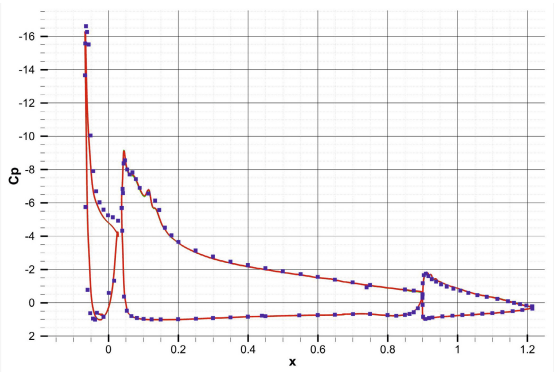


Fig. 28. Computed pressure coefficient distribution (red line) over the airfoil for final adapted mesh of WUT in comparison to experimental data (blue symbols)

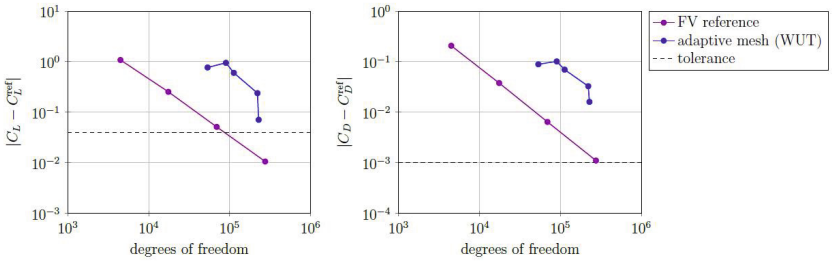


Fig. 29. Comparison between mesh convergence for the WUT adaptation computations and reference results for testcase U.3

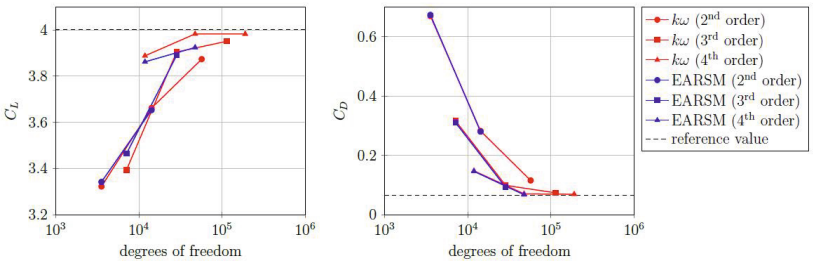


Fig. 30. UNIBG mesh convergence comparisons between $k\omega$ and EARSM turbulence models for the U.3 testcase

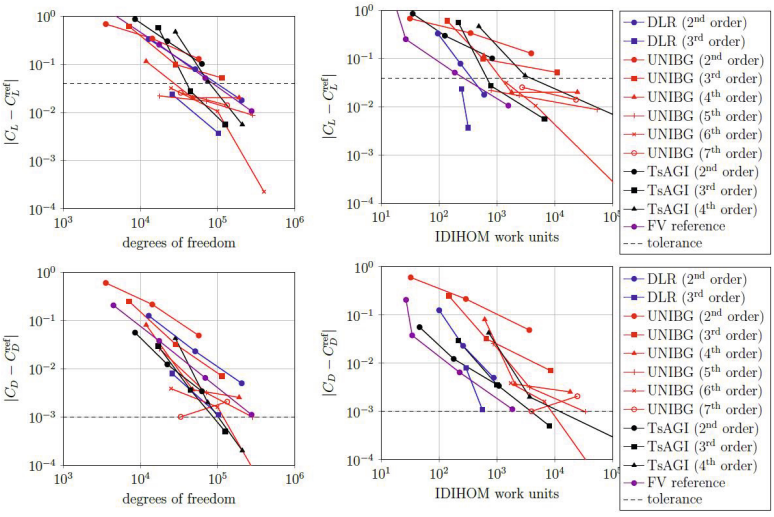


Fig. 31. Mesh convergence of error in lift and drag coefficients for reference, baseline and high order computations w.r.t. degrees of freedom and computational time, testcase U.3

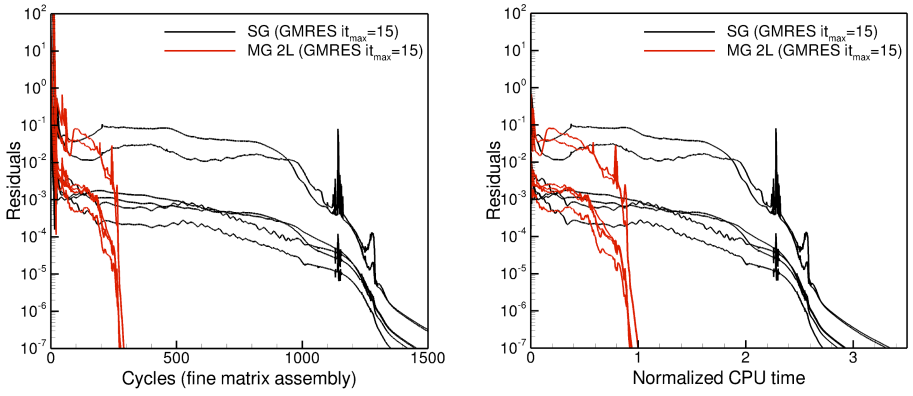


Fig. 32. Comparison on UNIBG's h-multigrid (red line) performance with a single-grid solver (black line) w. r. t. non-linear iterations (left) and CPU time (right), testcase U.3

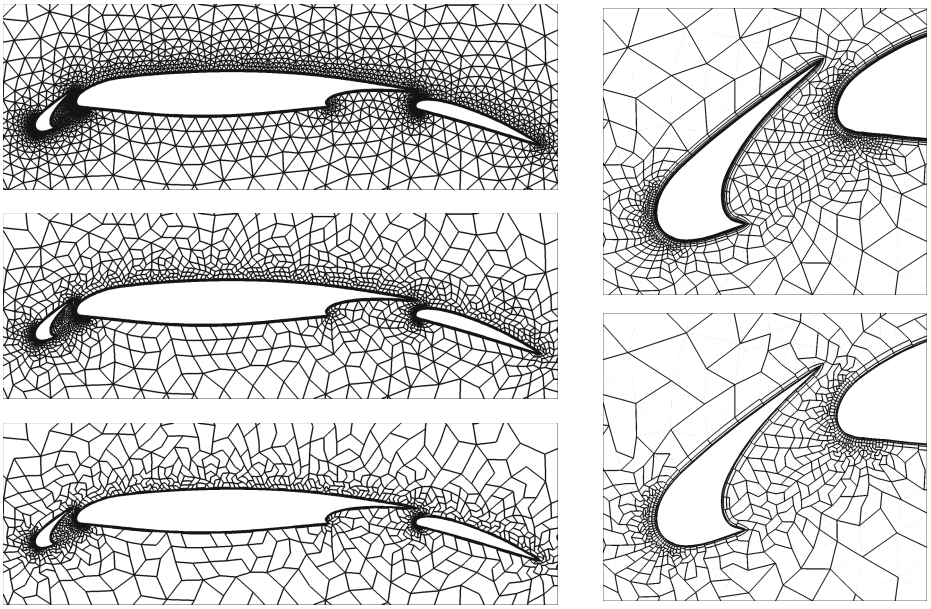


Fig. 33. Unstructured mesh of UNIBG and two agglomeration levels (left), as well as details of the two agglomerated meshes in the slat region (right), testcase U.3

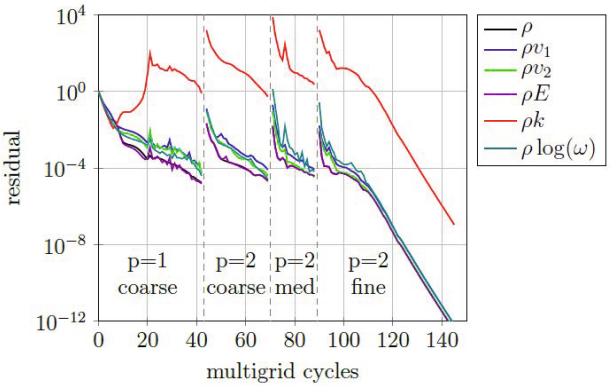


Fig. 34. Iterative convergence of third order computation on UNIBG mesh for DLR's h–multigrid algorithm on testcase U.3

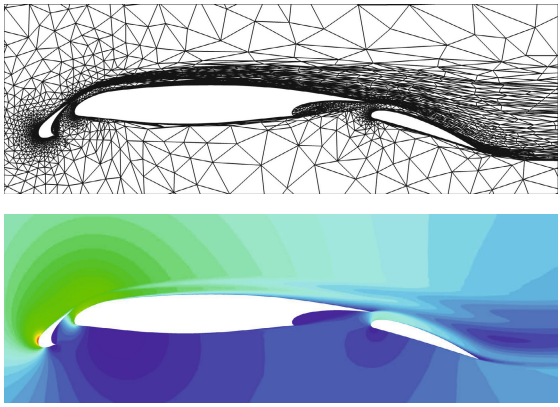


Fig. 35. High order mesh of WUT (top) and Mach number distribution computed by DLR with a third order DG method (bottom), testcase U.3

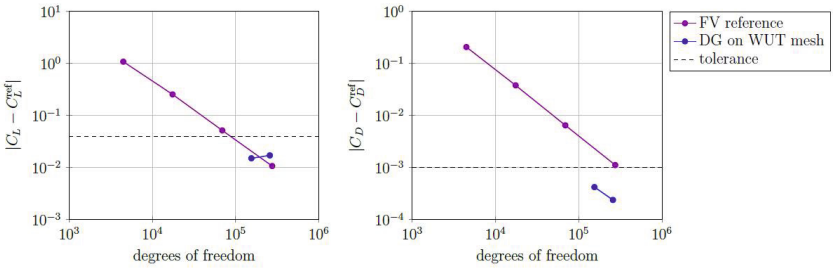


Fig. 36. DLR higher order results for testcase U.3 computed on a WUT adapted mesh, compared with industrial reference data

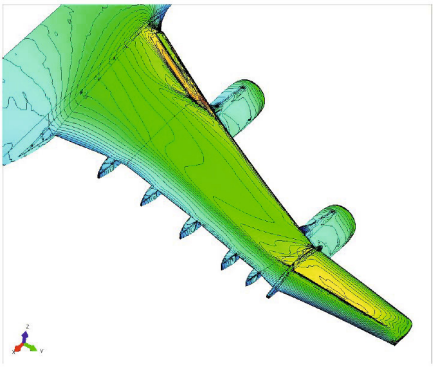


Fig. 37. Reference computation using a standard FV approach for the A1 testcase

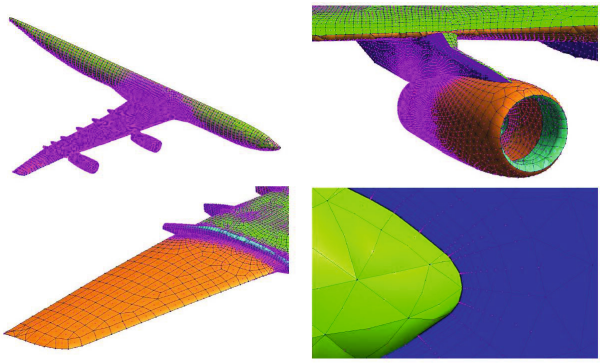


Fig. 38. Higher order surface mesh generated by ARA for the A1 testcase

2.4 Case A1: Clean Sky Aircraft

This application testcase was coordinated by DLR. The geometry considered is the Airbus laminar wing testcase, consisting of a modified A340 aircraft with a reduced–sweep outer wing section, Figure 37. Even though the case was given priority both by industry and the project reviewers, a higher order unstructured mesh could not be produced and thus no computations could be performed in IDIHOM for this configuration. Some illustrations of the higher order surface mesh generated during the project are shown in Figure 38.

2.5 Case A2: FA5 Aircraft

This application testcase was coordinated by Airbus D&S. The configuration considered is a delta–wing fighter aircraft with a complex geometry and flow topology. Two subcases were defined; case A2a is a subsonic testcase for which hot wire anemometry measurements have been conducted [11], and the transonic high angle of attack case A.2b, representing a flow condition of importance to industry. A thorough quantitative analysis into the improvements obtainable by higher order discretization was not expected; rather this testcase serves as an indication of the robustness and modeling accuracy currently obtainable by higher order implementations.

For case A.2a the flow condition $M = 0.125$, $\alpha = 15^\circ$ and $Re = 2.78 \cdot 10^6$ is considered, while for A.2b $M = 0.85$, $\alpha = 24^\circ$ and $Re = 4.65 \cdot 10^7$. The industrial reference was provided by Airbus D&S, employing the DLR Tau code on unstructured meshes, The higher order unstructured meshes, originally planned to be provided by ARA, were, due to technical difficulties, not made available during the project. The higher order computations were thus conducted on a structured mesh provided by Airbus D&S, converted into a curvilinear mesh using an agglomeration algorithm developed by UNIBG, Figure 39.

For the A.2a case, UNIBG have conducted preliminary X–LES computations, Figure 40. UNIBG have also provided results for the A.2b case, compared with the industrial reference in Figure 41. Some convergence problems caused by shocks were reported. The surface pressure plots are comparable to the industrial baseline for this very coarse mesh, indicating the potential of higher order solvers. The capability of solving complicated flows was also demonstrated even though the robustness of the new solvers have not yet reached an industrial level for this testcase.

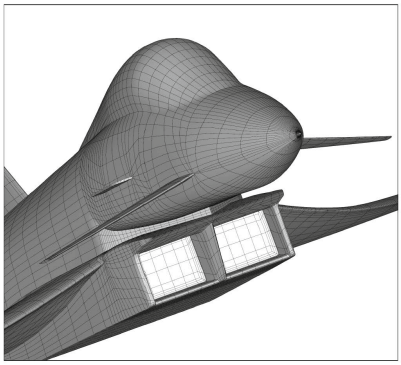


Fig. 39. Agglomerated structured higher order mesh used for the A2 case, consisting of $1.6 \cdot 10^5$ 50–node hexahedral elements.

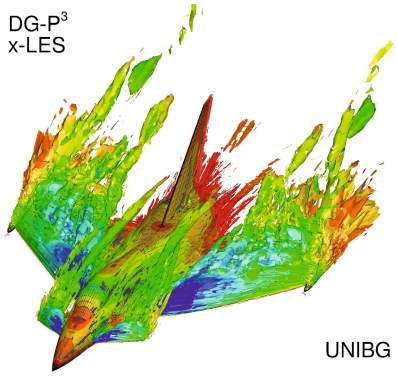


Fig. 40. Illustration of the A2a testcase, as computed by UNIBG

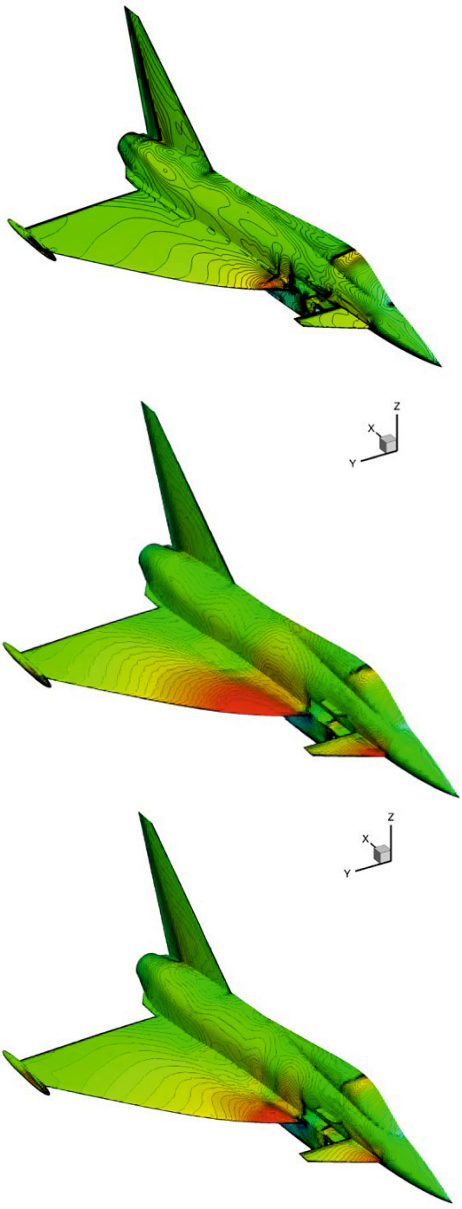


Fig. 41. Surface pressure plots for the A.2b testcase. From top to bottom: Airbus D&S Tau reference computation, UNIBG p=1 computation, UNIBG p=2 computation.

2.6 Case A3: High Lift Prediction Workshop Case 1

This application testcase features the NASA trapezoidal wing (trap wing) under high lift conditions. The same case has also been considered in the First AIAA CFD High Lift Prediction Workshop and involves a simplified geometry consisting of just a slat, a main wing and a flap attached to a body-type wind tunnel mounting without modeling of brackets between the elements.

This case uses the geometry and flow conditions defined as *configuration 1* for the First AIAA CFD High Lift Prediction Workshop. Detailed information is well documented on the corresponding web site [12]. The flow is treated as symmetric, thus a half model is applied. Two subcases are considered, where the relevant flow parameters for subcase A are $M = 0.2$, $\alpha = 13^\circ$ and $Re = 4.3 \cdot 10^6$, and $M = 0.2$, $\alpha = 28^\circ$ and $Re = 4.3 \cdot 10^6$ for subcase B. The cases are treated as fully turbulent. The turbulence model to be used for the computations was not specified. WUT used Spalart–Allmaras, whereas DLR and UNIBG employed the Wilcox-$k\omega$ model. TsAGI performed computations with their EARSM model.

Multi-block structured grids generated by Boeing are available from the original workshop web site. Due to embedded structured coarse levels, the baseline grid can further be coarsened up to a coarse mesh of just 93 088 elements. The finer grid levels can be exploited to deliver interpolation points for high order meshes, see Figure 42.

DLR created meshes at various resolution levels with quartic edge mappings and offered them to the consortium, in particular ONERA received the meshes. UNIBS used a similar approach to create meshes based on a quadratic mapping. TsAGI created multi-block structured linear meshes at different resolution levels, Figure 43.

The coarse and medium mesh were used with the DG code, while the extra fine mesh is used for a reference computation with a finite volume code. ARA worked on the mesh generation for this case and has seen some issues when creating a high-order volume mesh, while a surface mesh could be created. Due to a shift in priorities, the volume mesh was not completed. ICL has tried to extend the high order meshing capabilities from the single wing ONERA-M6 case to this multi element wing case. The creation of a linear volume mesh was not successful, thus the high order mesh could not be delivered.

A finite volume reference solution on a fine mesh has been performed by TsAGI. DLR provided baseline solutions on the coarsest structured mesh at second and third order with DLR's DG code. Figures 44 and 45 illustrate the results obtained. In particular, the topology of the flow in the symmetry plane behind the body changes considerably between second and third order.

WUT successfully applied the developed mesh adaptation strategy to both sub-cases A and B. The obtained results are in very good agreement with both experimental data and finite volume results from the workshop. At 6.8 and 11.4 million degrees of freedom for the linear regime and maximum lift case, respectively, the associated final adapted meshes are quite coarse for a second order flow solver. Figure 46 shows details of the adapted mesh for case A, while Figure 47 illustrates the quality of the results. The deviation between computational

results and experimental data for the pressure distribution is similar to what is observed for the high order results. Unfortunately, coarse curvilinear versions of the adapted meshes were not generated. Thus, they could not be used together with high order flow solvers.

Figure 48 illustrates the results that TsAGI obtained with second order DG on the coarse and medium mesh and with third order DG on the coarse mesh in comparison with the finite volume reference data. Even though the degree of freedom count is lower for the third order computation on the coarse mesh compared to the second order computation on the medium mesh, the solution is visually superior, compared to the finite volume reference. However, all three high order computations suffer from too little resolution and are far from mesh convergent. Computations on finer meshes or with higher order did not converge.

The pressure distribution in a cut at 95 % span is compared against the reference and against experimental data in Figure 49, which also illustrates the third order results from UNIBS on a globally refined level of the coarse workshop mesh. The reference finite volume solution shows a reasonable agreement with experimental data, although some clear differences can be observed. TsAGI's DG results approach the reference solution for increasing mesh density and increasing scheme order. The fact that the third order computation on the coarse mesh is more accurate than the second order result on the medium mesh is verified.

With over 7.4 million degrees of freedom at third order the computation of UNIBS is much better resolved than the third order DG result of TsAGI by a factor of more than 20 in degrees of freedom. Correspondingly, the overall agreement with experimental data is much better for the UNIBS results. In fact, the overall agreement seems even better than for the reference solution. However, some spurious local oscillations present in the refined mesh results indicate the need for further mesh (or order) refinement. For this outboard section of the wing it is difficult to obtain good agreement between experiments and numerical solutions. For all computations, the results are much closer in other sections, as illustrated in Figure 50.

The lift and drag coefficient values of the high order results are plotted vs. degrees of freedom for the results from TsAGI and UNIBS (second to fourth order) in Figure 51. The lift seems to converge towards the experimental value under mesh or order refinement, but the absolute error is still considerable. It is interesting to see that the results from UNIBS show a greater deviation from the experimental value than those of TsAGI, while the corresponding pressure distribution has been found to be in better agreement. This further illustrates the need for better resolution.

The situation is not as clear for drag. While the results of UNIBS seem to converge towards a lower value than the experiments, no clear convergence of the drag coefficient can be observed for the results from TsAGI. The generation of high order unstructured meshes seems to be rather difficult already for the relatively moderate geometrical complexity of this case. The exploitation of multi-block structured meshes may still be considered state-of-the-art for this type of application.

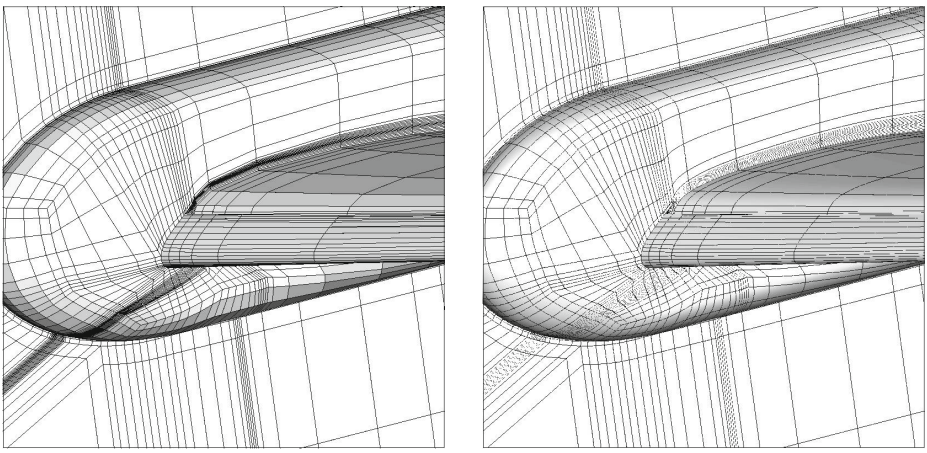


Fig. 42. Coarse structured workshop mesh for NASA trap wing (testcase A3): linear (left) and quartic mapping (right)

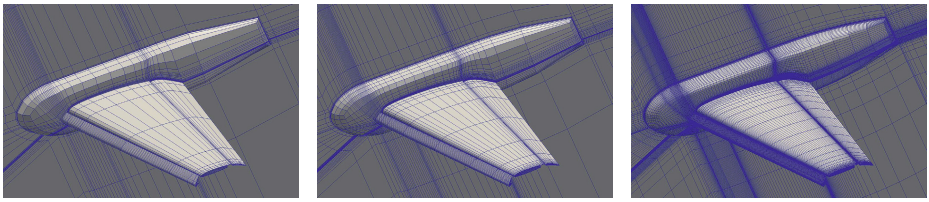


Fig. 43. Structured meshes of TsAGI used for testcase A3: coarse mesh with 32 771 elements (left), medium mesh with 121 014 elements (middle) and extra fine mesh with 2 097 344 elements (right)

Concerning the flow solver side, this case was originally defined as a detailed comparison case. However, all partners (TsAGI, DLR and UNIBS) have encountered convergence problems. The best DG results from UNIBS are still quite far from the experimental integral values. No mesh convergence study is available. Thus, no detailed analysis is possible. Still, the general applicability of the developed high order flow solvers to this application challenge of a 3D high lift configuration has been demonstrated with the presented results. Unfortunately, all results are for the lower angle of attack in the linear regime of the lift polar whereas no high order results have been presented for the angle of attack close to maximum lift.

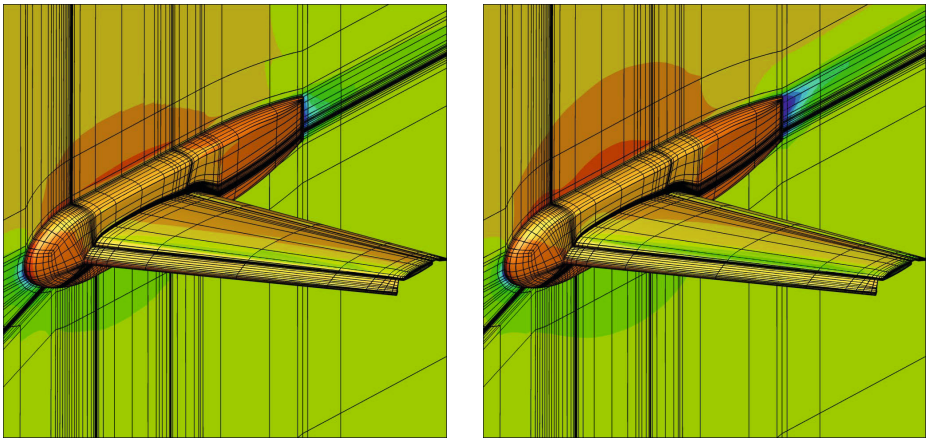


Fig. 44. Baseline results for testcase A3: Mach number in the symmetry plane and pressure coefficient on the wing and body at second (left) and third order (right)

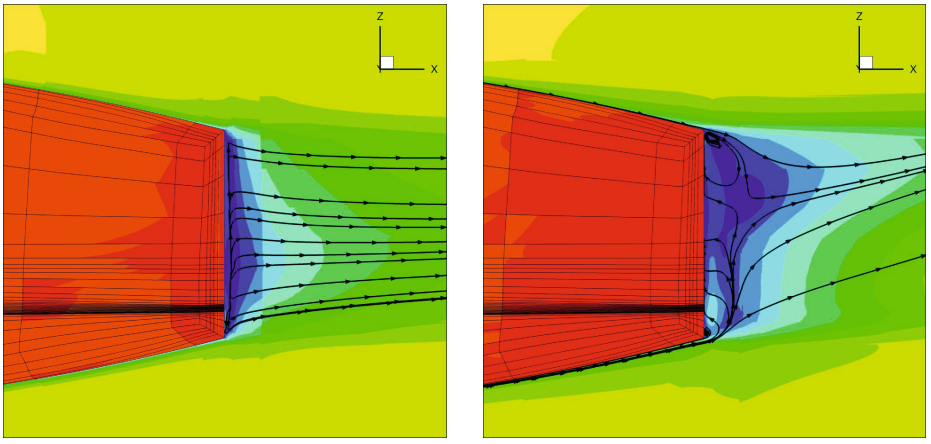


Fig. 45. Baseline results for testcase A3: Streamlines in the symmetry plane behind the body at second (left) and third order (right)

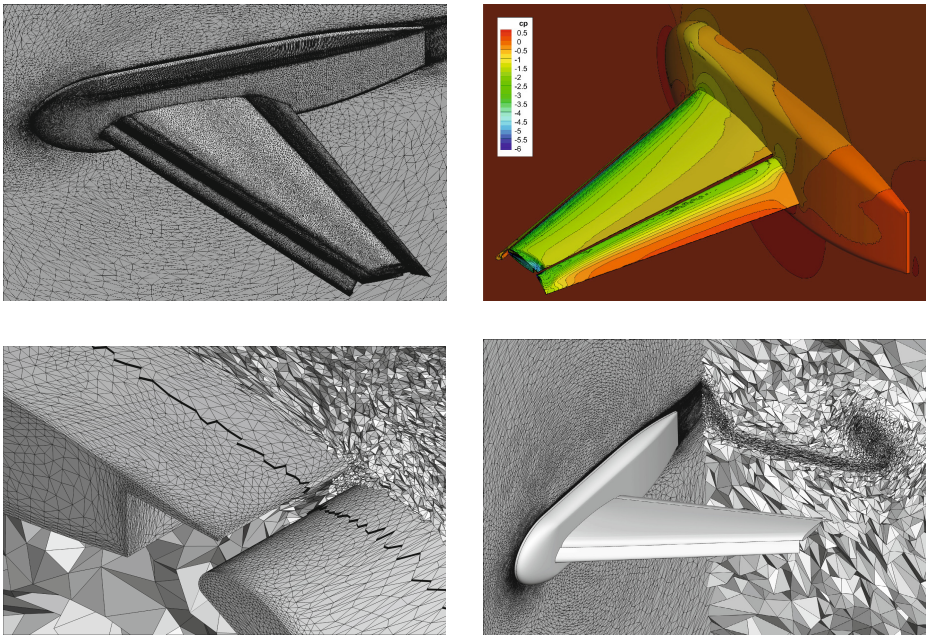


Fig. 46. Final adapted mesh of WUT for testcase A3, A: surface mesh (upper left), details of the mesh in the main wing / slat region (lower left) and the wake (lower right), as well as computed pressure distribution (upper right)

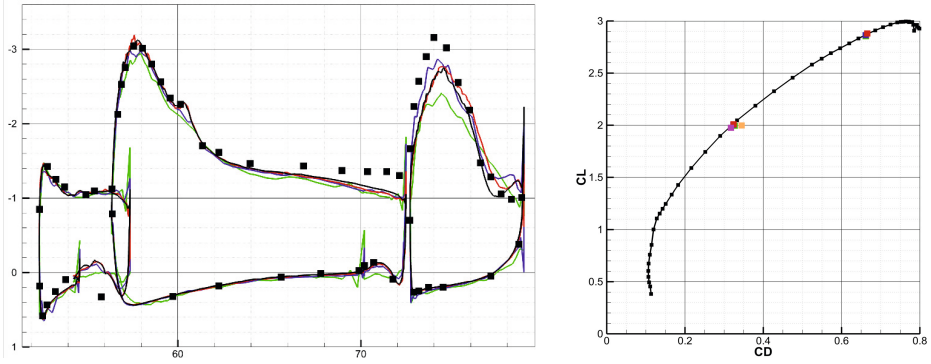


Fig. 47. Second order results on the adapted meshes of WUT for testcase A3: pressure coefficient in a cut at 95 % span for case A (left) and results for cases A and B compared to the experimental polar (right). Results on the final adapted mesh are represented by black lines and red boxes, respectively, while experimental results are represented by black boxes.

