Universitext

James F. Blowey
Alan W. Craig
(Eds.)

Frontiers of Numerical Analysis

Durham 2004

Springer

James F. Blowey
Department of Mathematical Sciences
University of Durham
South Road
DH1 3LE Durham
United Kingdom
E-mail: j.f.blowey@durham.ac.uk

Alan W. Craig
Department of Mathematical Sciences
University of Durham
South Road
DH1 3LE Durham
United Kingdom
E-mail: alan.craig@durham.ac.uk

Mathematics Subject Classification: 35-XX, 65-XX, 70-08

Library of Congress Control Number: 2005927222

ISBN-10 3-540-23921-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-23921-5 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springeronline.com
© Springer-Verlag Berlin Heidelberg 2005
Printed in The Netherlands

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: by the authors and TechBooks using a Springer LaTeX macro package

Cover design: Cover design: *design & production* GmbH, Heidelberg

Printed on acid-free paper SPIN: 11357292 46/TechBooks 5 4 3 2 1 0

Preface

The Eleventh LMS-EPSRC Computational Mathematics and Scientific Computing Summer School was held at the University of Durham, UK, from the 4th of July to the 9th of July 2004. This was the third of these schools to be held in Durham, having previously been hosted by the University of Lancaster and the University of Leicester. The purpose of the summer school was to present high quality instructional courses on topics at the forefront of computational mathematics and scientific computing research to postgraduate students. The main speakers were Emmanuel Candes, Markus Melenk, Joe Monaghan and Alex Schweitzer.

This volume presents written contributions three of our speakers which are more comprehensive versions of the high quality lecture notes which were distributed to participants during the meeting. We are also extremely pleased that Angela Kunoth was able to make an additional contribution from the ill-fated first week.

At the time of writing it is now more than two years since we first contacted the guest speakers and during that period they have given significant portions of their time to making the summer school, and this volume, a success. We would like to thank all of them for the care which they took in the preparation and delivery of their material.

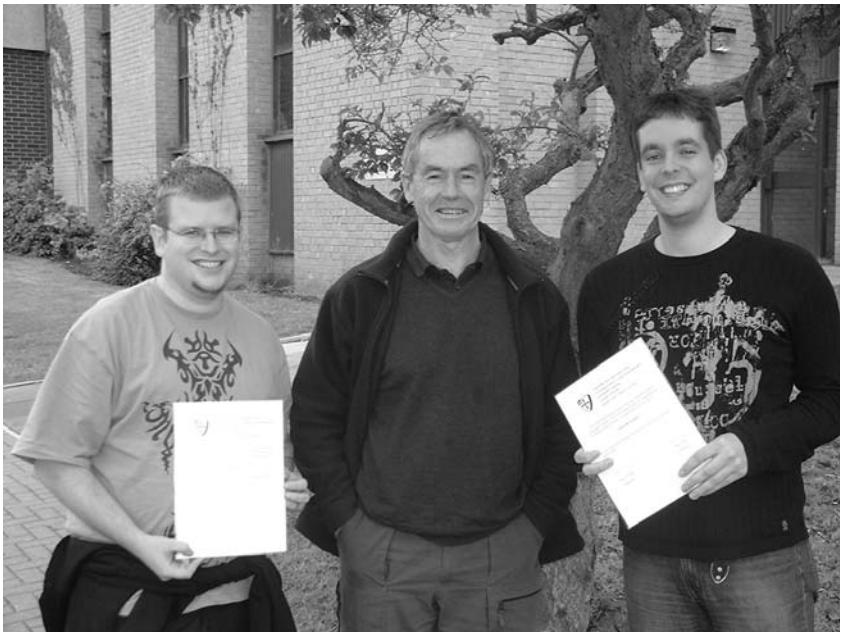
Instrumental to the school were the two tutors who ran a very successful tutorial programme (Peter Johnson & Angela Mihai). There was also a successful programme of contributed talks from eleven students in the afternoons. The UKIE section of SIAM contributed prizes for the best talks given by graduate students. The invited speakers took on the bulk of the task of judging these talks. After careful and difficult consideration, and after canvassing opinion from other academics present, the prizes were awarded to Patrick Lechner (Bath University) and Richard Welford (Sussex University), see figure below. The general quality of the student presentations was impressively high promising a vibrant future for the subject.

The audience covered a broad spectrum, thirty-seven participants ranging from research students to academics. As always, one of the most important aspects of the summer school was providing a forum for numerical analysts, both young and old, to meet for an extended period and exchange ideas.

We would also like to thank the Grey College for their hospitality in hosting the participants, the Durham postgraduates who together with those who had attended the previous Summer School ran the social programme, Rachel Duke, Fiona Giblin and Mary Bell for their secretarial support and our families for supporting our efforts.

We thank the LMS and the Engineering and Physical Sciences Research Council for their financial support which covered all the costs of the main speakers, tutors, plus the accommodation costs of the participants.

James F. Blowey and Alan W. Craig
Durham, April 2005



Patrick Lechner & Richard Welford being presented with the UKIE section of SIAM prizes for the best talks given by graduate students by Joe Monaghan.

Contents

Preface V
Contents VII

Wavelet Methods for Stationary PDEs and PDE-Constrained Control Problems 1
Angela Kunoth
1 Introduction 1
2 Problem Classes 4
2.1 An Abstract Operator Equation 4
2.2 Elliptic Boundary Value Problems 5
2.3 Saddle Point Problems Involving Boundary Conditions 7
2.4 PDE-Constrained Control Problems: Distributed Control 10
2.5 PDE-Constrained Control Problems: Dirichlet Boundary Control 12
3 Wavelets 13
3.1 Basic Properties 13
3.2 Norm Equivalences and Riesz Maps 16
3.3 Representation of Operators 18
3.4 Multiscale Decomposition of Function Spaces 19
4 Problems in Wavelet Coordinates 33
4.1 Elliptic Boundary Value Problems 33
4.2 Saddle Point Problems Involving Boundary Conditions 35
4.3 Control Problems: Distributed Control 37
4.4 Control Problems: Dirichlet Boundary Control 42
5 Iterative Solution 44
5.1 Finite Systems on Uniform Grids 44
5.2 Adaptive Schemes 51
References 60

On Approximation in Meshless Methods 65
Jens Markus Melenk
1 Introduction 65
1.1 Notation 66
1.2 The Notion of Optimality 68
2 Polynomial Reproducing Systems 69
2.1 Motivation 69
2.2 Approximation Properties of Systems Reproducing Polynomials 70
2.3 Construction of Shape Functions with the Moving Least Squares Procedure 76
2.4 Bibliographical Remarks 87
3 Approximation Properties of Radial Basis Functions 87
3.1 Analysis of a Class of RBFs 88

3.2 Bibliographical Remarks ... 92
4 Partition of Unity Method and Generalized FEM ... 93
4.1 Approximation Theory ... 93
4.2 Example: Polynomial Local Approximation Spaces ... 95
5 Examples of Operator Adapted Approximation Spaces ... 96
5.1 A One-Dimensional Example ... 97
5.2 Laplace's Equation ... 100
5.3 Helmholtz Equation ... 102
5.4 Linear Elasticity ... 103
5.5 Further Examples ... 105
5.6 Local Approximation Spaces Obtained Numerically ... 105
5.7 Bibliographical Remarks ... 106
6 Augmenting Classical FEM Spaces ... 106
6.1 Singular Functions ... 106
6.2 Crack Propagation Problems ... 108
6.3 Further Examples: The Generalized FEM ... 111
6.4 Bibliographical Remarks ... 111
7 Enforcement of Essential Boundary Conditions ... 111
7.1 Conforming Methods ... 112
7.2 Non-Conforming Methods: Lagrange Multiplier Methods and Collocation Techniques ... 115
7.3 Non-Conforming Methods: Penalty Method ... 116
7.4 Non-Conforming Methods: Nitsche's Method ... 119
A Results from Analysis ... 122
B Properties of Polynomials ... 123
C Approximation with Adapted Function Systems ... 126
C.1 The Theory of Bergman and Vekua ... 126
C.2 Proof of Theorems 5.3, 5.4 ... 127
C.3 Two-Dimensional Elasticity ... 130
References ... 136

Theory and Applications of Smoothed Particle Hydrodynamics 143
Joseph J. Monaghan
1 Introduction ... 143
2 Integral and Summation Interpolants ... 144
2.1 Errors in the Integral Interpolant ... 148
2.2 Errors in the Summation Interpolant ... 149
2.3 Errors when the Particles are Disordered ... 152
3 Euler Equations ... 155
3.1 The SPH Continuity Equation ... 156
3.2 The SPH Acceleration Equation ... 157
3.3 The Thermal Energy Equation ... 158
3.4 Dispersion of Sound Waves ... 159
4 Tests of the SPH Euler Equations ... 161
4.1 The Force Law in One Dimension ... 162

4.2 The Equations of Motion . . . 163
4.3 Oscillations . . . 164
4.4 SPH Results for Small Oscillations . . . 165
5 Lagrangian SPH . . . 167
5.1 The Lagrangian . . . 167
5.2 Conservation Laws . . . 168
5.3 The Lagrangian with Constraints . . . 172
5.4 Resolution Varying in Space and Time . . . 174
6 SPH Heat Conduction . . . 177
6.1 Derivatives from Integrals . . . 178
6.2 Does the Entropy Increase? . . . 179
6.3 Discontinuous Thermal Conductivity . . . 180
6.4 Diffusion of Matter . . . 181
7 Viscosity . . . 183
7.1 A Simple Artificial Shock Viscosity . . . 183
7.2 Invariance Properties . . . 185
7.3 Effective Pressure and Viscosity . . . 186
7.4 The Sign of the Dissipation Term . . . 186
8 Applications to Shock and Rarefaction Problems . . . 187
8.1 Rarefaction Waves . . . 187
8.2 The Sod Shock Tube . . . 187
References . . . 193

Implementation and Parallelization of Meshfree Methods . . . 195
Marc Alexander Schweitzer
1 Introduction . . . 195
2 Partition of Unity Method . . . 197
2.1 Construction of a Partition of Unity Space . . . 197
2.2 Variational Formulation and Boundary Conditions . . . 204
2.3 Galerkin Discretization . . . 209
2.4 Solution of Resulting Linear System . . . 210
3 Efficient Implementation . . . 212
3.1 Cover Construction . . . 212
3.2 Numerical Integration . . . 217
3.3 Multilevel Solution of Linear System . . . 231
4 Parallelization . . . 243
4.1 Parallel Data Decomposition . . . 244
4.2 Load Balancing with Space Filling Curves . . . 248
4.3 Parallel Cover Construction . . . 250
4.4 Parallel Neighbour Search . . . 252
4.5 Parallel Matrix Assembly . . . 254
4.6 Parallel Multilevel Solution . . . 254
4.7 Computational Complexity . . . 257
References . . . 258

Wavelet–Based Multiresolution Methods for Stationary PDEs and PDE-Constrained Control Problems

Angela Kunoth

Universität Bonn, Institut für Angewandte Mathematik, Wegelerstr. 6, 53115 Bonn, Germany
email: kunoth@iam.uni-bonn.de

Abstract These notes are concerned with numerical analysis issues arising in the solution of certain classes of stationary linear variational problems. The standard examples are second order elliptic boundary value problems, where particular emphasis is placed on the treatment of essential boundary conditions. These operator equations serve as a core ingredient for control problems where in addition to the state, the solution of the PDE, a control is to be determined which together with the state minimizes a certain tracking-type objective functional. Having assured that the variational problems are well-posed, we propose numerical schemes based on wavelets as a particular multiresolution discretization methodology. The guiding principle is to devise fast and efficient solution schemes which are optimal in the number of arithmetic unknowns. Issues that are dealt with are optimal conditioning of the system matrices, numerical stability of discrete formulations and adaptive approximation.