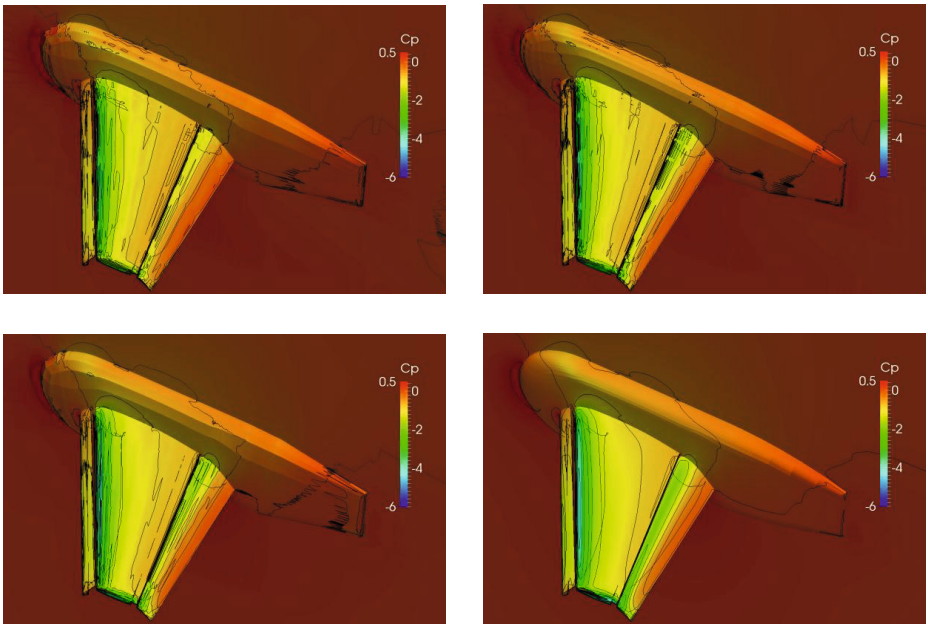


Fig. 48. Pressure distribution results from TsAGI: second order on coarse (upper left) and medium mesh (upper right) as well as third order solution on coarse mesh (lower left) and finite volume reference solution on extra fine mesh (lower right)

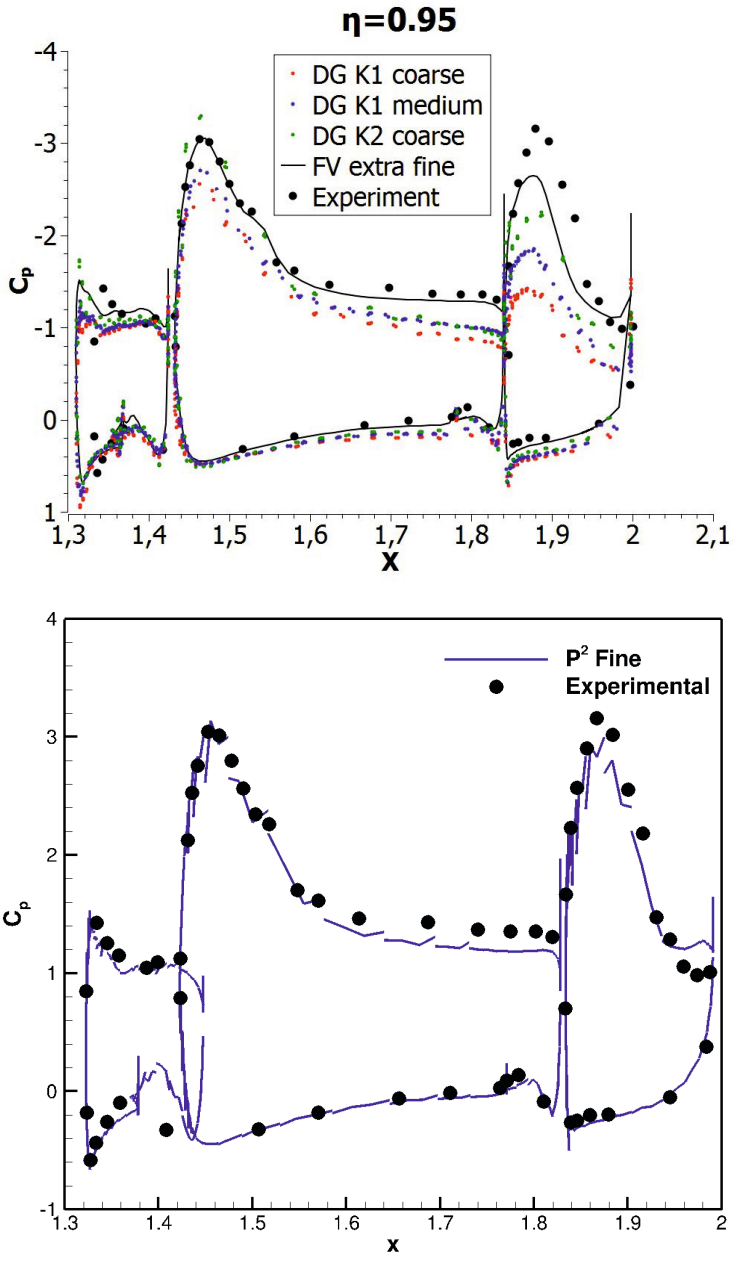


Fig. 49. Comparison of pressure coefficients for testcase A3, showing various computations and experimental results in a cut at 95 % span: results from TsAGI (top) and UNIBS

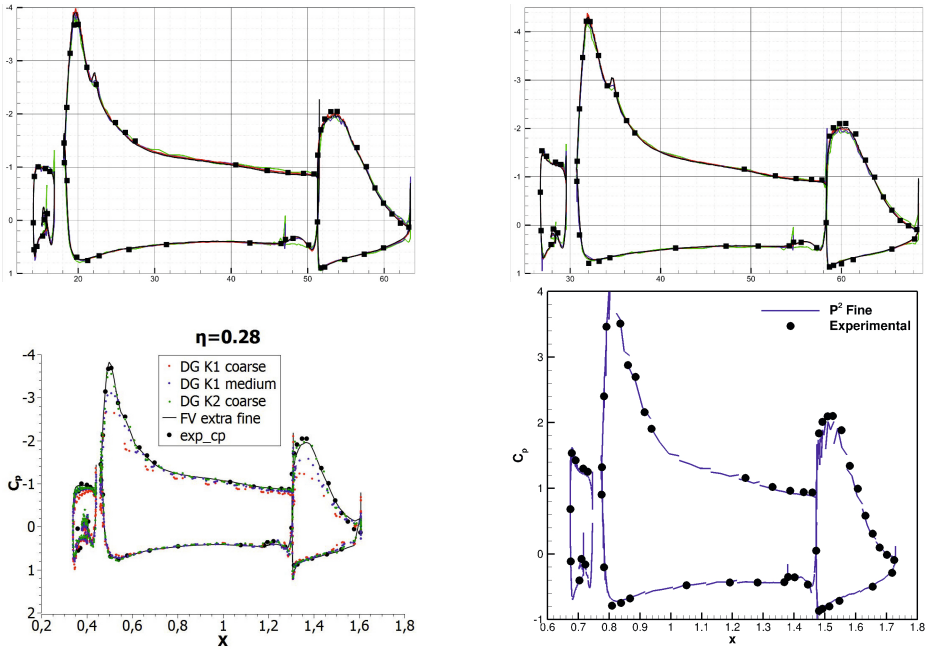


Fig. 50. Comparison of pressure coefficients for testcase A3, showing various computations and experimental results in mid-wing sections: a) cut at 28 % span with results from WUT (upper left) and TsAGI (lower left), b) cut at 50 % span with results from WUT (upper right) and UNIBS (lower right)

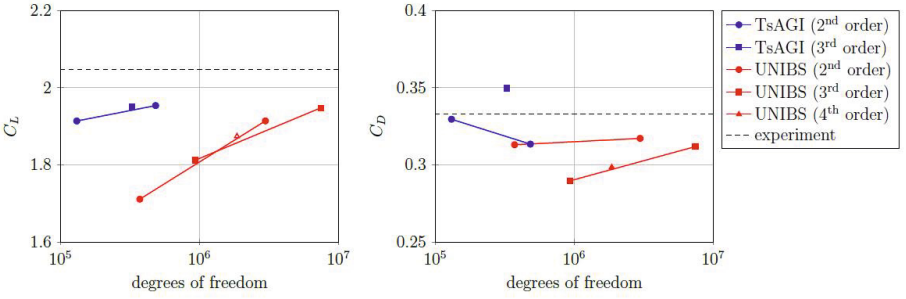


Fig. 51. Lift and drag mesh convergence results for testcase A3

2.7 Case A4: Generic Falcon Aircraft

The Generic Falcon Test Case A.4b was designed in order to give access to a complex aircraft configuration to IDIHOM partners in addition to Dassault Aviation. Originally four partners intended to compute the test case. Unfortunately, only two partners were able to produce reference results for this test case (DASSAV and NUMECA), and a single partner (DASSAV) was able to produce higher-order results. The configuration is presented in Figure 52. It is a twin-engine Falcon-like aircraft with flow-through nacelles. The flow conditions considered were: $M_\infty = 0.80$, $\alpha = 2.0°$, alt $= 40,000$ ft, which corresponds to $Re = 14,512,000$ based on the mean aerodynamic chord.

Reference meshes were generated by NUMECA using their automatic hexahedric unstructured mesh tool Hexpress™. Four meshes were created; coarse, fine and their optimized versions. Surface views of the optimized coarse mesh are presented in Figure 53. These meshes were further refined in the wake with identical refinement for the coarse and the fine grids. The coarse and fine meshes, optimized for orthogonality, respectively contained $13,015,824$ and $51,256,490$ cells. Reference computations were performed by NUMECA using the FINE/Open™ code with the Spalart-Allmaras turbulence model (extended wall functions and a $y^+ = 50$ for the first cell), central second order scheme, and 4 levels multigrid.

DASSAV generated a state-of-the-art reference tetrahedral P1 mesh, according to current industrial standards, containing 8.4 million grid points, Figure 54. For all computations, Dassault Aviation used its industrial stabilized finite element code `AeTher`. It relies on *continuous* isoparametric Lagrange polynomials of any order computed on unstructured grids as finite element shape and trial function spaces (solutions are C^0 continuous and the same degree of interpolation is used for both the solution variables and the space coordinates). So far, only tetrahedral elements have been implemented in 3-D up to P3. Reference 2nd-order computations were performed by DASSAV using the $K - \varepsilon$ and Spalart-Allmaras turbulence models.

Figure 55 presents pressure coefficient plots along the wing at eight stations along the wing span. It is observed that DASSAV $K - \varepsilon$ and Spalart-Allmaras computations are virtually undistinguishable. The reference computation by NUMECA shows some differences, especially in the midsection of the wing. This may be due to the use of wall functions and a Parasolid tesselation, consisting of 1.6 million triangles, to represent the geometry.

In order to build higher-order grids, the current strategy at Dassault Aviation is to add higher-order degrees of freedom to a coarse P1 "skeleton" mesh. The idea is to make this "skeleton" grid coarse enough, so that the higher-order mesh would contain fewer degrees of freedom than the reference P1 mesh and take advantage of higher-degree polynomials to represent curved regions of the model

with fewer elements, typically the leading edges of the aerodynamic surfaces and the nacelle lips. A "skeleton" mesh was generated with a $y^+ = 2$ for the first element layer and an aggressive growing ratio in the boundary layer, resulting in only 250,000 grid points. P2 and P3 grids (and the corresponding nested P1 grids) were built based on this "skeleton" mesh. The P2 mesh contains just below 2,000,000 degrees of freedom, Figure 56.

The resulting higher-order mesh requires "curving". This is performed in two steps: first the surface mesh is projected onto the CAD definition of the geometry; then the displacement is pushed into the volume mesh with some deformation technique to make the higher-order mesh valid. All tested deformation techniques failed and it was decided to refine the surface mesh in highly curved regions, viz. along the leading edges of the wing (4×), the empennage (2×), and the nacelle pylons (2×). The leading edge of the vertical tail and the lips of the nacelle were not modified. The refined P2 3-D mesh is only slightly bigger than the original, containing 75,000 additional degrees of freedom. The best combination of curving techniques was applied (P2 elastic deformation followed with an optimization with the UCL Gmsh code) yielding a valid mesh. The P2 surface mesh after curving with Gmsh can be seen in Figure 57. It can be observed that the nacelle lips, although not refined, are particularly well represented with P2 curved elements.

Computations were performed on the uncurved P1, P2 and P3 meshes, and the valid curved P2 mesh produced in interaction with UCL. The latter is compared with the reference solution in Figure 58. The acceleration of the flow over the leading edge of the wing is much better represented than in the solution obtained

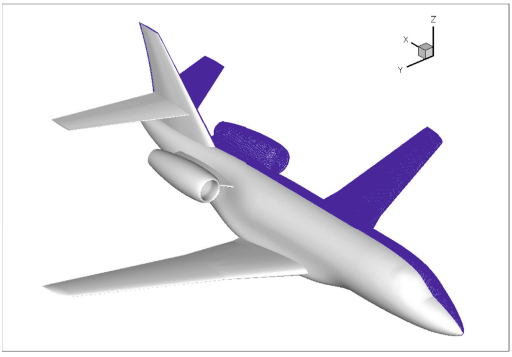


Fig. 52. Case A.4b. Test case geometry

on the uncurved mesh. It is anticipated that this is due to the curving, rather than to the mesh refinement of the leading edge.

Pressure plots for the reference and higher-order solutions at different stations along the wing span are compared in Figure 59. Although on an uncurved grid, the P2 solution is much closer in terms of pressure distribution to the reference computations than the P1 solution with the same number of degrees of freedom. The agreement with the reference solution provided by DASSAV is remarkable, although the 3rd-order P2 solution was obtained with 4 times fewer degrees of freedom (2 vs 8.4 million).

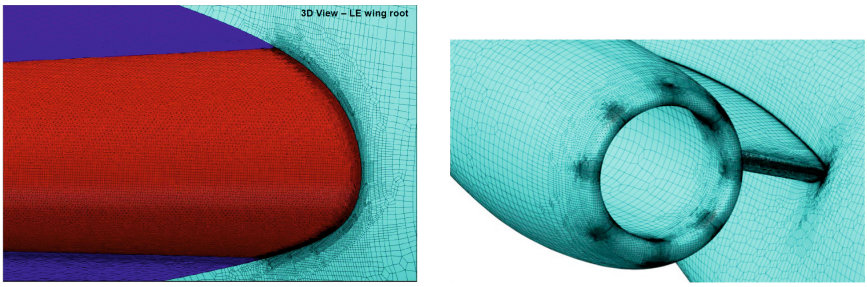


Fig. 53. Case A.4b. NUMECA mesh

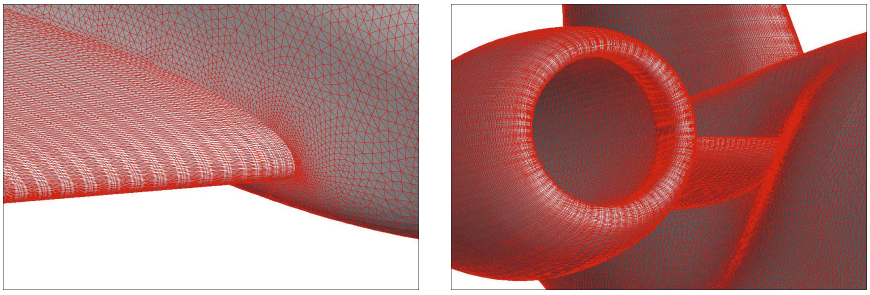


Fig. 54. Case A.4b. Views of the reference P1 surface mesh used by DASSAV

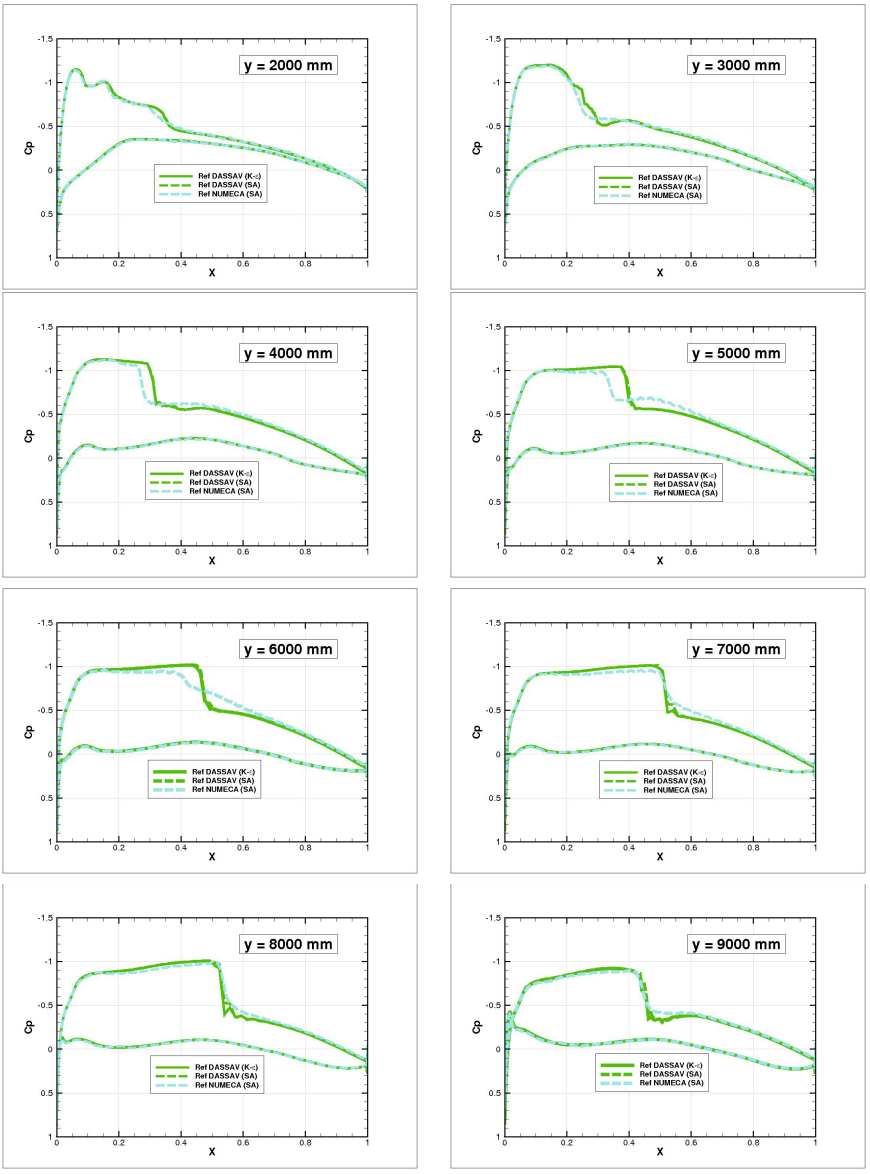


Fig. 55. Case A.4b. Comparison of C_p plots of the reference solutions at different stations along the wing span

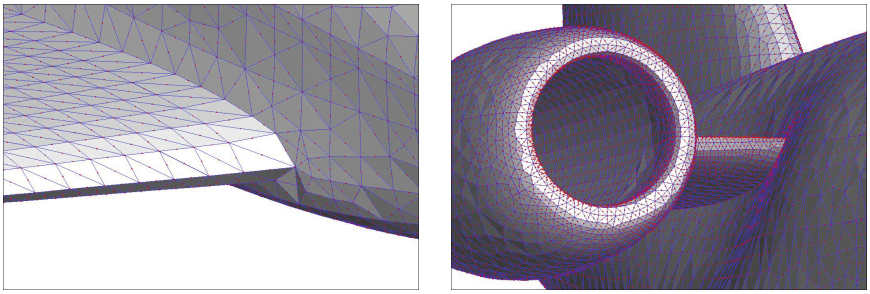


Fig. 56. Case A.4b. Views of the uncurved P2 mesh used by DASSAV

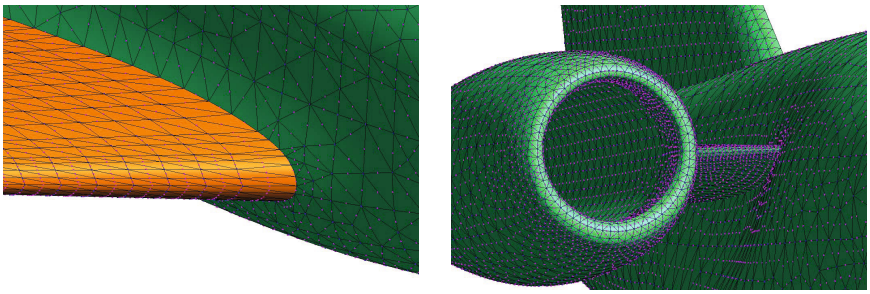


Fig. 57. Case A.4b. Views of P2 mesh used by DASSAV, refined alond the leading edges, curved and optimized with Gmsh

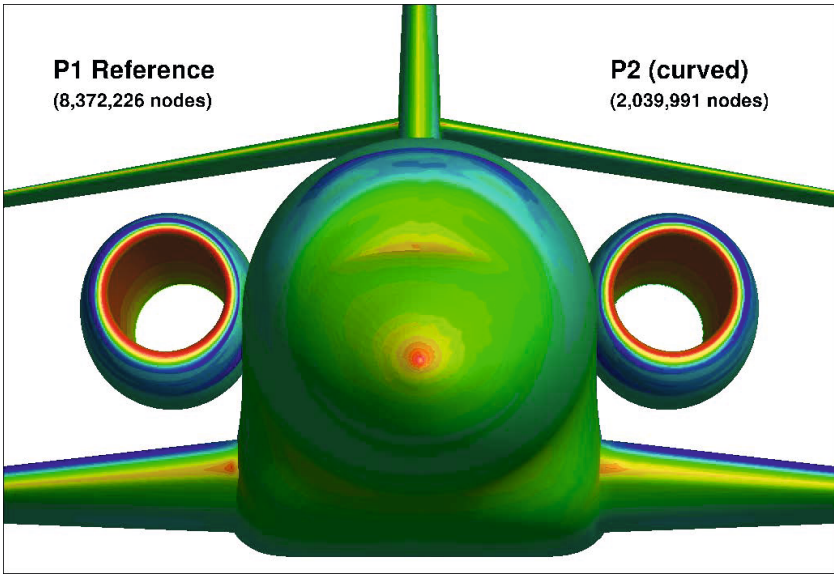


Fig. 58. Case A.4b. Pressure contours: comparison of reference P1 computation with P2 result on **curved** mesh

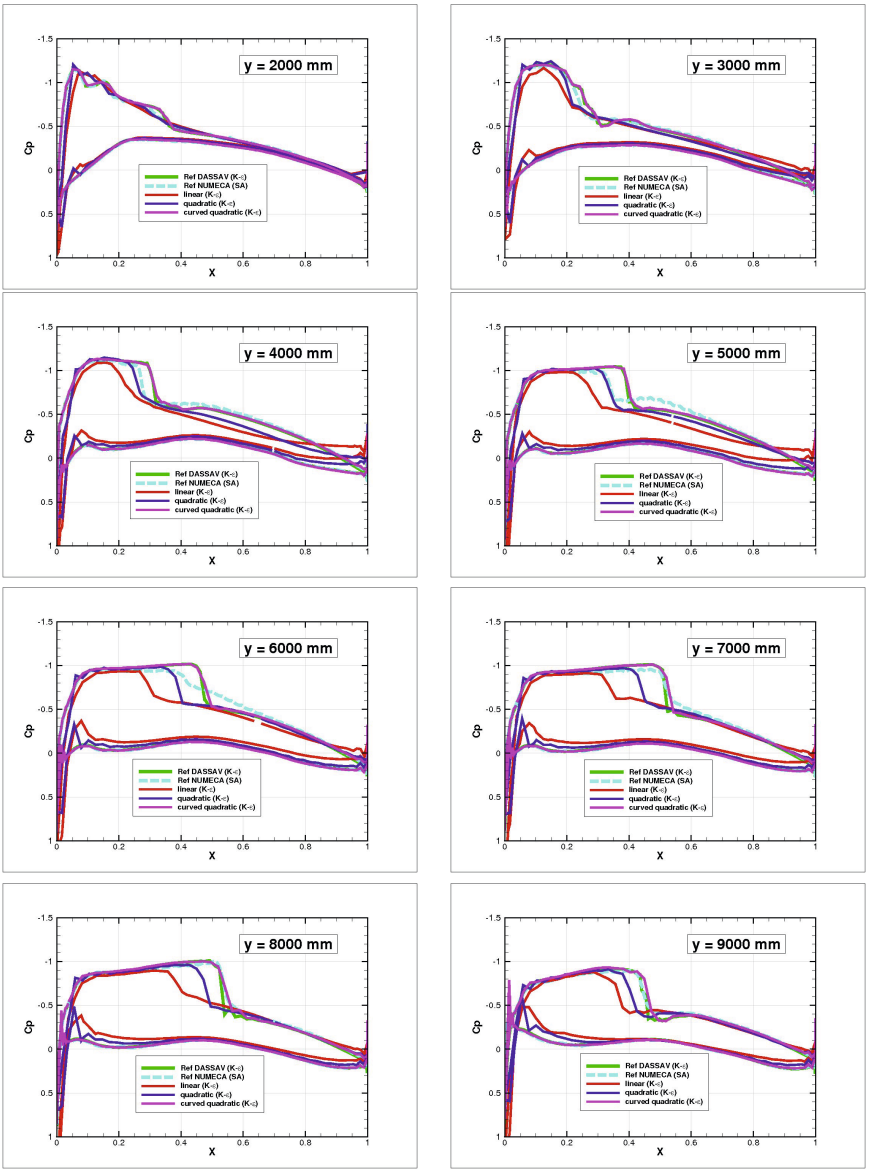


Fig. 59. Case A.4b. Comparison of C_p plots at different stations along the wing span

2.8 Case A6: Train Head

This application testcase was coordinated by UNIBG. The geometry under consideration is the Bombardier Aerodynamic Train Model (ATM). UBS and UNIBG contributed to the testcase, focusing on assessing the benefits of using the EARSM turbulence model for the simulation of vortical flow structures around the body and on comparing the computational efficiency of the Linearized Backward Euler (LBE) and p-multigrid (p-MG) solvers in terms of CPU time and memory usage.

The freestream flow conditions of the test case are wind velocity $U_w = 70m/s$ and Reynolds number based on the reference length $Re_L = 1.2 \cdot 10^6$. The effectiveness of EARSM compared to the linear $k - \omega$ model was assessed by computing the flow field at three different yaw angles, $\beta = 5°, 10°, 15°$. No experimental data have been made available for this geometry. UNIBG generated

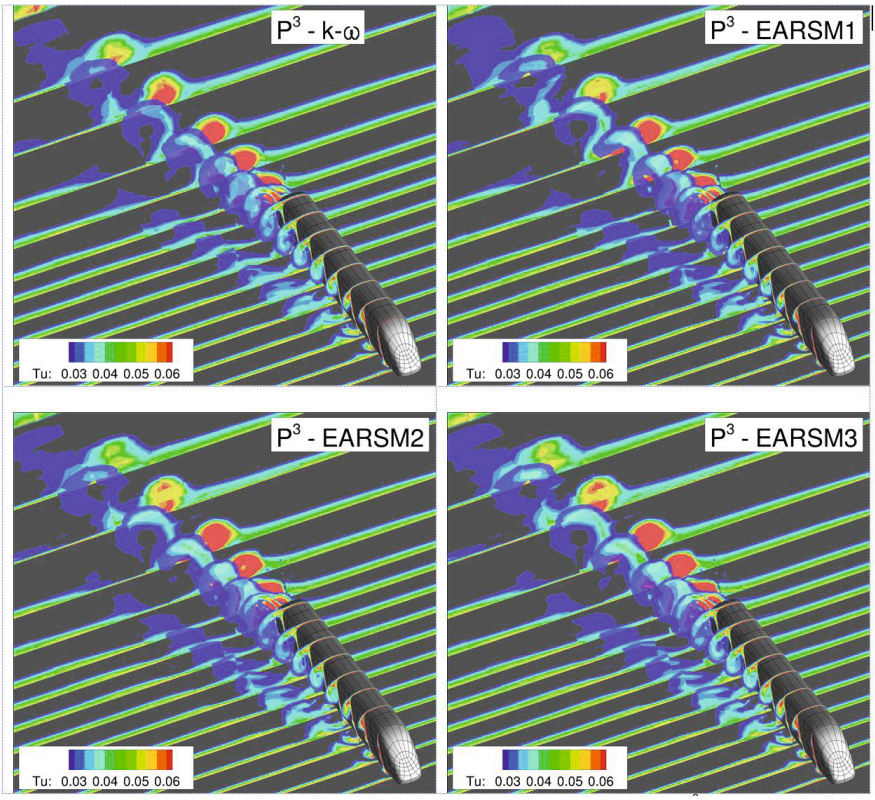


Fig. 60. Turbulence intensity contour $k - \omega$ and EARSM1-3, $\beta = 15°$, $p = 3$ solution for the A6 testcase

fine (linear) structured grids suitable to produce high-order grids by means of an in-house agglomeration software. The high-order results presented in this report were computed on an agglomerated grid with 7776 50-node hexahedral serendipity elements (quartic edges).

EARSM solutions with different contributions to the anisotropy tensor were produced and compared with the results of the $k-\omega$ model, Figure 60, where it is seen that the EARSM computation exhibits a sharper definition of the vortices flowing on the ground, next to and behind the train. The LBE (UNIBG) and p-MG (UNIBS) solvers have been compared in terms of IDIHOM work units and memory requirements needed to compute the $\beta = 10°$ case using the $k-\omega$ model. Both solvers have been implemented using the same multilevel data structure developed on top of the DG code MIGALE. In Figure 61 the convergence behavior of the LBE and p-MG(LBE) solvers is shown. For this case, p-MG converged about 10 to 14% faster than LBE at the cost of an additional 19 to 39% memory usage.

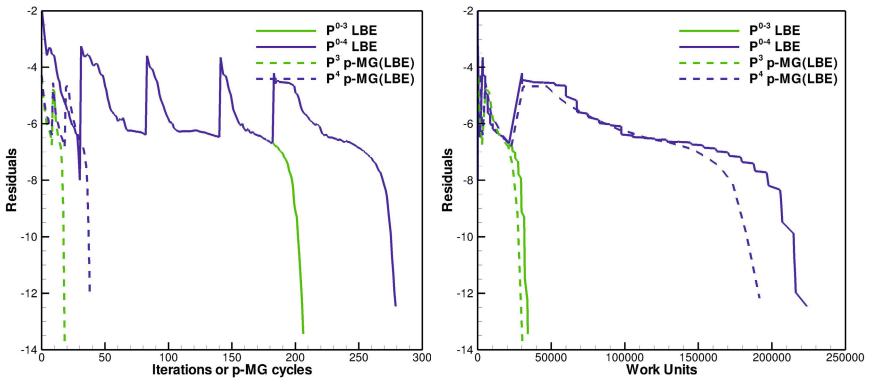


Fig. 61. LBE vs. p-MG(LBE), convergence history (L2 norm of the density residual) in terms of Newton steps or p-MG cycles (left) and non-dimensional CPU time (right), $p = 3$ and 4 solutions for the A6 testcase

2.9 Case A12: HART–II Rotorcraft

This application testcase was coordinated by USTUTT and consists of two subcases; hover and forward flight. The testcase was originally planned to demonstrate the implementation of aeroelastic higher order computations. However, the majority of the computations performed in the project were conducted for rigid blades, these purely aerodynamic computations are thus presented in this chapter. The rotational speed for both hover and forward flight is set to 1042 min^{-1}. The forward flight speed is set to 32.9 m/s.

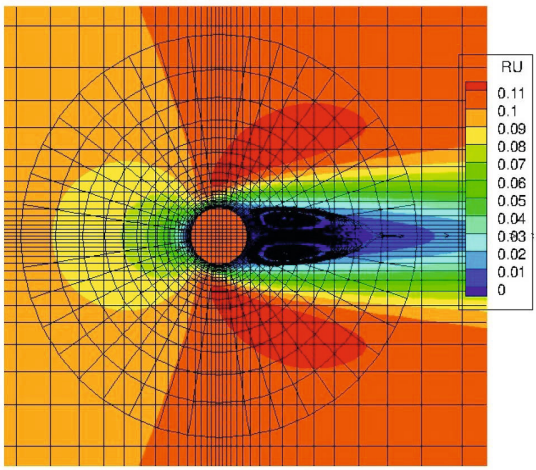


Fig. 62. Illustration of the USTUTT Chimera implementation used for the A12 hover case

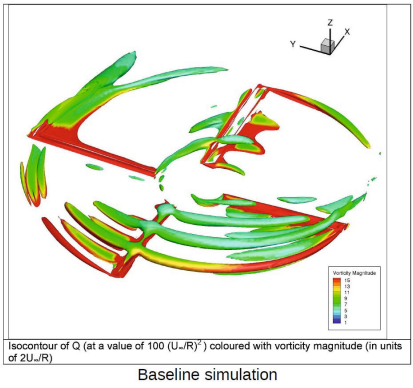


Fig. 63. Results from the NLR A12 forward flight baseline computation

Reference solutions were generated outside of the project using the DLR FLOWer solver for both hover and forward flight cases. The higher order computations however proved to be a challenge. USTUTT performed hover computations using a chimera approach, Figure 62. In Figures 63 and 64, the results of the NLR computations for second and fourth order can be compared, illustrating the reduced dissipation obtained by the higher order computation. In Figure 65, a comparison of the vertical force coefficient between baseline and higher order is shown.

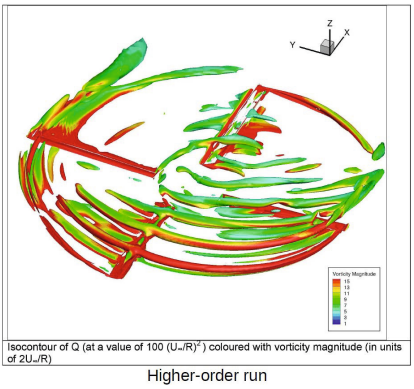


Fig. 64. Results from the NLR A12 forward flight fourth order computation

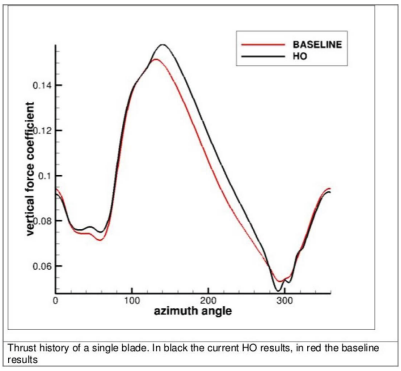


Fig. 65. Comparison between vertical force coefficients for the baseline and higher order computations for the NLR A12 simulations

3 Conclusions

A significant number of external–aerodynamic testcases have been computed in the project, some yielding insight into the improvement potential of higher order solvers for industrial applications. Also demonstrated are however the current limitations and problems associated with the higher order schemes. A major issue turned out to be the generation of unstructured higher order meshes, necessary for the efficient treatment of many problems of industrial complexity. This seriously hampered the planned activities concerning mesh cascade based efficiency analysis of the new approaches compared with the industrial reference. Nonetheless, by introducing further assumptions on the scheme behavior, some of the data gathered could be used for quantitative statements. In general, there are strong indications that higher order schemes have the potential to reduce the

discrete system size required to converge within industrial accuracy for many cases. This reduction seems to typically reach around one order of magnitude. It appears that such improvements are more likely for smooth flowfields devoid of shocks, but also transonic cases have shown reductions in the number of DOFs required for convergence. This reduction in problem size does however not translate to a reduced computational time for most cases, typically showing that the higher order solvers require at least one order of magnitude more computational resources per DOF to converge. This may indicate that the increased stiffness of higher order discrete systems is a major hindrance to the attractiveness of higher order approaches. A more optimistic interpretation is that the relatively low maturity level of the higher order solvers has not yet allowed for the development of efficient solvers. Which of these is true is not clear, the substantial amount of effort in the project invested into improving solver performance has however not resulted in algorithms competitive to the industrial reference on a discrete system size basis.

A frequently stated argument promoting DG methods is the locality of the discretization, allowing for a very efficient parallelization of the approach, also demonstrated in the project. The importance of this for external aerodynamic computations in industry is however not necessarily high. The bulk of aerodynamic work involves the computation of a large number of flow solutions for a given configuration, allowing for several runs to be performed in parallel and thus permitting the efficient usage of the hardware without requiring extreme levels of scalability. For single–point designs, more common in internal aerodynamics, the reduction of wall clock time required for a single computation may however be of importance. Future computer hardware developments may also increase the attractiveness of extreme parallelization for external aerodynamics.

Another selling point of the higher order approaches is the possibility of efficient implementation of adjoint adaptation, also demonstrated in IDIHOM. The efficiency improvement shown easily exceeds one order of magnitude in wall clock time compared with second order finite volume computations not applying adaptation. These approaches should however be compared with similar schemes applied on industrial–type methods to enable a clear statement of the advantages of the higher order methods in this context.

Regarding robustness of the codes, progress has been made, allowing for the routine computation of many complex flow cases within the pre–defined application areas of the codes. Some problems are however still remaining for issues such as shock capturing, and for the most complex cases attempted, convergence and stability problems were reported. The applicability of the higher order approaches have thus not yet reached the level of the industrial codes, there are however no indications that an industrial stability level of the codes is unobtainable.

References

1. Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., van der Ven, H., Sørensen, K. (eds.): ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications. NNFM, vol. 113. Springer, Heidelberg (2010)
2. Krietemeyer, M., Merz, M. (eds.): IPACS-Benchmark Integrated Performance Analysis of Computer Systems (IPACS). Logos Verlag, Berlin (2006), www.ipacs-benchmark.org
3. Chu, J., Luckring, J.M.: Experimental surface pressure data obtained on 65° delta wing across Reynolds number and Mach number ranges. NASA TM 4645 (1996)
4. Konrath, R., Klein, C., Schröder, A.: PSP and PIV investigations on the VFE-2 configuration in sub– and transonic flow. AIAA Paper2008-0379 (2008)
5. Furman, A., Breitsamter, C.: Experimental investigations on the VFE–2 configuration at TU Munich. Summary Report of RTO Task Group AVT–113, RTO (2009)
6. Gerhold, T., Friedrich, O., Evans, J., Galle, M.: Calculation of Complex Three–Dimensional Configurations Employing the DLR–Tau Code. AIAA 97-0167 (1997) Summary Report of RTO Task Group AVT–113, RTO (2009)
7. Boerstoel, J.W., Kassies, A., Kok, J.C., Spekreijse, S.P.: ENFLOW, a full functionality system of CFD codes for industrial Euler/Navier–Stokes computations, NLR CR 96286 L (1996)
8. Schmitt, V., Charpin, F.: Pressure Distributions on the ONERA–M6–WING at Transonic Mach numbers. AGARD AR 138 Appendix B-1
9. http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html
10. Sermeus, K., Deconinck, H.: Solution of steady Euler and Navier–Stokes equations using residual distribution schemes. 33rd Lecture Series on Computational Fluid Dynamics - Novel Methods for Solving Convection Dominated Systems. VKI LS2003-05 (2003)
11. Breitsamter, C.: Turbulente Strömungsstrukturen an Flugzeugkonfigurationen mit Vorderkantenwirbeln. PhD Dissertation, Technical University Munich, Herbert Utz Verlag, ISBN 3-897675-202-4 (1997)
12. http://hiliftpw.larc.nasa.gov/index-workshop1.html

Internal Aerodynamic Test Cases

K. Hillewaert[1], A. Garcia-Uceda[2], C. Hirsch[2],
J. Kok[3], M. de la Llave Plata[4],
and F. Renac[4]

[1] Cenaero, Rue des frères Wright 29, B6041 Gosselies, Belgium
koen.hillewaert@cenaero.be
[2] Numeca Int., Brussels, Belgium
{antonio.garcia-uceda,charles.hirsch}@numeca.be
[3] National Aerospace Laboratory NLR,
Flight Physics and Loads Department, Amsterdam, The Netherlands
jkok@nlr.nl
[4] Onera The French Aerospace Lab, 92320 Châtillon Cedex, France
{Marta.De_La_Llave_Plata,florent.renac}@onera.fr

Abstract. This section gathers the descriptions of all test cases concerning internal flows, as well as the most important results obtained by the participants in the IDIHOM project. The test cases comprise a transonic compressor (NASA Rotor 37), a subsonic nozzle (JEAN), low pressure turbine cascades in turbulent (T106A) and transitional conditions (T106C) and finally 2 validation cases, namely a bump flow from the DESIDER project, and the periodic flow over a 2D hill from the ERCOFTAC QNET CFD database. Depending on the test case, RANS, LES and even DNS computations were performed.

Keywords: Internal flows, RANS, LES, DNS, Compressor, Turbine, Nozzle.

1 NASA Rotor 37

The NASA Rotor 37 is a highly loaded transonic compressor. It was designed and studied experimentally at the NASA Lewis Research Center (now NASA Glenn) [1]. The rotor was tested in isolation with a circumferentially uniform inlet flow so as to provide steady flow in the rotor relative frame. This test case was used as a CFD test case at the 1994 ASME Gas Turbine Conference [2] and the AGARD Working Group 26 [4].

1.1 Conditions

The performance characteristic gives a measuring choking flow of 20.93 Kg/s, and calculations are aimed at a point corresponding to 98% of this flow rate.

The inlet conditions specify total pressure p_t and temperature T_t profiles, together with the absolute flow direction, which is assumed axial. The operating points are hence defined by the downstream static pressure p_2. The turbulent

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on
Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_27

quantities at the entrance are defined in order to have a 3% turbulence intensity and a turbulent to molecular viscosity ratio of the order of 10. Thus, the values of the turbulent kinetic energy and dissipation rate are $k = 44\ m^2/s^2$ and $\epsilon = 106\ m^2/s^3$, respectively. The walls are assumed adiabatic, with only a small part of the hub up- and downstream of the rotor rotating.

1.2 Grids and Computations

ARTS have performed simulations on three different grids, using three different schemes, namely the *Residual Based Compact* schemes of orders of convergence 2 and 3, as well as Jameson's scheme. The finest mesh has 1,480,704 cells; the medium and coarse meshes are obtained through successive agglomeration of neighbouring cells (eight by eight) of the finest one, resulting in 185,088 and 23,136 cells respectively. The coarsest grid is depicted in Figure 1.

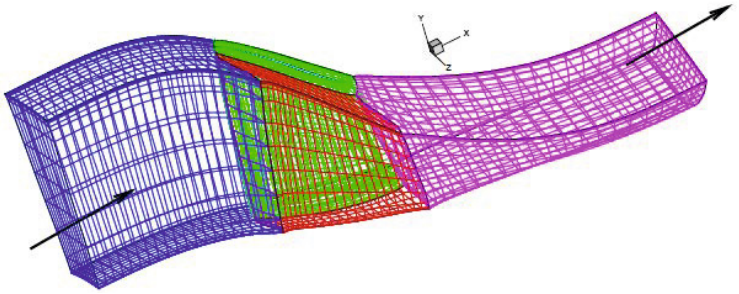
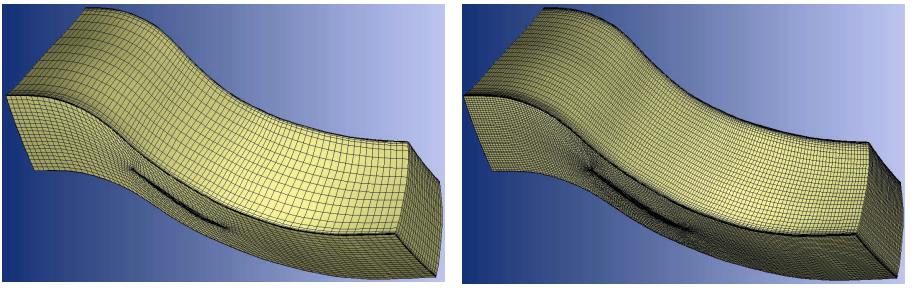


Fig. 1. RANS of the NASA Rotor 37: coarsest mesh used by ARTS, comprising 23,136 cells

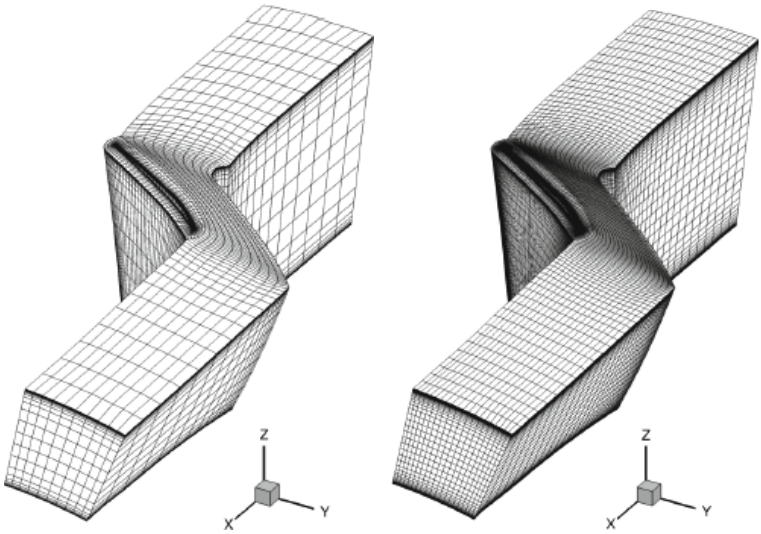
NUMECA has only been able to obtain converged solutions for this case with interpolation order p=1, due to the lack of a shock capturing tool in its DG code. Two different grids have been generated, consisting of 641,524 and 79,092 cells,



(a) Medium - 79,092 cells (b) Fine - 641,524 cells

Fig. 2. RANS of the NASA Rotor 37: grids used by NUMECA

where the second is obtained halving the number of points of the first mesh in each direction. These grids are shown in figure 2.
ONERA has performed four simulations on two different grids, namely a fine mesh consisting of 644,096 cells, and a coarse mesh with half the number of points in each direction, with a total of 80,512 cells. A modal DG scheme was used with interpolation orders 1 and 2, corresponding respectively to second and third order of accuracy. The grids are depicted in Figure 3.



(a) Coarse - 80,512 cells (b) Fine - 644,096 cells

Fig. 3. RANS of the NASA Rotor 37: grids used by ONERA

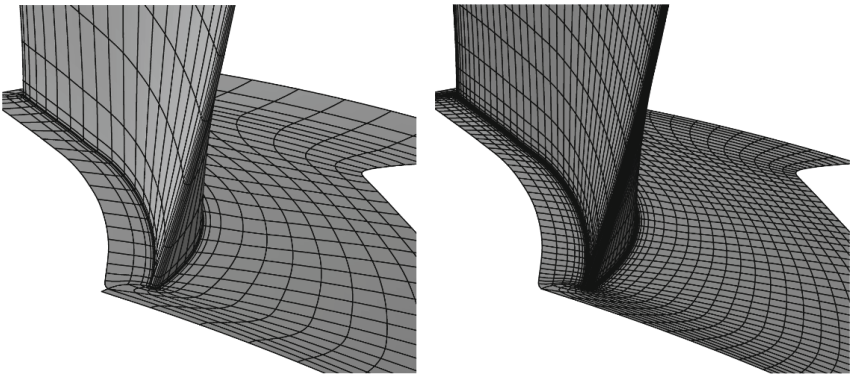
The university of Brescia (UNIBS) has performed five simulations on two different grids, a fine mesh consisting of 160,512 cells, and a coarse mesh with half the number of points in each direction, with a total of 20,064 cells. A DG scheme with interpolation orders 1 through 3 was used, corresponding to second through fourth order of accuracy. The grids are shown in Figure 4.

1.3 Reference Solution

The solutions have been compared to experimental data available from NASA. These correspond to performance curves and spanwise distributions at a given axial station of the rotor. These should be obtained at an operation point of 98% of the choking mass flow rate. Furthermore, the blade-to-blade Mach number distributions is compared at 30%, 50% and 70% of the blade span.

1.4 Assessment of High-Order Solutions

Details of the solvers used by the different participants are summarized in Table 1. Figures 5-8 compare the flow fields, in casu Mach number and static pressure



(a) Coarse - 20,064 cells (b) Fine - 160,512 cells

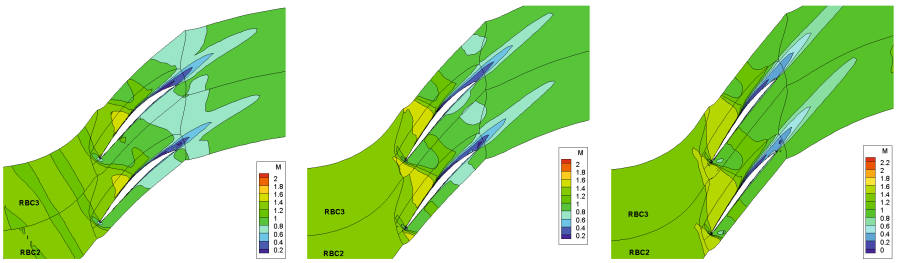
Fig. 4. RANS of the NASA Rotor 37: grids used by the university of Brescia

Table 1. RANS of the NASA Rotor 37: solver specifications

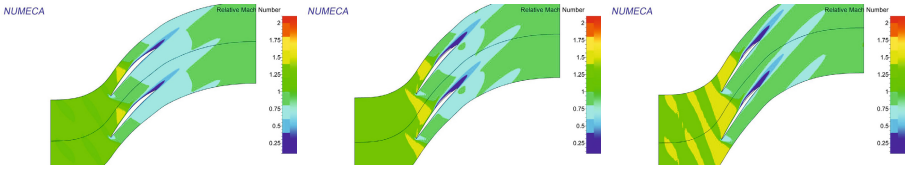
	Discretisation	Iterative	Turbulence model
ARTS	RBC3		Spalart-Allmaras
	RBC2 (baseline)		Spalart-Allmaras
NUMECA	DG(p=1), Roe, BR2	RK hpMG	EARSM
ONERA	DG(p=1,2), Lax-Friedrichs, BR2		Spalart-Allmaras
UNIBS	DG(p=1-3), Riemann solver, BR2	implicit	Wilcox $k-\omega$

at three different spanwise sections. From those figures, we can see that partners ARTS, ONERA, and UNIBS have provided very similar high-order solutions. In particular NUMECA suffered from severe instabilities due to the shocks present in the flow domain, which prevented from obtaining converged solution for interpolation order $p > 1$, as no shock capturing strategy has been implemented.

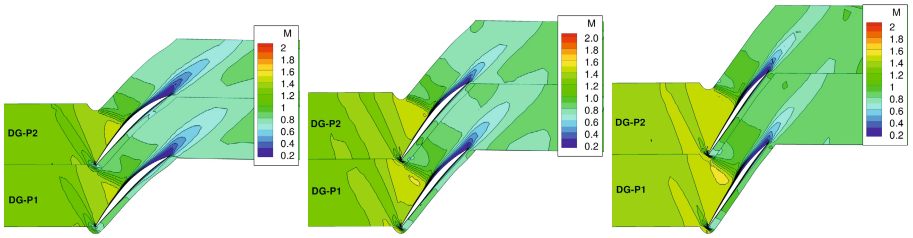
Looking at the iso-contour plots, we see that high-order solutions on the coarse mesh converge, as the order p is increased, to the solutions on the fine mesh. It should be noted that these plots have been obtained on the volume mesh (solution at the cell centres) which does not take into account the high-order character of the solution. A high-order representation of these iso-contours is expected to accentuate this trend. This demonstrates that high-order methods on coarse grids have the potential to provide a solution that is as accurate as the one obtained by widespread used second order finite volume methods on fine grids. Of course, a certain level of h-refinement will always be required in order to exploit the nominal high-order of the method, and not to pollute the solution with inaccuracies due to insufficient geometry representation.



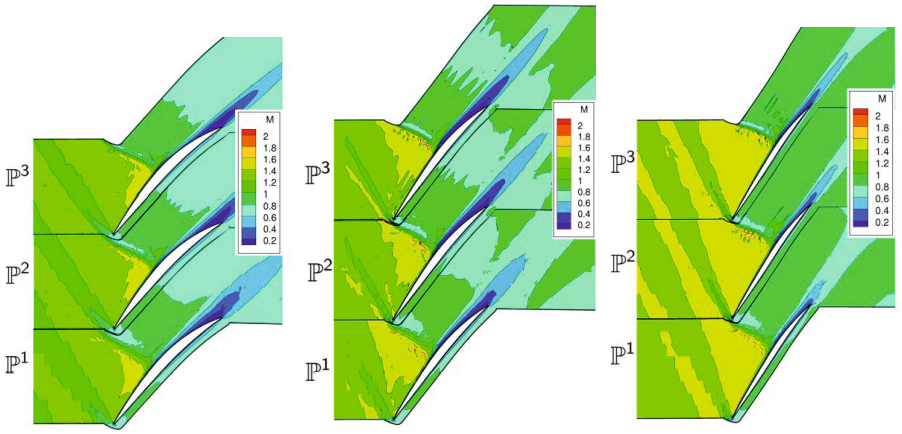
(a) ARTS, RBC2/3 2nd and 3rd order accuracy



(b) NUMECA, DG(p=1), 2nd order accurate

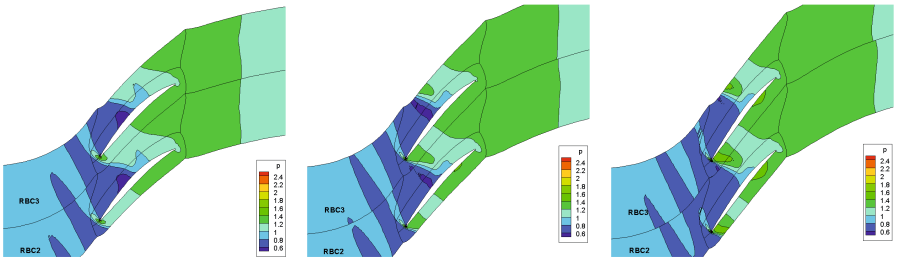


(c) ONERA, DG(p=1-2), 2nd to 3rd order accurate

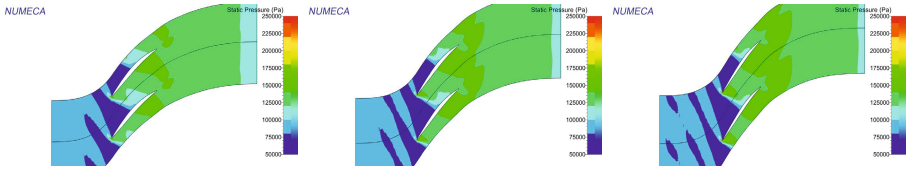


(d) UNIBS, DG(p=1-3), 2nd to 4th order accurate

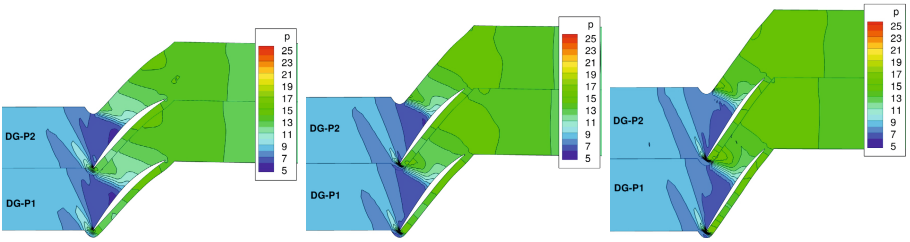
Fig. 5. RANS of the NASA Rotor 37: Mach number contours obtained on the coarse mesh



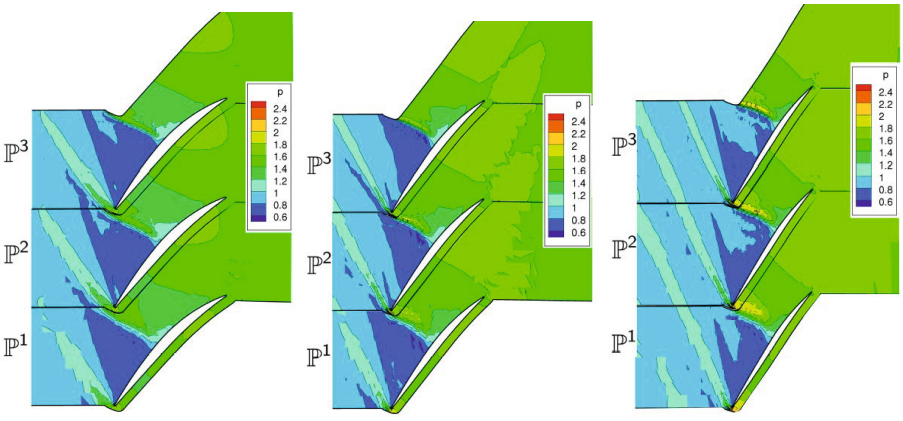
(a) ARTS, RBC2/3 2nd and 3rd order accuracy



(b) NUMECA, DG(p=1), 2nd order accurate

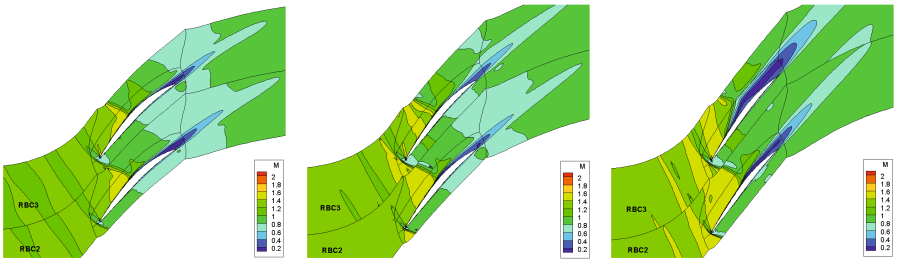


(c) ONERA, DG(p=1-2), 2nd to 3rd order accurate

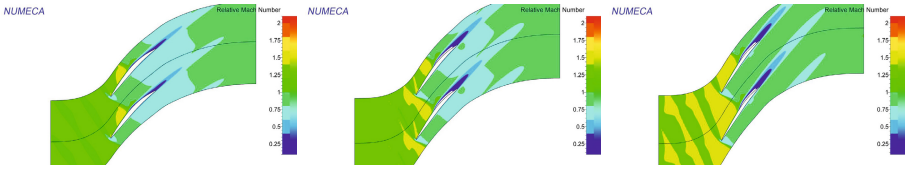


(d) UNIBS, DG(p=1-3), 2nd to 4th order accurate

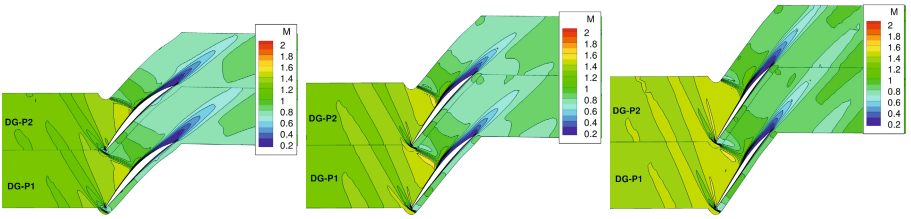
Fig. 6. RANS of the NASA Rotor 37: pressure contours obtained on the coarse mesh



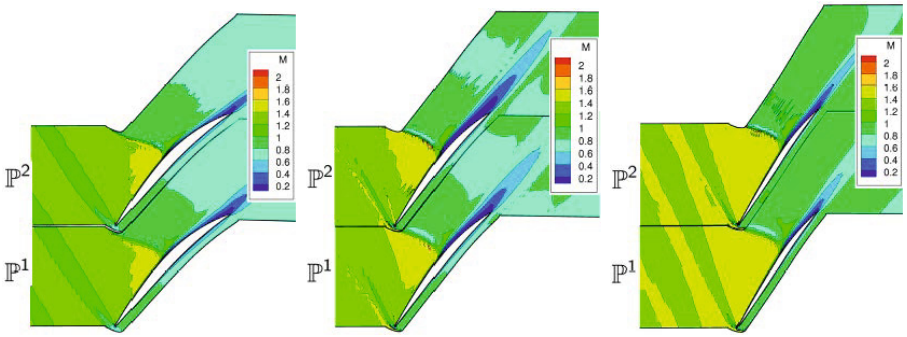
(a) ARTS, RBC2/3 2nd and 3rd order accuracy



(b) NUMECA, DG(p=1), 2nd order accurate

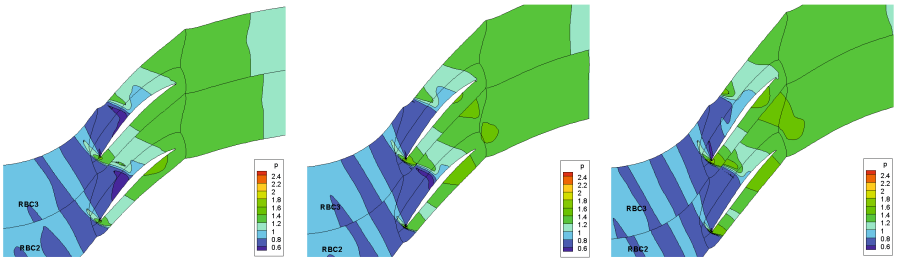


(c) ONERA, DG(p=1-2), 2nd to 3rd order accurate

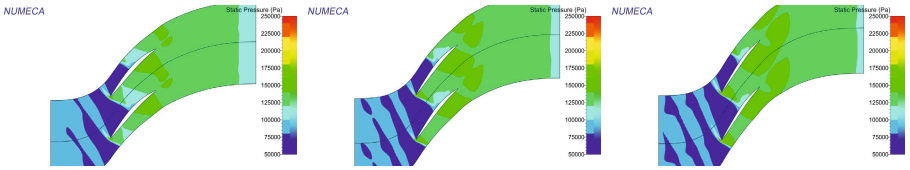


(d) UNIBS, DG(p=1-3), 2nd to 4th order accurate

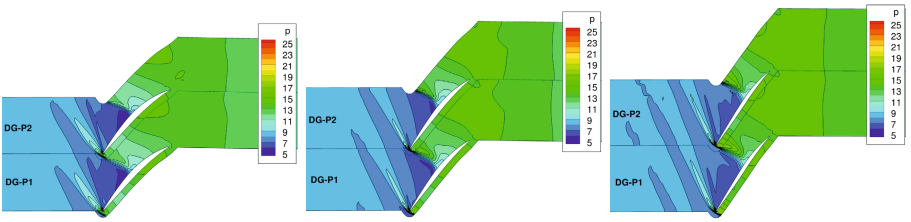
Fig. 7. RANS of the NASA Rotor 37: Mach number contours obtained on the fine mesh



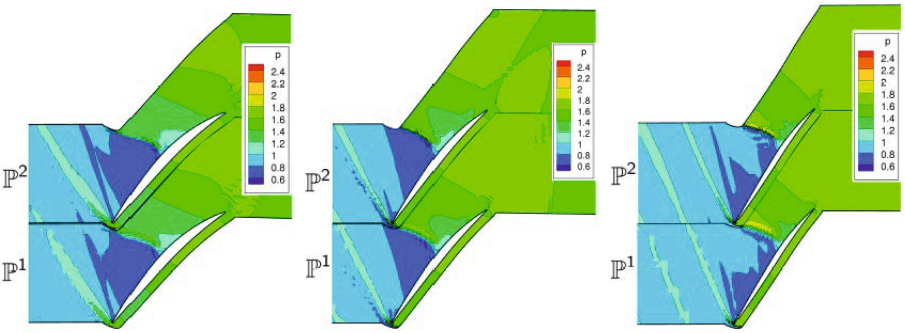
(a) ARTS, RBC2/3 2nd and 3rd order accuracy



(b) NUMECA, DG(p=1), 2nd order accurate

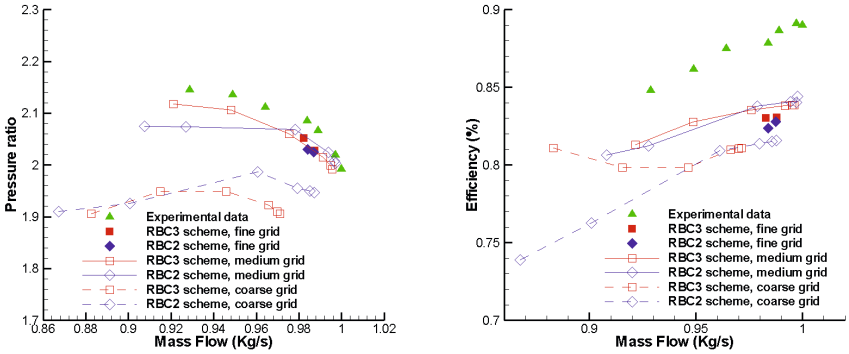


(c) ONERA, DG(p=1-2), 2nd to 3rd order accurate

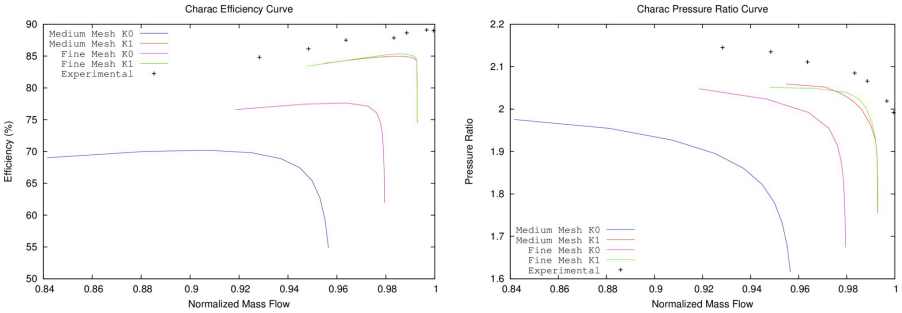


(d) UNIBS, DG(p=1-3), 2nd to 4th order accurate

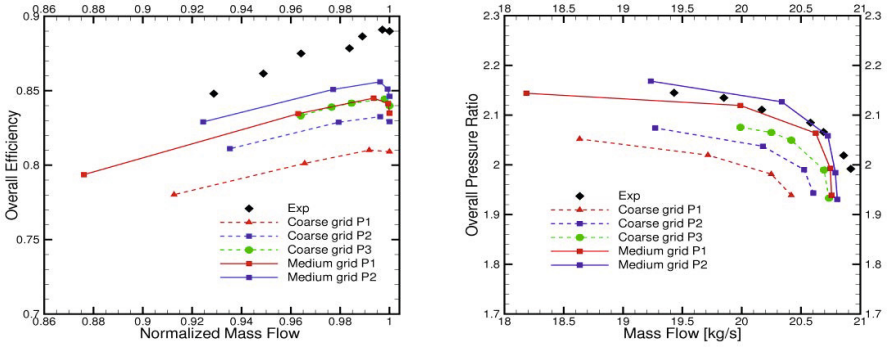
Fig. 8. RANS of the NASA Rotor 37: pressure contours obtained on the fine mesh



(a) ARTS

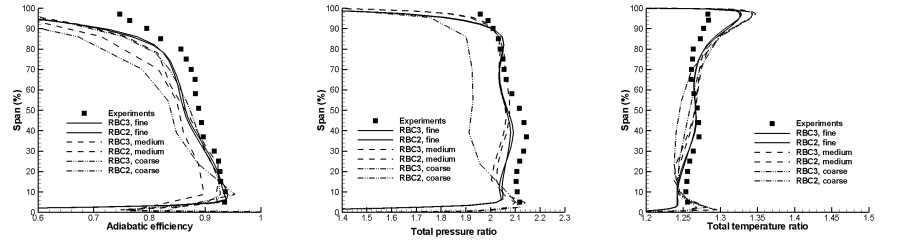


(b) NUMECA

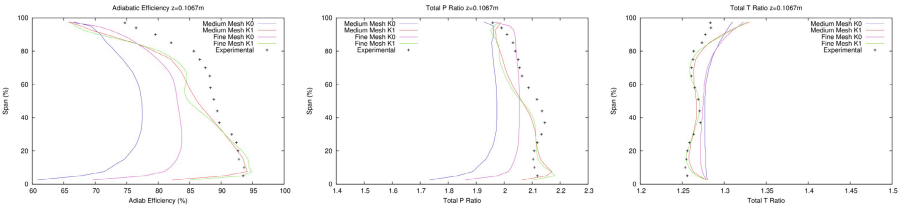


(c) UNIBS

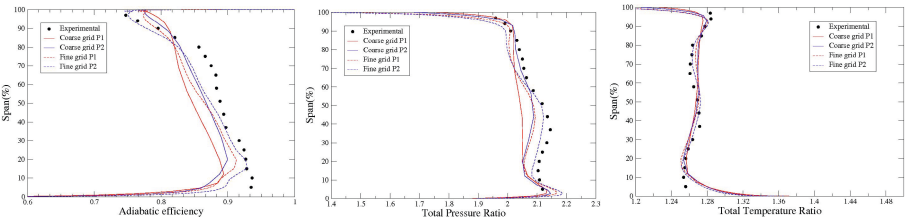
Fig. 9. RANS of the NASA Rotor 37: global isentropic efficiency (left) and total pressure (right) characteristics



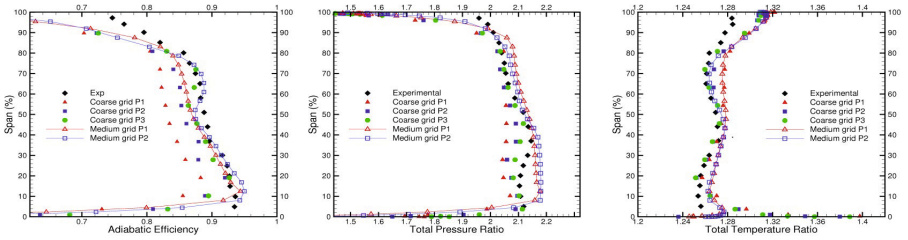
(a) ARTS



(b) NUMECA



(c) ONERA



(d) UNIBS

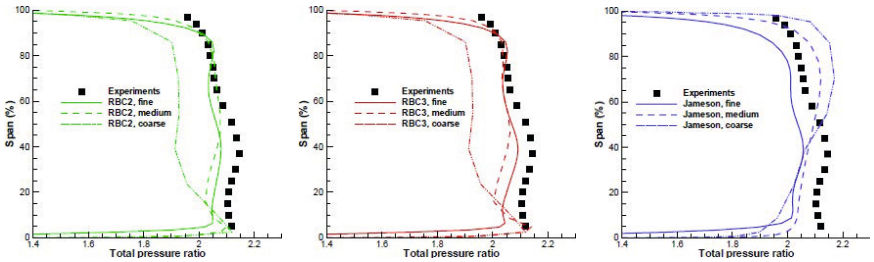
Fig. 10. RANS of the NASA Rotor 37: radial variation of isentropic efficiency (left), total pressure (middle) and total temperature ratio (right)

The same conclusions can be deduced from the figures 9 which show the computed performance characteristics, and Figures 10 which show the radial profiles of isentropic efficiency, total pressure and total temperature. We can see from ARTS and UNIBS results how the high-order solutions on the coarse grids

seem to converge to the same profiles, as the interpolation order is increased. From ONERA's results we can see that the profiles provided by the 3rd-order simulation (DG-p2) on the coarse grid are very similar to those obtained from the 2nd-order simulation (DG-p1) on the fine grid. The gain in terms of degrees of freedom is over 50% when considering the DG-p2 simulation on the coarse with respect to the fine grid simulation DG-p1.

The differences with the experimental data can be attributed to the flow model/insufficient resolution. For instance, it is known that the flow solution is very sensitive to the correct representation of the gap between the blade and the shroud (see Shabhir *et al.* [3]).