1 Introduction

For the solution of elliptic partial differential equations (PDEs), multilevel ingredients have proved to achieve more efficient solution methods for a variety of problems than methods based on approximating on a single scale. This is due to the fact that solutions often exhibit a multiscale behaviour which one naturally wants to exploit. Perhaps the first such schemes were multigrid methods where a fixed discretization, with respect to some underlying uniform fine grid, leads a large ill-conditioned system of linear equations to solve. The basic idea of multigrid schemes is to successively solve smaller versions of the linear system which can be interpreted as discretizations with respect to coarser grids. Here 'efficiency of the scheme' means that one can solve the problem with respect to the fine grid with a number of arithmetic operations which is proportional to the number of unknowns on the finest grid. This in turn means that multigrid schemes provide an asymptotically optimal *preconditioner* for the original system on the finest grid. The search for such optimal preconditioners was one of the major topics in the solution of elliptic boundary value problems for many years. Another multiscale preconditioner which has this property is the BPX-preconditioner proposed first

in [BPX] which was proved to be asymptotically optimal with techniques from Approximation Theory in [DK1, O].
Wavelets as a particular example of a multiscale basis were constructed with compact support in the 1980's [Dau]. While mainly used for signal analysis and image compression, they were also discovered to provide optimal preconditioners in the above sense for elliptic boundary value problems [DK1, J]. It was soon realized that biorthogonal spline-wavelets developed in [CDF] are better suited for the numerical solution of elliptic PDEs since they allow one to work with piecewise polynomials instead of the implicitly defined original wavelets [Dau] (in addition to the fact that orthogonality with respect to L_2 of the Daubechies wavelets is only a minor advantage for elliptic PDEs). The principal ingredient that allows one to prove optimality of the preconditioner are certain *norm equivalences* between Sobolev norms and sequence norms of weighted wavelet expansion coefficients, and optimal conditioning of the resulting linear system of equations can be achieved by applying the Fast Wavelet Transform together with a weighting in terms of an appropriate diagonal matrix. The terminology 'wavelets' here and in the sequel is to mean that these are not necessarily Daubechies' wavelets, but rather classes of such multiscale bases with three main properties:

(R) Riesz basis property for the underlying function spaces;
(L) locality of the basis functions;
(CP) cancellation properties;

all of which are detailed in Section 3.1.
After these initial results, research on using wavelets for solving elliptic PDEs numerically has gone into many different directions. Since the original constructions in [Dau, CDF] and many others are based on using the Fourier transform, these constructions provide bases for function spaces only on all of $\mathbb{R}$ or $\mathbb{R}^n$. In order for these tools to be applicable for the solution of PDEs which naturally live on a bounded domain $\Omega \subset \mathbb{R}^n$, there arose the need for having available constructions on bounded intervals without, of course, losing the aforementioned properties (R), (L) and (CP). The first such systematic construction of biorthogonal spline-wavelets on $[0, 1]$ (and, by tensor products, on $[0, 1]^n$) was provided in [DKU]. At the same time, techniques for satisfying essential boundary conditions were investigated in the context of wavelets in [K1].
Aside from the investigations to provide appropriate bases, the built-in potential of *adaptivity* for wavelets has played a prominent role when solving PDEs, on account of the fact that wavelets provide a locally supported Riesz basis for a whole function space. Here the issue is to approximate the solution of the variational problem on an infinite-dimensional function space by the fewest number of degrees of freedom up to a certain prescribed accuracy. Most approaches use wavelet coefficients in a heuristic way, i.e., judging approximation quality by the size of the wavelet coefficients together

with thresholding. In the past few years *convergence* of wavelet-based adaptive methods for stationary variational problems was investigated systematically [CDD1, CDD2, CDD3]. In particular, these schemes are designed to also provide *optimal complexity* of the schemes, meaning that these algorithms provide the solution in a total number of arithmetic operations which is comparable to the wavelet-best N-term approximation of the solution. Here the guidelines are, given a prescribed tolerance, find a sparse representation of the solution by extracting the largest N expansion coefficients of the solution during the solution process.

As soon as one aims at numerically solving a variational problem which can no longer be formulated in terms of a single elliptic operator equation such as a saddle point problem, one is faced with the problem of numerical stability. This means that finite approximations of the continuous well-posed problem may be ill-posed, obstructing its efficient numerical solution. This issue will also be addressed below.

Along these lines, I would like to discuss in these notes the potential of by wavelet methods for the following classes of problems. First, we will be concerned with second order elliptic PDEs with a particular emphasis placed on treating essential boundary conditions. Then *PDE-constrained control problems* guided by elliptic boundary value problems are considered, leading to a *system* of elliptic PDEs. The starting point for contriving efficient solution schemes are wavelet representations of continuous well-posed problems in their variational form. Viewing the numerical solution of such a discretized, yet still infinite-dimensional operator equation as an approximation helps to reveal multilevel preconditioners for elliptic PDEs which yield *uniformly bounded condition numbers*. *Stability issues* like the LBB condition for saddle point problems are also discussed in this context. In addition, the compact support of the wavelets allows for sparse representations of the implicit information contained in systems of PDEs, the *adaptive approximation* of their solution.

More information and extensive literature on applying wavelets for more general PDEs addressing, among other things, the connection between adaptivity and nonlinear approximation and the evaluation of nonlinearities may be found in [Co, D2, D3].

This paper is structured as follows. In Section 2 a number of well–posed variational problem classes are compiled to which later several aspects of the wavelet methodology are applied. The simplest example is a linear elliptic boundary value problem for which we derive two forms of an operator equation, the simplest one consisting of just one equation for homogeneous boundary conditions and a more complicated one in the form of a saddle point problem where nonhomogeneous boundary conditions are treated by means of Lagrange multipliers. Both formulations are then employed for the following classes of PDE-constrained control problems. In the *distributed control problems* in Section 2.4 the control is exerted through the right hand side

of the PDE, while in *Dirichlet boundary control problems* in Section 2.5 the Dirichlet boundary condition serves this purpose. Section 3 is devoted to assembling necessary ingredients and basic properties of wavelets which are required in the sequel. In particular, Section 3.4 collects the essential construction principles for wavelets on bounded domains which do not rely on Fourier techniques, namely, multiresolution analyses of function spaces and the concept of stable completions. In Section 4, we formulate the problem classes introduced in Section 2 in wavelet coordinates and in particular derive the resulting systems of linear equations for the control problems arising from the optimality conditions. Section 5 is devoted to the iterative solution of these systems. We shall fully investigate iterative schemes on uniform grids and show that the resulting systems can be solved in the wavelet framework together with a nested iteration strategy with a number of arithmetic operations which is proportional to the total number of unknowns on the finest grid. Finally, in Section 5.2 a wavelet-based adaptive scheme for the distributed control problem will be derived together with convergence results and complexity estimates, relying on techniques from Nonlinear Approximation Theory.

Throughout these notes we will employ the following notational convention: the relation $a \sim b$ will always stand for $a \lesssim b$ and $b \lesssim a$ where the latter inequality means that b can be bounded by some constant times a uniformly in all parameters on which a and b may depend. Norms and inner products are always indexed by the corresponding function space. For $1 \le p \le \infty$, $L_p(\Omega)$ are the usual Lebesgue spaces on a domain Ω, and for $k \in \mathbb{N}$, $W_p^k(\Omega) \subset L_p(\Omega)$ denote the Sobolev spaces of functions whose weak derivatives up to order k are bounded in $L_p(\Omega)$. For $p = 2$, we write as usual $H^k(\Omega) = W_2^k(\Omega)$.

2 Problem Classes

The variational problems to be investigated here will first be formulated in the following abstract form.

2.1 An Abstract Operator Equation

Let $\mathcal{H}$ be a Hilbert space with norm $\|\cdot\|_{\mathcal{H}}$ and let $\mathcal{H}'$ be the normed dual of $\mathcal{H}$ endowed with the norm

$$\|w\|_{\mathcal{H}'} := \sup_{v \in \mathcal{H}} \frac{\langle v, w\rangle}{\|v\|_{\mathcal{H}}} \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the dual pairing between $\mathcal{H}$ and $\mathcal{H}'$.

Given $F \in \mathcal{H}'$, we seek a solution to the operator equation

$$\mathcal{L}\, U = F \tag{2.2}$$

where $\mathcal{L} : \mathcal{H} \to \mathcal{H}'$ is a linear operator which is assumed to be a bounded bijection, that is,

$$\|\mathcal{L}V\|_{\mathcal{H}'} \sim \|V\|_{\mathcal{H}}, \qquad V \in \mathcal{H}. \tag{2.3}$$

We call the operator equation *well-posed* since (2.2) implies for any given data $F \in \mathcal{H}'$ the existence and uniqueness of the solution $U \in \mathcal{H}$ which depends continuously on the data.

In the following subsections, we describe some problem classes which can be placed into this framework. In particular, these examples will have the format that $\mathcal{H}$ is a product space

$$\mathcal{H} := H_{1,0} \times \cdots \times H_{m,0} \tag{2.4}$$

where each of the $H_{i,0} \subseteq H_i$ is a Hilbert space (or a closed subspace of a determined Hilbert space H_i, e.g., by homogeneous boundary conditions). The spaces H_i will be Sobolev spaces living on a domain $\Omega \subset \mathbb{R}^n$ or on (part of) its boundary. According to the definition of $\mathcal{H}$, the elements $V \in \mathcal{H}$ will consist of m components $V = (v_1, \ldots, v_m)^T$, and we define $\|V\|_{\mathcal{H}}^2 := \sum_{i=1}^m \|v_i\|_{H_i}^2$. The dual space $\mathcal{H}'$ is then endowed with the norm

$$\|W\|_{\mathcal{H}'} := \sup_{V \in \mathcal{H}} \frac{\langle V, W\rangle}{\|V\|_{\mathcal{H}}} \tag{2.5}$$

where $\langle V, W\rangle := \sum_{i=1}^m \langle v_i, w_i\rangle_i$ in terms of the dual pairing $\langle \cdot, \cdot \rangle_i$ between H_i and H_i'.

We next formulate four problem classes which fit into this format. The first two concern elliptic boundary value problems with included essential boundary conditions, and elliptic boundary value problems formulated as saddle point problem with boundary conditions treated by means of Lagrange Multipliers. For an introduction on elliptic boundary value problems and saddle point problems together with the functional analytic background one can, e.g., resort to [B]. Based on these formulations, we introduce certain control problems afterwards. A recurring theme in the derivation of the system of operator equation is the minimization of a quadratic functional subject to linear constraints.