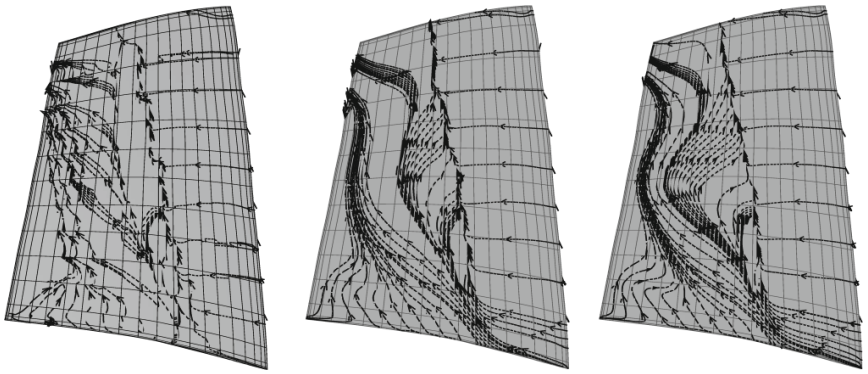
ARTS has performed a comparison of the grid convergence of its high-order RBC scheme, to that of a reference second order finite volume solver based on the Jameson scheme. These are shown in Figure 11. The computational times for ARTS are indicated in Table 2, whereas those for the DG method of UNIBS are given in Table 3. The main conclusion from ARTS is that, although RBC schemes are more expensive per degree of freedom than Jameson's scheme, they are definitely more accurate, since they provide almost grid converged solutions using about 1/8 of the DOF required by Jameson's scheme. Also RBC3 is slightly more expensive than RBC2, but this extra cost is compensated by a somewhat better accuracy.



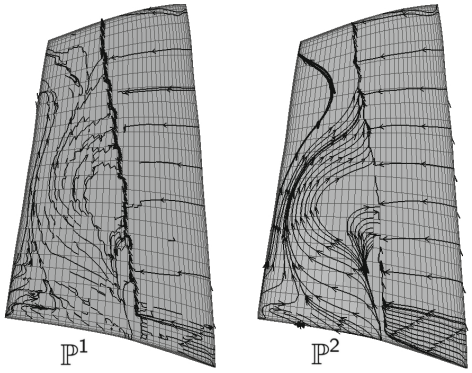
(a) RBC2 (b) RBC3 (c) Jameson

Fig. 11. RANS of the NASA Rotor 37: grid convergence of the radial profile of total pressure ratio for different schemes. (ARTS)

Finally, looking at the skin friction lines computed by UNIBS in Figure 12, we see how a third and fourth order solution on a coarse grid provides a very similar solution as a second order solution on a fine mesh, which has approximately 8 times as many cells.



(a) Coarse mesh, p=1,2,3



(b) Fine mesh, p=1,2

Fig. 12. RANS of the NASA Rotor 37: UNIBS. Skin friction lines on the suction side.

Table 2. RANS of the NASA Rotor 37: CPU times for ARTS

	Fine grid		Medium grid		Coarse grid	
Scheme	Total CPU	CPU/dof	Total CPU	CPU/dof	Total CPU	CPU/dof
RBC3	4531	0.00306	566	0.00306	70.8	0.00306
RBC2	3720	0.00251	465	0.00251	58.1	0.00251
Jameson	2054	0.00139	257	0.00139	32.1	0.00139

1.5 Conclusions

ARTS, ONERA and UNIBS have successfully obtained high-order solutions for this complex case, with a good agreement with the experimental data. NUMECA had serious stability issues for this case, due to shocks. This demonstrates the need of developing shock treatment tools for DG methods.

Mesh convergence is observed for high-order solutions. For UNIBS we see that the fourth order solution on the coarse mesh gives very similar solution as a

Table 3. RANS of the NASA Rotor 37: CPU/DOF in work units (taub bench) for UNIBS

Coarse grid			Fine grid	
p=1	p=2	p=3	p=1	p=2
0.119	0.423	11.15	0.689	1.868

second order solution on the fine mesh (which has 8 times more cells). A certain mesh resolution is always needed, supporting the need of hp-refinement.

ONERA's results equally show that by increasing the polynomial order it is possible to have important savings in terms of number of degrees of freedom with respect to a low-order simulation for a comparable level of accuracy. Indeed, the results obtained from the 3rd-order computation on the coarse grid are very close to those obtained from the 2nd-order computation on the fine grid.

The comparison of ARTS high-order solution with the reference Jameson's scheme demonstrates that low dissipative scheme developed in IDIHOM do represent a significant improvement over the state of the art.

2 JEAN Nozzle

This test case concerns the flow of a turbulent isothermal and subsonic jet exhausting from a nozzle, as shown in Figure 13. The test case has been first defined in the European project JEAN (Jet Exhaust Aerodynamics and Noise) for prediction of noise by jet flows [5, 6]. The jet is isothermal with a static temperature in the nozzle exit $T_j = T_0$ where T_0 is the ambient temperature. The Mach number and Reynolds number, based on isentropic conditions, in particular velocity u_j, at the nozzle exit and the diameter D are respectively $M = u_j/\sqrt{\gamma RT_j} = 0.75$ and $Re = \rho_j u_j D/\mu(T_j) = 5 \times 10^4$. In these conditions, the flow is subsonic and fully turbulent. The total values imposed at the inlet, the ambient states imposed at the freestream boundaries and the other physical parameters of the simulation are presented in Tab. 4.

2.1 Geometry

Figure 14 presents the geometry of the test case. Contraction and nozzle geometries are given by analytical (cubic) laws. Each part of the convergent is given by two cubic functions

$$f_1(x) = a_1X^3 + b_1X^2 + c_1X + d_1, \quad \text{for } X \leq X_1,$$
$$f_2(x) = a_2X^3 + b_2X^2 + c_2X + d_2, \quad \text{for } X > X_1,$$

where $X = x - x_0$ and x_0 is the longitudinal axial position of the left side of either the convergent or the nozzle; X_1 represents the abscissa of the common inflexion point for both functions f_1 and f_2 (see Fig. 14). Table 5 gives the values of the numerical parameters.

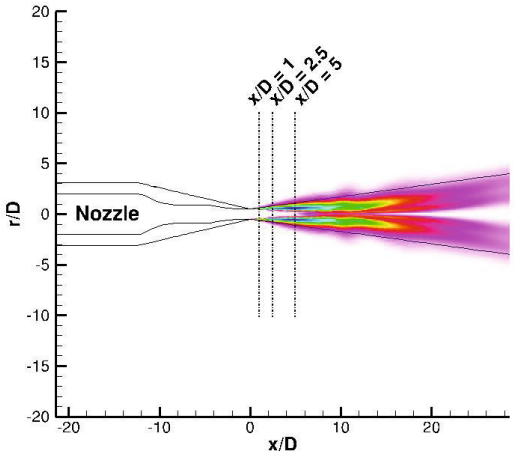


Fig. 13. JEAN nozzle: geometry, potential core and mixing layer of the jet and axial location of the measurement planes

Table 4. JEAN nozzle: physical parameters of the simulation

Ambient pressure	p_0	100380.0 Pa
Ambient temperature	T_0	283.15 K
Nozzle inlet total pressure	p_i^t	144400 Pa
Nozzle inlet total temperature	T_i^t	314.15 K
Dynamic viscosity at the jet	$\mu(T_j)$	3.12757×10^{-4} kg/m^2s
Nozzle inlet diameter	D_i	0.2 m
Nozzle outlet diameter	D_j	0.05 m

2.2 Reference Results

Experimental data have been obtained by Jordan *et al.* [7] during the JEAN project. Both axial and radial profiles at at three axial positions: *i.e.* at $x/D = 1$, $x/D = 2.5$, and $x/D = 5$ of the average axial velocity and its auto-correlation are provided. The data have been obtained at a much higher Reynolds number of $Re_{D,is} = 9 \times 10^5$. It is thereby assumed that the Reynolds number dependence at $Re_{D,is} = 5 \times 10^4$ sufficiently weak.

2.3 Meshes

Two meshes were provided by UCL. One was used by Onera for RANS computations, the other was used by Cenaero for LES computations. The first mesh

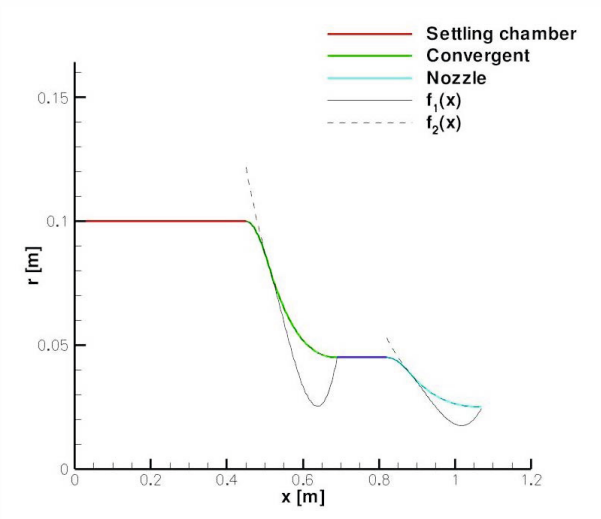


Fig. 14. JEAN nozzle: geometry of the nozzle walls (coloured lines) and analytical laws for the definition of convergent geometry (black lines)

Table 5. JEAN nozzle: parameters of the functions f_1 and f_2 describing the analytical geometry

	convergent	nozzle
X_0	0.45	0.82
X_1	0.06	0.0625
a_1	22.2800925925926	7.168
b_1	-6.30208333333334	-2.12
c_1	0	0
d_1	0.1	0.045
a_2	-3.18287037037035	-1.024
b_2	2.86458333333332	0.96
c_2	-0.825	-0.288
d_2	0.122	0.053

consists of 179,180 quadratic prisms and hexahedra. The second one contains 572,096 quadratic elements (also prisms and hexahedra) and was refined from RANS results obtained on the first mesh. The meshes are detailed in the sections specific to both computations.

2.4 RANS Results

RANS results were obtained with the Aghora code from Onera, based on the discontinuous Galerkin discretisation (DGM), with the one equation turbulence model from Spalart-Allmaras [8]. Two orders of accuracy have been used, namely p=1 (2nd order accurate) and p=2 (3rd order accurate).

Meshes. The RANS mesh consists of 179, 180 quadratic prisms and hexahedra, and is illustrated in Fig. 15, together with the distributions of non-dimensional first cell height y^+, verifying $y^+ < 3$ at the inner wall and $y^+ < 6$ at the outer wall.

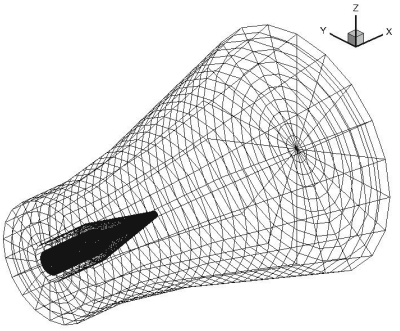
Boundary Conditions. Total pressure, total temperature and velocity direction are imposed at the inlet of the settling chamber. No-slip and adiabatic conditions are applied at the walls, and non-reflecting boundary conditions are applied at the free artificial boundaries.

Mean Flow Field. Figure 16 presents the Mach number and turbulent variable contours in a vertical plane crossing the jet axis obtained at steady-state. We clearly see the structure of the potential core of the jet and the development of turbulence in the mixing layer downstream of the nozzle lips. The turbulence intensity is stronger with the third-order discretization indicating a lower diffusion of the turbulence transport in the flow. Figure 17 compares the distributions of the mean values of the axial velocity component with experimental data obtained by LEA Poitiers during the JEAN project [7]. We observe a good agreement between numerical and experimental results in the potential core and velocity around the jet axis.

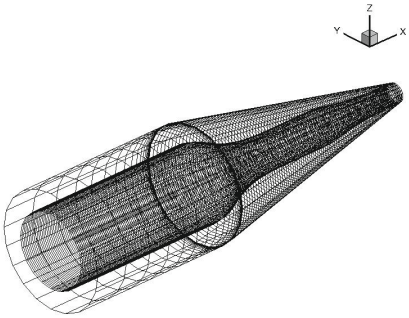
Nevertheless, the potential core length is underpredicted and discrepencies occur for the radial distribution away from the axis. We attribute most of these differences to the very low mesh density in the mixing layer region (see Fig. 15) which prevents the correct capture of mass transfer associated to the global conservation of momentum in the jet flow. Some of these differences could also be attributed to a small remaining dependence on the Reynolds number, which is about 20 times smaller in the computations. However, results should be improved using the second, more refined mesh.

2.5 LES Results

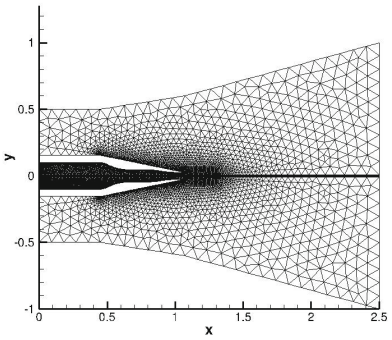
Mesh Definition. The computation is performed on a mesh composed of 582, 464 hexahedra and 23, 808 prisms using the compressible DGM solver Argo. With a fourth order solution ($p = 3$), this leads to 38M of *dof* per equation. The resolution of the mesh is considered fine enough to perform a wall-resolved ILES with a $y^+ < 4$ everywhere (studies have shown that accurate results can be obtained with DGM for $y^+ < 8$). As the mesh is constructed by the rotational extrusion of a two-dimensional mesh composed of quadrangles, all the prisms are located on the jet axis. To avoid reflections, the boundaries of the domain



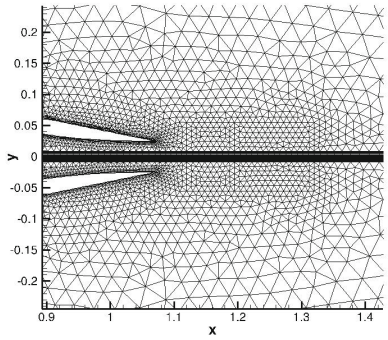
(a) Farfield mesh



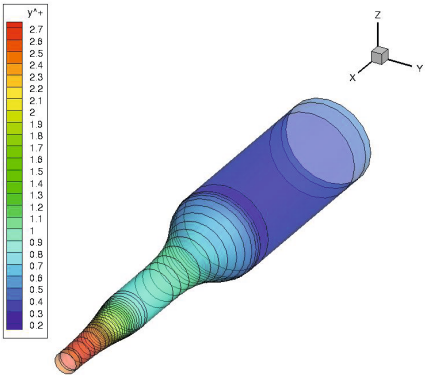
(b) Nozzle mesh



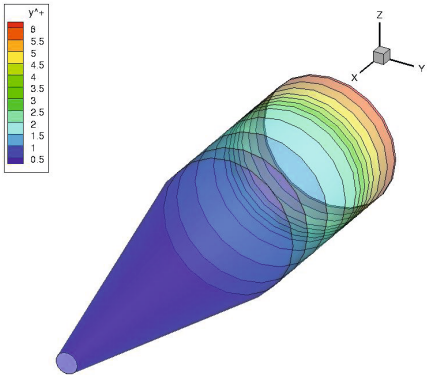
(c) Extrusion plane - global



(d) Extrusion plane - zoom



(e) Internal wall resolution



(f) External wall resolution

Fig. 15. JEAN nozzle, RANS computations: views of the mesh

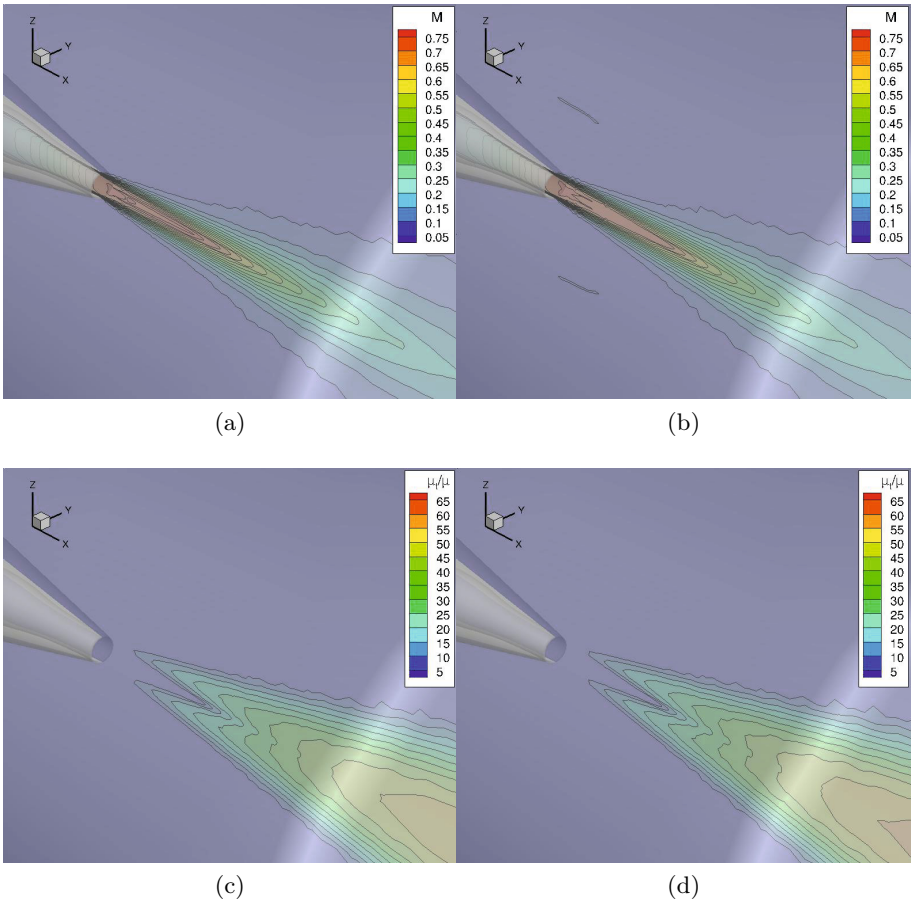
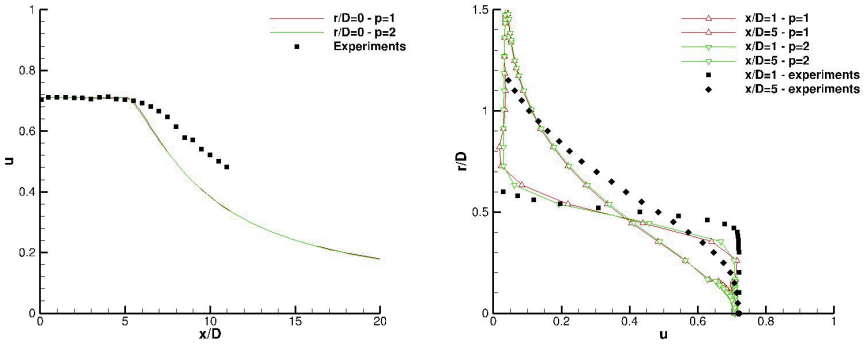


Fig. 16. JEAN nozzle, RANS computations: Mach number (top) and ratio of turbulent to dynamic viscosity coefficients μ_t/μ (bottom) contours for second- (left) and third-order (right) approximations

are located far away from the nozzle, *i.e.* at $x/D = 300$ and $r/D = 100$. As a very coarse mesh is used in the far field, this large domain does not increase significantly the size of the problem. Figure 18 shows a section of the extruded mesh close to the jet exit. The boundary layer mesh is prolonged into the domain at the exit of the nozzle to capture the shear layers and the Kelvin-Helmholtz instabilities downstream. Two refinement boxes are placed in the wake of the jet, allowing to capture the complex turbulent interactions up to $x/D = 14$ downstream from the nozzle exit (up to $x/D = 5$ for the finest refinement box).

Results. Instantaneous flow fields computed with the fourth order accurate version of Argo DGM are shown in Fig. 19. The flow is fully turbulent downstream



(a) (b)

Fig. 17. JEAN nozzle, RANS computations: mean axial velocity component along the jet axis (a) and at axial stations $x/D = 1$ and $x/D = 5$ for second- and third-order approximations

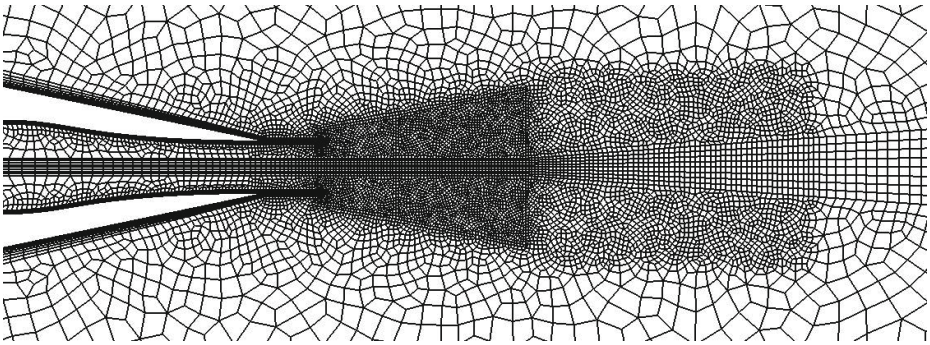


Fig. 18. JEAN nozzle, LES computations: section of the mesh, and zoom on the nozzle exit

of the nozzle. Small Kelvin-Helmholtz instabilities can clearly be observed at the nozzle exit. Those Kelvin-Helmholtz instabilities rapidly transition to turbulence, creating very small vortical structures downstream.

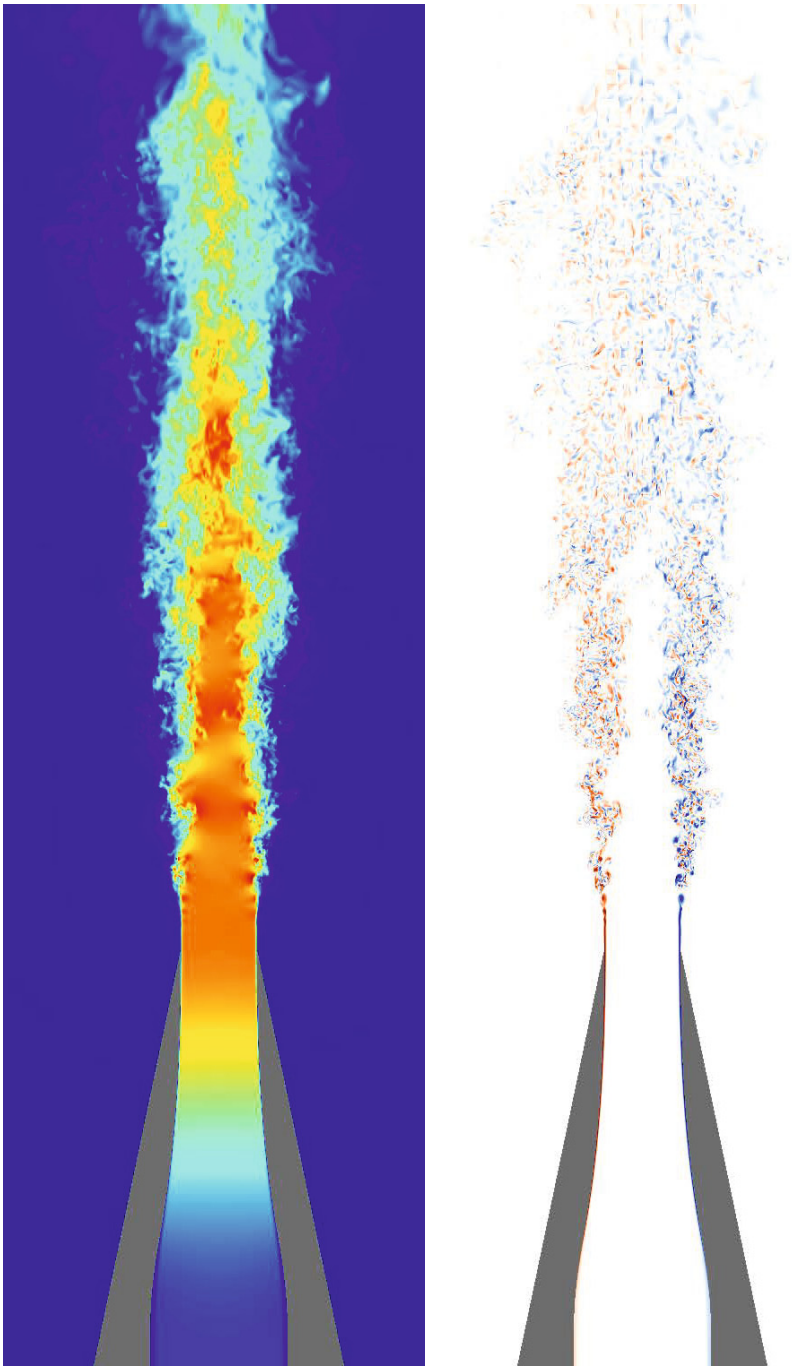


Fig. 19. JEAN nozzle, LES computations: snapshot of the Mach number (top) and the z-component of the vorticity (bottom) fields from the DGM/ILES computation

Experimental data are available along the axis and at different axial positions as well as the LES results of Andersson *et al.* [9]. All the following results are scaled with the obtained velocity at the centre of the nozzle exit. Figure 20 shows the mean axial velocity $u_x(x)$ and the mean axial fluctuations $u'_{rms}(x)$ along the nozzle axis. The overall behaviour of the computation is close to that of the

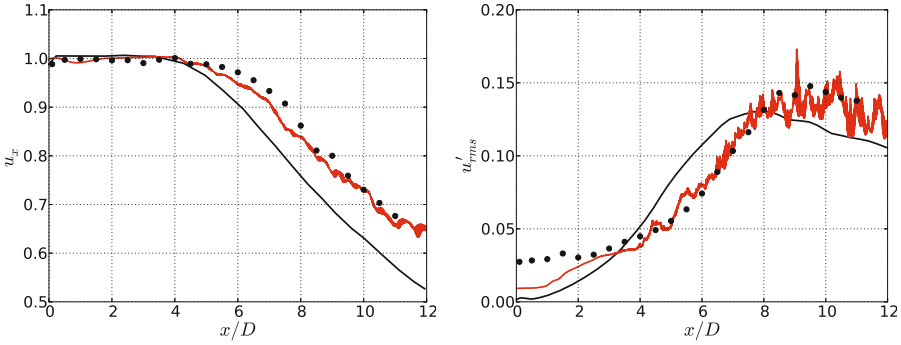


Fig. 20. JEAN nozzle, LES computations: mean axial velocity along the nozzle axis. Experimental results (black circle), LES of Andersson *et al.* [9] (black line) and the DGM/ILES computation (red line).

experiment with the end of the potential core located around $x/D = 4$. The drop of velocity in the wake is also globally well represented. On the fluctuations, the effect of the absence of inlet turbulence can clearly be seen, resulting in an under-prediction of the fluctuations at the nozzle exit. Nevertheless, the fluctuations matches well those of the experiment downstream the potential core, *i.e.* after $x/D = 4$. The results improve on earlier LES by Andersson *et al.* [9].

Figure 21 shows the mean axial velocity and the mean axial RMS fluctuations at the axial positions $x/D = 1$, $x/D = 2.5$, and $x/D = 5$. At each station, the solution is also averaged in rotation to accelerate the convergence. It can be seen that the diffusion of the jet is slightly more important in the computation, as the computation is performed at a relatively low Reynolds number compared to the experiment. For the fluctuations, a wider curve is obtained together with a more pronounced peak at the first station. Nevertheless, a fair agreement is obtained between the computation and the experiment. The widening of the shear layer is furthermore in good qualitative agreement with the results obtained by Anderson *et al.* [9].

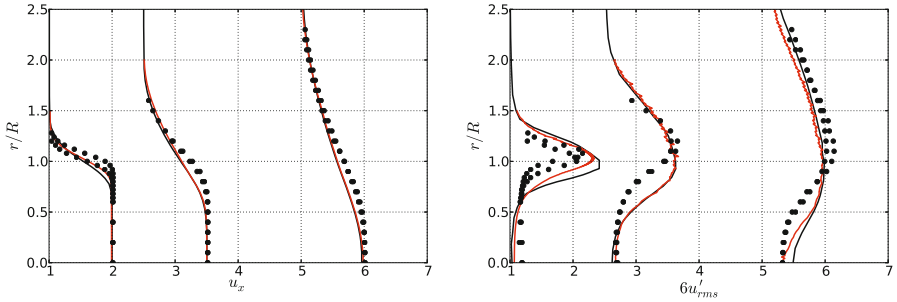


Fig. 21. JEAN nozzle, LES computations: Mean axial velocity and mean axial RMS fluctuations at the axial stations $x/D = 1$, $x/D = 2.5$, and $x/D = 5$. Experimental results (black circle), LES of Andersson *et al.* [9] (black line) and DGM/ILES (red line). Curves are shifted and scaled for clarity.

Further downstream, some discrepancies can be observed, together with few oscillations due to the solution jumps at the interface. It also shows that more time would be required to fully converge the statistics. Here again, the effects of the inlet turbulence can clearly be seen at the first station where the fluctuations in the core are three times more important in the experiment.

2.6 Conclusion

The present results constitute preliminary results on the compressible flow of a Mach 0.75 jet at Reynolds number 5×10^4. The results demonstrate the potential interest in using high-order DG methods for both RANS and LES modelling of this kind of test-case exhibiting complex flow features such as vortical structures, turbulent patches, etc. Nevertheless, the trend highlighted in this study need to be confirmed by an in-depth analysis with hp-refinement in order to achieve or ascertain grid convergence on results. This would allow to quantify the impact of the RANS turbulence model and the remaining dependence on the Reynolds number for the LES.

3 Turbulent Flow in the T106A Cascade

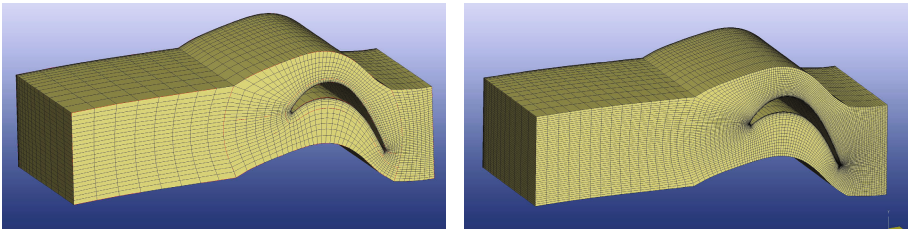
The MTU Cascade T106A is representative of low pressure turbine blades. The flow conditions are chosen such that fully turbulent flow is obtained, such that this test case is used as a benchmark for the implementation and convergence of RANS model with high order methods. This test case has been computed by Numeca and the university of Brescia.

3.1 Conditions

The inlet conditions are prescribed as constant, with stagnation pressure $P_1^t = 126562\ Pa$ and temperature $T_1^t = 306.2\ K$ and inlet flow angle $\alpha_1 = 127.7°$. The imposed outlet static pressure is $p_2 = 100000\ Pa$. These conditions correspond to an isentropic exit Mach number $M_{2,is} = 0.59$, and a corresponding Reynolds number of $Re_{2,is} = 500000$ approximately. The flow is assumed fully turbulent.

3.2 Grids and Discretisation

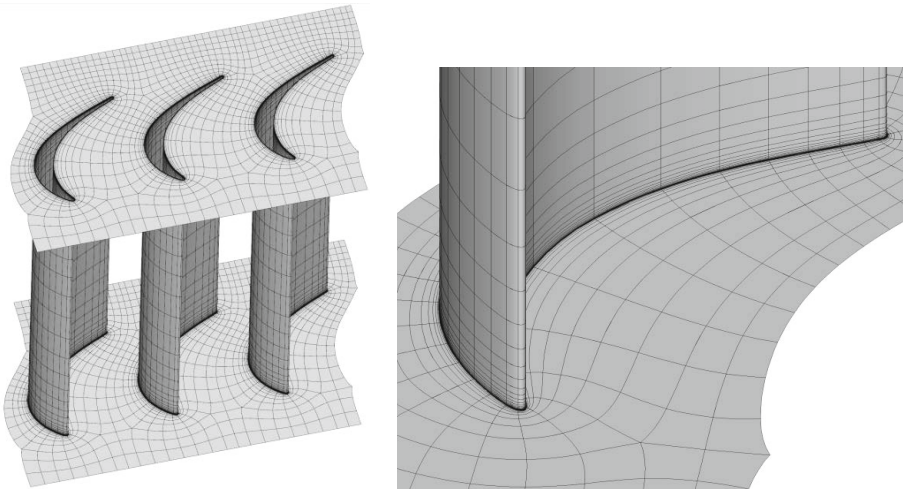
NUMECA. DGM simulations of 2nd to 4th order of accuracy have been performed on two different grids, generated by NUMECA, and curved accordingly by means of the in-house tool implemented in its DG solver. The finest mesh has 223745 cells. The coarse one is obtained by agglomeration of four-by-four neighbouring cells of the fine mesh in the blade to blade plane whereas the resolution in the spanwise direction is kept, resulting in 55170 cells. Both meshes are shown in figure 22.



(a) Coarse - 55170 cells (b) Fine - 223745 cells

Fig. 22. RANS of the turbulent flow in the T106A cascade: grids used by Numeca

University of Brescia. Five DGM simulations, with 3rd and 4th order of accuracy, have been performed on two different grids, respectively of 77300 and 43000 cells, respectively. As example the coarsest grid is shown in Figure 23.



(a) Global view (b) Close up near the blade root

Fig. 23. RANS of the turbulent flow in the T106A cascade: coarse grid used by the University of Brescia

3.3 Reference Results

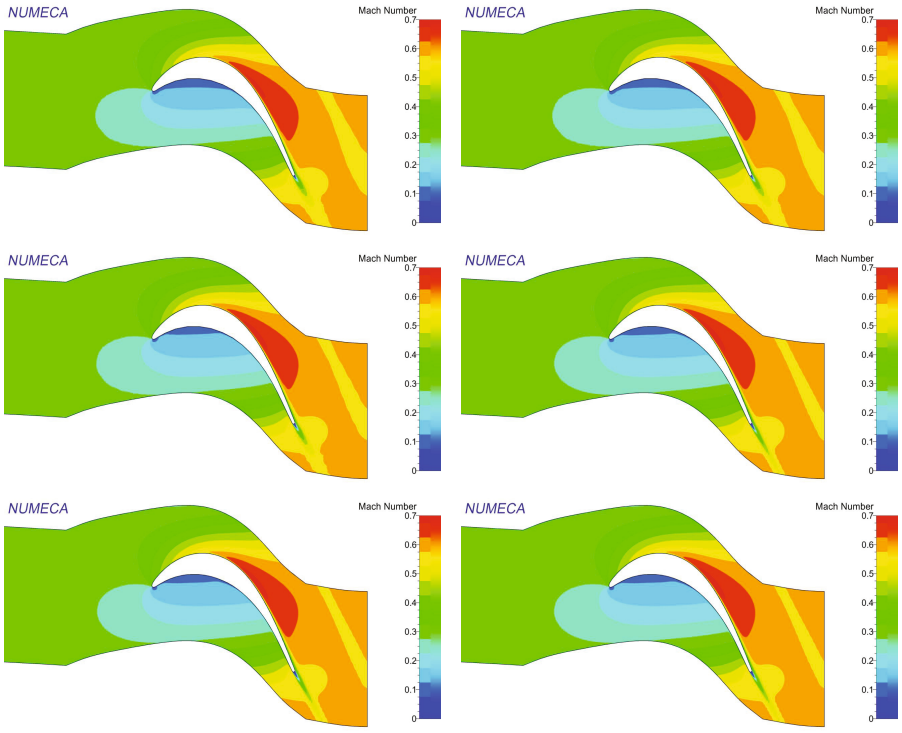
The high-order solutions obtained by each partner have been validated by experimental data available from MTU, corresponding to the pressure coefficient distribution on the surface of the blade.