2.2 Elliptic Boundary Value Problems

Let $\Omega \subset \mathbb{R}^n$ be a bounded domain with piecewise smooth boundary $\partial\Omega := \Gamma \cup \Gamma_N$. We consider the scalar second order boundary value problem

$$\begin{aligned} -\nabla \cdot (\mathbf{a}\nabla y) + cy &= f \quad \text{in } \Omega, \\ y &= g \quad \text{on } \Gamma, \\ (\mathbf{a}\nabla y) \cdot \mathbf{n} &= 0 \quad \text{on } \Gamma_N, \end{aligned} \tag{2.6}$$

where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the outward normal at $\mathbf{x} \in \Gamma$, $\mathbf{a} = \mathbf{a}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ is uniformly positive definite and bounded on Ω and $c \in L_\infty(\Omega)$. Moreover, f

and g are some given right hand side and boundary data. With the usual definition of the bilinear form

$$a(v,w) := \int_\Omega (\mathbf{a}\nabla v \cdot \nabla w + cvw)\, d\mathbf{x}, \tag{2.7}$$

the weak formulation of (2.6) requires in the case $g \equiv 0$ to find $y \in \mathcal{H}$ where

$$\mathcal{H} := H^1_{0,\Gamma}(\Omega) := \{v \in H^1(\Omega) :\ v|_\Gamma = 0\}, \tag{2.8}$$

or

$$\mathcal{H} := \{v \in H^1(\Omega) : \int_\Omega v(\mathbf{x})\, d\mathbf{x} = 0\} \quad \text{when } \Gamma = \emptyset, \tag{2.9}$$

such that

$$a(y,v) = \langle v, f\rangle, \quad v \in \mathcal{H}. \tag{2.10}$$

The Neumann–type boundary conditions on Γ_N are implicitly satisfied in the weak formulation (2.10), therefore called *natural boundary conditions*. In contrast, the Dirichlet boundary conditions on Γ have to be posed explicitly, for this reason called *essential boundary conditions*. The easiest way to achieve this for homogeneous Dirichlet boundary conditions when $g \equiv 0$ is to include them in the solution space as above in (2.8). In the nonhomogeneous case $g \not\equiv 0$ on Γ in (2.6) and $\Gamma \neq \emptyset$, one can reduce the problem to a problem with homogeneous boundary conditions by *homogenization* as follows. Let $w \in H^1(\Omega)$ be such that $w = g$ on Γ. Then $\tilde{y} := y - w$ satisfies

$$a(\tilde{y}, v) = a(y,v) - a(w,v) = \langle v, f\rangle - a(w,v) =: \langle v, \tilde{f}\rangle$$

for all $v \in \mathcal{H}$ defined in (2.8), and on Γ one has $\tilde{y} = g - w \equiv 0$, that is, $\tilde{y} \in \mathcal{H}$. Thus, it suffices to consider the weak form (2.10) with eventually modified right hand side. (A second possibility which allows one to treat inhomogeneous boundary conditions explicitly in the context of saddle point problems will be discussed below in Section 2.3.)

The crucial property is that the bilinear form defined in (2.7) is continuous and elliptic on $\mathcal{H}$,

$$a(v,v) \sim \|v\|^2_{\mathcal{H}} \qquad \text{for any } v \in \mathcal{H}, \tag{2.11}$$

for example see [B].

By Riesz' representation theorem, the bilinear form defines a linear operator $A : \mathcal{H} \to \mathcal{H}'$ by

$$\langle w, Av\rangle := a(v,w), \qquad v, w \in \mathcal{H}, \tag{2.12}$$

which is under the above assumptions of being a bounded linear bijection, that is,

$$c_A\|v\|_{\mathcal{H}} \ \le\ \|Av\|_{\mathcal{H}'} \ \le\ C_A\|v\|_{\mathcal{H}} \qquad \text{for any } v \in \mathcal{H}. \tag{2.13}$$

Here we only consider the case where A is symmetric. With corresponding alterations, the material in the subsequent sections can also be derived for the

nonsymmetric case with corresponding changes with respect to the employed algorithms.
The relation (2.13) implies that given any $f \in \mathcal{H}'$, there exists a unique $y \in \mathcal{H}$ which solves the linear system

$$Ay = f \qquad \text{in } \mathcal{H}' \tag{2.14}$$

derived from (2.10). This linear operator equation, where the operator defines a bounded bijection in the sense of (2.13), is the simplest case of a well-posed variational problem (2.2). Adhering to the notation in Section 2.1, here we have $m = 1$ and $\mathcal{L} = A$.

2.3 Saddle Point Problems Involving Boundary Conditions

A collection of saddle point problems or, more general, multiple field formulations, including first order system formulations of the elliptic boundary value problem (2.6) and the three field formulation of the Stokes problem with inhomogeneous boundary conditions, have been rephrased as well-posed variational problems in the above sense in [DKS], see also further references cited therein.
Here a particular saddle point problem derived from (2.6) shall be considered which will be recycled later in the context of control problems. In fact, this formulation is particularly appropriate for handling essential Dirichlet boundary conditions.
Recall from, e.g., [B], that the solution $y \in \mathcal{H}$ of (2.10) is also the unique solution of the minimization problem

$$\inf_{v \in \mathcal{H}} \mathcal{J}(v), \qquad \mathcal{J}(v) := \frac{1}{2} a(v, v) - \langle v, f \rangle. \tag{2.15}$$

This means that y is a zero for its first order variational derivative of $\mathcal{J}$, that is, $\delta \mathcal{J}(y; v) = 0$. We denote here and in the following by $\delta^m \mathcal{J}(v; w_1, \ldots, w_m)$ the m–th variation of $\mathcal{J}$ at v in directions $w_1, \ldots, w_m$, see e.g. [Z]. In particular, for $m = 1$

$$\delta \mathcal{J}(v; w) := \lim_{t \to 0} \frac{\mathcal{J}(v + tw) - \mathcal{J}(v)}{t} \tag{2.16}$$

is the (Gateaux) derivative of $\mathcal{J}$ at v in direction w.
In order to generalize (2.15) to the case of nonhomogeneous Dirichlet boundary conditions g, we formulate this as minimizing J over $v \in H^1(\Omega)$ subject to constraints in form of the essential boundary conditions $v = g$ on Γ. Using techniques from nonlinear optimization theory, one can employ a *Lagrange multiplier* p to append the constraints to the optimization functional J defined in (2.15). Satisfying the constraint is guaranteed by taking the supremum over all such Lagrange multipliers before taking the infimum. Thus,

minimization subject to a constraint leads to the problem of finding a *saddle point* (y,p) of the *saddle point problem*

$$\inf_{v\in H^1(\Omega)} \sup_{q\in(H^{1/2}(\Gamma))'} \mathcal{J}(v) + \langle v-g, q\rangle_\Gamma. \tag{2.17}$$

Some comments on the choice of the Lagrange multiplier space and the dual form $\langle\cdot,\cdot\rangle_\Gamma$ in (2.17) are in order. The boundary expression $v = g$ actually means taking the *trace* of $v \in H^1(\Omega)$ to $\Gamma \subseteq \partial\Omega$ which we explicitly write from now on as $\gamma v := v|_\Gamma$. Classical trace theorems which may be found in [Gr] state that for any $v \in H^1(\Omega)$ one loses '$\frac{1}{2}$ order of smoothness' when taking traces so that one ends up with $\gamma v \in H^{1/2}(\Gamma)$. Thus, when the data $g \in H^{1/2}(\Gamma)$, the expression in (2.17) involving the dual form $\langle\cdot,\cdot\rangle_\Gamma := \langle\cdot,\cdot\rangle_{H^{1/2}(\Gamma)\times(H^{1/2}(\Gamma))'}$ is well–defined, and so is the selection of the multiplier space $(H^{1/2}(\Gamma))'$. In the case of Dirichlet boundary conditions on the whole boundary of Ω, i.e., the case $\Gamma \equiv \partial\Omega$, one can identify $(H^{1/2}(\Gamma))' = H^{-1/2}(\Gamma)$.

The formulation (2.17) above was first investigated in [Ba1]. Another standard technique from optimization to handle minimization problems under constraints is to append the constraints to $J(v)$ by means of a *penalty parameter* ε as follows, cf. [Ba2]. For the case of homogeneous Dirichlet boundary conditions, one could introduce the functional $J(v)+(2\varepsilon)^{-1}\|\gamma v\|^2_{H^{1/2}(\Gamma)}$. (The original formulation in [Ba2] uses the term $\|\gamma v\|^2_{L_2(\Gamma)}$.) Although the linear system derived from this formulation is still elliptic — the bilinear form is of the type $a(v,v)+\varepsilon^{-1}(\gamma v,\gamma v)_{H^{1/2}(\Gamma)}$ — the spectral condition number of the corresponding operator A_ε depends on ε. The choice of ε is typically attached to the discretization of an underlying grid with grid spacing h for Ω of the form $\varepsilon \sim h^\alpha$ when $h \to 0$ for some exponent $\alpha > 0$ chosen such that one retains the optimal approximation order of the underlying scheme. Thus, the spectral condition number of the operators in such systems depends polynomially on (at least) $h^{-\alpha}$. Consequently, iterative solution schemes such as the conjugate gradient method converge as slow as without preconditioning for A, and so far no optimal preconditioniers for this situation are known.

It should also be mentioned that the way of treating essential boundary conditions by Lagrange multipliers can be extended to *fictitious domain methods* which may be used for problems with changing boundaries such as shape optimization problems [HM,KP]. There one embeds the domain Ω into a larger, simple domain $\Box$, and formulates (2.17) with respect to $H^1(\Box)$ and dual form on the changing boundary Γ [K3]. One should note, however, that for Γ a proper subset of $\partial\Omega$, some ambiguity may occur in the relation between the fictitious domain formulation and the corresponding strong form (2.6).

In order to bring out the role of the trace operator, in addition to (2.7) we define a second bilinear form on $H^1(\Omega)\times(H^{1/2}(\Gamma))'$ by

$$b(v,q) := \int_\Gamma (\gamma v)(s)\, q(s)\, ds \tag{2.18}$$

so that the saddle point problem (2.17) may be rewritten as

$$\inf_{v\in H^1(\Omega)} \sup_{q\in(H^{1/2}(\Gamma))'} \mathcal{J}(v,q), \quad \text{where} \quad \mathcal{J}(v,q) := J(v) + b(v,q) - \langle g, q\rangle_\Gamma. \tag{2.19}$$

Computing zeroes of the first order variations of $\mathcal{J}$, now with respect to both v and q, yields the system of equations that a saddle point (y,p) has to satisfy

$$\begin{aligned} a(y,v) + b(v,p) &= \langle v, f\rangle, & v &\in H^1(\Omega),\\ b(y,q) &= \langle g, q\rangle_\Gamma, & q &\in (H^{1/2}(\Gamma))'. \end{aligned} \tag{2.20}$$