3.4 Assessment of the High Order Methods

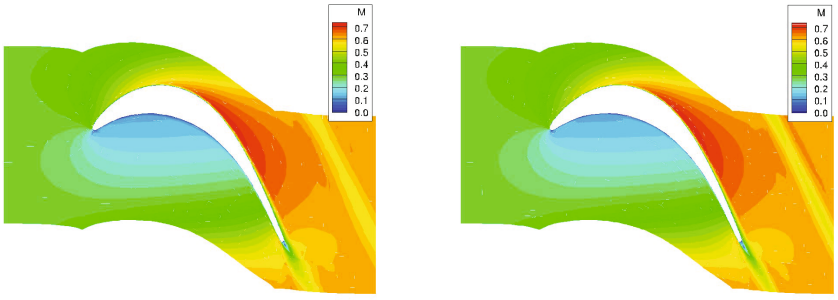
The solver specifications are given in table 6.

Table 6. RANS of the turbulent flow in the T106A cascade: high order solver specifications

Partner	Discretisation	Iterative Method	Model
NUMECA	DG, Roe/BR2	explicit hp-MG	EARSM
University of Brescia	DG, exact Riemann solver/BR2	implicit	EARSM

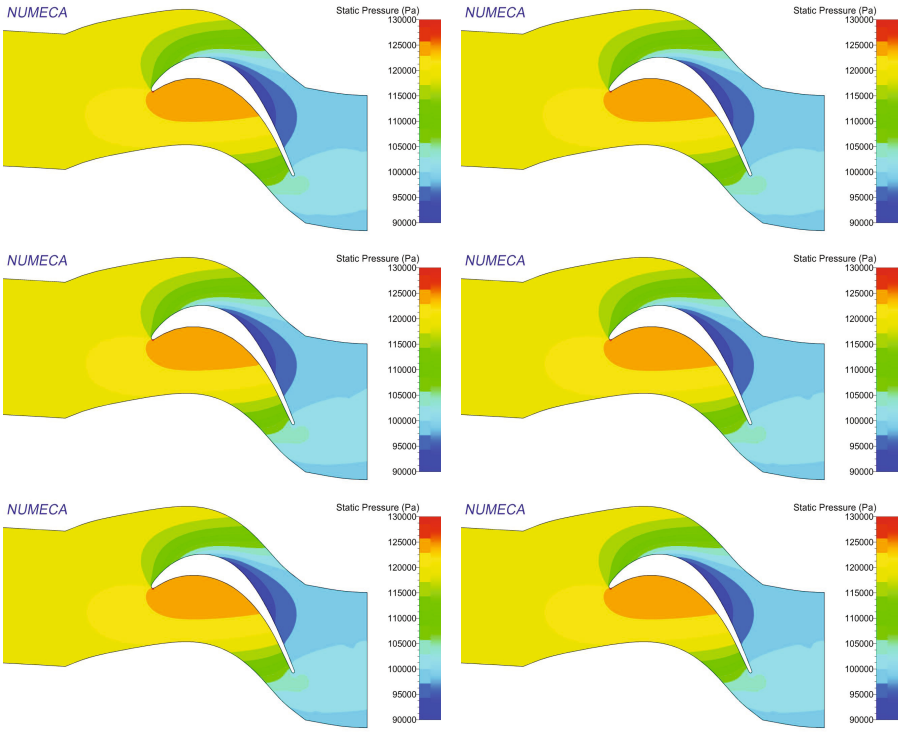


(a) Numeca, coarse (left) and fine mesh (right), orders p=1 to 3 (top to bottom)

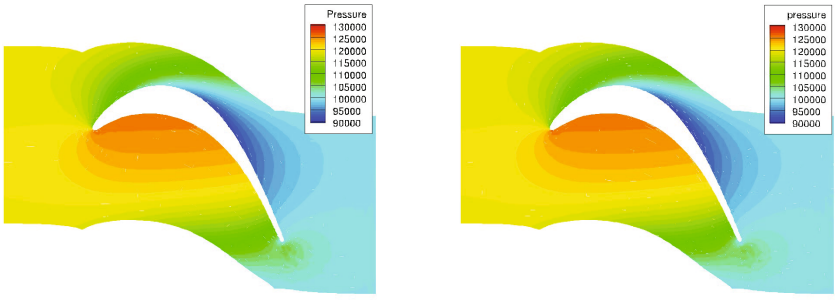


(b) University of Brescia, p=2 (left) and p=3 (right)

Fig. 24. RANS of the turbulent flow in the T106A cascade: computed Mach number contours



(a) Numeca, coarse (left) and fine mesh (right), orders p=1 to 3 (top to bottom)



(b) University of Brescia, p=2 (left) and p=3 (right)

Fig. 25. RANS of the turbulent flow in the T106A cascade: computed static pressure contours

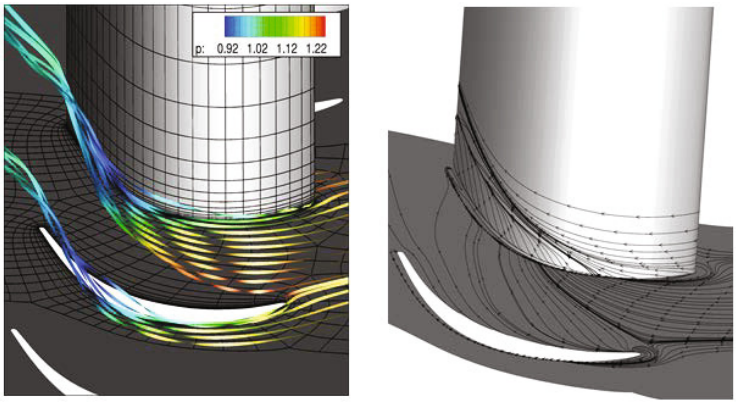


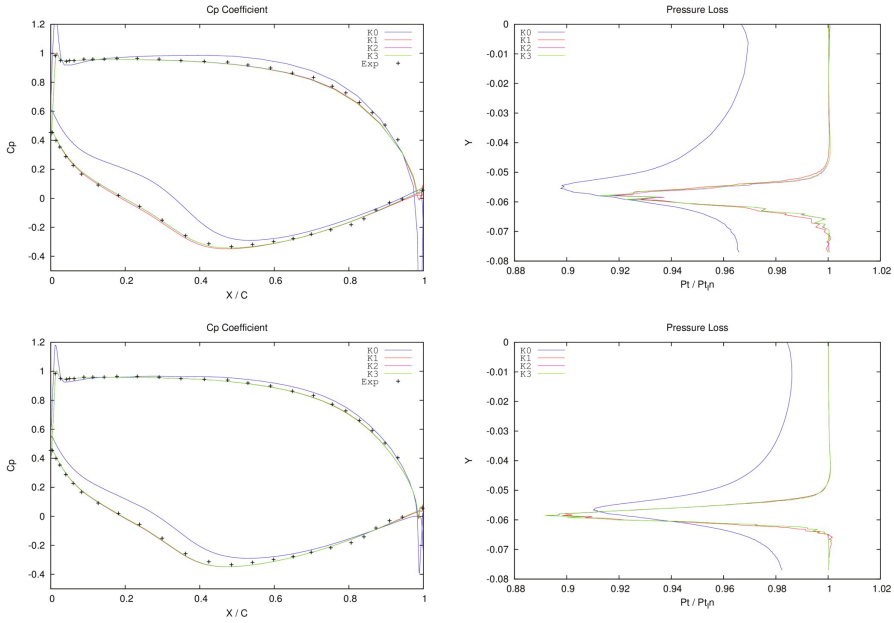
Fig. 26. RANS of the turbulent flow in the T106A cascade: streamlines and skin friction computed by the university of Brescia using 4th order accurate DGM

The computed flow fields, respectively Mach number and static pressure, are shown in the figures 24 and 25. Figure 26 shows the stream traces and skin friction lines, computed by the university of Brescia. Quantitative results, namely the pressure coefficient distribution and the loss distribution in the wake, are shown respectively in figure 27. It can be seen that both NUMECA and UNIBS have obtained very similar solutions. In fact, the agreement with the experimental data provided for the pressure coefficient is esentially perfect, for all orders p=1,2,3, even on the coarse mesh. Thus, the mesh convergence can be asserted. For orders p=1, 2, 3, the distributions of pressure coefficient and pressure loss essentially match each other, even on the coarsest mesh, indicating grid convergence. Both partners have obtained very accurate high-order solutions, essentially equal to each other.

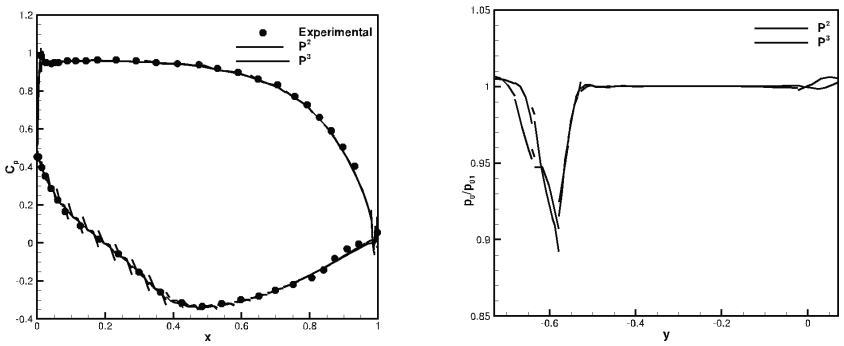
Relatively few convergence issues were met. A comparison of the computation times has not been performed, due to the very different iterative methods, NUMECA's code being based on explicit hp-multigrid, whereas UNIBS's solver is based on implicit iterations.

3.5 Conclusions

This flow case turned out to be straightforward for DG methods. High-order curved meshes were also relatively easy to generate, even by in-house algorithms from NUMECA and UNIBS. This allows to perform computations with very high order p on very coarse meshes.



(a) Numeca, coarse (top) and fine mesh (bottom), interpolation orders $p = 0 \ldots 3$



(b) University of Brescia, p=2 (left) and p=3 (right)

Fig. 27. RANS of the turbulent flow in the T106A cascade: computed pressure coefficient (left) and wake loss (right) distributions

4 Transitional Flow in the T106C Cascade

This test case was also included as C3.7 at the second International Workshop for High Order Methods in CFD [12]. Experimental data were provided by Prof. Arts of the von Karman Institute.

The flow conditions are defined with respect to the isentropic exit conditions, corresponding to the ratio of inlet total to exit static pressure, and correspond to $M_{2,is} = 0.59$ and $Re_{2,is} = 80.000$. These conditions lead to laminar separation, followed by transition in the shear layer and reattachment zone. The boundary conditions impose total pressure and temperature, as well as flow angle at the inlet, while static pressure is imposed at the outlet. The flow is assumed to be periodic in both the pitch- and spanwise directions, with the latter corresponding to 20 % of the axial chord C_{ax}. The air is furthermore assumed to behave as an ideal gas, and dynamic viscosity and conductivity are fixed to the values evaluated by Sutherland's law at exit isentropic conditions. The inlet was assumed to be free of turbulence, as natural transition is expected.

Cenaero was the only to compute this test case, with results published in the proceedings of the ASME Turbo Expo 2014 [11]. Fig. 28 illustrates the computational domain and the mesh, generated by Gmsh [10]. This two-dimensional mesh is extruded on 20 regular layers, leading to $262k$ hexahedra. Using a fourth order solution ($p = 3$), this gives $16.8M$ *degrees of freedom (dof)*.

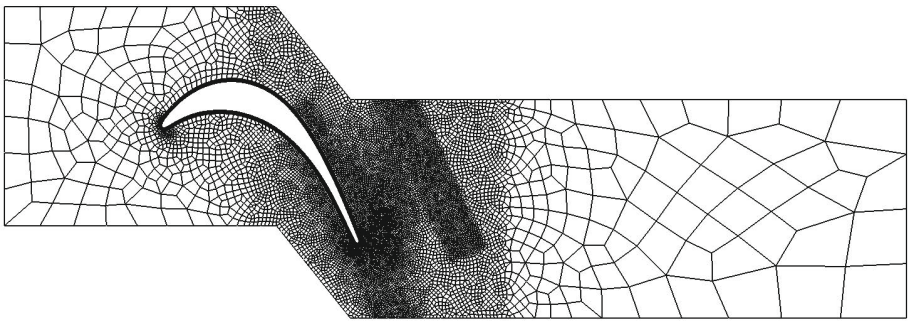
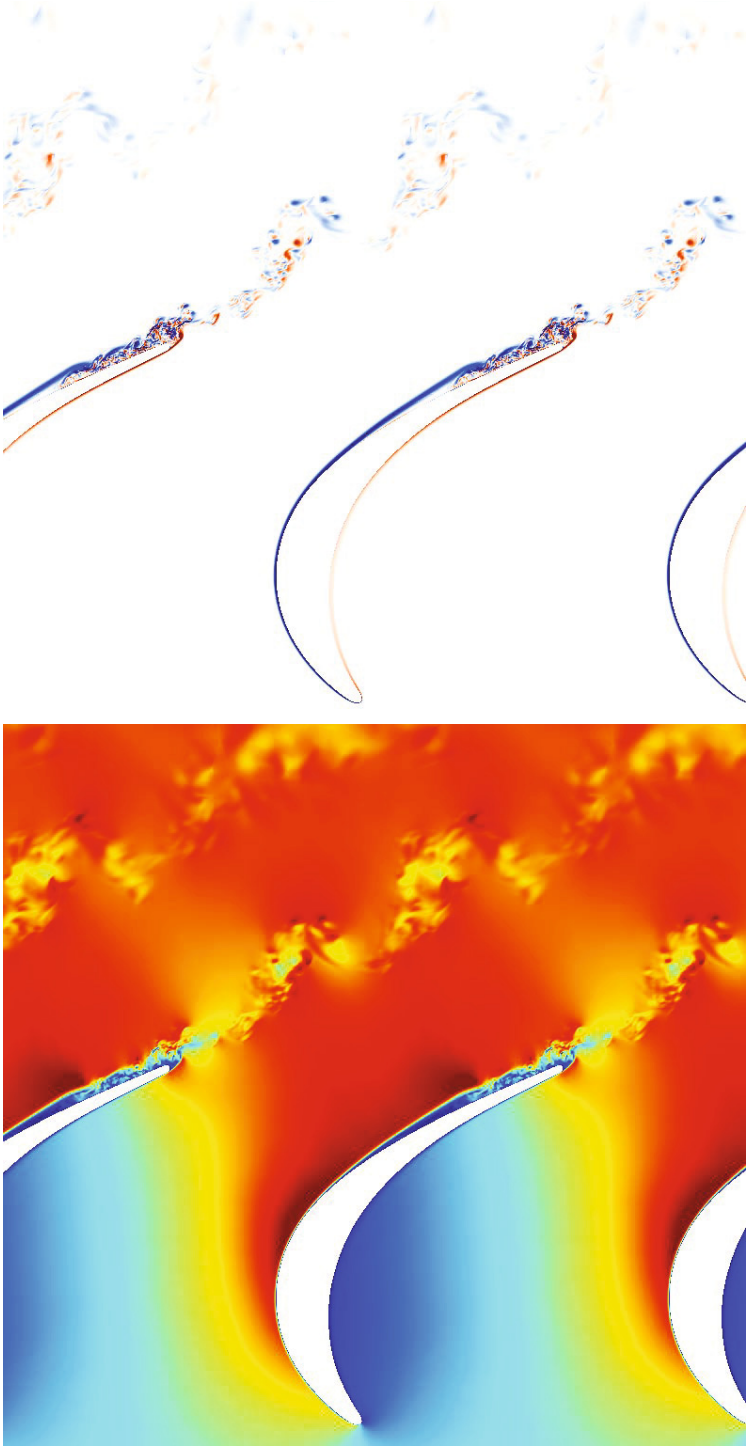


Fig. 28. ILES of the transitional flow in the T106C cascade: global view of the computational domain and mesh

Fig. 29 shows snapshots of the Mach number field and the spanwise components of the vorticity field. One sees that the flow field downstream of the separation point, located at $x/C_{ax} \approx 0.5C_{ax}$, is highly intermittent, alternating burst regions with calm flow. The width of the wake is firstly determined by these bursts, and evolves only slowly downstream. Transition occurs about $30\%C_{ax}$ downstream of the separation point in the recirculation bubble and reattachment region.



(a) Mach number

(b) Spanwise vorticity

Fig. 29. ILES of the transitional flow in the T106C cascade: global overview of the flow field on the spanwise periodic plane. The passage has been duplicated for clarity.

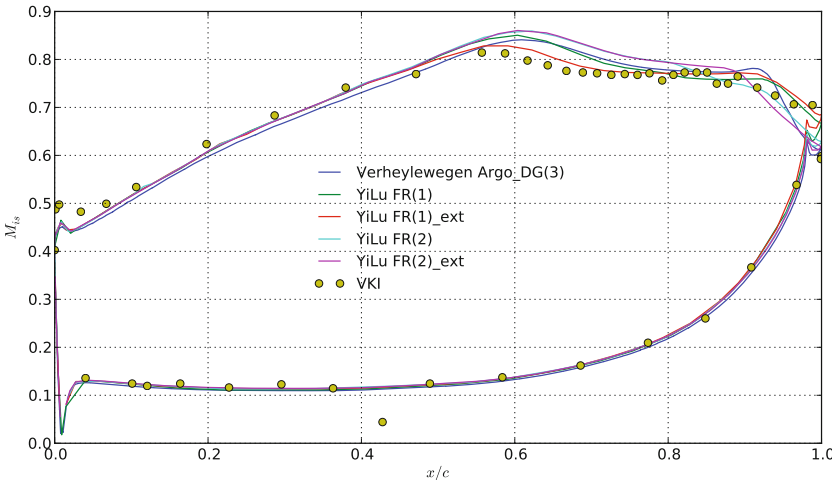


Fig. 30. ILES of the transitional flow in the T106C cascade: computed chordwise distribution of the isentropic Mach number

Fig 30 shows the resulting averaged isentropic Mach number distribution along the blade together with the results of the participants of the workshop [12] and the experimental data provided by the VKI. It is seen that all computations have a consistent difference with the experiment at the front suction side. Further LES computations, performed with a standard FVM code at Cenaero and RANS computations at VKI, showed a similar trend. Based on this consistency between the different computations and codes, it was conjectured that there was potentially a small difference in the flow conditions or the cascade geometry. This mismatch between computational and experimental setup is currently investigated at the VKI in collaboration with Cenaero.

5 Bump in a Square Duct

This test case was proposed in the framework of the European research project DESider, dealing with hybrid RANS-LES modelling [13]. It corresponds to a water tunnel experiment carried out by ONERA/Toulouse [14] in order to investigate flow separation, reattachment and post-reattachment recovery in a highly three-dimensional configuration. More detailed information about this configuration can be found in the DESider Project website [15].

5.1 Test Case Description

The geometry of ONERA's water tunnel is shown in Fig. 31. The channel has a length $L = 2.367m$ and a cross-sectional area of $0.5m \times 0.162m$ $(W \times H_c)$ at the inlet and $0.5m \times 0.3m$ $(W \times H)$ at the outlet. The shape of the bump was

chosen so that flow separation takes place in the middle region of the tunnel, independent of the geometry of the bump. The bump has a height $H_b = 0.138m$ at $x = -0.367m$, and ends at $x = 0$ ($L_b = 0.367$). At the inlet, the velocity profiles from the experiment are prescribed, which have a centreline velocity of approximately $7m/s$. Furthermore, water with a density of $997kg/m^3$ and a dynamic viscosity of $0.89 \times 10^3 Pa$ is considered. In case a compressible flow solver is used for solving this incompressible flow, an inflow Mach number of $M = 0.1$ is chosen.

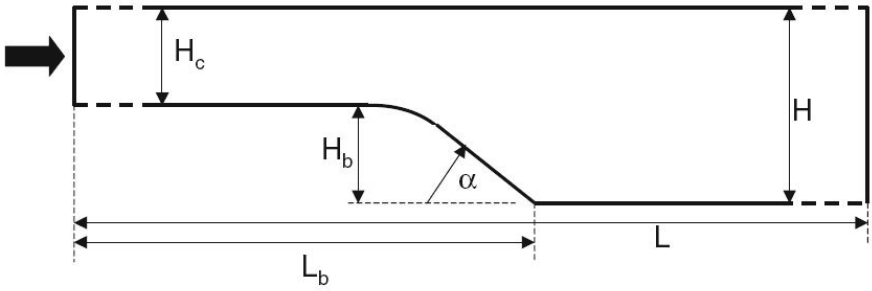


Fig. 31. Hybrid RANS-LES of the DESider bump: geometric configuration of the water tunnel

5.2 Grids

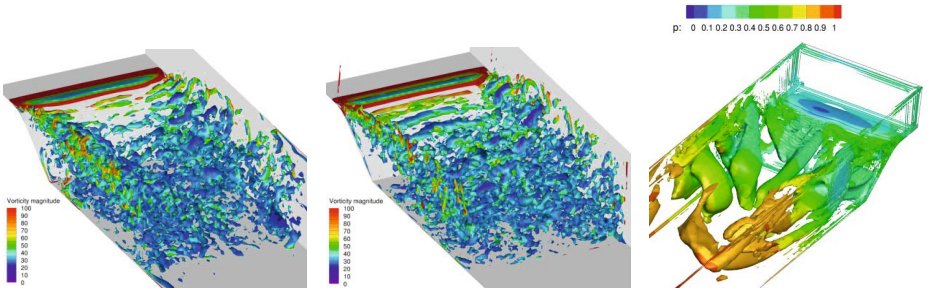
Two types of grids were used. NLR used a baseline structured grid of $142 \times 60 \times 76$ (647,520) cells and a grid with local grid refinement (LGR) and 355,560 cells. Compared to the baseline structured grid, the LGR grid is coarsened away from the separated flow region that is captured with LES, saving 40 to 45% of the grid points. UNIBG used a grid with curved cells (quartic edges) suitable for high-order methods, which was obtained by means of an in-house agglomeration software. The original linear mesh is a block-structured grid provided by ONERA, a different spacing has been used along the span wise direction. The resulting high-order DG mesh is made of 29,700 50-node hexahedral elements.

5.3 Baseline Solutions

As baseline solution, NLR performed an X-LES computation with its fourth-order finite-volume method (FVM) on the baseline structured grid. The obtained result is consistent with the results found in the DESider project, where a smaller flow separation region was found compared to the experiment.

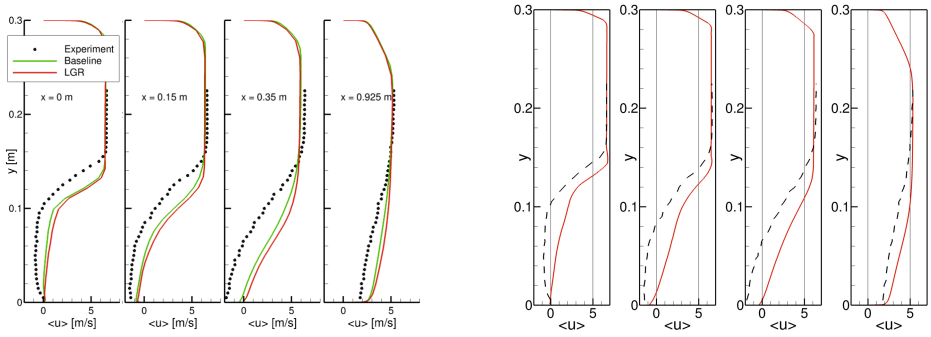
5.4 Assessment of High-Order Solutions

NLR successfully performed X-LES computations with its fourth-order finite-volume method on the LGR grid, maintaining stability and accuracy compared



(a) X-LES on baseline grid (b) X-LES on LGR grid (c) SA-DES on DG grid

Fig. 32. Hybrid RANS-LES of the DESider bump: instantaneous solution of DES-type computations with high-order methods. Figures (a) and (b): iso-contours of Q-criterion coloured by the vorticity magnitude. Figure (c) iso-contours of λ_2-criterion coloured by the pressure.



(a) X-LES with $4th$-order FVM (b) SA-DES with DG-p3 method (red - computation; black - experiment)

Fig. 33. Hybrid RANS-LES of the DESider bump: mean velocity profile at four stations in the span-wise centre plane ($z = 0$)

to the baseline computations. UNIBG successfully performed SA-DES computations for the first time using a discontinuous Galerkin (DG) fourth-order method (DG-p3) on the high-order grid. Instantaneous solutions are compared in Fig. 32. The NLR X-LES results show substantially finer turbulent structures than the UNIBG SA-DES results. Considering the number of degrees of freedom, which is of the same order for both approaches, one would expect a similar fineness of the turbulent structures. The UNIBG results are typical for first-time DES-type

computations - too large turbulent structures are found possibly due to some source of dissipation (e.g., too large filter width). As expected, more work will be needed to improve these results further. Nevertheless, being able to perform DES-type computations with a high-order DG method is already a success in itself.

The mean velocity profiles at four stations are shown in Fig. 33 . As mentioned above, the computations predict a significantly smaller separation region than the experiment. Compared to the experiment, all computations are comparatively close to each other, predicting reattachment at around $x = 0.35m$. Thus, despite the clear differences in terms of resolved structures, the mean flow does not appear to be strongly affected.

5.5 Conclusions

For the fourth-order FVM, stable and accurate X-LES computations have been performed on a grid with block-wise LGR. The solution is comparable to the baseline solution, although the separation region is slightly smaller. For the DG-p3 method, stable SA-DES computations have been performed. Although the resolved structures are clearly coarser than the baseline computation, similar mean velocity profiles are obtained.

6 Periodic Flow over a 2D Hill

This test case was proposed in an ERCOFTAC/IAHR Workshop in 1995 [16].

This configuration represents the flow over periodically arranged hills. As shown in Fig. 34, the flow presents separation from the curved surface, reattachment, and post-reattachment recovery. This test case is of interest for the evaluation of high-order methods in the context of DNS and LES, due to the periodic boundary conditions and the 2D character of the geometry which significantly reduces the computational costs.

Most of the technical information supplied here has been extracted from the references provided, and in particular from [26].

The geometry of the channel is shown in Fig. 35. This geometry was introduced by Mellen *et al.* [23] and is a modification of the geometry studied experimentally by Almeida *et al.* [16]. Denoting by h the hill height, the channel length is $L_x = 9h$ (inter-hill distance), the height is $L_y = 3.035h$ and the spanwise extent of the computational domain is $L_z = 4.5h$. The analytical shape of the $3.857h$ long 2D hill can be found in [26].

Experimental and numerical reference data are available at the following spatial locations: $x/h = 0.05$, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0 8.0.

In [17] the authors used an incompressible second-order Finite Volume (FV) solver, *LESOCC*, on a curvilinear grid composed of 13.1 million points to perform a DNS at $Re = 2,800$, as well as an LES at $Re = 10,595$. These results will be used as reference for the cross-comparison of results presented in Sections 6.1 and 6.2. Results from LES simulations at $Re = 10,595$ have also been reported in the literature by Frölich *et al.* [21] and Chaouat *et al.* [19, 20].

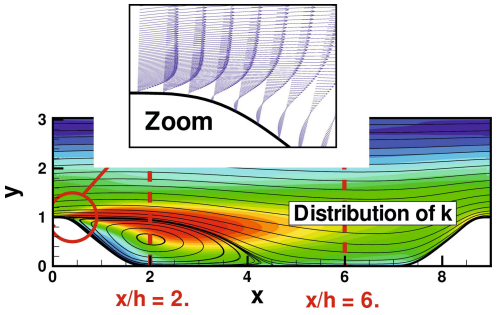


Fig. 34. DNS and LES of the periodic flow over a 2D hill: streamlines of the time-averaged flow field. Figure reproduced from the test case description on the QNET-CFD wiki [26]

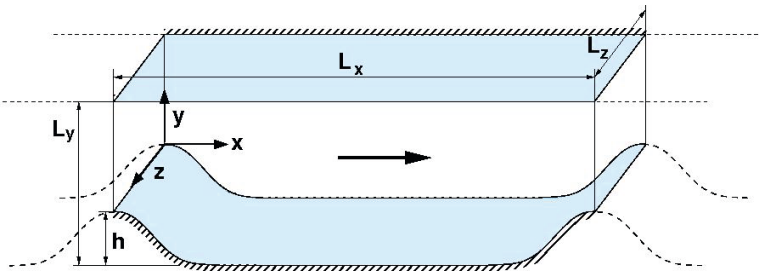


Fig. 35. DNS and LES of the periodic flow over a 2D hill: geometric configuration of the channel. Figure reproduced from the test case description on the QNET-CFD wiki [26]

Required Grid Resolution. This benchmark was proposed as test case C3.6 in the framework of *2nd International Workshop on High-Order CFD Methods* (see test-case summary in [22]). Two computations were proposed, namely DNS at $Re_b = 2800$ and LES at $Re_b = 10595$, both at the same equivalent spatial resolution, compensated for the subcell resolution of polynomial methods such as DGM, RDS or SDM. A series of second-order hexahedral grids were proposed, obtained by successive coarsening of a reference straight mesh with resolution of $512 \times 256 \times 256$. Figure 36 shows the coarsest curved mesh proposed in this workshop consisting of $32 \times 16 \times 16$ elements, used for DGM(p=4). The simulations in the IDIHOM project have followed the same specifications.

Flow Conditions and Forcing Term. The Reynolds number Re_b is defined based on the bulk velocity u_b and on the hill height at the crest h, namely

$$Re_b = \frac{u_b h}{\nu} \qquad u_b = \frac{1}{2.035} \int_h^{3.035h} u(y) dy \tag{1}$$

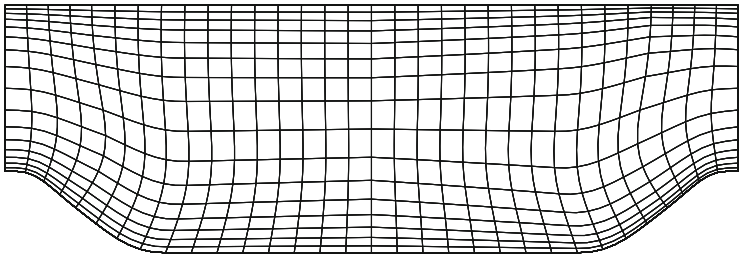


Fig. 36. DNS and LES of the periodic flow over a 2D hill: second-order mesh used in the *2nd International Workshop on High-Order CFD Methods*

As for the turbulent plane channel flow configuration, a forcing term is added to the streamwise momentum and the energy equations. This body force $f_p(t)$ therefore drives the flow and represents a pressure gradient in the streamwise direction. Following the approach proposed in [18], also described in [22], this source term is updated at each time step as

$$f_p^{n+1}(t+\Delta t) = f_p^n(t) - \frac{1}{\Delta t}\left(Q_0 - 2Q^n + Q^{n-1}\right) \tag{2}$$

where Q^n is the mass flow rate per unit surface computed at each time step $t + n\Delta t$ and Q_0 its target value. The forcing term also appears in the energy equation as uf_p.

In the case a compressible flow solver is used it is necessary to specify a low value for the Mach number (e.g. $M_0 = 0.1 - 0.2$) so that the flow regime is quasi-incompressible. The temperature at the wall T_w is therefore computed as,

$$T_w = \frac{u_b^2}{\gamma R M_0^2} \tag{3}$$

in which $\gamma = 1.4$ is the isentropic coefficient and R the gas constant.

The flow is periodic in the streamwise and spanwise directions, and isothermal no-slip boundary conditions are imposed on the lower and upper walls.

6.1 Comparison of DNS Results at $Re = 2,800$

Three IDIHOM partners have reported results for this test case: Cenaero, ICL and USTUTT. As mentioned above, the results are compared against the DNS simulation performed by Breuer *et al.* [17] using the 2nd-order FV solver *LESOCC*. For completeness, the results obtained by H. Lüdeke (DLR) using a 4*th*-order Finite Difference (FD) code, presented in the *2nd International Workshop on High-order CFD methods*, have also been included for completeness.

Table 7. DNS and LES of the periodic flow over a 2D hill: summary of the DNS computations at $Re = 2,800$. DG-pN stands for Discontinuous Galerkin method with polynomial order p=N. SEM/PS denotes the use of a spectral element method in the streamwise and vertical directions and a pseudo-spectral Fourier method in the spanwise direction.

Author	*Num. Method*	*Solver*	DOF	$\Delta t/10^{-3}$	T_{avg}
Breuer *et al.* [17]	2nd-order FVM Exp. RK3	Incomp.	13.1 M	2.0	1249
Cenaero	DG-p3 Imp. 3BDF	Comp.	33.5 M	10.	150
ICL	6th-order SEM/PS Exp./Impl.	Incomp.	20.9 M	1.	512
USTUTT	DGSEM-p9 Exp. RK4	Comp.	8.2 M	0.134	500
Lüdeke (DLR)	4th-order FD Exp. RK4	Comp.	4.1 M	0.05	270

Table 8. DNS and LES of the periodic flow over a 2D hill: summary of grid resolutions used for the DNS and the LES computations

Author	*Grid resolution*	*# Elements*	*Grid order*
Breuer *et al.* [17]	-	12.4 M	1
ARTS	64×32×32	65,536	4
Cenaero	128×64×64	524,288	2
ICL	74 ×49×160	3626 × 160 Fourier	6
USTUTT	32 ×16×16	8192	4
Lüdeke (DLR)	256 ×128×128	4.19M	1

Table 7 summarises the basic properties of the simulations being compared as well as the specific numerical method employed. For further details on the discretisations, the reader is referred to the individual partner contributions. The grid resolutions used in these simulations are ment to provide DNS resolution and are compiled in Table 8.

Figures 37 show the profiles of mean streamwise and vertical velocity and velocity correlations. As to be expected, very good results have been obtained

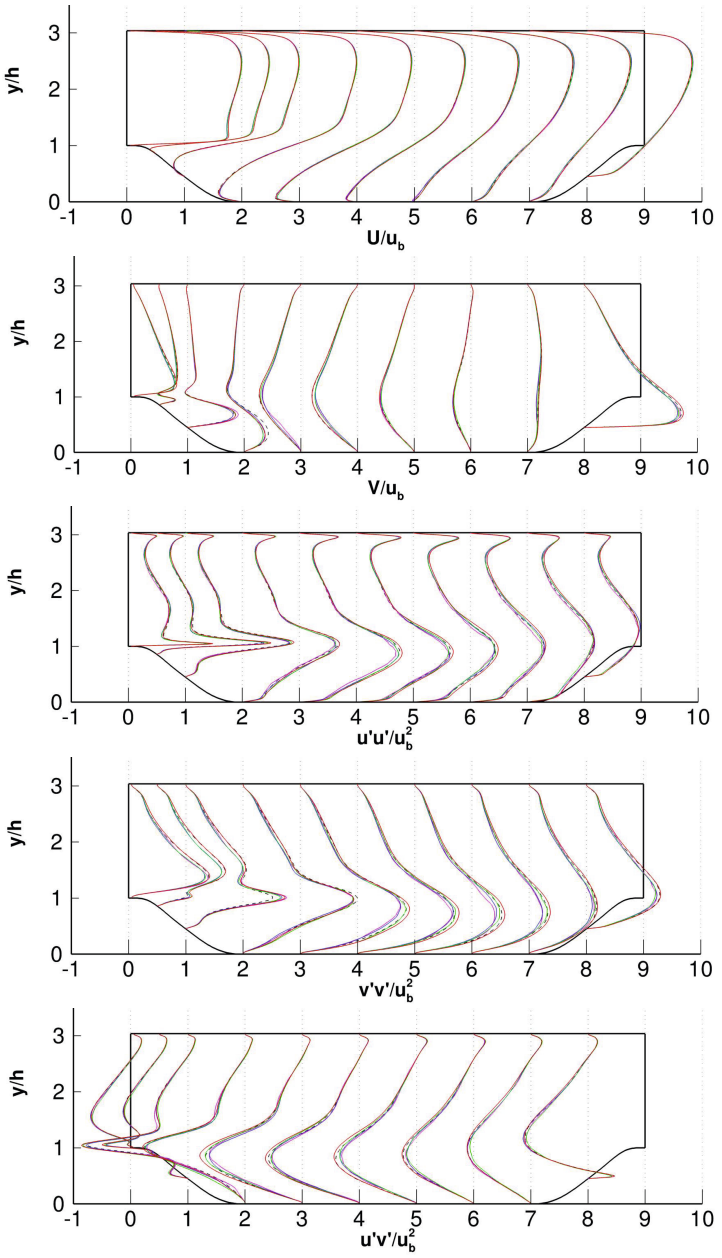


Fig. 37. DNS and LES of the periodic flow over a 2D hill: comparison of the computed profiles of the mean velocity and Reynolds stress components for the DNS computations at $Re = 2,800$. The dashed black lines correspond to the reference results of Breuer *et al.* The solid lines correspond to the results from the different codes : Cenaero, ICL, USTUTT, Lüdeke.

on all quantities. USTUTT has performed additional, less resolved computations, also leading to a good correspondence to reference results. Details can be found in the individual contribution in chapter 2.

6.2 Comparison of LES Results at $Re = 10,595$

ARTS, USTUTT and Cenaero have provided results for this configuration. The computational setup is given in Tab. 9. The mesh resolutions are again found in Tab. 8. Additional computations are described in the partner-specific sections in chapter 2.

Table 9. DNS and LES of the periodic flow over a 2D hill: summary of the LES computations at $Re = 10,595$. DG-pN stands for Discontinuous Galerkin method with polynomial order p=N.

Author	*Num. Method*	*Solver*	DOF	*LES strategy*
Breuer *et al.* [17]	2nd-order FVM Exp. RK3	Incomp.	13.1 M	Smagorinsky Van Driest wall damping
ARTS	RBC3 Imp. Gear (2nd)	Comp.	65,536	No model
Cenaero	DG-p3 Imp. 3BDF	Comp.	33.5 M	No model
USTUTT	DGSEM-p9 Exp. RK4	Comp.	8.2M	Overintegration

The comparison of these results with the reference simulation of Breuer *et al.* and with the available experimental data [17,24] is shown in Fig. 38. Overall, very good correspondence between computations and reference results is found for the time-averaged velocities for both USTUTT and Cenaero, usually providing a better match with the reference computations. In spite of the very coarse resolution, the results of ARTS are still surprisingly fair. For the Reynolds stresses good results were only obtained by USTUTT; however, it should be noted that at the time of the writing Cenaero had not yet obtained statistical convergence for these quantities. Both Cenaero and USTUTT performed further coarser computations, which are detailed in their individual contributions in chapter 2. These resulted in an almost as good correspondence with the reference results as shown here. For Cenaero this includes a much better match for the correlations due to the longer averaging time.

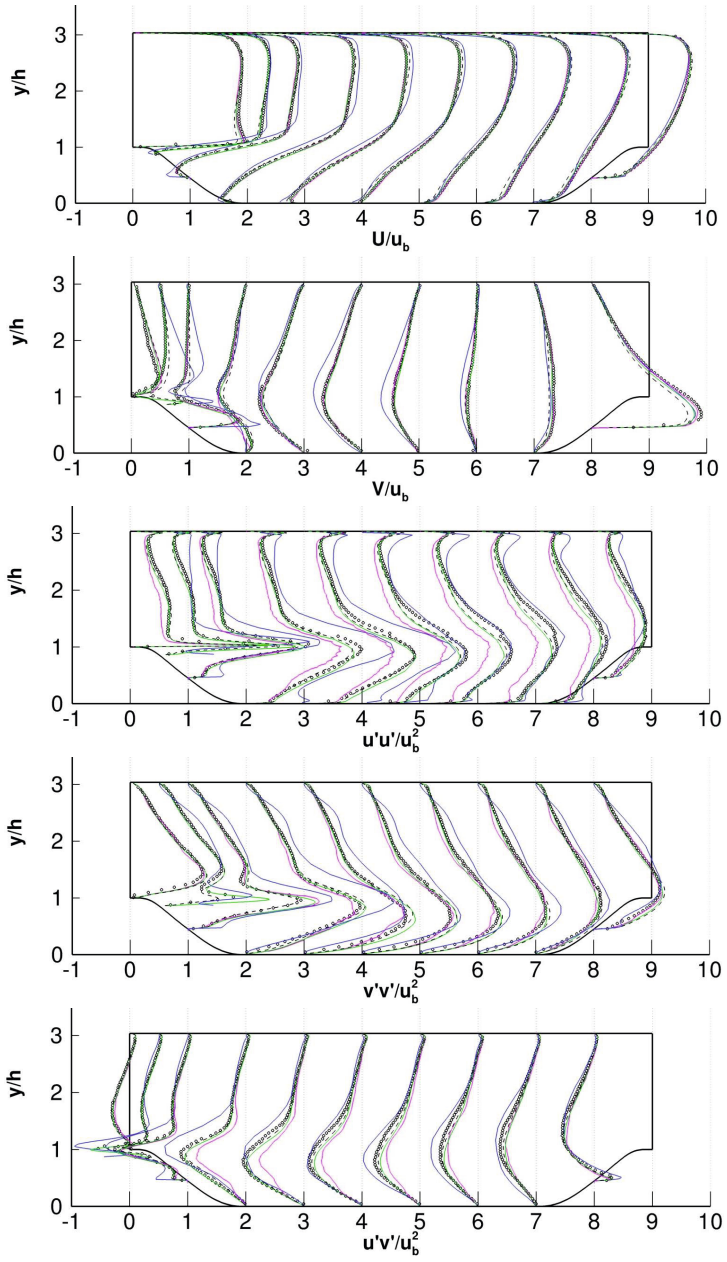


Fig. 38. DNS and LES of the periodic flow over a 2D hill: comparison of the computed profiles of the mean velocity and Reynold stress components for the LES computations at $Re = 10,595$. Dashed black lines correspond to the reference results of Breuer et al. and circles to experimental data. Solid lines correspond to the results of Cenaero, ARTS, USTUTT. Cenaero's Reynolds stresses have not yet fully converged statistically.

References

1. Reid, L., Moore, R.D.: Design and overall performance of four highly loaded, high-speed inlet stages for an advanced high-pressure-ratio core compressor. NASA TP 1337 (1978)
2. ASME Turbomachinery Committee, CFD Code Assessment in Turbomachinery - Author's Information Package (June 1993)
3. Shabbir, A., Celestina, M.L., Adamczyk, J.J., Strazisar, A.J.: The effect of hub leakage on two high speed axial flow compressor rotors. ASME Paper 97-GT-346 (1997)
4. AGARD Advisory Report 355, CFD Validation for Propulsion System Components, Report of the Propulsion and Energetics Panel Working Group 26 (1998)
5. Home page of the European FP5 research project JEAN (Jet Exhaust Aerodynamics and Noise), `http://www.ist-world.org/ProjectDetails.aspx?ProjectId=14e67d914b3141cabcf1b1499e404a0a`
6. JEAN project page on the web site of the X-Noise project, `http://www.xnoise.eu/index.php?id=162`
7. Jordan, P., Gervais, Y., Valire, J.-C., Foulon, H.: Final results from single point measurements, Project deliverable D3.4. JEAN EU 5th Framework Programme, G4RD-CT-2000-00313 (2002)
8. Spalart, P.R., Allmaras, S.R.: A one-equation turbulence model for aerodynamic flows. La Recherche Aérospatiale 1, 5–21 (1994)
9. Andersson, N., Eriksson, L.-E., Davidson, L.: Large-Eddy Simulation of a Mach 0.75 Jet, AIAA paper 2003-3312. In: 9th AIAA/CEAS Aeroacoustics Conference (2003)
10. Geuzaine, C., Remacle, J.-F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79(11), 1309–1331 (2009)
11. Hillewaert, K., Carton de Wiart, C., Verheylewegen, G., Arts, T.: Assessment of a high-order discontinuous Galerkin method for the direct numerical simulation of transition at low Reynolds number in the T106C high-lift low pressure turbine cascade. In: Proceedings of the ASME Turbine Technical Conference and Exhibition, Number GT2014-26739. ASME Turbo Expo (2014)
12. Second International Workshop on High-Order CFD Methods, Cologne, Germany, May 27-28 (2013), `http://www.dlr.de/as/desktopdefault.aspx/tabid-8170/13999_read-35550/`
13. Haase, W., Braza, M., Revell, A.: DESider – A European Effort on Hybrid RANS-LES Modelling. NNFM, vol. 103, pp. 2004–2007. Springer, Heidelberg (2009)
14. Aupoix, B., Barricau, P., Egorov, Y., Geiler, C., Gilliot, A., Menter, F.R., Monnier, J.-C., Pailhas, G., Perret, L., Stanislas, M., Touvet, Y.: The DESider Bump Experiment. In: Proc. 7th Int. ERCOFTAC Symp. Eng. Turb, Turkey, June 4-6 (2008)
15. `http://cfd.mace.manchester.ac.uk/desider/`
16. Almeida, G.P., Durao, D.F., Heitor, M.V.: Wake flows behind two dimensional model hills. Experimental Thermal and Fluid Science 7, 87–101 (1993)
17. Breuer, M., Peller, N., Rapp, C., Manhart, M.: Flow over periodic hills - Numerical and experimental study in a wide range of Reynolds numbers. Computers & Fluids 38(2), 433–457 (2009)
18. Benocci, C., Pinelli, A.: The role of the forcing term in the large eddy simulation of equilibrium channel flow. In: Proceedings of the International Symposium on Engineering, Turbulence Modelling, and Measurements, Yugoslavia (1999)

19. Chaouat, B.: Subfilter-Scale Transport Model for Hybrid RANS/LES Simulations Applied to a Complex Bounded Flow. Journal of Turbulence 11(51), 1–30 (2010)
20. Chaouat, B., Schiestel, R.: Hybrid RANS-LES Simulations of the Turbulent Flow over Periodic Hills at High Reynolds Number. Computers & Fluids 84, 279–300 (2013)
21. Fröhlich, J., Mellen, C.P., Rodi, W., Temmerman, L., Leschziner, M.A.: Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. Journal of Fluid Mechanics 526, 19–66 (2005)
22. Hillewaert, K., Carton de Wiart, C., Bricteux, L.: DNS and LES of the flow over a 2D periodic hill (Test case 3.6). In: 2nd International Workshop on High-order CFD Methods, Köln (May 2013), http://www.dlr.de/as/desktopdefault.aspx/tabid-8170/13999_read-35550/
23. Mellen, C.P., Fröhlich, J., Rodi, W.: Large-eddy simulation of the flow over periodic hills. In: Deville, M., Owens, R. (eds.) Proceedings of the IMACS World Congress, Lausanne (2000)
24. Rapp, C.: Experimentelle Studie der turbulenten Strömung über periodische Hegel. Mitteilungen, Nr. 75, Fachgebiet Hydromechanik, TUM (2009)
25. Rodi, W., Bonnin, J.C., Buchal, T. (eds.): Proceedings of the ERCOFTAC Workshop on Data Bases and Testing of Calculation Methods for Turbulent Flows, in association with 4th ERCOFTAC/IAHR Workshop on Refined Flow Modeling, April 3-7. University of Karlsruhe, Germany (1995)
26. ERCOFTAC QNET-CFD wiki, Test case UFR 3-30: 2D periodic hill flow, http://qnet-ercoftac.cfms.org.uk/w/index.php/UFR3-30

Aeroelastic Testcases in IDIHOM Project

K. Kotecki[1], H. van der Ven[2], J. Kok[2], H. Hausa[1], M. Nowak[1], W. Stankiewicz[1], R. Roszak[1], and M. Morzyński[1]

[1] Poznań University of Technology, Division of Virtual Engineering, 60-965, Poznań, Poland
krzysztof.kotecki@put.poznan.pl
[2] Netherlands National Aerospace Laboratory, Flight Physics and Loads, Amsterdam, The Netherlands

Abstract. In the IDIHOM Project three aeroelastic testcases have been calculated. Two of them - LANN Wing and DLR-F6 wing-body configuration - have been conducted by PUT. The last one, the HART II rotor has been prepared by NLR. Each partner has used different technology and software tools, hence each of the testcases describes the results of the simulation and technology which has been used.

Keywords: Aeroelasticisty, high order, CFD.

1 Introduction

With the development of new techniques in CFD like Finite Element Method, Discontinuous Galerkin Method, High Order Methods, Adjoint Mesh Refinement also they find use in aeroelastic systems. This transition in technology has to be accompanied in complete revision of CFD tools and methods, ranging from curvilinear grid generation to visualization of high order discontinuous results.

The new developed systems and tools require to be validated using common and well known testcases. In the IDIHOM Project three aeroelastic testcases have been calculated. Two of them - LANN Wing and DLR-F6 wing-body configuration - have been conducted by PUT. The last one, the HART II rotor - example of rotating geometry has been prepared by NLR.

2 LANN Wing

One of the most widely used examples used in CFD is, for many years, LANN wing proposed by AGARD group [2]. The wing has a semispan 1m in length, a planform area of 0.25m^2, a root chord of 0.361m, a tip chord of 0,144m, has a 1/4-chord sweep angle of 25° and a linear twist from root to tip of 4.8°. The CFD simulation has been prepared with flow parameters: Ma = 0.4, Re = $4.9 \cdot 10^6$ and angle of attack equal 2.56° [3].

The calculation has been conducted on the Aeroelastic System developed within IDIHOM Project Aeroelastic System (IDIHOM AE System). The loosely

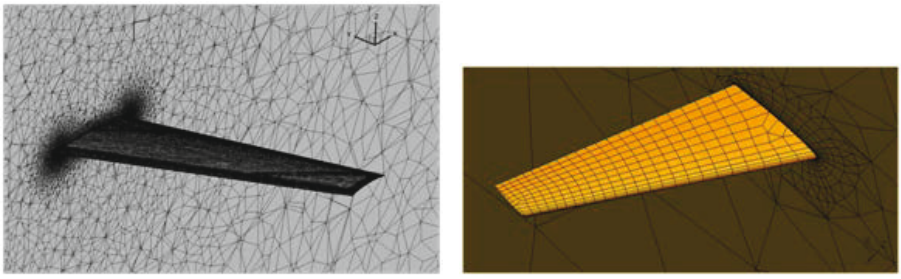
N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_28

coupled technique bases on PADGE Code - Discontinuous Galerkin High Order CFD Solver delivered by DLR. The CSM part has been solved using modal approach. Other tools for interpolation data between models, deformation CFD grid and interface for discontinuous solution have been developed by PUT and are described in previous chapters. The results have been compared with the reference one obtained from TAURUS AE System [1]. TAURUS AE System bases on TAU Code which is second order FV CFD solver. In TAURUS AE System the linear CFD mesh has been used, unlike to DIHOM AE System.

2.1 Grids

CAD Geometry has been delivered by Aircraft Research Association (ARA). The model has been used as a coupling surface for exchanging information between CFD and CSM models.

The linear mesh (fig.1a) is unstructured hybrid grid with prism elements in boundary layer. The curvilinear grid (fig.1b) for high-order computations consists of prism, pyramid, hexahedral and tetrahedral elements. The details of both meshes are given in tab.1.



(a) linear (b) curvilinear

Fig. 1. LANN grids

Table 1. Grids details

Grid	**linear**	**curvilinear**
No of nodes	2.6M	98 469
No of elements	10.5M	19 981
- prism	2.5M	432
- pyramid	17k	673
- tetrahedral	7.5M	5 762
- hexahedral	0	12 114

2.2 Solution

The convergence of aeroelastic steady state computations is obtained after five iterations (fig. 2, 3). There is a good agreement between the results of first-order (TAURUS) and high-order (IDIHOM) AE systems.

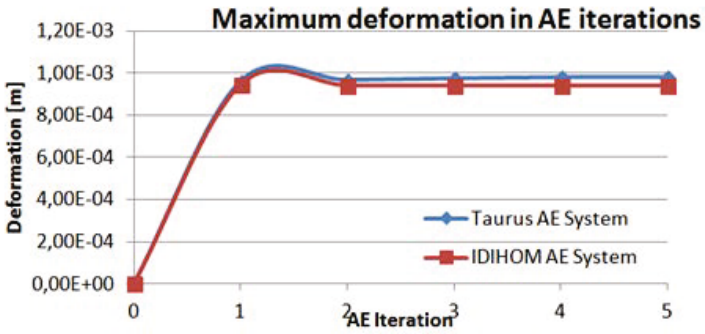


Fig. 2. Aeroelastic steady state LANN wing deformation computed using first order (TAURUS) AE System and high order (IDIHOM) AE System

The deformation computed using High-Order methods is 0.942mm. Comparing to the reference solution, where the deformation is 0.980mm, the difference is 3.96%.

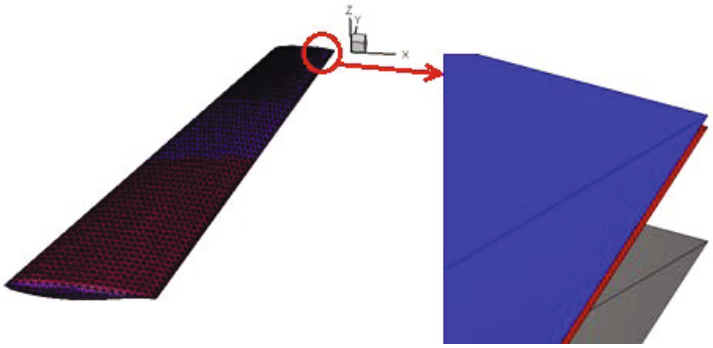
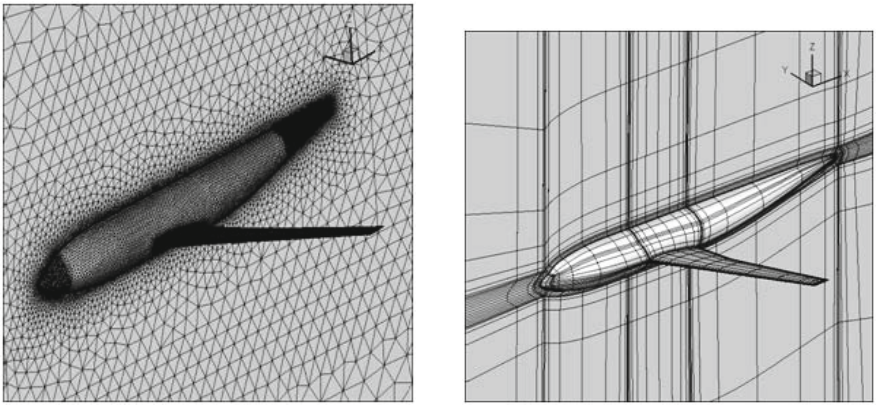


Fig. 3. Deformed wing: grey - non-deformed structural model, blue - deformed model calculated in TAURUS AE System, red - deformed model calculated in IDIHOM AE System

3 DLR-F6 Wing-Body Configuration

The second testcase was chosen the DLR-F6 wing body configuration as the full-scale industrial testcase. [4]. In this case, for high order computations structured grid has been used. In the reference computation, similarly as LANN wing, hybrid grid has been used. Both the grids are depicted in the fig.4 and described in tab.2.



(a) linear (b) curvilinear

Fig. 4. DLR-F6 grids

Table 2. Grids details

Grid	linear	curvilinear
No of nodes	4.0M	3.3M
No of elements	10.2M	50 618
- prism	6.1M	0
- pyramid	11k	0
- tetrahedral	5.1M	0
- hexahedral	0	50 618

Again, the aeroelastic steady state computations converge after five iterations (fig.5).

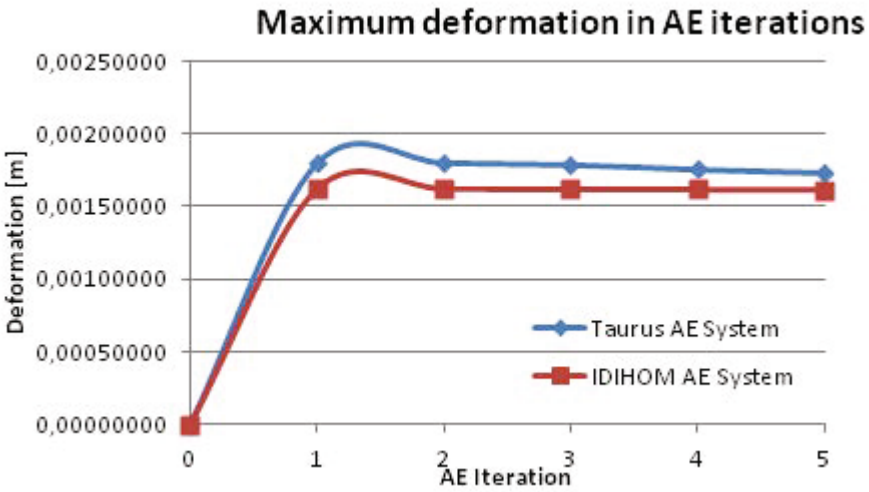


Fig. 5. Aeroelastic steady stare DLR-F6 aircraft deformation

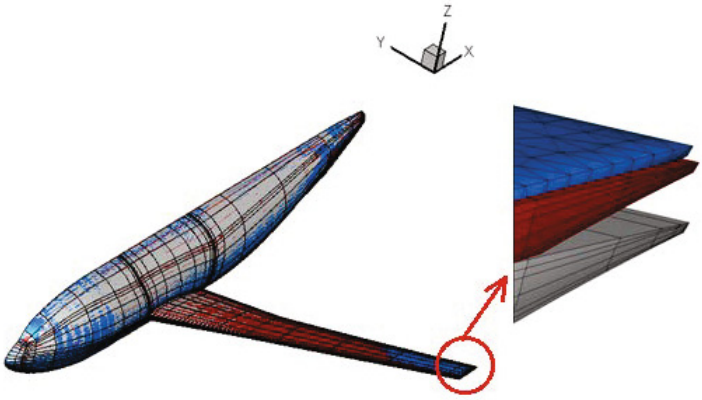


Fig. 6. Deformed aircraft: grey - non-deformed structural model; blue - deformed model calculated in TAURUS AE System; red - deformed model calculated in IDIHOM AE System

This time, the difference between reference solution (deformation: 1.733mm) and High-Order solution (deformation: 1.619mm) is 6.58% (fig.6).

More details about the technology used for the aeroelastic simulations are presented in article: "Aeroelastic system for large scale computations with High Order Discontinuous Galerkin Flow Solver"

4 HART II Rotor - Trimmed Aeroelastic Simulation

In this chapter, aeroelastic simulations are performed for an isolated four-bladed rotor in one of the flow conditions of the HART-II experiment. The HART-II experiment is described in detail in Van der Wall et al. [10]. The chosen flow condition is a slow descent flight where multiple BVI events take place during a rotor revolution. The rotor radius of the windtunnel model is 2 meters. Rotational frequency is 109.12 rad/s. Forward speed of the rotor is 32.9 m/s. Effective shaft angle is 4.5 degrees (tilted backward).

4.1 Grid

For rotor flows the convection of the tip vortices in the wake of the blades is important for the capture of the blade-vortex interaction. As the vortices move through the wake, the easiest way of obtaining a mesh with sufficient resolution in the vortex regions is to uniformly refine the mesh in a cylinder around the rotor blades. So the definition of the refinement region in this case is a cylinder centred at the rotor hub with radius equal to the rotor radius and sufficient height to contain the vortices.

The target mesh width is $0.01R$, where R is the rotor radius. This corresponds to 16% chord, which is a rather coarse resolution. The original mesh has 13 million cells, the refined mesh has 26 million cells. Details of the mesh can be found in Kok et al. [8].

4.2 Rigid-Blade Trim

As a pre-cursor computation the rotor is trimmed with rigid blades. The rotor has been trimmed to the experimental values of thrust, rolling moment, and pitching moment. The trim results are tabulated in Table 3.

4.3 Aeroelastic Trim

To account for the effects of blade flexibility to the trim, mutual interaction between blade aerodynamics and structural dynamics is modelled by solving the aeroelastic equations in modal space. The aeroelastic deformations are expressed in a limited degrees-of-freedom model built upon the vibration modes of the rotating blade. The required modal data, that is, the natural frequencies and vibrations modes, are computed with respect to the deformed shape due to centrifugal loads. This means that the deformation is linearized around this deformed shape, see Figure 7.

The structural dynamics of the blade is represented using a finite element modelling. A NASTRAN model of the blade is generated based on the available elasto-mechanical properties reported in the documentation of the HARTII experiment [10]. First, static deformation of the blade is determined using a nonlinear approach. Subsequently normal mode analysis is carried out starting from this deformed shape, taking into account the effects of the centrifugal force

in the stiffness matrix. The natural frequencies of the blade at various rotational speeds are shown in Figure 8. Measured frequencies at zero rotational speed and computed frequencies using CAMRAD at the reference rotational speed are also presented. Good agreement with both data is observed. Hence, no attempts have been made to improve the correlation by model-updating.

During the simulations on the original grid it became clear that the response of the first lead-lag mode (at a frequency of 0.64/rev) was hardly damped by the aerodynamic forces. As the corresponding period of the mode is almost one-and-a-half revolution, the coupled fluid-structure problem displayed a period of three revolutions. From period to period the response was damped a little bit, as can be seen in the first eight revolutions of the response plotted in Figure 9. To resolve the issue, this mode was damped in the structural model, and a periodic solution of the fluid-structure problem was quickly obtained (see the next four revolutions in Figure 9). The trim values are tabulated in Table 3.

Subsequently the flow results were interpolated onto the refined mesh and the simulation continued with the same trim settings. As can be seen from the next five revolutions in Figure 9, the structural response did not change significantly. The aerodynamic forces, however, deviated from the experimental values. Hence, the rotor was retrimmed, to the control angles shown in Table 3 in order to obtain the correct aerodynamic forces. As can be seen in the last five revolutions in Figure 9, the response from the first flap mode changed significantly, and obviously this can have a definite impact on the miss distance between blades and vortices.

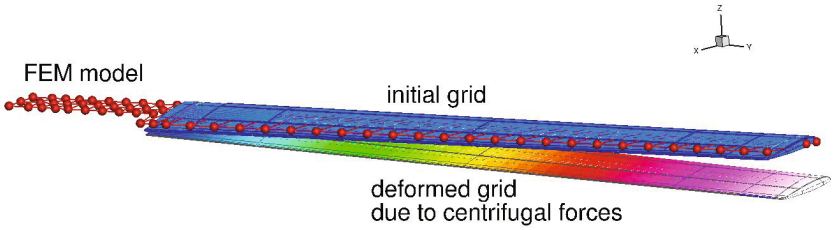


Fig. 7. Overview of the initial grid of the blade and deformed grid due to centrifugal force, about which the linearisation of the deformation is defined

4.4 Flow Results

The turbulent flow is modeled using the Reynolds-averaged Navier-Stokes equations with the TNT k-ω turbulence model [6]. All simulations have been run using the fourth-order accurate finite volume scheme [7]. Simulations have been performed on the original mesh of 13 million elements, and the locally refined mesh of 26 million elements. The discretization algorithm on locally refined meshes

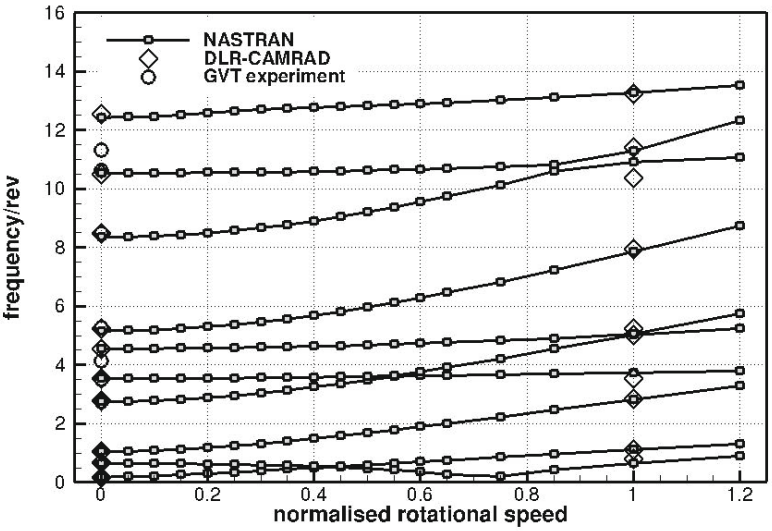


Fig. 8. Campbell diagram of the blade showing the influence of the rotational speed to the natural frequencies of the blade (lines with square symbols). The normalised rotational speed on the horizontal axis is normalised by the rotational speed of the HART-II experiment. Measured frequencies at zero rotational speed (GVT; circles with black outline) and computed frequencies using CAMRAD (DLR-CAMRAD; diamonds with black outline) at the reference rotational speed are taken from [10].

Table 3. Trim values for the simulations. The pitch angle θ of the blade is computed as $\theta = \theta_0 + \theta_{1c}\cos(\psi) + \theta_{1s}\sin(\psi)$ where ψ is the azimuth angle.

model	grid	θ_0	θ_{1c}	θ_{1s}	thrust [N]	roll moment [Nm]	pitch moment [Nm]
experiment	–	3.8°	1.94°	−1.34°	3300	20	-20
rigid blades	original	3.07°	1.92°	−1.61°	3630	30	-6
flexible blades	original	4.9°	1.7°	−0.15°	3350	42	-24
flexible blades	refined	4.4°	1.3°	−0.15°	3265	4	-14

is described in [8]. In the current simulations a second order interpolation algorithm is applied, rather than the fourth-order algorithm described in [8]. The time step corresponds to 0.5° azimuth for all simulations. An impression of the vortex signature in the simulation is given in [8].

The sectional vertical force coefficient at 87% for the two simulations with flexible blades is shown in Figure 10.

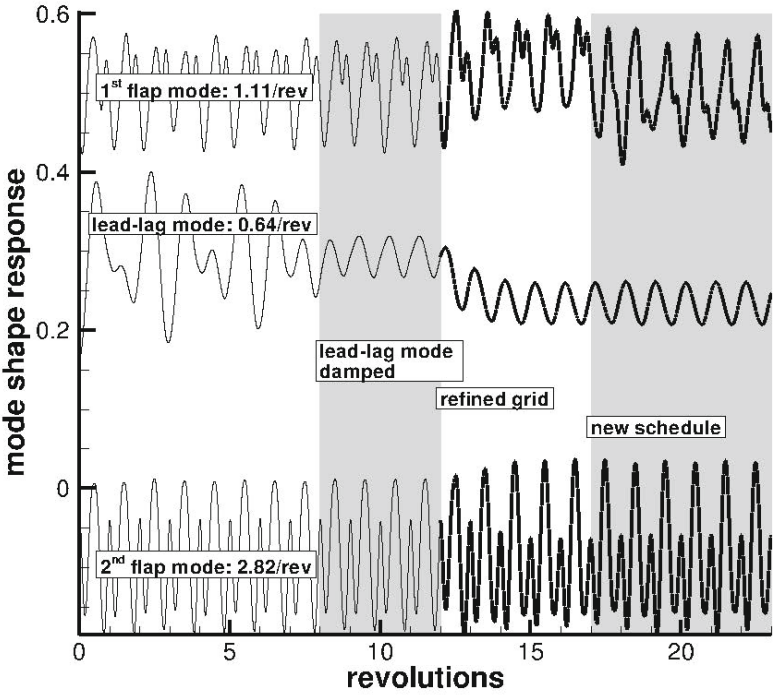


Fig. 9. The response of the first three modes for the simulations on original and refined meshes

First, the global behaviour of the force history is discussed. The simulation on the original grid generally agrees with the experiment on the retreating side (azimuth angles greater than 200 degrees). On the advancing side the agreement is less than satisfactory. The retrim on the refined mesh improves the agreement with experiment for azimuth angles greater than 120 degrees, but for smaller angles no improvement is seen. This suggests that the (controlled) blade motion in the simulation is different from the experiment, especially since the control angles of simulation and experiment are different (see Table 3). However, comparing the simulation results with the results of Jayaraman et al. [5] suggests otherwise. Figure 11 shows their results combined with the results of the current paper. The control angles in the simulations of Jayaraman match very well with the experimental values. The global behaviour of the sectional lift for all simulations (current and Jayaraman's) is generally the same, especially for the BVI regions on both advancing and retreating side. One of the conclusions of the HART-II workshop (Smith et al. [9]) is that the BVI events at the advancing

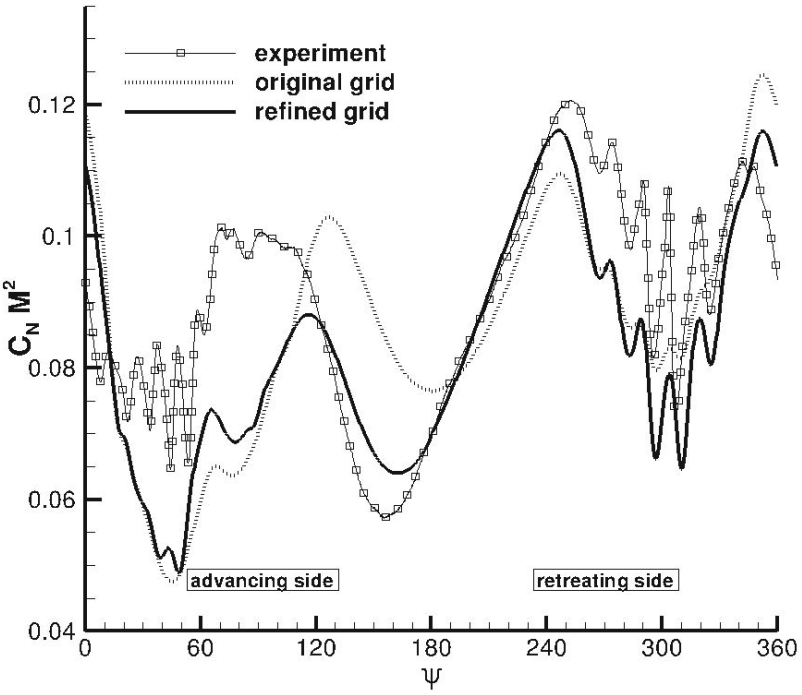


Fig. 10. Vertical force coefficient at 87% span for the simulations with flexible blades. Black line with symbols: experiment; dotted line: original grid; black line: refined grid.

side are influenced by the fuselage, which is not taken into account in the current simulation.