Defining the linear operator $B : H^1(\Omega) \to H^{1/2}(\Gamma)$ and its adjoint $B' : (H^{1/2}(\Gamma))' \to (H^1(\Omega))'$ by

$$\langle Bv, q\rangle_\Gamma = \langle v, B'q\rangle_\Gamma := b(v,q),$$

this can be rewritten as the linear operator equation from $\mathcal{H} := H^1(\Omega) \times (H^{1/2}(\Gamma))'$ to $\mathcal{H}'$ as follows:
Given $(f,g) \in \mathcal{H}'$, find $(y,p) \in \mathcal{H}$ that solves

$$\begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{2.21}$$

It can be shown that the Lagrange multiplier is given by $p = -\mathbf{n}\cdot\mathbf{a}\nabla y$ and can here be interpreted as a *stress force* on the boundary [Ba1].
Let us briefly investigate the properties of B representing the trace operator. Classical trace theorems from, e.g., [Gr], state that for any $f \in H^s(\Omega)$, $1/2 < s < 3/2$, one has

$$\|f|_\Gamma\|_{H^{s-1/2}(\Gamma)} \lesssim \|f\|_{H^s(\Omega)}. \tag{2.22}$$

Conversely, for every $g \in H^{s-1/2}(\Gamma)$, there exists some $f \in H^s(\Omega)$ such that $f|_\Gamma = g$ and

$$\|f\|_{H^s(\Omega)} \lesssim \|g\|_{H^{s-1/2}(\Gamma)}. \tag{2.23}$$

Note that the range of s extends accordingly if Γ is more regular. Estimate (2.22) immediately implies for $s = 1$ that $B : H^1(\Omega) \to H^{1/2}(\Gamma)$ is continuous. Moreover, the second property (2.23) means B is surjective, i.e., $\operatorname{range} B = H^{1/2}(\Gamma)$ and $\ker B' = \{0\}$, which yields that the *inf–sup condition*

$$\inf_{q\in(H^{1/2}(\Gamma))'} \sup_{v\in H^1(\Omega)} \frac{\langle Bv, q\rangle_\Gamma}{\|v\|_{H^1(\Omega)}\,\|q\|_{(H^{1/2}(\Gamma))'}} \gtrsim 1 \tag{2.24}$$

is satisfied.
At this point it will be more convenient to consider (2.21) as a saddle point problem in abstract form on $\mathcal{H} = Y \times Q$. Thus, we identify $Y = H^1(\Omega)$ and $Q = (H^{1/2}(\Gamma))'$ and linear operators $A : Y \to Y'$ and $B : Y \to Q'$.
The abstract theory of saddle point problems states there exists and unique solution pair $(y,p) \in \mathcal{H}$ if A and B are continuous, A is invertible on

$\ker B \subseteq Y$ and the range of B is closed in Q', for example see [B, BF, GR]. The properties for B and the continuity for A have been assured above. In addition, we will always deal here with operators A which are invertible on $\ker B$, which cover the standard cases of the Laplacian ($\mathbf{a} = I$ and $c \equiv 0$) and the Helmholtz operator ($\mathbf{a} = I$ and $c = 1$).
Consequently,

$$\mathcal{L} := \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} : \mathcal{H} \to \mathcal{H}' \tag{2.25}$$

is a linear bijection, and one has the mapping property

$$\left\| \mathcal{L} \begin{pmatrix} v \\ q \end{pmatrix} \right\|_{\mathcal{H}'} \sim \left\| \begin{pmatrix} v \\ q \end{pmatrix} \right\|_{\mathcal{H}} \tag{2.26}$$

for any $(v, q) \in \mathcal{H}$ with constants depending on upper and lower bounds for A, B. Thus, the operator equation (2.21) is established to be a well-posed variational problem in the sense of Section 2.1: for given $(f, g) \in \mathcal{H}'$, there exists a unique solution $(y, p) \in \mathcal{H} = Y \times Q$ which depends continuously on the data.

2.4 PDE-Constrained Control Problems: Distributed Control

A class of problems where the numerical solution of systems (2.14) is required repeatedly are certain control problems with PDE-constraints described next. Adhering to the notation from Section 2.2, consider as a guiding model for the subsequent discussion the objective to minimize a quadratic functional of the form

$$\mathcal{J}(y, u) = \frac{1}{2}\|y - y_*\|_{\mathcal{Z}}^2 + \frac{\omega}{2}\|u\|_{\mathcal{U}}^2, \tag{2.27}$$

subject to linear constraints

$$Ay = f + u \qquad \text{in } H' \tag{2.28}$$

where $A : H \to H'$ is defined as above in (2.12) satisfying (2.13) and $f \in H$ is given. Reserving the symbol $\mathcal{H}$ for the resulting product space, in view of the notation in Section 2.1, the space H in this subsection is defined as in (2.8) or in (2.9). In order for a solution y of (2.28), the *state* of the system, to be well–defined, the problem formulation has to ensure that the unknown *control* u appearing on the right hand side is at least in H'. This can be achieved by choosing the *control space* $\mathcal{U}$ whose norm appears in (2.27) such that it is as least as smooth as H'. The second ingredient in the functional (2.27) is a data fidelity term which tries to match the system state y to some prescribed target state y_*, measured in some norm which is typically weaker than $\|\cdot\|_H$. Thus, we require that the *observation space* $\mathcal{Z}$ and the control space $\mathcal{U}$ are such that the continuous embeddings

$$\|v\|_{H'} \lesssim \|v\|_{\mathcal{U}}, \quad v \in \mathcal{U}, \qquad \|v\|_{\mathcal{Z}} \lesssim \|v\|_H, \quad v \in H, \tag{2.29}$$

hold. Mostly the simplest cases of norms which occur for $\mathcal{U} = \mathcal{Z} = L_2(\Omega)$ have been investigated and which are covered by these assumptions [Li]. The parameter $\omega \geq 0$ balances the norms in (2.27).
Since the control appears in all of the right hand side of (2.28), such control problems are termed problems with *distributed* control. Although their practical value is of a rather limited nature, distributed control problems help to bring out the basic mechanisms. Note that when the observed data are *compatible* in the sense that $y_* \equiv A^{-1}f$, the control problem has the trivial solution $u \equiv 0$ which yields $\mathcal{J}(y, u) \equiv 0$.
Solution schemes for the control problem (2.27) subject to the constraints (2.28) can be based on the system of operator equations derived next by the same variational principles as employed in the previous section, using a Lagrange multiplier p to enforce the constraints. Defining the Lagrangian functional

$$\mathrm{Lagr}(y, p, u) := \mathcal{J}(y, u) + \langle p, Ay - f - u\rangle \tag{2.30}$$

on $H \times H \times H'$, the first order necessary conditions or *Karush-Kuhn-Tucker (KKT) conditions* $\delta\,\mathrm{Lagr}(x) = 0$ for $x = p, y, u$ can be derived as

$$\begin{aligned} Ay &= f + u, \\ A'p &= -S(y - y_*), \\ \omega Ru &= p. \end{aligned} \tag{2.31}$$

Here the linear operators S and R can be interpreted as Riesz operators defined by the inner products $(\cdot,\cdot)_{\mathcal{Z}}$ and $(\cdot,\cdot)_{\mathcal{U}}$. The system (2.31) may be written in saddle point form as

$$\mathcal{L}\,V := \begin{pmatrix} \mathcal{A} & \mathcal{B}' \\ \mathcal{B} & 0 \end{pmatrix} V := \begin{pmatrix} S & 0 & A' \\ 0 & \omega R & -I \\ A & -I & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} Sy_* \\ 0 \\ f \end{pmatrix} =: F \tag{2.32}$$

on $\mathcal{H} := H \times H \times H'$.

Remark 2.1.
We can also allow for $\mathcal{Z}$ in (2.27) to be a *trace space* on part of the boundary $\partial\Omega$ as long as the corresponding condition (2.29) is satisfied [K4].
The class of control problems where the control is exerted through Neumann boundary conditions can also be written in this form, since in this case the control still appears on the right hand side of a single operator equation of a form like (2.28), see [DK3]. ∎

Well-posedness of the system (2.32) can now be established by applying the conditions for saddle point problems stated in Section 2.3. However, for the control problems here and below we will follow a different route which supports efficient numerical solution schemes better. The idea is as follows, while the PDE constraints (2.28) that govern the system are fixed, there is in many applications some ambiguity with respect to the choice of the spaces $\mathcal{Z}$ and

$\mathcal{U}$. L_2 norms are easily realized in finite element discretizations, although in some applications like glass cooling smoother norms for the observation $\|\cdot\|_{\mathcal{Z}}$ are desirable [PT]. Once $\mathcal{Z}$ and $\mathcal{U}$ are fixed, there is only a single parameter ω to balance the two norms in (2.27). *Modelling* the objective functional is therefore an issue where more flexibility may be advantageous. Specifically in a multiscale setting, one may want to weight contributions on different scales by multiple parameters.
The wavelet setting which we describe below allows for this flexibility. It is based on formulating the objective functional in terms of weighted wavelet coefficient sequences which are equivalent to $\mathcal{Z}$, $\mathcal{U}$ and which, in addition, support an efficient numerical implementation. Once wavelet discretizations are introduced, we formulate control problems with such objective functionals below.

2.5 PDE-Constrained Control Problems: Dirichlet Boundary Control

Even more involved than the control problems with distributed control encountered in the previous section are those problems with Dirichlet boundary control which, however, are practically more relevant.
An illustrative guiding model for this case is the problem to minimize for some given data y_* the quadratic functional

$$\mathcal{J}(y,u) = \frac{1}{2}\|y - y_*\|_{\mathcal{Z}}^2 + \frac{\omega}{2}\|u\|_{\mathcal{U}}^2, \tag{2.33}$$

where, adhering to the notation in Section 2.2, the state y and the control u are coupled through the linear second order elliptic boundary value problem

$$\begin{aligned} -\nabla\cdot(\mathbf{a}\nabla y) + ky &= f \qquad \text{in } \Omega,\\ y &= u \qquad \text{on } \Gamma,\\ (\mathbf{a}\nabla y)\cdot\mathbf{n} &= 0 \qquad \text{on } \Gamma_N. \end{aligned} \tag{2.34}$$

The appearance of the control u as a Dirichlet boundary condition in (2.34) is referred to as a *Dirichlet boundary control*. In view of the treatment of essential Dirichlet boundary conditions in the context of saddle point problems derived in Section 2.3, we write the PDE constraints (2.34) in the operator form (2.21) on $Y \times Q$ where $Y = H^1(\Omega)$ and $Q = (H^{1/2}(\Gamma))'$. The model control problem with Dirichlet boundary control then reads as follows:
For given data $y_* \in \mathcal{Z}$ and $f \in Y'$ minimize the quadratic functional

$$\mathcal{J}(y,u) = \frac{1}{2}\|y - y_*\|_{\mathcal{Z}}^2 + \frac{\omega}{2}\|u\|_{\mathcal{U}}^2 \tag{2.35}$$

subject to

$$\begin{pmatrix} A & B' \\ B & 0 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ u \end{pmatrix}. \tag{2.36}$$

In view of the problem formulation in Section 2.4 and the discussion of the choice of the observation space $\mathcal{Z}$ and the control space, analogously here we require that $\mathcal{Z}$ and $\mathcal{U}$ are such that the continuous embeddings

$$\|v\|_{Q'} \lesssim \|v\|_{\mathcal{U}}, \quad v \in \mathcal{U}, \qquad \|v\|_{\mathcal{Z}} \lesssim \|v\|_Y, \quad v \in Y, \tag{2.37}$$

hold. In view of Remark 2.1, the case of observations on part of the boundary $\partial\Omega$ can also be taken into account [K5]. Part of the numerical results are for such a situation shown in Figure 5.2.

Remark 2.2.
It should be noted that the simple choice of $\mathcal{U} = L_2(\Gamma)$, which is used in many applications of Dirichlet control problems, is *not* covered here. The problem of well-posedness may arise in this case which we briefly discuss. Note that the constraints (2.34) or, in weak form (2.21), guarantee a unique weak solution $y \in Y = H^1(\Omega)$ provided that the boundary term u satisfies $u \in Q' = H^{1/2}(\Gamma)$. Therefore, in the framework of control problems, this smoothness of u has to be required either by the choice of $\mathcal{U}$ or by the choice of $\mathcal{Z}$ (such as $\mathcal{Z} = H^1(\Omega)$) which would assure $By \in Q'$. In the latter case, we could relax condition (2.37) on $\mathcal{U}$. ∎

In the context of flow control problems, an H^1 norm on the boundary for the control has been used in [GL].
Similarly as stated at the end of Section 2.4, we can now derive by variational principles the first order necessary conditions for a coupled *system* of saddle point problems. Well-posedness of this system can again be established by applying the conditions for saddle point problems from Section 2.3 where the inf-sup condition for the saddle point problem (2.21) yields an inf-sup condition for the exterior saddle point problem of interior saddle point problems [K2]. However, also in this case, we follow the ideas mentioned at the end of Section 2.5 and pose a corresponding control problem in terms of wavelet coefficients.