Second, the prediction of BVI is discussed. Comparing the results for the original and refined grid, the increase in vortex resolution on the refined grid is evident from several blade-vortex interactions on the retreating blade. Comparing the refined grid simulation with the experiment, the number of BVI on the retreating side are the same for simulation and experiment. The amplitude of the interactions in the simulation is only half that of the experiment. Comparing with the simulation of Jayaraman in Figure 11 the phase of the BVI are remarkably the same. On the advancing side the current simulation shows almost no evidence of BVI. The simulation of Jayaraman do exhibit BVI on the advancing side. As their mesh has a background resolution of 10% chord, the most plausible explanation of the lack of BVI on the advancing side in the current simulations is lack of resolution.

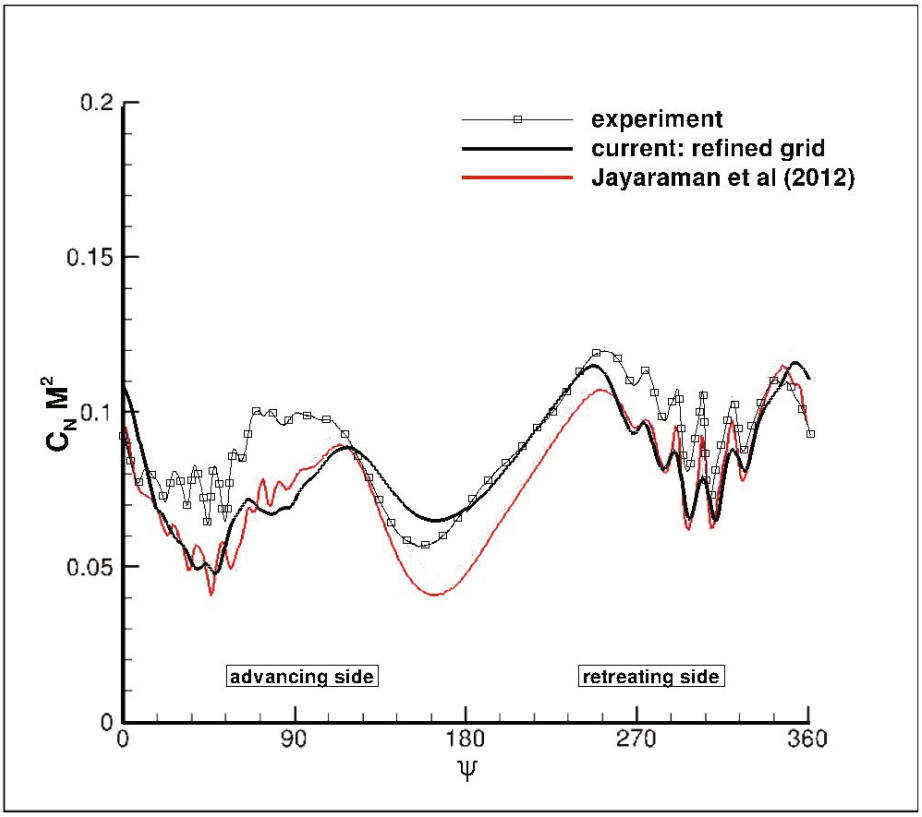


Fig. 11. Vertical force coefficient at 87% span for the simulations with flexible blades. Black line with symbols: experiment; black line: refined grid; red line: results from Jayaraman et al. [5], Figure 6b, fine grid results.

4.5 Conclusions

The uniform resolution of the refined mesh in the rotor wake improves the resolution of the tip vortices. As a consequence, the prediction of the BVI phenomenon on the retreating side has improved, compared to the simulation on the unrefined mesh. On the advancing side the resolution of the mesh is insufficient to capture BVI.

5 Summary

In the article, three aeroelastic testcases results with High-Order CFD solvers analyzed by two different AE Systems have been presented. The LANN wing has been prepared for developed by PUT Aeroelastic System verification while the DLR-F6 is generic aircraft configuration. The last testcase is HART-II rotor aeroelastic simulation prepared by NLR. The all testcases prove readiness of

the developed systems for aeroelastic simulation of full configuration high scale industrial cases.

The presented testcase are the result of development of the necessary software for aeroelastic simulation using high-order curvilinear grids such as parallelized deformation grid tool. Also the tool responsible for interpretation results from Discontinuous Galerkin CFD Solvers. The adaptation tool for high order tools significantly have increased the accuracy of the results.

Developed by PUT in IDIHOM project Aeroelastic System base on loosely-coupled technique. This solution and developed software responsible for exchanging information between independent solvers allow for applying any CFD high order solver.

References

1. Roszak, R., Posadzy, P., Stankiewicz, W., Morzyski, M.: Fluid-structure interaction for large scale complex geometry and non-linear properties of structure. Archives of Mechanics 61(1), 3–27 (2009)
2. Compendium of unsteady aerodynamic measurements. AGARD Report No. 702 Addendum 1 (May 1985)
3. Ruo, S.Y., Malone, J.B., Horsten, J.J., Houwink, R.: The LANN program. An experimental and theoretical study of steady and unsteady transonic airloads on a supercritical wing, AIAA Paper 1983-1686
4. Brodersen, O., Strmer, A.: Drag Prediction of EngineAirframe Interference Effects using Unstructured Navier-Stokes Calculations. In: 19th AIAA Applied Aerodynamics Conference, Anaheim, CA, USA, June 11-14, AIAA Paper 2001-2414 (2001)
5. Jayaraman, B., Wissink, A.M., Lim, J.W., Potsdam, M., Dimanlig, A.C.B.: Helios prediction of blade-vortex interaction and wake of the HART II rotor. AIAA-2012-0714 (2012)
6. Kok, J.C.: Resolving the dependence on freestream values for the k-ω turbulence model. AIAA Journal 38(7), 1292–1294 (2000)
7. Kok, J.C.: A high-order low-dispersion symmetry-preserving finite-volume method for compressible flow on curvilinear grids. J. Comput. Phys. 228, 6811–6832 (2009)
8. Kok, J.C., van der Ven, H.: A high-order finite-volume method with block-structured local grid refinement. This book
9. Smith, M.J., Lim, J.W., van der Wall, B.G., Baeder, J.D., Biedron, R.T., Douglas Boyd Jr., D., Jayaraman, B., Jung, S.N., Min, B.-Y.: An assessment of CFD/CSD prediction state-of-the-art using the HART II international workshop data. Proceedings of the 2012 AHS forum (2012)
10. van der Wall, B.G.: 2nd HCC aeroacoustics rotor test (HART II) Part I: Test documentation. Technical Report IB 111-2003/19, DLR (2003)

Aeroacoustic Test Cases

L. Koloszar[1], N. Villedieu[1], H. Deconinck[1], I.S. Bosnyakov[2], S.V. Mikhaylov[2], A.N. Morozov[2], V.Y. Podaruev[2], A.I. Troshin[2], V.V. Vlasenko[2], A.V. Wolkov[2], P. Eliasson[3], T. Bolemann[4], and F. Chalot[5]

[1] von Karman Institute for Fluid Dynamics, B-1640, Rhode-St-Genese, Belgium
lilla.koloszar@vki.ac.be
[2] Central Institute of aero- and hydrodynamics (TsAGI), Zhukovsky, Russia
andrey.wolkov@tsagi.ru
[3] The Swedish Defense Research Agency (FOI), SE-164 90 Stockholm, Sweden
peter.eliasson@foi.se
[4] University of Stuttgart, Germany
bolemann@iag.uni-stuttgart.de
[5] Dassault Aviation, Cedex 300, 92552 Saint-Cloud Cedex, France
frederic.chalot@dassault-aviation.com

Abstract. In this article, the results of aeroacoustic cases obtained by high-order methods are presented. One internal and one external case are considered. The internal case is a transonic cavity. It has been computed by three parters, namely FOI-LiU (Linköping University), University of Stuttgart (USTUTT) and Dassault Aviation (DASSAV). FOI-LiU has computed with their higher order accurate finite difference solver using a block structured grid. Comparisons are made to reference calculations using an unstructured grid with Edge. USTUTT uses a high order discontinuous Galerkin spectral element code for the final high-order computations and a mixed modal/nodal DG code for the baseline computations. Dassault Aviation uses their in-house stabilized finite element code Aether, both for the reference and the higher-order computations. In both instances unstructured tetrehedral grids are used. The external case is a quasi-two dimensional generic wing and flap configuration defined in the VALIANT FP7 project. Two institutions were involved in the high-order simulation of this low-Mach number test case, $M = 0.15$, Central Institute of Aero- and Hydrodynamics, TsAGI, and the von Karman Institute, VKI. In the following, the acoustic predictions corresponding to the two test cases will be described.

Keywords: CFD, CAA, hybrid method, higher order.

1 Introduction

Flows over an open cavity may arise over the under-carriage wheel well and over an embedded weapon bay during the store release operation of a military aircraft. The acoustic tones for open cavity flows are interpreted as a consequence of the interaction between the shear layer that bridges the cavity and the aft

N. Kroll et al. (eds.), *IDIHOM: A Top-Down Approach*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, DOI: 10.1007/978-3-319-12886-3_29

cavity wall on which the shear layer impinges. A better understanding and prediction capability of this type of flow is of practical importance to improve the design methodology and to avoid possible flow-induced operation problems in aeronautic applications. Three partners consider the test case with turbulent flow over a rectangular cavity. FOI and USTUTT are performing hybrid RANS/LES calculations without a subgrid-scale model. Dassault Aviation is using a zonal DES (ZDES) approach.

During aircraft approach and landing, a significant portion of noise is generated by the high-lift devices. Reducing the noise during landing is very important for the comfort of the residents living nearby an airport. The airframe noise during take-off maneuver is less important, since the slope of the take-off is much steeper, and the engines are in full trust, louder than any other component. Among the various noise sources during approach and landing, the landing gear, leading-edge slat and the side edges of the flap where identified as the main airframe noise sources. The current test case considering a generic wing and flap, more precisely the noise propagation around such geometry. Two partners, namely TsAGI and VKI were contributiong to this industrial demonstrator test case. VKI used the Large-Eddy Simulation performed within the VALIANT project as noise sources while TsAGI injected stochastic perturbations in order to represent the effect of the turbulent flow.

2 Transonic Cavity Case

The test case is denoted Test Case A15 within IDIHOM and is a test case with transonic flow over a rectangular cavity; it is also known as M219 in the literature. This test case is suitable for LES or hybrid RANS/LES calculations due to the turbulent fluctuations over the cavity. Experimental, time dependent data exist on the cavity walls and ceiling and are available for IDIHOM. This test case has been measured and computed in the past with many references available, e.g. [1]– [3]. Several different cavity geometries exist; the one used in IDIHOM is the cavity with 5:1:1 length-to-depth-to-width relation. The geometry as well as the locations of the pressure probes are reproduced here in Figure 1.

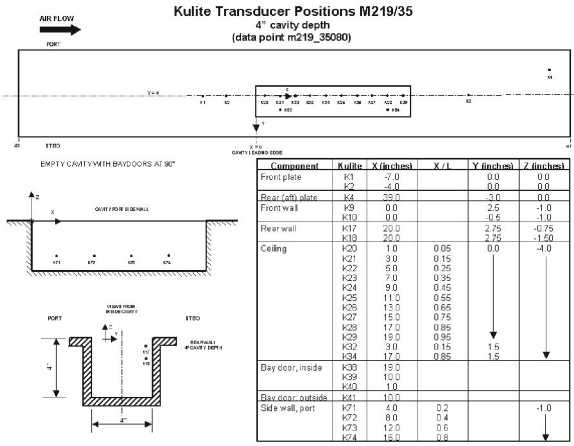
The free stream values are

$$M_\infty = 0.85;\ Re = 6.8 \times 10^6 \tag{1}$$

where the Reynolds number is based on the cavity length (20 inches).

2.1 Computational Domains and Grids

FOI-LiU. For the higher order calculations with the finite difference solver Essense, calculations were carried out on a flat plate where the cavity is embedded on the flat plate. A two block structured grid was created for these calculations



Component	Kulite	X (inches)	X / L	Y (inches)	Z (inches)
Front plate	K1	-7.0		0.0	0.0
	K2	-4.0		0.0	0.0
Rear (aft) plate	K4	33.0		-3.0	0.0
Front wall	K9	0.0		2.5	-1.0
	K10	0.0		-0.5	-1.0
Rear wall	K17	20.0		2.75	-0.75
	K18	20.0		2.75	-1.50
Ceiling	K20	1.0	0.05	0.0	-4.0
	K21	3.0	0.15		
	K22	5.0	0.25		
	K23	7.0	0.35		
	K24	9.0	0.45		
	K25	11.0	0.55		
	K26	13.0	0.65		
	K27	15.0	0.75		
	K28	17.0	0.85		
	K29	19.0	0.95		
	K32	3.0	0.15	1.5	
	K34	17.0	0.85	1.5	
Bay door, inside	K39	19.0			
	K38	10.0			
	K40	1.0			
Bay door, outside	K41	10.0			
Side wall, port	K71	4.0	0.2		-1.0
	K72	8.0	0.4		
	K73	12.0	0.6		
	K74	16.0	0.8		

Fig. 1. Cavity geometry, experimental setup and location of the pressure probes recording unsteady pressure fluctuations

where one block is located inside the cavity. To have a single boundary condition per block side, the block on top of the cavity block was split up resulting in 10 blocks all together. For the reference calculations with Edge [4], FOI-LiU has computed on two different grids, one hybrid unstructured grid that has been generated by EADS and contains a grid over the cavity, the device on which the cavity is integrated and the entire test section of the wind tunnel. The other grid is a structured grid generated by FOI-LiU over the cavity and a limited domain of the flat plate outside of the cavity, very similar to the structured grid. The structured grid is displayed in Figure 2 and the main data are summarized in Table 1.

Table 1. Details of FOI-LiU computational grids for A15 Cavity

Grid	Solver	# vol. nodes	# boundary nodes in cavity	# volume nodes in cavity	Near wall distance
Structured	Essense	2.6×10^6	41×10^3	0.73×10^6	1.2×10^{-5} m
EADS grid	Edge	6.2×10^6	77×10^3	2.0×10^6	4.0×10^{-6} m
FOI grid	Edge	1.2×10^6	16×10^3	0.5×10^6	4.0×10^{-6} m

USTUTT. The mesh used for the cavity computations by the University of Stuttgart is a modified version of the mesh provided by FOI. The mesh contains about 62,000 elements, of which about 50% are placed either inside the cavity or at a distance of less then 0.2D to the cavity. In the flat plate region the effective $y^+, (y^+ = y_1^+/(P+1))$ has been chosen to be 10, considering the schemes sub cell

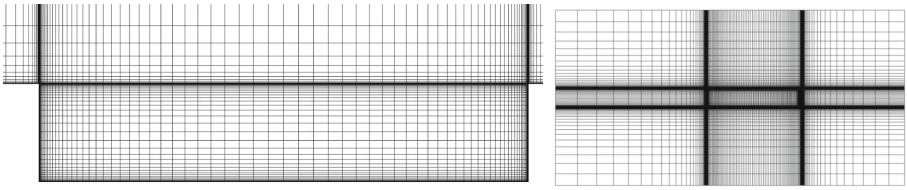


Fig. 2. Mesh of the cavity test case by FOI-LiU using Essense

resolution. Here y_1^+ denotes the height of the first grid cell and P the polynomial degree. The magnitude of y^+ could be confirmed by the computations. The mesh is displayed in Figure 3.

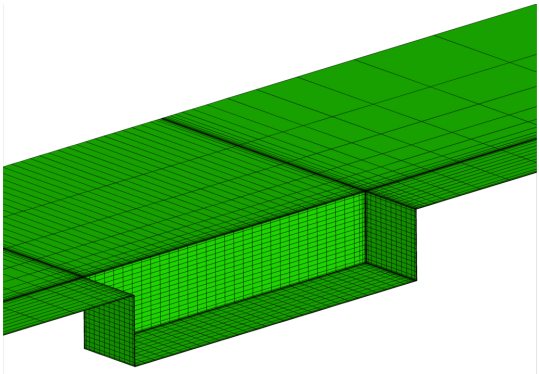


Fig. 3. Cavity mesh used by USTUTT, cut at plane through the right wall

DASSAV. Contrary to the other two contributors, Dassault Aviation has computed the M219 cavity in a configuration close to its original experimental setup, as can be seen in Figure 4. The Reference linear mesh used by Dassault Aviation is an unstructured tetrahedral mesh build on top of a triangular surface mesh with a characteristic edge size of 2 mm. In the boundary layer, the mesh is constructed with a first layer of elements of 10 μm height and a growing ratio of 1.25. In the mixing layer, an unstructured block of tetrahedra with edge lengths od 2 mm in the x and y directions and 2.5 mm in the z direction is inserted. The grid contains slightly less than 3.4 million grid points. Higher order grids are generated by adding degrees of freedom to a coarse P1 grid. The difficulty is to build a "skeleton" grid which is coarse enough to yield higher-order grids with a controlled number of grid points and still preserve the qualities in geometric representation, point density, and element distorsion of the original reference grid. Fortunately for this test case, all surfaces of the model are flat which alleviates the issue of curving the elements in the volume. A discussion about this aspect of higher-order grid generation can be found in the A04 Generic Falcon Test Case in the External Aerodynamic Test Case section [5] and also in Chapter II.2

in the section dedicated to the numerical method developed by DASSAV [6]. With these constraints, the skeleton P1 mesh built on a different surface grid with a characteristic edge size of 4 mm could not be made coarser than about 450,000 grid points; consequently the corresponding P2 grid (and the matching P1 grid) contains 3.5 million grid points. Coarsening techniques developed later on for the A04 Falcon test case could probably have been similarly applied to the cavity to produce yet coarser higher-order grids [5].

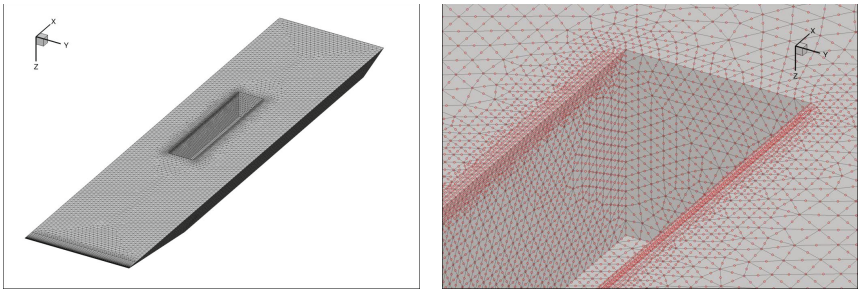


Fig. 4. Configuration and surface view of the P2 mesh used by DASSAV

2.2 Numerical Methods, Reference and Higher-Order Solutions

FOI-LiU. For the higher order accurate calculations FOI-LiU use the finite difference solver Essense that has been further developed and validated within IDIHOM. The higher order code is based on Summation By Parts (SBP) combined with Simultaneous Approximation Term (SAT) approach with penalty terms that guarantee accuracy and stability [7], [8]. The code is able to handle arbitrary order of spatial accuracy, currently limited to 5^{th} order. The code uses explicit time stepping with a 4^{th} order accurate additive Runge-Kutta scheme.

The unsteady calculations were initiated from poorly converged steady state calculation with local time steps. The intention was to use the same hybrid RANS/LES model as in Edge. For this purpose this algebraic model has been implemented in Essense. In the calculations, however, the model imposed a severe restriction on the time step due to small cells away from the wall caused by the H-topology of the employed mesh. This model was therefore abandoned and the calculations presented here have been computed without a model. Only one calculation was completed during IDIHOM. This calculations employs a scheme that is 4^{th} order accurate in the interior, 2^{nd} order accurate at the boundaries making it globally 3^{rd} order accurate. Adiabatic weak wall boundary conditions were used inside the cavity and on the flat plate. Far field characteristic boundary conditions were used elsewhere. The higher order calculation progress for about 60 through flows, the solutions from last 40 through flow are used for the statistics.

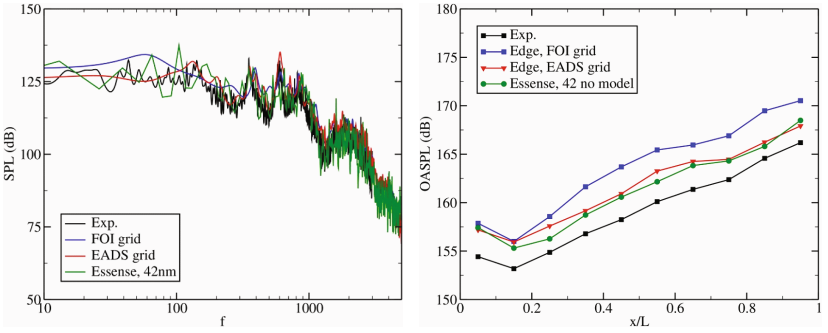
The reference calculations for this test cases were carried out some time ago and have been repeated for IDIHOM. The 2^{nd} order backward difference implicit

method is used in time which is A-stable, i.e. stable for all sizes of physical time step. In dual time multigrid with 3 levels accelerate the convergence. The calculations progress for about 120 through flows, i.e. L/U_∞, where L is the cavity length and U_∞ is the free stream velocity. Some computational parameters are given in Table 2.

Table 2. Sizes of time steps and number of inner iterations for the reference cavity calculations

Grid	Solver	Δt	Δt per T	N inner iterations
Structured	Essense	1.0×10^{-8}	182×10^3	1
EADS grid	Edge	2.0×10^{-5}	91	32
FOI grid	Edge	1.0×10^{-5}	182	40

In Figure 5 the overall the sound pressure level (SPL) and the overall sound pressure level (OASPL) are displayed and compared to experimental values. The higher order results compare reasonably well to the experimental values of SPL. The main tonal peaks are captured. The higher order results have a tendency to have somewhat larger amplitudes of the oscillations compared to the reference results. This may be due to the lack of RANS/LES model for the higher order calculations or possibly a too short and coarse sampling interval. The OASPL is obtained by integrating SPL for all frequencies. The computed OASPL from the higher order scheme and from the reference calculations with the EADS grid agree best with the experimental values, the over prediction of OASPL is common.



(a) SPL at kulits k21, $x/L = 15\%$ (b) OASPL

Fig. 5. Local and Overall Sound Pressure Level by FOI-LiU

USTUTT. USTUTT computed baseline higher order computations for the cavity testcase within the ADIGMA project [9] computed with USTUTTs mixed

modal/nodal hp-adaptive DG code HALO. The calculations were carried out for a different Reynolds number (Re=2×10^5) and have hence been left out here.

The cavity computations have been performed using the high order discontinuous Galerkin spectral element code Flexi developed by the Numerical Research Group (NRG) at the University of Stuttgart [10]. A polynomial degree of $P = 4$ has been chosen for the computations. For the computation a model-free approach has been chosen, however an incomplete polynomial de-aliasing has been applied using over integration. The de-aliasing reduces unphysical oscillations of the solution and thus provides 'clean numerics' and in our case also significantly reduces destabilizing effects caused by the high Reynolds and Mach number. For the overintegration a polynomial degree of $P_{Over} = 6$ has been chosen, which was regarded as a good compromise between speed and accuracy, where full overintegration would require $P_{Over} = 9$. During the computation occasionally occurring shocklets have been observed, therefore a low amount of artificial viscosity has been used to provide additional stability. The amount of artificial viscosity has been controlled by a Persson-type modal pressure indicator. Compared to the baseline results, only 20% of the amount of artificial viscosity was sufficient for this computation.

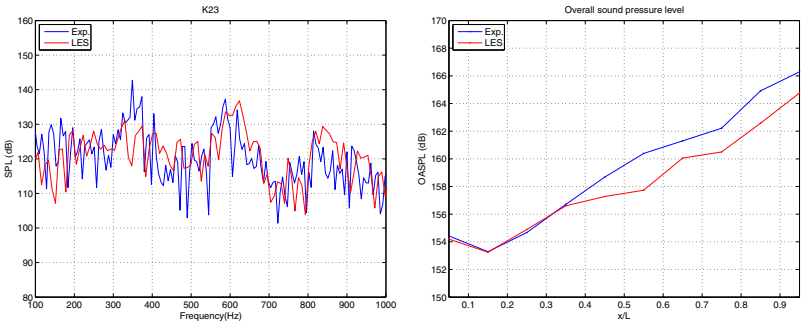
For the time discretization USTUTT has employed a 4^{th} order explicit Runge-Kutta scheme. The average time step of the simulation was typically in the range between $\Delta t = 1.4 \times 10^{-7} - 1.6 \times 10^{-7}$. The data at the Kulites was collected every 4^{th} time step, resulting in a very high sampling rate of about 1.6 Mhz. From $t = 0.0s - 0.035s$ the polynomial degree has been gradually increased from $P = 2$ with an increased amount of artificial viscosity to the final simulation setup with $P = 4$. The collection of the data started at t=0.05s, which equals to 27 through-flows. The total averaging time was 0.11s being about 60 through-flows.

The domain has been chosen to be periodic in spanwise direction. For the flat plate and the cavity itself adiabatic no-slip boundary conditions have been applied. At the inflow and the top characteristic conditions have been used, producing a boundary layer thickness of 0.1D at the cavity leading edge, to have comparable results to the which is identical to the results by Chen et al. [1]. At the outflow characteristic non-reflecting boundary conditions have been applied.

The overall computation contains 7.7×10^6 DOFs, with an additional 13.4×10^6 DOFs used for the de-aliasing. The simulation has been carried out on the Cray XE6 cluster Hermit at HLRS in Stuttgart with on 4096 cores. The number of DOFs/core was about twice the number of which the code reaches its peak efficiency (2000-3000 DOFs/core), more cores could not be used for availability reasons on the cluster.

For the evaluation of our computations we compare the sound pressure level spectra and the overall sound pressure level along the Kulites K20-K29 placed equidistantly in streamwise direction on the cavity ceiling, the results on one of the Kulites is displayed in Figure 6.

The plots of the sound pressure level reveal that all cavity modes are predicted slightly too high in frequency, while the amplitude prediction differs somewhat



(a) SPL at kulits k23, $x/L = 35\%$ (b) OASPL

Fig. 6. Local and Overall Sound Pressure Level by USTUTT

between the Kulites. It is clearly visible, that the 3^{rd} and 4^{th} mode are predicted best by the LES computation, with respect to both frequency and amplitude. Even some of the higher modes are well resolved, which can be best seen at Kulite 23. The predictions for the first mode suffer from the relatively short averaging time compared to the experiment. To fully capture this mode a largely longer averaging time would be required. Nevertheless both amplitude and frequency prediction are in an acceptable region. The SPL for the 2^{nd} mode can be regarded too low for some Kulites, for others the 2^{nd} mode is nearly missing. While the reason for this behaviour is unknown, many simulations from literature suffer from the same problem e.g. [2]. The SPL prediction at Kulites 20-23 at the beginning of the cavity matches the experimental results somewhat better, compared to the Kulites at the end. In an overall view the mode prediction is satisfying, with exception of the 2^{nd} mode.

Table 3. Modal frequency and amplitude at K29, results from USTUTT

Mode	1	2	3	4	5
Rossiter	159 Hz	371 Hz	582 Hz	794 Hz	1005 Hz
Experiment	150 Hz	350 Hz	590 Hz	810 Hz	991 Hz
	142 dB	153 dB	146 dB	135 dB	129 dB
LES-N4	190 Hz	380 Hz	623 Hz	840 Hz	1010 Hz
	140 dB	140 dB	145 dB	137 dB	127 dB

Regarding the overall sound pressure level the computations do in general match the experimental results very well as displayed in Figure 6. Especially in the first 3^{rd} of the cavity the OASPL is nearly identical to the measured results. In the region near $x/L = 0.4$ where the pressure oscillations caused by the shear layer reach their maximum the results start to slightly diverge up to a maximum

difference of 3 dB, which then decreases again to 1-2 dB difference till the end of the cavity.

DASSAV. For all computations, Dassault Aviation used its industrial stabilized finite element code `Aether`. It relies on *continuous* isoparametric Lagrange polynomials of any order computed on unstructured grids as finite element shape and trial function spaces (solutions are C^0 continuous and the same degree of interpolation is used for both the solution variables and the space coordinates). So far, only tetrahedral elements have been implemented in 3-D up to P3. More details about the numerical method and its higher-order implementation can be found in [6].

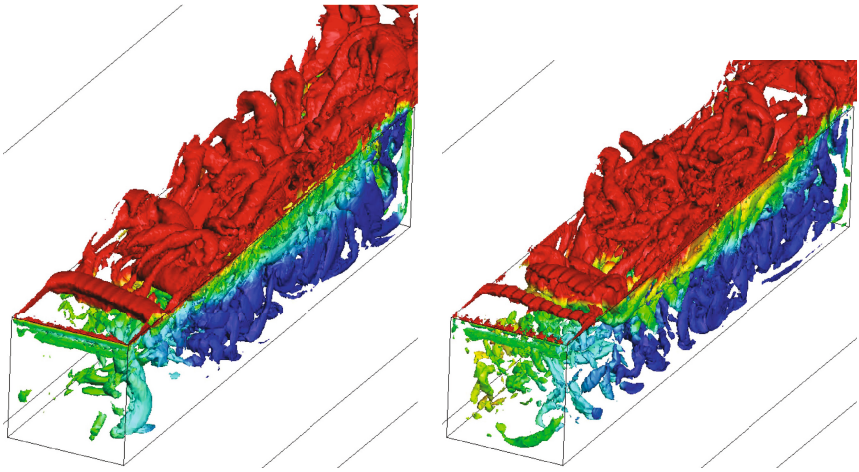


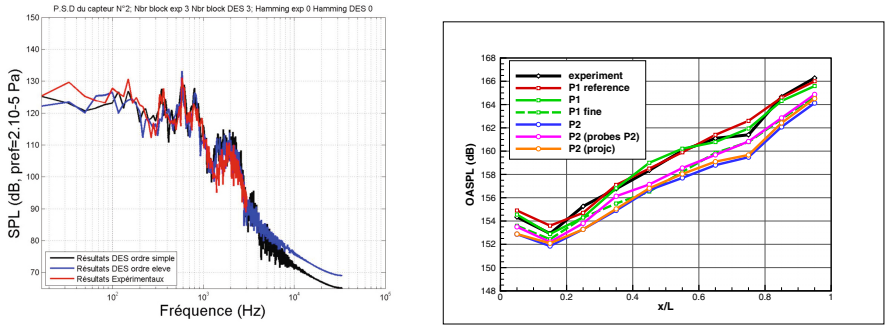
Fig. 7. Turbulent structures in the QinetiQ M219 cavity computed by DASSAV: standard 2nd-order linear P1 elements (left) vs. 3rd-order quadratic P2 elements (right).

For unsteady calculations fully implicit time integration with dual time stepping is performed based on the standard second-order Gear's scheme. The same time step of 1.5×10^{-5} s (that is 121 time steps per through flow over the cavity length) is used for all computations whatever the order of the space integration. Third order calculation may suffer from a time integration which is only second order accurate. Our experience with DES calculations and in particular with this test case tells that time accuracy should be sufficient with such a time step. The effect of a time step reduction is shown in section [6]. Nevertheless, keeping everything equal, including the time integration scheme, enables a fair comparison between 2nd and 3rd order spatial accuracy.

For all simulations, the flowfield is initialized with a steady RANS computations using the Spalart-Allmaras turbulence model. The computation is pursued in a unsteady mode with a Zonal DES approach [11]. After a settling time of 195 ms, unsteady data is acquired for another 225 ms, which accounts for a total

simuation time of 420 ms (i.e., 230 through flows). 186 ms are post-processed to produce the energy spectra and the OASPL curves. For a better comparison, especially at low frequencies, the experimental pressure history is post-processed over the same time interval. All calculations were performed on 1024 cores of an IBM BlueGene/P.

In Figure 7, isovalues of the Q-criterion colored by the Mach number are presented for the 2nd-order linear P1 and the 3rd-order quadratic P2 solutions computed on nested grids and thus containing the same number of degrees of freedom. The higher-order simulation transitions sooner to full 3-D turbulence and contains much finer turbulent structures.

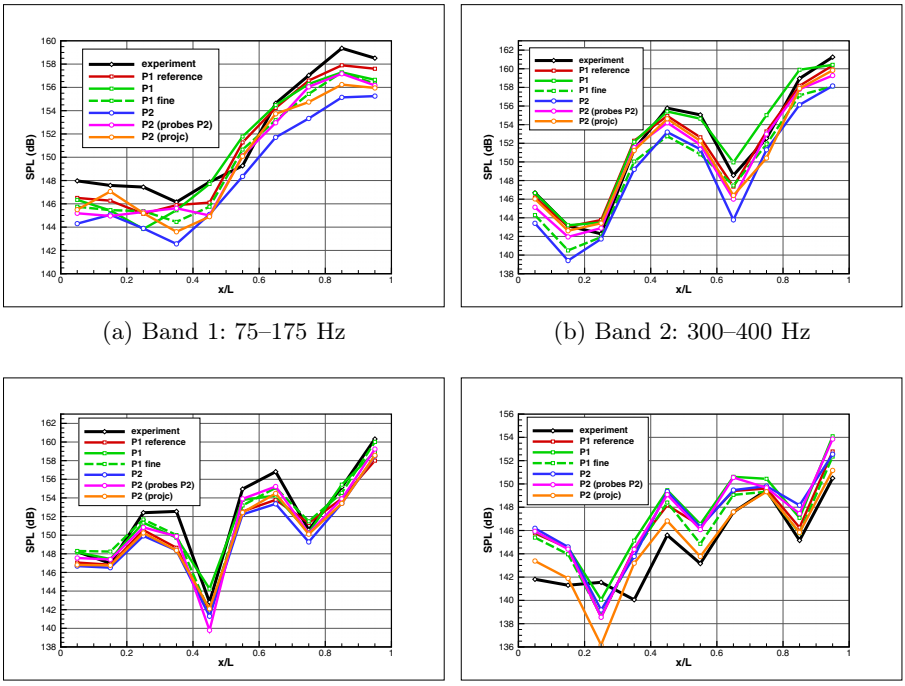


(a) SPL at Kulite K21, $x/L = 15\%$ (b) OASPL

Fig. 8. Local and Overall Sound Pressure Level by DASSAV

A typical SPL frequency spectrum corresponding to Kulite #21 is displayed in Figure 8a. It shows a reasonable agreement between the experiment (in red) and the different simulations both in terms of Rossiter peak locations and amplitude. One must note that the higher-order solution (in blue) exhibits more energy in the higher part of the spectrum.

In Figure 8b, OASPL plots corresponding to different realizations of the simulation are presented. The red and green curves represent respectively the Reference solution and the P1 solution computed with the same degrees of freedom as the higher-order solutions. Both solutions are quite similar and are in good agreement with the experiment shown in black (a few tenth of a dB on the average, see Table 4). Note that the Reference solution corresponds to a longer simulation time of 350 ms. This confirms that the statistics are converged for the newer computations which account for simulations times of 225 ms. The P2 third-order solution is plotted in blue. It shows a drop in OASPL 1 to 2 dB below the experiment. We believe that the more precise P2 computation reveals some of the limitations of the underlying Smagorinsky subgrid scale model in the DES. A finer P1 simulation, shown in a green dashed line, exhibits the same tendency of decreased OASPL's; the corresponding mesh is the same as the P1 and P2 meshes, except in the cavity region where the typical grid size has been



(a) Band 1: 75–175 Hz

(b) Band 2: 300–400 Hz

(c) Band 3: 550–650 Hz

(d) Band 4: 750–850 Hz

Fig. 9. Band-integrated Sound Pressure Levels by DASSAV

reduced from 2 to 1 mm; this mesh contains close to 3.9 million grid points. A more detailed discussion about the effects of mesh refinement, subgrid scale model, time step and scheme order can be found in [6].

Finally, two additional curves are presented in Figure 8b. For the sake of simplicity, earlier higher-order computations were carried out with the same sampling routine as P1 computations, that is the pressure signal was linearly interpolated at the location of the experimental Kulites. The effect of the actual higher-order interpolation is shown in the pink curve. Depending on the pressure tap, an increase of up to 1 dB is observed in OASPL, bringing the higher-order P2 results very close to the refined mesh P1 results. This stresses the importance of postprocessing higher-order solutions with adapted higher-order techniques. This remark is also valid for line and contour plots. The orange curve in Figure 8b represents the effect of the second order Navier-Stokes derivatives in the SUPG/GLS stabilization term of DASSAV residual-based stabilized finite element code. This term is zero for 2nd-order linear solutions and was dropped so-far for higher-order computations. The effet is not so sensitive on OASPL's but is more significant at certain frequencies as can be seen in the band-integrated SPL's in Figure 9. Frequency bands are defined for this test case according to the analysis of Larchevêque [12]. Again, the difference in SPL with respect to

the orginal P2 solution is significant, and the experimental level can even be retrieved for the higher frequency band. Unfortunately, the combination of both higher-order interpolation for pressure probes and of the higher-order term in the stabilization could not be tested during the duration of the project.

The band-integrated sound pressure levels (SPL's) in Figure 9, plotted in the same colors as the OASPL's, show that the main physics is captured in all simulations over the complete range of frequencies. As could be anticipated, higher frequencies are better captured on P1 refined meshes and higher-order solutions with the best numerical ingredients. More on that aspect can be found in [6].

2.3 Assessment of High-Order Solutions

In an attempt to quantify the deviation from the experimental OASPL and to make a cross comparison between partners results we define the normalized deviation D as

$$D = \frac{\parallel O_{CFD} - O_{exp} - \overline{\delta O} \parallel_2}{\overline{O}_{exp}} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_{i,CFD} - O_{i,exp} - \overline{\delta O})^2}}{\overline{O}_{exp}} \tag{2}$$

$$\overline{\delta O} = \frac{1}{N}\sum_{i=1}^{N}(O_{i,CFD} - O_{i,exp}) \tag{3}$$

and where $\overline{O}_{exp} = 159.2dB; N = 10$. The intention with the derived formula for the deviation is to define a measure that gives a zero value if the shape of OASPL is identical to the shape of the experimental OASPL. It allows for a shift in absolute level though.

In Figure 10 the OASPL by FOI-LiU, USTUTT, and DASSAV using HOM are displayed. In Table 4 below we give the deviation for the different calculations. The deviation obtained by FOI-LiU between the higher order and reference results is very similar indicating that the two solutions follow experimental OASPL equally well. The slightly higher deviation obtained by USTUTT is due to the

Table 4. Computed deviation to experimental values

Partner	Solution	D	$\overline{\delta O}$ (dB)
FOI-LiU	Reference EADS grid, 2^{nd} order	3.0×10^{-3}	2.480
FOI-LiU	HOM, 3^{rd} order	3.0×10^{-3}	2.077
USTUTT	Flexi, 5^{th} order	5.4×10^{-3}	-1.027
DASSAV	Aether, Reference, 2^{nd} order	3.1×10^{-3}	0.200
DASSAV	Aether, 2^{nd} order	3.0×10^{-3}	-0.074
DASSAV	Aether, 3^{rd} order	2.7×10^{-3}	-1.936

fact that their shape deviates somewhat more from the shape of the experimental results although there is a very good experimental match upstream in the cavity. In the case of DASSAV, both 2nd-order solutions present similar deviations; the 3rd-order simulation produces however a slightly lower deviation, but at the price of a larger shift (-2 dB), as previously mentioned.

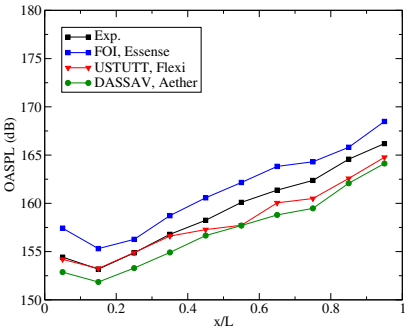


Fig. 10. Overall Sound Pressure Level (OASPL) using higher order methods (HO) by FOI-LiU, USTUTT, and DASSAV on the cavity ceiling (Kulites k20 - k29)

Next is a summary of the computational resources required to obtain the considered solutions. The reference calculation on the FOI grid has been left out. It should be noted that the simulation time is different for the calculations. The higher order simulation by FOI-LiU is obviously more than an order of magnitude more expensive than the FOI-LiU reference calculation. This is mainly due to small time steps and the lack of convergence acceleration in ESSENSE. The higher order calculation by USTUTT is more efficient and comparable to the reference solution by FOI. It is to be noted, that while the effective N_{DOF} is 7.7×10^6, the overall N_{DOF} including overintegration is about three times higher. DASSAV 3rd-order P2 simulations last about 2.5 times as long as a standard 2nd-order P1 simulation (6 days compared to 2.5 days on 1024 cores of an IBM BlueGene/P). Higher-order computations use 10 integration points per tetrahedron, whereas a single point is used for 2nd-order runs. It is quite possible that on this configuration where all elements have uncurved edges, a four-point integration rule would suffice, further reducing the cost of the higher-order simulation.

3 VALIANT Flap Case

The VALIANT flap testcase represents a wing+flap configuration in approach condition. This case is a generic geometry defined in the FP7 VALIANT project. The flap is placed under the wing with a bit of overlap. This configuration was installed in the anechoic wind tunnel of Ecole Central de Lyon and both flow and

Table 5. Computational information about the cavity test case

Partner/ case	Simulation time	N_{DOF}	N_{CPU}	T_{wall}	T_{bench}	$T_{wall} \times N_{CPU}/(T_{bench} \times N_{DOF})$
FOI-LIU HO	0.080 s	2.6×10^6	300	$7.70 \times 10^6 s$	16.0 s	56.00
FOI-LIU Ref.	0.200 s	6.2×10^6	64	$2.44 \times 10^6 s$	16.0 s	1.60
USTUTT HO	0.110 s	7.7×10^6	4096	$1.75 \times 10^5 s$	15.1 s	6.10
DASSAV Ref.	0.225 s	3.5×10^6	1024	$2.12 \times 10^5 s$	131.5 s	0.47
DASSAV HO	0.225 s	3.5×10^6	1024	$5.48 \times 10^5 s$	131.5 s	1.20

acoustic measurements were performed in the framework of VALIANT (VALidation and Improvement of Airframe Noise prediction Tools) FP-7 European project [13]. The main characteristics defining the flow are the free-stream velocity $U_{\inf} = 51 m/s$, which corresponds to approach condition with $Ma = 0.15$. The resulting Reynolds number, $Re = 1.36 \times 10^6$, is approximately one order of magnitude less than the Reynolds numbers corresponds to a real aircraft wing during approach. Roughness elements (sandpaper ISO P150) were placed on both sides of the wing in order to trigger an established incoming turbulent boundary layer. The experimental database obtained by ECL includes the following data: time-dependent microphone signals, time-dependent wall pressure signals, time-dependent series of the velocities acquired from hot wire measurements [14].

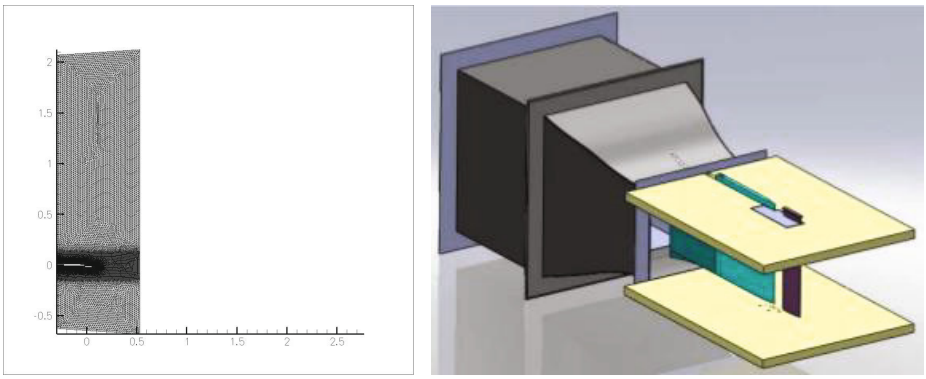
3.1 Computational Domains and Grids

As a first step towards the characterization of the noise propagation from this generic wing flap configuration, the computational domain of the LES simulation was extended for the LEE simulation. Acoustic measurement data are available 2 meters above the wing trailing edge, so this point was considered in the LEE domain to be resolved. The boundary of the computational domain, therefore, was truncated a bit further above this measurement point in order to limit the reflections arising due to this artificial domain restriction. The computational domain can be seen in Figure 11. This 2D mesh consists of 36k triangular elements, heavily clustered in the noise production region in order to have a good representation of the reconstructed source terms transferred from the LES simulation.

The maximum mesh size was determined based on the maximum frequency resolution we intended to achieve. Previous investigations showed that with P1 representation, the LDA scheme needs 10 points per wavelength. The maximum frequency resolution we can have from the LES source data can be computed as:

$$f = 1/\Delta t_{LES} = 1/10^{-6} = 10^3 kHz \tag{4}$$

Due to the space-time discretization, we need ten points in time, as well, to resolve the waves in time, giving the maximum assessable frequency of



(a) Computational domain of VKI (b) Measurement setup

Fig. 11. Quasi-2D wing and flap configuration

$f_{max} = 10^2 kHz$. This frequency is still far above the maximum frequency of the measurements ($20kHz$), so we decided to limit the maximum frequency resolution to this value, giving the maximum cell size:

$$\Delta x = \frac{\lambda}{100} = 0.1\frac{c}{f} = 2mm \tag{5}$$

TsAGI performed calculations of this test as well in 2D (flat) formulation. 2D basis functions were used in DG method. Initially, RANS calculation has been performed on appropriate grid with detailed resolution of turbulent boundary layer (for uniform inflow). The converged flowfield, obtained in RANS calculation, have been used as basic (aerodynamic) flowfield.

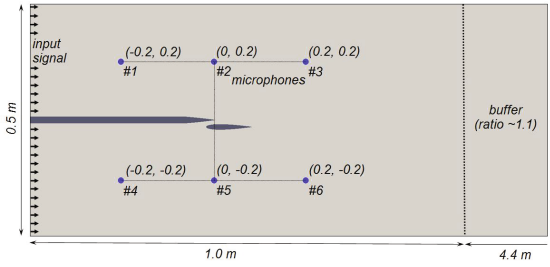


Fig. 12. Geometry of computational domain and position of control points for test A14 by TsAGI

Acoustic calculations have been performed using DG method $K = 3$ for four quasi-uniform grids with quasi-quadratic cells, with 4, 6, 9 and 14 cells per the shortest wavelength in the incoming sound. TsAGI used a smaller domain in order to be able to perform parametric studies and grid convergence.

3.2 Numerical Methods and Baseline Solutions

The von Karman Institute used their second- and third-order Residual Distribution solver to deal with this noise propagation problem. The Residual Distribution or Fluctuation Splitting Method is somewhere between the Finite Element and Finite Volume Methods. The idea was introduced in 1982 by P.L. Roe and later extended for the solution of conservation laws on unstructured meshes. In the last decades several multidimensional upwind schemes have been developed, and proved to be accurate and robust. A new strategy in the computation of the residuals has been designed involving contour integrations and leading to conservative discretization even for problems where it is not possible to linearize the system of equations. Thanks to this improvement the application of RDS to high order discretisation was possible. In our case, since we want to distribute to the downwind nodes it is necessary to split the high order elements in linear elements where we know how to use a multidimensional upwind distribution. Several methods have been developed to solve unsteady problems. In this work we used the space-time method, where the time is considered as a third dimension yielding a space-time element and was first presented by Ricchiuto et al. [15]

In the present work we consider the Linearized Euler Equations (LEE) in two spatial directions, as derived by Bailly et al. for inhomogeneous mean flow, written in conservative variables. At the boundaries non-reflective boundary conditions are used. This boundary treatment relies on the characteristic theory. It is well known that the number of physical conditions which has to be prescribed at the boundary depends on the sign of the eigenvalues of the characteristic system. Only the information coming from outside has to be imposed, all the others are provided naturally by the inner domain. In case of acoustic problems the first difficulty arising is to know what kind of conditions should be fixed. In the present Linearized Euler Equations, the solution variables are the fluctuations of density, velocity and pressure. At an inflow boundary condition three of them should be given, for an outflow just one. But none of the conservative variables are known, in general. It is much easier acting on the waves themselves.

In order to perform noise propagation simulation with the help of Linearized Euler Equations, the background flow (time-averaged) needs to be provided to the solver. VKI was using the time-averaged mean flow provided by the LES, extrapolated to the regions, what the LES domain do not cover. The non-uniform mean flow expected to effect the noise propagation, therefore its correct representation is an important step of the current procedure. The noise production calculation of the generic wing and flap configuration was set up and validated within the FP7 VALIANT project. This simulation was continued to collect equivalent noise sources for the LEE solver used for noise propagation simulation. The noise sources produced by the open source OpenFOAM solver were collected for a time span of $T = 0.04s$ and saved to be able to transfer them to our in house code COOLFluiD. In order to reduce the data amount of these volumetric sources, only the mid-span averaged data was extracted in every time-step.

Based on the incompressible flow quantities, equivalent acoustic sources were reconstructed as:

$$\mathbf{S} = \begin{bmatrix} 0 \\ S_i - \bar{S}_i \\ 0 \end{bmatrix} \tag{6}$$

where $S_i = -\frac{\partial \rho_0 u_i' u_j'}{\partial x_j}$. These source terms were directly used as volumetric source terms, but the source domain was truncated to a disk of 10cm around the wing trailing edge, where the dominant source terms are located, in order to reduce the numerical noise due to interpolation of the sources from the fine LES mesh to the coarser LEE mesh.

In the previously discussed manner the reconstructed sources from the LES simulation were introduced in the LEE simulation. A total simulation time of $T = 0.04s$ was covered by the propagation simulation limiting the statistically converged low frequency resolution to $f = 6kHz$. The effect of the non uniform flow is clearly visible in Figure 13, where the instantaneous divergence of velocity is plotted.

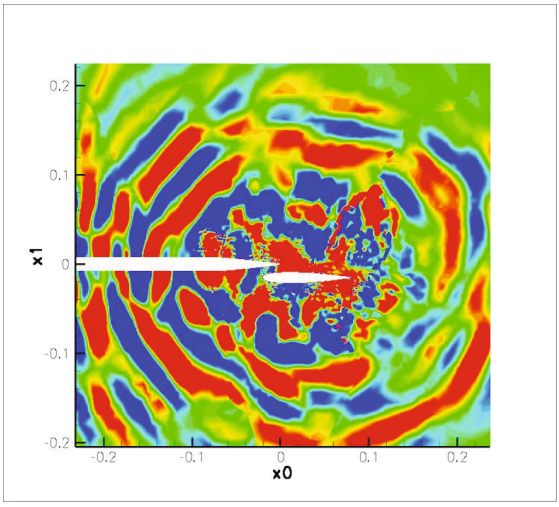


Fig. 13. Contours of divergence of velocity indicating the noise propagation pattern

In the present simulation, only spanwise averaged data were stored and a 2D propagation simulation was performed. According to Manoha *et al.* [16] only the zeroth spanwise wavenumber of the spanwise, Fourier-transformed source needs to be considered. Based on this observation Ewert *et al.* [17] used a reduced 2D source in a pure 2D acoustic simulation to correct the sound pressure levels from 2D to 3D. In the present case the method suggested by Ewert *et al.* is followed.

The integral length scale of the acoustic source in the spanwise direction is small compared to the acoustic wavelength for low Mach numbers, so the source

is compact in this direction. Therefore, an acoustic simulation with a spanwise extension L_z and periodic boundary conditions is equivalent to a 2D acoustic simulation with a spanwise averaged acoustic source [17]:

$$\widehat{s_x} = \frac{1}{L_z} \int_{-L_z/2}^{L_z/2} s_x dz \tag{7}$$

If these averaged source terms are used in a 2D acoustic simulation, the sound-pressure correction form 2D to 3D becomes:

$$\widehat{p}(0, R, \theta, \omega) \backsimeq \widehat{p}(R, \theta, \omega) \frac{1 - \imath}{2} \sqrt{\frac{k\Delta^2}{\pi R}} \tag{8}$$

where $\widehat{p}(0, R, \theta, \omega)$ denotes the 3D, $\widehat{p}(R, \theta, \omega)$ the 2D , frequency-related Fourier transform of the sound pressure, ω is the angular frequency and $k = \omega / c_0$ is the wavenumber. This correction affects the final SPL distribution, but has no impact on the θ-dependent directivity. The SPL correction based on Equation 8 is:

$$SPL_{3D,L_z} = SPL_{2D} + 10log\left(\frac{f\Delta^2}{Rc_0}\right) \tag{9}$$

So, the 2D sound-pressure spectrum is shifted 3 dB/octave towards the higher frequencies by the correction. The SPL_{3D,L_z} is the sound radiated by the slice of airfoil simulated by LES. For a finite spanwise extension L_{span} an additional correction is needed:

$$SPL_{3D,L_{span}} = SPL_{3D,L_z} + 10log\left(\frac{L_{span}}{\Delta}\right) \tag{10}$$

This correction is based on the assumption that all slices of L_z along the wingspan are uncorrelated, whereas the spanwise extension is small compared to the distance between the source and the listener (R).

There are several approaches for solving acoustic tasks. The approach, which was considered by TsAGI, is known as "perturbation method" [18]. It requires relatively not much computational costs and consists of three general steps. At the first step a basic aerodynamic flowfield is calculated independently using Reynolds averaged Navier-Stokes equations. Then near acoustic field is simulated directly as generation and propagation of small acoustic perturbations over the aerodynamic field. This direct numerical simulation of sound is performed on the basis of linearized Euler equations. And finally far acoustic field is estimated using special methods (Kirchhoff or Ffowcs-Williams Hawkings methods), which consider radiation of sound by control surface over practically uniform aerodynamic field. Acoustic characteristics of the control surface are taken from the calculation of near acoustic field. TsAGI work within IDIHOM project was concentrated on direct simulation of sound propagation in near acoustic field. Theoretically, it is possible to neglect the influence of viscosity on sound propagation and to consider this process as adiabatic. To minimize essentially the quantity of arithmetic operations, it was decided to calculate sound propagation on the basis

of Isentropic Linearized Euler Equations (ILEE). This equation system may be obtained from full linearized Euler equations, if we replace the last differential equation (for energy) by isentropic relation $p' = c_a^2 \rho'$, where c is speed of sound and index a corresponds to basic (aerodynamic) flow.

ILEE equation system may be represented in the following form:

$$\frac{\partial U'}{\partial t} + \frac{\partial F_i'\left(U', U_a\right)}{\partial x_i} = 0,$$

where

$$U' = \begin{pmatrix} U_0' \\ U_1' \\ U_2' \\ U_3' \end{pmatrix} = \begin{pmatrix} \rho' \\ \rho_a u' + \rho' u_a \\ \rho_a v' + \rho' v_a \\ \rho_a w' + \rho' w_a \end{pmatrix}, \quad F_i'\left(U', U_a\right) = \begin{pmatrix} U_i' \\ U_1' u_{ia} + \rho_a u_a u_i' + c_a^2 U_0' \delta_{i1} \\ U_2' u_{ia} + \rho_a v_a u_i' + c_a^2 U_0' \delta_{i2} \\ U_3' u_{ia} + \rho_a w_a u_i' + c_a^2 U_0' \delta_{i3} \end{pmatrix}.$$

Here Cartesian coordinates $x_1 = x$, $x_2 = y$, $x_3 = z$; velocity components of basic (aerodynamic) flow $u_{1a} = u_a$, $u_{2a} = v_a$, $u_{3a} = w_a$; velocity perturbations $u_1' = u'$, $u_2' = v'$, $u_3' = w'$.

To construct DG method, the vector of primitive variables $Q' = [\rho'; u'; v'; w']^{\mathrm{T}}$ in each cell of computational grid is represented as a linear combination of local polynomial basis functions $\varphi_j(\mathbf{x})$:

$$Q' = \sum_{j=1}^{K_f} q_j(t) \varphi_j(\mathbf{x}),$$

Polynomial basis functions with maximal degree K provide (theoretically) the accuracy order $K + 1$ in space.

Coefficients of this expansion, $q_j(t)$, are the main unknown values in DG method. Equation system for the determination of $q_j(t)$ may be represented as follows:

$$\sum_{j=1}^{K_f} \left(\int_{\Omega} \Gamma \varphi_i(\mathbf{x}) \varphi_j(\mathbf{x}) d\Omega \right) \frac{dq_j}{dt} + \oint_{\Sigma} F_k' n_k \varphi_i(\mathbf{x}) d\Sigma = \int_{\Omega} F_k' \frac{\partial \varphi_i}{\partial x_k} d\Omega, \quad i = 1, \ldots, K_f.$$

Here Ω is volume of computational grid, Σ is its surface, $\mathbf{n} = [n_1; n_2; n_3]^{\mathrm{T}}$ is unit outer normal vector to the surface element $d\Sigma$, $\Gamma = \partial U'/\partial Q'$. This set of ordinary differential equations is solved using explicit four-stage Runge-Kutta method that can be described by Butcher tableau

Fluxes at the cell faces F_k' are calculated using Roe linearized solution of Riemann problem about the decay of discontinuity between two flows adjoining to current point of cell surface. Approximate Roe matrix is used; it coincides with Jacobian of ILEE, computed using arithmetic averages of aerodynamic flows from two sides of the cell face.

Numerical experiments with 1D propagation of sine wave show that DG method with $K = 3$ allows to calculate propagation of sound waves on grids with only 3 cells per wavelength.

0	
1/4	1
1/3	0 1
1/2	0 0 1
	0 0 0 1

3.3 Assessment of High-Order Solutions

The final comparison with the measurements in the measurement location 2m above the wing trailing edge can be seen in Figure 4. Both P1 and P2 results are represented in this graph. The comparison shows a good agreement both for P1 and P2 discretization. In our RDS code the P2 discretozation was not giving significant improvement, but increased the simulation time. Thanks to the good performance of the P1 element (10 point/wavelength) the P2 discretization do not have a lot of room of improvement. Though, it must be stressed, that the P2 discretization can catch the wave propagation speed a bit better, since it introduces less dispersion error, than the P1 element.

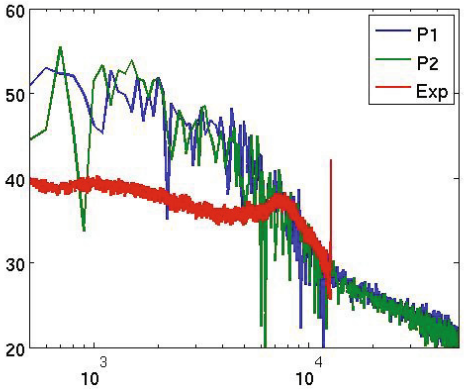


Fig. 14. LEE simulation compared with experimental data

In the main series of calculations by TsAGI, stochastic perturbations were continuously introduced from the inflow boundary. Stochastic perturbations with nearly uniform flat distribution (in 1/3rd octave presentation) within the frequency range 100-12000 Hz were used. Instant field of pressure perturbations in the time moment $t = 0.00615s$ is shown in Figure 15. One may see that incoming sound waves propagate (practically without damping) along the computational domain. Their diffraction and reflection from the flap tip may be seen.

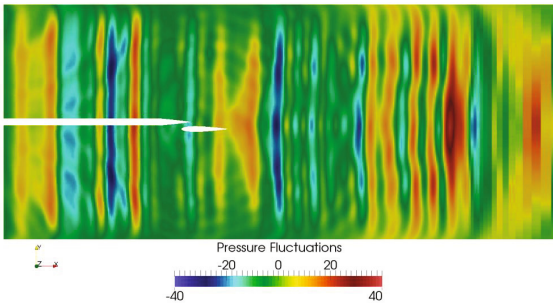


Fig. 15. Instant field of pressure fluctuations in the case of stochastic perturbations at the inflow

After that, an attempt has been made to estimate the convergence order using the same procedure as in test with 1D propagation of a sine wave. Dependencies were registered in six control points that are shown in Figure 12. RMS value of pressure perturbations was determined. Using sequences of for 3 grids, convergence order has been estimated. But this attempt has appeared to be unsuccessful. In some points the convergence order was undetermined, in other points it was far from the expected values, and in different points these values were very different.

To understand this result, additional series of calculation have been performed. Now the flat sine waves of maximal frequency (12000 Hz) were introduced continuously through the left boundary. Maximal frequency was chosen to have the maximal approximation errors and to avoid the influence of truncation errors on the results of Richardson extrapolation. Instant field of pressure perturbations in the same time moment is shown in Figure 16. Attempt to estimate the convergence order has been performed again. And it appeared to be unsuccessful, too. Analysis of Figure 6 allows to assume that the reason of the unsuccess in the convergence order determination is the interference of the main flat waves (propagating to the left) with the waves, reflected from the flap tip and from the upper or lower boundaries of computational domain. Even if all

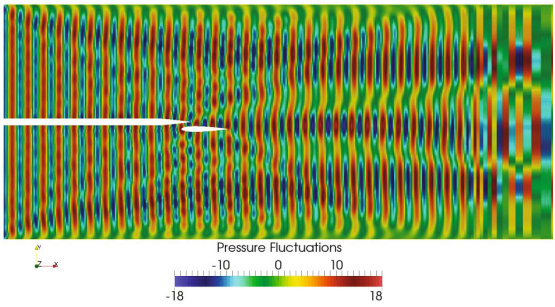


Fig. 16. Instant field of pressure fluctuations in the case of monochrome sinusoidal perturbations at the inflow

these waves have the same frequency, the behavior of signal in control points depends not only upon the grid resolution but also upon the phase shift between the interfering waves. It is important to note that in a test with 1D propagation of the same sine wave on comparable grids the convergence order is determined successfully; it appears to be close to the expected value −4.

4 Conclusion

For the cavity test case higher order accurate solutions over the cavity have been provided by FOI-LIU, USTUTT, and DASSAV. The quality of the solutions is good in terms of SPL and OASPL with similar quality as obtained by a reference solution with about the same number of degrees of freedom. The flow physics are well captured in both the reference and higher order accurate solutions. The computational costs for the higher order results are somewhat higher than those of the reference solution. Further calculations and comparisons on somewhat coarser grids would be needed to judge if the higher order computations pay off compared to a reference solution.

The baseline simulation of the VALIANT testcase showed the potential of hybrid LES/LEE method for noise propagation. The procedure starts with a high fidelity Large-Eddy Simulation in the noise propagation region. After statistical convergence is reached, instantaneous volumetric/spanwise averaged noise sources can be extracted. These sources then introduced to the Linearized Euler Equation solver in order to simulate noise propagation for larger distances. With a high fidelity computation this would be impossible due to the associated high computational cost.

The bottleneck of the procedure is the data transfer between the two solvers. This requires a huge amount of data to be written by the LES solver and read by the LEE solver. Even in the case of the 3D LES to 2D LEE it meant several Gb storage and file I/O. In order to overcome this limitation, it is recommended that the two set of equations are solved simultaneously and the source terms are transferred directly between the solvers.

It was shown that in case of our Residual Distribution solver, the P1 and P2 discretization gave the same quality in terms of noise prediction. For this study two meshes with approximately the same degree of freedom was generated. Though, the P2 simulation took 6 times longer than the P1 (48 CPU hour / 0.001s simulation time for the P1). As a onclusion, the higher order method we are using clearly has higher accuracy for academic cases, but is not justifiable for industrial usage.

Explicit high-order Discontinuous Galerkin method (K=2-3) is implemented for solution of unsteady Isentropic Linearized Euler equations for solution of aeroacoustical tasks by TsAGI. Method with K=3 allows to calculate propagation of sound waves on grids with only 3 cells per wavelength. Grid convergence order of the method is close to the expected value , if tasks without interference of waves with different frequency and phase shift are considered. In task with interference of waves, standard approach to determination of the convergence order seems to be inapplicable.

References

1. Chen, X., Sandham, N.D., Zhang, X.: Cavity Flow Noise Predictions, Report No. AFM-07/05, School of Engineering Sciences, University of Southampton (2007)
2. Ashworth, R.M.: Prediction of acoustic resonance phenomena for weapon bays using detached eddy simulation. Aeronautical Journal, 631–638 (2005)
3. Peng, S.-H.: Simulation of Flow Past a Rectangular Open Cavity using DES and unsteady RANS, AIAA Paper 2006-2827 (2006)
4. Eliasson, P., Weinerfelt, P.: Recent Applications of the Flow Solver Edge. In: Proceedings to 7th Asian CFD Conference, Bangalore, India (2007)
5. Sørensen, K.A., Johnston, C., Leicht, T., Chalot, F., Eliasson, P., Bassi, F., Couaillier, V., Kessler, M.: External Aerodynamic Test Cases. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach. NNFM, vol. 128, pp. 559–614. Springer, Heidelberg (2015)
6. Chalot, F., Dagrau, F., Mallet, M., Normand, P.E., Yser, P.: Higher-order RANS and DES in an industrial stabilized finite element code. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach. NNFM, vol. 128, pp. 495–525. Springer, Heidelberg (2015)
7. Nordström, J., Gong, J., van der Weide, E., Svärd, M.: A Stable and Conservative High Order Multi-block Method for the Compressible Navier-Stokes Equations. Journal of Computational Physics 228, 9020–9035 (2009)
8. Berg, J., Nordström, J.: Stable Robin Solid Wall Boundary Conditions for the Navier-Stokes Equations. Journal of Computational Physics 230, 7519–7532 (2011)
9. Taube, A., Munz, C.D.: Test Case Report CTC 1 Cavity. ADIGMA project report (2009)
10. Hindenlang, F., Gassner, G., Altmann, C., Beck, A., Staudenmaier, M., Munz, C.-D.: Explicit discontinuous Galerkin methods for unsteady problems. Computers & Fluids 61, 86–93 (2012)
11. Deck, S.: Recent improvements in the Zonal Detached Eddy Simulation (ZDES) formulation. Theor. Comput. Fluid. Dyn. 26, 523–550 (2012)
12. Larchevêque, L., Sagaut, P., Lê, T.-H., Comte, P.: Large-Eddy Simulation of a compressible flow in a three-dimensional open cavity at high Reynolds number. Journal of Fluid Mechanics 516, 265–301 (2004)
13. `http://www.cimne.com/websasp/valiant/`
14. Lemoine, B., Roger, M., Legriffon, I.: Aeroacoustics of a Model Non-Lifting Wing-Flap System in a Parallel Flow. In: 17th AIAA/CEAS Aeroacoustics Conference, Portland OR, June 6-8, pp. 2011–2735. AIAA paper 2011-2735 (2011)
15. Ricchiuto, M., Csik, A., Deconinck, H.: Residual distribution for general time dependent conservation laws. J. Comput. Phys. 209, 249–289 (2005)
16. Manoha, E., Delahay, C., Ben Khelil, S., Guillen, P., Sagaut, P., Mary, Y.: Numerical prediction of the unsteady flow and radiated noise from a 3d lifting airfoil. AIAA 2001-2133 (2001)
17. Ewert, R.: Acoustic perturbation equations based on flow decomposition via source filtering. J. of Comp. Physics 188 (2003)
18. Vlasenko, V., Bosniakov, S., Mikhailov, S., Morozov, A., Troshin, A.: Computational approach for investigation of thrust and acoustic performances of present-day nozzles. In: Progress in Aerospace Sciences, vol. 46, pp. 141–197 (2010)

Author Index