3 Wavelets

The numerical solution of the classes of problems introduced above hinges on the availability of appropriate wavelet bases for the function spaces under consideration which are all particular Hilbert spaces. First we introduce the three basic properties that we require our wavelet bases to satisfy.
Afterwards, construction principles for wavelets based on multiresolution analysis of function spaces on bounded domains will be given.

3.1 Basic Properties

In view of the problem classes considered above, we need to have a wavelet basis at our disposal for each occurring function space. A *wavelet basis* for a

Hilbert space H here is understood as a collection of functions

$$\Psi_H := \{\psi_{H,\lambda} : \lambda \in I\!I_H\} \subset H \tag{3.1}$$

which are indexed by elements λ from an infinite index set $\in I\!I_H$. Each of the λ comprises different information $\lambda = (j, \mathbf{k}, \mathbf{e})$ such as the *refinement scale* or *level of resolution* j and a spatial location $\mathbf{k} = \mathbf{k}(\lambda) \in \mathbb{Z}^n$. In more than one space dimensions, the basis functions are built from taking tensor products of certain univariate functions, and in this case the third index $\mathbf{e}$ contains information on the *type* of wavelet. We will frequently use the symbol $|\lambda| := j$ to have access to the resolution level j. In the univariate case on all of $\mathbb{R}$, $\psi_{H,\lambda}$ is typically generated by means of shifts and dilates of a single function ψ, i.e., $\psi_\lambda = \psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$, $j, k \in \mathbb{Z}$, normalized with respect to $\|\cdot\|_{L_2}$. On bounded domains, the structure of the functions is essentially the same up to modifications near the boundary.
The three crucial properties that we will assume the wavelet basis to have for the sequel are the following.

(R) Riesz basis property
Every $v \in H$ has a unique expansion in terms of Ψ_H,

$$v = \sum_{\lambda \in I\!I_H} v_\lambda \, \psi_{H,\lambda} =: \mathbf{v}^T \, \Psi_H, \quad \mathbf{v} := (v_\lambda)_{\lambda \in I\!I_H}, \tag{3.2}$$

and its expansion coefficients satisfy a *norm equivalence*, that is, for any $\mathbf{v} = \{v_\lambda : \lambda \in I\!I_H\}$ one has

$$c_H \, \|\mathbf{v}\|_{\ell_2(I\!I_H)} \;\le\; \|\mathbf{v}^T \Psi_H\|_H \;\le\; C_H \, \|\mathbf{v}\|_{\ell_2(I\!I_H)}, \quad \mathbf{v} \in \ell_2(I\!I_H), \tag{3.3}$$

where $0 < c_H \le C_H < \infty$. This means that wavelet expansions induce *isomorphisms* between certain function spaces and sequence spaces. It will be convenient in the following to abbreviate ℓ_2 norms without subscripts as $\|\cdot\| := \|\cdot\|_{\ell_2(I\!I_H)}$ when the index set is clear from the context. If the precise format of the constants does not matter, we write the norm equivalence (3.3) shortly as

$$\|\mathbf{v}\| \;\sim\; \|\mathbf{v}^T \Psi_H\|_H, \quad \mathbf{v} \in \ell_2(I\!I_H). \tag{3.4}$$

(L) Locality
The functions $\psi_{H,\lambda}$ have compact support which decreases with increasing level $j = |\lambda|$, i.e.,

$$\operatorname{diam}(\operatorname{supp} \psi_{H,\lambda}) \;\sim\; 2^{-|\lambda|}. \tag{3.5}$$

(CP) Cancellation property
There exists an integer $\tilde{m} = \tilde{m}_H$ such that

$$\langle v, \psi_{H,\lambda} \rangle \lesssim 2^{-|\lambda|(n/2-n/p+\tilde{m})} |v|_{W_p^{\tilde{m}}(\operatorname{supp}\ \psi_{H,\lambda})}. \tag{3.6}$$

Thus, integrating against a wavelet has the effect of taking an $\tilde{m}$'th order difference which annihilates the smooth part of v. This property is for wavelets defined on Euclidean domains typically realized by constructing Ψ_H in such a way that it possesses a *dual* or *biorthogonal* basis $\tilde{\Psi}_H \subset H'$ such that the multiresolution spaces $\tilde{S}_j := \mathrm{span}\{\tilde{\psi}_{H,\lambda} : |\lambda| < j\}$ contain all polynomials of order $\tilde{m}$. Here *dual basis* means that $\langle \psi_{H,\lambda}, \tilde{\psi}_{H,\nu} \rangle = \delta_{\lambda,\nu}$ where $\lambda, \nu \in I\!I_H$.

A few remarks on these properties are in order. In (R), the norm equivalence (3.4) is crucial since it means complete control over a function measured in $\|\cdot\|_H$ from above and below by its expansion coefficients: small changes in the coefficients only causes small changes in the function which, together with the locality (L), also means that local changes stay local. This stability is an important feature which is used for deriving optimal preconditioners and driving adaptive approximations where, again, the locality is crucial. Finally, the cancellation property (CP) implies that smooth functions have small wavelet coefficients which, on account of (3.3), may be neglected in a controllable way. Moreover, (CP) can be used to derive quasi–sparse representations of a wide class of operators.

By duality arguments one can show that (3.3) is equivalent to the existence of a biorthogonal collection which is *dual* or *biorthogonal* to Ψ_H,

$$\tilde{\Psi}_H := \{\tilde{\psi}_{H,\lambda} : \lambda \in I\!I_H\} \subset H', \quad \langle \psi_{H,\lambda}, \tilde{\psi}_{H,\mu} \rangle = \delta_{\lambda,\mu}, \qquad \lambda, \mu \in I\!I_H, \quad (3.7)$$

which is a Riesz basis for H', that is, for any $\tilde{v} = \tilde{\mathbf{v}}^T \tilde{\Psi}_H \in H'$ one has

$$C_H^{-1} \|\tilde{\mathbf{v}}\| \;\le\; \|\tilde{\mathbf{v}}^T \tilde{\Psi}_H\|_{H'} \;\le\; c_H^{-1} \|\tilde{\mathbf{v}}\|, \quad (3.8)$$

see [D1,D3,K2]. Here, and in the sequel, the tilde expresses that the collection $\tilde{\Psi}_H$ is a dual basis to a primal one for the space identified by the subscript, so that $\tilde{\Psi}_H = \Psi_{H'}$.

Above in (3.3), we have already introduced the following shorthand notation which simplifies the presentation of many terms. We will view Ψ_H, as in (3.1), as a *collection* of functions as well as a (possibly infinite) column *vector* containing all functions always assembled in some fixed unspecified order. For a countable collection of functions Θ and some single function σ, the term $\langle \Theta, \sigma \rangle$ is to be understood as the column vector with entries $\langle \theta, \sigma \rangle$, $\theta \in \Theta$, and correspondingly $\langle \sigma, \Theta \rangle$ the row vector. For two collections Θ, Σ, the quantity $\langle \Theta, \Sigma \rangle$ is then a (possibly infinite) matrix with entries $(\langle \theta, \sigma \rangle)_{\theta \in \Theta,\ \sigma \in \Sigma}$ for which $\langle \Theta, \Sigma \rangle = \langle \Sigma, \Theta \rangle^T$. This also implies that for a (possibly infinite) matrix $\mathbf{C}$ that $\langle \mathbf{C}\Theta, \Sigma \rangle = \mathbf{C}\langle \Theta, \Sigma \rangle$ and $\langle \Theta, \mathbf{C}\Sigma \rangle = \langle \Theta, \Sigma \rangle \mathbf{C}^T$.

In this notation, the *biorthogonality* or *duality conditions* (3.7) can be reexpressed as

$$\langle \Psi, \tilde{\Psi} \rangle = \mathbf{I} \quad (3.9)$$

with the infinite identity matrix $\mathbf{I}$.

Wavelets with the above properties can actually be obtained in the following way. This concerns, in particular, a scaling depending on the regularity of the space under consideration. In our case, H will always be a Sobolev space $H^s = H^s(\Omega)$ or a closed subspace of $H^s(\Omega)$ determined by homogeneous boundary conditions, or its dual. For $s < 0$, H^s is interpreted as above as the dual of H^{-s}. One typically obtains the wavelet basis Ψ_H for H from an *anchor basis* $\Psi = \{\psi_\lambda : \lambda \in I\!I = I\!I_H\}$ which is a Riesz basis for $L_2(\Omega)$, meaning that Ψ is scaled such that $\|\psi_\lambda\|_{L_2(\Omega)} \sim 1$. Moreover, its dual basis $\tilde{\Psi}$ is also a Riesz basis for $L_2(\Omega)$. Ψ and $\tilde{\Psi}$ are constructed in such a way that rescaled versions of *both bases* $\Psi, \tilde{\Psi}$ form Riesz bases for a whole range of (closed subspaces of) Sobolev spaces H^s, for $0 < s < \gamma, \tilde{\gamma}$, respectively. Consequently, one can derive that for each $s \in (-\tilde{\gamma}, \gamma)$ the collection

$$\Psi_s := \{2^{-s|\lambda|}\psi_\lambda : \lambda \in I\!I\} =: \mathbf{D}^{-s}\Psi \tag{3.10}$$

is a Riesz basis for H^s, see [D1]. This means that there exist positive finite constants c_s, C_s such that

$$c_s \|\mathbf{v}\| \;\le\; \|\mathbf{v}^T\Psi_s\|_{H^s} \;\le\; C_s \|\mathbf{v}\| \quad \mathbf{v} \in \ell_2(I\!I), \tag{3.11}$$

holds for each $s \in (-\tilde{\gamma}, \gamma)$. Such a scaling represented by a diagonal matrix $\mathbf{D}^s$ introduced in (3.10) will play an important role later on. The analogous expression in terms of the dual basis reads

$$\tilde{\Psi}_s := \{2^{s|\lambda|}\,\tilde{\psi}_\lambda : \lambda \in I\!I\} = \mathbf{D}^s\,\tilde{\Psi}, \tag{3.12}$$

where $\tilde{\Psi}_s$ forms a Riesz basis of H^s for $s \in (-\gamma, \tilde{\gamma})$. This implies the following fact. For $t \in (-\tilde{\gamma}, \gamma)$ the mapping

$$D^t : v = \mathbf{v}^T\Psi \mapsto (\mathbf{D}^t\mathbf{v})^T\Psi = \mathbf{v}^T\mathbf{D}^t\Psi = \sum_{\lambda\in I\!I} v_\lambda\, 2^{t|\lambda|}\psi_\lambda \tag{3.13}$$

acts as a shift operator between Sobolev scales which means that

$$\|D^t v\|_{H^s} \sim \|v\|_{H^{s+t}} \sim \|\mathbf{D}^{s+t}\mathbf{v}\|, \quad \text{if} \;\; s,\; s+t \in (-\tilde{\gamma}, \gamma). \tag{3.14}$$

Concrete constructions of wavelet bases with the above properties for parameters $\gamma, \tilde{\gamma} \le 3/2$ on a bounded Lipschitz domain Ω can be found in [DKU,DSt]. This suffices for the aforementioned examples where the relevant Sobolev regularity indices range between -1 and 1.

3.2 Norm Equivalences and Riesz Maps

As we have seen, the scaling provided by $\mathbf{D}^{-s}$ is an important feature to establish norm equivalences (3.11) for the range $s \in (-\tilde{\gamma}, \gamma)$ of Sobolev spaces H^s. However, there are several other norms which are *equivalent* to $\|\cdot\|_{H^s}$ which may be used later in the objective functional (2.27) in the context of

control problems. This issue addresses the *mathematical model* which we now briefly discuss.

We first consider norm equivalences for the L_2 norm. As before let Ψ be the anchor wavelet basis for L_2 for which the *Riesz operator* $\mathbf{R} = \mathbf{R}_{L_2}$ is the (infinite) Gramian matrix with respect to the inner product $(\cdot,\cdot)_{L_2}$ defined as

$$\mathbf{R} := (\Psi, \Psi)_{L_2} = \langle \Psi, \Psi \rangle. \tag{3.15}$$

Expanding Ψ in terms of $\tilde{\Psi}$ and recalling the duality (3.9), this implies

$$\mathbf{I} = \langle \Psi, \tilde{\Psi} \rangle = \left\langle \langle \Psi, \Psi \rangle \tilde{\Psi}, \tilde{\Psi} \right\rangle = \mathbf{R} \langle \tilde{\Psi}, \tilde{\Psi} \rangle \qquad \text{or} \qquad \mathbf{R}^{-1} = \langle \tilde{\Psi}, \tilde{\Psi} \rangle. \tag{3.16}$$

$\mathbf{R}$ may be interpreted as the transformation matrix for the change of basis from $\tilde{\Psi}$ to Ψ, that is, $\Psi = \mathbf{R}\tilde{\Psi}$.

For any $w = \mathbf{w}^T \Psi \in L_2$, we now obtain the identities

$$\|w\|_{L_2}^2 = (\mathbf{w}^T \Psi, \mathbf{w}^T \Psi)_{L_2} = \mathbf{w}^T \langle \Psi, \Psi \rangle \, \mathbf{w} = \mathbf{w}^T \mathbf{R} \mathbf{w} = \|\mathbf{R}^{1/2} \mathbf{w}\|^2 =: \|\hat{\mathbf{w}}\|^2. \tag{3.17}$$

Expanding w with respect to the basis $\hat{\Psi} := \mathbf{R}^{-1/2}\Psi = \mathbf{R}^{1/2}\tilde{\Psi}$, that is, $w = \hat{\mathbf{w}}^T \hat{\Psi}$, yields $\|w\|_{L_2} = \|\hat{\mathbf{w}}\|$. On the other hand, we get from (3.11) with $s = 0$

$$c_0^2 \, \|\mathbf{w}\|^2 \;\le\; \|w\|_{L_2}^2 \;\le\; C_0^2 \, \|\mathbf{w}\|^2. \tag{3.18}$$

From this we can derive the *condition number* $\kappa(\Psi)$ of the wavelet basis in terms of the extreme eigenvalues of $\mathbf{R}$ by defining

$$\kappa(\Psi) := \left(\frac{C_0}{c_0} \right)^2 = \frac{\lambda_{\max}(\mathbf{R})}{\lambda_{\min}(\mathbf{R})} = \kappa(\mathbf{R}) \sim 1, \tag{3.19}$$

where $\kappa(\mathbf{R})$ also denotes the spectral condition number of $\mathbf{R}$ and the last relation is assured by the asymptotic estimate (3.18). However, the absolute constants will have an impact on numerical results in specific cases.

For a Hilbert space H, denote by Ψ_H a wavelet basis for H satisfying (R), (L), (CP) with a corresponding dual basis $\tilde{\Psi}_H$. The (infinite) Gramian matrix with respect to the inner product $(\cdot,\cdot)_H$ inducing $\|\cdot\|_H$ which is defined by

$$\mathbf{R}_H := (\Psi_H, \Psi_H)_H \tag{3.20}$$

will be also called *Riesz operator*. The space L_2 is covered trivially by $\mathbf{R}_0 = \mathbf{R}$. For any function $v := \mathbf{v}^T \Psi_H \in H$ we then have the identity

$$\begin{aligned} \|v\|_H^2 = (v, v)_H &= (\mathbf{v}^T \Psi_H, \mathbf{v}^T \Psi_H)_H = \mathbf{v}^T (\Psi_H, \Psi_H)_H \, \mathbf{v} \\ &= \mathbf{v}^T \mathbf{R}_H \mathbf{v} = \|\mathbf{R}_H^{1/2} \mathbf{v}\|^2. \end{aligned} \tag{3.21}$$

Note that in general $\mathbf{R}_H$ may not be explicitly computable, in particular, when H is a fractional Sobolev space.

Again referring to (3.11), we obtain as in (3.19) for the more general case

$$\kappa(\Psi_s) := \left(\frac{C_s}{c_s}\right)^2 = \frac{\lambda_{\max}(\mathbf{R}_{H^s})}{\lambda_{\min}(\mathbf{R}_{H^s})} = \kappa(\mathbf{R}_{H^s}) \sim 1 \qquad \text{for each } s \in (-\tilde{\gamma}, \gamma). \tag{3.22}$$

Thus, all Riesz operators on the applicable scale of Sobolev spaces are spectrally equivalent. Moreover, comparing (3.22) with (3.19), we get

$$\frac{c_s}{C_0}\,\|\mathbf{R}^{1/2}\mathbf{v}\| \;\le\; \|\mathbf{R}_{H^s}^{1/2}\mathbf{v}\| \;\le\; \frac{C_s}{c_0}\,\|\mathbf{R}^{1/2}\mathbf{v}\|. \tag{3.23}$$

Of course, in practice, the constants appearing in this equation may be much sharper, as the bases for Sobolev spaces with different exponents are only obtained by a diagonal scaling which preserves much of the structure of the original basis for L_2.
We summarize these results for further reference.

Proposition 3.1. *In the above notation, we have for any $v = \mathbf{v}^T\Psi_s \in H^s$ the norm equivalences*

$$\|v\|_{H^s} = \|\mathbf{R}_{H^s}^{1/2}\mathbf{v}\| \sim \|\mathbf{R}^{1/2}\mathbf{v}\| \sim \|\mathbf{v}\| \qquad \textit{for each } s \in (-\tilde{\gamma}, \gamma). \tag{3.24}$$

3.3 Representation of Operators

A final ingredient concerns the *wavelet representation* of linear operators in terms of wavelets. Let H, V be Hilbert spaces with wavelet bases Ψ_H, Ψ_V and corresponding duals $\tilde{\Psi}_H$, $\tilde{\Psi}_V$, and suppose that $\mathcal{L} : H \to V$ is a linear operator with dual $\mathcal{L}' : V' \to H'$ defined by $\langle v, \mathcal{L}'w\rangle := \langle \mathcal{L}v, w\rangle$ for all $v \in H$, $w \in V$.
We shall make frequent use of this representation and its properties.

Remark 3.1.
The wavelet representation of $\mathcal{L} : H \to V$ with respect to the bases $\Psi_H, \tilde{\Psi}_V$ of H, V', respectively, is given by

$$\mathbf{L} := \langle \tilde{\Psi}_V, \mathcal{L}\Psi_H\rangle, \qquad \mathcal{L}v = (\mathbf{L}\mathbf{v})^T\Psi_V. \tag{3.25}$$

Thus, the expansion coefficients of $\mathcal{L}v$ in the basis that spans the range space of $\mathcal{L}$ is obtained by applying the *infinite* matrix $\mathbf{L} = \langle \tilde{\Psi}_V, \mathcal{L}\Psi_H\rangle$ to the coefficient vector of v. Moreover, boundedness of $\mathcal{L}$ implies boundedness of $\mathbf{L}$ in ℓ_2, i.e.,

$$\|\mathcal{L}v\|_V \lesssim \|v\|_H,\ v \in H, \ \text{ implies } \ \|\mathbf{L}\| := \sup_{\|\mathbf{v}\|_{\ell_2(\mathcal{I}_H)} \le 1} \|\mathbf{L}\mathbf{v}\|_{\ell_2(\mathcal{I}_V)} \lesssim 1. \tag{3.26}$$

■

Proof. Any image $\mathcal{L}v \in V$ can naturally be expanded with respect to Ψ_V as $\mathcal{L}v = \langle \mathcal{L}v, \tilde{\Psi}_V \rangle \Psi_V$. In addition expanding v in the basis Ψ_H, $v = \mathbf{v}^T \Psi_H$ yields

$$\mathcal{L}v = \mathbf{v}^T \langle \mathcal{L}\Psi_H, \tilde{\Psi}_V \rangle \Psi_V = (\langle \mathcal{L}\Psi_H, \tilde{\Psi}_V \rangle^T \mathbf{v})^T \Psi_V = (\langle \tilde{\Psi}_V, \mathcal{L}\Psi_H \rangle \mathbf{v})^T \Psi_V. \tag{3.27}$$

As for (3.26), we can infer from (3.3) and (3.25) that

$$\|\mathbf{L}\mathbf{v}\|_{\ell_2(\mathcal{I}_V)} \sim \|(\mathbf{L}\mathbf{v})^T \Psi_V\|_V = \|Lv\|_V \lesssim \|v\|_H \sim \|\mathbf{v}\|_{\ell_2(\mathcal{I}_H)},$$

which confirms the claim. □

3.4 Multiscale Decomposition of Function Spaces

In this section, the basic construction principles of the biorthogonal wavelets with properties (R), (L) and (CP) are summarized, for example see [D2]. Their cornerstones are *multiresolution analyses* of the function spaces under consideration and the concept of *stable completions*. These concepts are free of Fourier techniques and can therefore be applied to derive constructions of wavelets on domains or manifolds which are subsets of $\mathbb{R}^n$.

Multiresolution of L_2

Practical constructions of wavelets typically start out with multiresolution analyses of function spaces. Consider a *multiresolution* $\mathcal{S}$ of L_2 which consists of closed subspaces S_j of L_2, called *trial spaces*, such that they are nested and their union is dense in L_2,

$$S_{j_0} \subset S_{j_0+1} \subset \ldots \subset S_j \subset S_{j+1} \subset \ldots L_2, \qquad \operatorname{clos}_{L_2}\Big(\bigcup_{j=j_0}^{\infty} S_j \Big) = L_2. \tag{3.28}$$

The index j is the refinement level which already appeared in the elements of the index set $\mathcal{I}$ in (3.1), starting with some coarsest level $j_0 \in \mathbb{N}_0$. For a finite subset $\Theta \subset L_2$ we abbreviate the linear span of Θ as

$$S(\Theta) = \operatorname{span}\{\Theta\}.$$

Typically the multiresolution spaces S_j have the form

$$S_j = S(\Phi_j), \qquad \Phi_j = \{\phi_{j,k} : k \in \Delta_j\}, \tag{3.29}$$

for some finite index set Δ_j, where the set $\{\Phi_j\}_{j=j_0}^{\infty}$ is *uniformly stable* in the sense that

$$\|\mathbf{c}\|_{\ell_2(\Delta_j)} \;\sim\; \|\mathbf{c}^T \Phi_j\|_{L_2}, \qquad \mathbf{c} = \{c_k\}_{k \in \Delta_j} \in \ell_2(\Delta_j), \tag{3.30}$$

holds uniformly in j. Again we have used the shorthand notation

$$\mathbf{c}^T \Phi_j = \sum_{k \in \Delta_j} c_k \phi_{j,k}$$

and Φ_j denotes both the (column) vector containing the functions $\phi_{j,k}$ as well as the set of functions (3.29).
The collection Φ_j is called a *single scale basis* since all of its elements only live on one scale j. In the present context of multiresolution analysis, Φ_j is also called a *generator basis* or shortly *generators* of the multiresolution. We assume that the $\phi_{j,k}$ are compactly supported with

$$\operatorname{diam}(\operatorname{supp}\phi_{j,k}) \sim 2^{-j}. \tag{3.31}$$

It follows from (3.30) that they are scaled such that

$$\|\phi_{j,k}\|_{L_2} \sim 1 \tag{3.32}$$

holds. It is known that nestedness (3.28) together with stability (3.30) implies the existence of matrices $\mathbf{M}_{j,0} = (m^j_{r,k})_{r\in\Delta_{j+1},k\in\Delta_j}$ such that the two-scale relation

$$\phi_{j,k} = \sum_{r\in\Delta_{j+1}} m^j_{r,k}\phi_{j+1,r}, \quad k \in \Delta_j, \tag{3.33}$$

is satisfied. We can essentially simplify the subsequent presentation of the material by viewing (3.33) as a matrix–vector equation which then attains the compact form

$$\Phi_j = \mathbf{M}^T_{j,0}\Phi_{j+1}. \tag{3.34}$$

Any set of functions satisfying an equation of this form, the *refinement* or *two–scale relation*, will be called *refinable*.
Denoting by $[X, Y]$ the space of bounded linear operators from a normed linear space X into the normed linear space Y, one has that

$$\mathbf{M}_{j,0} \in [\ell_2(\Delta_j), \ell_2(\Delta_{j+1})]$$

is *uniformly sparse* which means that the number of entries in each row or column is uniformly bounded. Furthermore, one infers from (3.30) that

$$\|\mathbf{M}_{j,0}\| = \mathcal{O}(1), \quad j \geq j_0, \tag{3.35}$$

where the corresponding operator norm is defined as

$$\|\mathbf{M}_{j,0}\| := \sup_{\mathbf{c}\in\ell_2(\Delta_j),\ \|\mathbf{c}\|_{\ell_2(\Delta_j)}=1} \|\mathbf{M}_{j,0}\mathbf{c}\|_{\ell_2(\Delta_{j+1})}.$$

Since the union of $\mathcal{S}$ is dense in L_2, a basis for L_2 can be assembled from functions which span any complement between two successive spaces S_j and S_{j+1}, i.e.,

$$S(\Phi_{j+1}) = S(\Phi_j) \oplus S(\Psi_j) \tag{3.36}$$

where

$$\Psi_j = \{\psi_{j,k} : k \in \nabla_j\}, \qquad \nabla_j := \Delta_{j+1} \setminus \Delta_j. \tag{3.37}$$

The functions Ψ_j are called *wavelet functions* or shortly *wavelets* if, among other conditions detailed below, the union $\{\Phi_j \cup \Psi_j\}$ is still uniformly stable in the sense of (3.30). Since (3.36) implies $S(\Psi_j) \subset S(\Phi_{j+1})$, the functions in Ψ_j must also satisfy a matrix–vector relation of the form

$$\Psi_j = \mathbf{M}_{j,1}^T \Phi_{j+1} \tag{3.38}$$

with a matrix $\mathbf{M}_{j,1}$ of size $(\#\Delta_{j+1}) \times (\#\nabla_j)$. Furthermore, (3.36) is equivalent to the fact that the linear operator composed of $\mathbf{M}_{j,0}$ and $\mathbf{M}_{j,1}$,

$$\mathbf{M}_j = (\mathbf{M}_{j,0}, \mathbf{M}_{j,1}), \tag{3.39}$$

is *invertible* as a mapping from $\ell_2(\Delta_j \cup \nabla_j)$ onto $\ell_2(\Delta_{j+1})$. One can also show that the set $\{\Phi_j \cup \Psi_j\}$ is uniformly stable if and only if

$$\|\mathbf{M}_j\|, \|\mathbf{M}_j^{-1}\| = \mathcal{O}(1), \quad j \to \infty. \tag{3.40}$$

The particular cases that will be important for practical purposes are when not only $\mathbf{M}_{j,0}$ and $\mathbf{M}_{j,1}$ are uniformly sparse but also the inverse of $\mathbf{M}_j$. We denote this inverse by $\mathbf{G}_j$ and assume that it is split into

$$\mathbf{G}_j = \mathbf{M}_j^{-1} = \begin{pmatrix} \mathbf{G}_{j,0} \\ \mathbf{G}_{j,1} \end{pmatrix}. \tag{3.41}$$

A special situation occurs when

$$\mathbf{G}_j = \mathbf{M}_j^{-1} = \mathbf{M}_j^T$$

which corresponds to the case of L_2 *orthogonal wavelets* [Dau]. A systematic construction of more general $\mathbf{M}_j$, $\mathbf{G}_j$ for spline-wavelets can be found in [DKU], see also [D2] for more examples, including the hierarchical basis.

Thus, the identification of the functions Ψ_j which span the complement of $S(\Phi_j)$ in $S(\Phi_{j+1})$ is equivalent to completing a given refinement matrix $\mathbf{M}_{j,0}$ to an invertible matrix $\mathbf{M}_j$ in such a way that (3.40) is satisfied. Any such completion $\mathbf{M}_{j,1}$ is called *stable completion* of $\mathbf{M}_{j,0}$. In other words, the problem of the construction of compactly supported wavelets can equivalently be formulated as an algebraic problem of finding the (uniformly) sparse completion of a (uniformly) sparse matrix $\mathbf{M}_{j,0}$ in such a way that its inverse is also (uniformly) sparse. The fact that inverses of sparse matrices are usually dense elucidates the difficulties in the constructions.

The concept of stable completions has been introduced in [CDP] for which a special case is known as the *lifting scheme* [Sw]. Of course, constructions that yield compactly supported wavelets are particularly suited for computations in numerical analysis.

Combining the two–scale relations (3.34) and (3.38), one can see that $\mathbf{M}_j$ performs a change of bases in the space S_{j+1},

$$\begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{j,0}^T \\ \mathbf{M}_{j,1}^T \end{pmatrix} \Phi_{j+1} = \mathbf{M}_j^T \Phi_{j+1}. \tag{3.42}$$

Conversely, applying the inverse of $\mathbf{M}_j$ to both sides of (3.42) results in the *reconstruction identity*

$$\Phi_{j+1} = \mathbf{G}_j^T \begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \mathbf{G}_{j,0}^T \Phi_j + \mathbf{G}_{j,1}^T \Psi_j. \tag{3.43}$$

Fixing a *finest resolution level* J, one can repeat the decomposition (3.36) so that $S_J = S(\Phi_J)$ can be written in terms of the functions from the coarsest space supplied with the complement functions from all intermediate levels,

$$S(\Phi_J) = S(\Phi_{j_0}) \oplus \bigoplus_{j=j_0}^{J-1} S(\Psi_j). \tag{3.44}$$

Thus, every function $v \in S(\Phi_J)$ can be written in its *single–scale representation*

$$v = (\mathbf{c}_J)^T \Phi_J = \sum_{k \in \Delta_J} c_{J,k} \phi_{J,k} \tag{3.45}$$

as well as in its *multiscale form*

$$v = (\mathbf{c}_{j_0})^T \Phi_{j_0} + (\mathbf{d}_{j_0})^T \Psi_{j_0} + \cdots + (\mathbf{d}_{J-1})^T \Psi_{J-1} \tag{3.46}$$

with respect to the *multiscale* or *wavelet basis*

$$\Psi^J := \Phi_{j_0} \cup \bigcup_{j=j_0}^{J-1} \Psi_j =: \bigcup_{j=j_0-1}^{J-1} \Psi_j. \tag{3.47}$$

Often the single–scale representation of a function may be easier to compute and evaluate while the multiscale representation allows one to separate features of the underlying function characterized by different length scales. Therefore since both representations are advantageous, it is useful to determine the transformation between the two representations, commonly referred to as the *Wavelet Transform*,

$$\mathbf{T}_J : \ell_2(\Delta_j) \to \ell_2(\Delta_j), \qquad \mathbf{d}^J \mapsto \mathbf{c}_J, \tag{3.48}$$

where

$$\mathbf{d}^J := (\mathbf{c}_{j_0}, \mathbf{d}_{j_0}, \ldots, \mathbf{d}_{J-1})^T.$$

The previous relations (3.42) and (3.43) indicate that this will involve the matrices $\mathbf{M}_j$ and $\mathbf{G}_j$. In fact, $\mathbf{T}_J$ has the representation

$$\mathbf{T}_J = \mathbf{T}_{J,J-1} \cdots \mathbf{T}_{J,j_0}, \tag{3.49}$$

where each factor has the form

$$\mathbf{T}_{J,j} := \begin{pmatrix} \mathbf{M}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^{(\#\Delta_J - \#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J)\times(\#\Delta_J)}. \tag{3.50}$$

Schematically $\mathbf{T}_J$ can be visualized as a pyramid scheme

$$\begin{array}{ccccccccccc} & & \mathbf{M}_{j_0,0} & & \mathbf{M}_{j_0+1,0} & & & & \mathbf{M}_{J-1,0} & & \\ \mathbf{c}_{j_0} & \longrightarrow & \mathbf{c}_{j_0+1} & \longrightarrow & \mathbf{c}_{j_0+2} & \longrightarrow \cdots & \mathbf{c}_{J-1} & \longrightarrow & \mathbf{c}_J & & \\ & \nearrow^{\mathbf{M}_{j_0,1}} & & \nearrow^{\mathbf{M}_{j_0+1,1}} & & \nearrow \cdots & & \nearrow^{\mathbf{M}_{J-1,1}} & & & \\ \mathbf{d}_{j_0} & & \mathbf{d}_{j_0+1} & & \mathbf{d}_{j_0+2} & & \mathbf{d}_{J-1} & & & & \end{array} . \tag{3.51}$$

Accordingly, the inverse transform $\mathbf{T}_J^{-1}$ can also be written in product structure (3.49) in reverse order involving the matrices $\mathbf{G}_j$ as follows:

$$\mathbf{T}_J^{-1} = \mathbf{T}_{J,j_0}^{-1} \cdots \mathbf{T}_{J,J-1}^{-1}, \tag{3.52}$$

where each factor has the form

$$\mathbf{T}_{J,j}^{-1} := \begin{pmatrix} \mathbf{G}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^{(\#\Delta_J - \#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J)\times(\#\Delta_J)}. \tag{3.53}$$

The corresponding pyramid scheme is then

$$\begin{array}{ccccccccccc} & & \mathbf{G}_{J-1,0} & & \mathbf{G}_{J-2,0} & & & & \mathbf{G}_{j_0,0} & & \\ \mathbf{c}_J & \longrightarrow & \mathbf{c}_{J-1} & \longrightarrow & \mathbf{c}_{J-2} & \longrightarrow \cdots & & \longrightarrow & \mathbf{c}_{j_0} & & \\ & \searrow^{\mathbf{G}_{J-1,1}} & & \searrow^{\mathbf{G}_{J-2,1}} & & \searrow \cdots & & \searrow^{\mathbf{G}_{j_0,1}} & & & \\ & & \mathbf{d}_{J-1} & & \mathbf{d}_{J-2} & & \mathbf{d}_{J-1} & & \mathbf{d}_{j_0} & & \end{array} . \tag{3.54}$$

Remark 3.2.
Property (3.40) and the fact that $\mathbf{M}_j$ and $\mathbf{G}_j$ can be applied in $(\#\Delta_{j+1})$ operations uniformly in j implies that the complexity of applying $\mathbf{T}_J$ or $\mathbf{T}_J^{-1}$ using the pyramid scheme is of order $\mathcal{O}(\#\Delta_J) = \mathcal{O}(\dim S_J)$ uniformly in J. For this reason, $\mathbf{T}_J$ is called the *Fast Wavelet Transform* (FWT). Note that there is no need to explicitly assemble $\mathbf{T}_J$ or $\mathbf{T}_J^{-1}$. ∎

In Table 3.1 spectral condition numbers for the Fast Wavelet Transform (FWT) for different constructions of biorthogonal wavelets on the interval computed in [P] are displayed.
Since $\cup_{j\geq j_0} S_j$ is dense in L_2, a basis for the whole space L_2 is obtained when letting $J \to \infty$ in (3.47),

$$\begin{aligned} \Psi &:= \bigcup_{j=j_0-1}^{\infty} \Psi_j = \{\psi_{j,k} : (j,k) \in I\!I\}, \qquad \Psi_{j_0-1} := \Phi_{j_0} \\ I\!I &:= \{\{j_0\} \times \Delta_{j_0}\} \cup \bigcup_{j=j_0}^{\infty} \{\{j\} \times \nabla_j\}. \end{aligned} \tag{3.55}$$

The next theorem from [D1] illustrates the relation between Ψ and $\mathbf{T}_J$.

Theorem 3.1. *The multiscale transformations* $\mathbf{T}_J$ *are well–conditioned in the sense*

$$\|\mathbf{T}_J\|, \|\mathbf{T}_J^{-1}\| = \mathcal{O}(1), \quad J \geq j_0, \tag{3.56}$$

if and only if the collection Ψ *defined by (3.55) is a Riesz basis for* L_2*, i.e., every* $v \in L_2$ *has unique expansions*

$$v = \sum_{j=j_0-1}^{\infty} \langle v, \tilde{\Psi}_j \rangle \Psi_j = \sum_{j=j_0-1}^{\infty} \langle v, \Psi_j \rangle \tilde{\Psi}_j, \tag{3.57}$$

where $\tilde{\Psi}$*, defined analogously as in (3.55), is also a Riesz basis for* L_2 *which is biorthogonal or dual to* Ψ*,*

$$\langle \Psi, \tilde{\Psi} \rangle = \mathbf{I} \tag{3.58}$$

such that

$$\|v\|_{L_2} \sim \|\langle \tilde{\Psi}, v \rangle\|_{\ell_2(I\!I)} \sim \|\langle \Psi, v \rangle\|_{\ell_2(I\!I)}. \tag{3.59}$$

Next we briefly explain how the functions in $\tilde{\Psi}$, called *wavelets dual to* Ψ, or *dual wavelets*, can be determined. Assume that there is a second multiresolution $\tilde{\mathcal{S}}$ of L_2 satisfying (3.28) where

$$\tilde{S}_j = S(\tilde{\Phi}_j), \qquad \tilde{\Phi}_j = \{\tilde{\phi}_{j,k} : k \in \Delta_j\} \tag{3.60}$$

and $\{\tilde{\Phi}_j\}_{j=j_0}^{\infty}$ is uniformly stable in j in the sense of (3.30). Let the functions in $\tilde{\Phi}_j$ also have compact support satisfying (3.31). Furthermore, suppose that the biorthogonality conditions

$$\langle \Phi_j, \tilde{\Phi}_j \rangle = \mathbf{I} \tag{3.61}$$

hold. We will often refer to Φ_j as the *primal* and to $\tilde{\Phi}_j$ as the *dual generators*. The nestedness of the $\tilde{S}_j$ and the stability again implies that $\tilde{\Phi}_j$ is refinable with some matrix $\tilde{\mathbf{M}}_{j,0}$, similar to (3.34),

$$\tilde{\Phi}_j = \tilde{\mathbf{M}}_{j,0}^T \tilde{\Phi}_{j+1}. \tag{3.62}$$

The problem of determining biorthogonal wavelets now consists of finding bases $\Psi_j, \tilde{\Psi}_j$ for the complements of $S(\Phi_j)$ in $S(\Phi_{j+1})$, and of $S(\tilde{\Phi}_j)$ in $S(\tilde{\Phi}_{j+1})$, such that

$$S(\Phi_j) \perp S(\tilde{\Psi}_j), \qquad S(\tilde{\Phi}_j) \perp S(\Psi_j) \tag{3.63}$$

and

$$S(\Psi_j) \perp S(\tilde{\Psi}_r), \quad j \neq r, \tag{3.64}$$

holds. The connection between the concept of stable completions and the dual generators and wavelets is made by the following result which is a special case from [CDP].

Proposition 3.2. *Suppose that the biorthogonal collections* $\{\Phi_j\}_{j=j_0}^{\infty}$ *and* $\{\tilde{\Phi}_j\}_{j=j_0}^{\infty}$ *are both uniformly stable and refinable with refinement matrices* $\mathbf{M}_{j,0}$, $\tilde{\mathbf{M}}_{j,0}$, *i.e.,*

$$\Phi_j = \mathbf{M}_{j,0}^T \Phi_{j+1}, \qquad \tilde{\Phi}_j = \tilde{\mathbf{M}}_{j,0}^T \tilde{\Phi}_{j+1}, \tag{3.65}$$

and satisfy the duality condition (3.61). Assume that $\check{\mathbf{M}}_{j,1}$ *is any stable completion of* $\mathbf{M}_{j,0}$ *such that*

$$\check{\mathbf{M}}_j := (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1}) = \check{\mathbf{G}}_j^{-1} \tag{3.66}$$

satisfies (3.40).
Then

$$\mathbf{M}_{j,1} := (\mathbf{I} - \mathbf{M}_{j,0}\tilde{\mathbf{M}}_{j,0}^T)\check{\mathbf{M}}_{j,1} \tag{3.67}$$

is also a stable completion of $\mathbf{M}_{j,0}$, *and* $\mathbf{G}_j = \mathbf{M}_j^{-1} = (\mathbf{M}_{j,0}, \mathbf{M}_{j,1})^{-1}$ *has the form*

$$\mathbf{G}_j = \begin{pmatrix} \tilde{\mathbf{M}}_{j,0}^T \\ \check{\mathbf{G}}_{j,1} \end{pmatrix}. \tag{3.68}$$

Moreover, the collections of functions

$$\Psi_j := \mathbf{M}_{j,1}^T \Phi_{j+1}, \qquad \tilde{\Psi}_j := \check{\mathbf{G}}_{j,1} \tilde{\Phi}_{j+1} \tag{3.69}$$

form biorthogonal systems,

$$\langle \Psi_j, \tilde{\Psi}_j \rangle = \mathbf{I}, \qquad \langle \Psi_j, \tilde{\Phi}_j \rangle = \langle \Phi_j, \tilde{\Psi}_j \rangle = \mathbf{0}, \tag{3.70}$$

so that

$$S(\Psi_j) \perp S(\tilde{\Psi}_r), \quad j \neq r, \qquad S(\Phi_j) \perp S(\tilde{\Psi}_j), \quad S(\tilde{\Phi}_j) \perp S(\Psi_j). \tag{3.71}$$

In particular, the relations (3.61), (3.70) imply that the collections

$$\Psi = \bigcup_{j=j_0-1}^{\infty} \Psi_j, \qquad \tilde{\Psi} := \bigcup_{j=j_0-1}^{\infty} \tilde{\Psi}_j := \tilde{\Phi}_{j_0} \cup \bigcup_{j=j_0}^{\infty} \tilde{\Psi}_j \tag{3.72}$$

are biorthogonal,

$$\langle \Psi, \tilde{\Psi} \rangle = \mathbf{I}. \tag{3.73}$$

Remark 3.3.
It is important to note that the properties needed in addition to (3.73) in order to ensure (3.59) are neither properties of the complements nor of their bases $\Psi, \tilde{\Psi}$ but of the multiresolution sequences $\mathcal{S}$ and $\tilde{\mathcal{S}}$. These can be phrased as approximation and regularity properties and appear in Theorem 3.2. ∎

We briefly recall yet another useful point of view. The operators

$$\begin{aligned} P_j v &:= \langle v, \tilde{\Phi}_j\rangle \Phi_j = \langle v, \tilde{\Psi}^j\rangle \Psi^j = \langle v, \tilde{\Phi}_{j_0}\rangle \Phi_{j_0} + \sum_{r=j_0}^{j-1} \langle v, \tilde{\Psi}_r\rangle \Psi_r \\ P'_j v &:= \langle v, \Phi_j\rangle \tilde{\Phi}_j = \langle v, \Psi^j\rangle \tilde{\Psi}^j = \langle v, \Phi_{j_0}\rangle \tilde{\Phi}_{j_0} + \sum_{r=j_0}^{j-1} \langle v, \Psi_r\rangle \tilde{\Psi}_r \end{aligned} \tag{3.74}$$

are projectors onto

$$S(\Phi_j) = S(\Psi^j) \qquad \text{and} \qquad S(\tilde{\Phi}_j) = S(\tilde{\Psi}^j) \tag{3.75}$$

respectively, which satisfy

$$P_r P_j = P_r, \quad P'_r P'_j = P'_r, \qquad r \le j. \tag{3.76}$$

Remark 3.4.
Let $\{\Phi_j\}_{j=j_0}^{\infty}$ be uniformly stable. The P_j defined by (3.74) are uniformly bounded if and only if $\{\tilde{\Phi}_j\}_{j=j_0}^{\infty}$ is also uniformly stable. Moreover, the P_j satisfy (3.76) if and only if the $\tilde{\Phi}_j$ are refinable as well. Note that then (3.61) implies

$$\mathbf{M}_{j,0}^T \tilde{\mathbf{M}}_{j,0} = \mathbf{I}. \tag{3.77}$$

■

In terms of the projectors, the uniform stability of the complement bases Ψ_j, $\tilde{\Psi}_j$ means that

$$\|(P_{j+1} - P_j)v\|_{L_2} \sim \|\langle \tilde{\Psi}_j, v\rangle\|_{\ell_2(\nabla_j)}, \quad \|(P'_{j+1} - P'_j)v\|_{L_2} \sim \|\langle \Psi_j, v\rangle\|_{\ell_2(\nabla_j)}, \tag{3.78}$$

so that the L_2 norm equivalence (3.59) is equivalent to

$$\|v\|_{L_2}^2 \sim \sum_{j=j_0}^{\infty} \|(P_j - P_{j-1})v\|_{L_2}^2 \sim \sum_{j=j_0}^{\infty} \|(P'_j - P'_{j-1})v\|_{L_2}^2 \tag{3.79}$$

for any $v \in L_2$, where $P_{j_0-1} = P'_{j_0-1} := 0$.
The whole concept derived so far is based on the availability of both Φ_j *and* $\tilde{\Phi}_j$. It should be pointed out that in the algorithms one does not actually need $\tilde{\Phi}_j$ explicitly for computations.
We recall next results that guarantee norm equivalences of the type (3.3) for Sobolev spaces.

Multiresolution of Sobolev Spaces
Now let $\mathcal{S}$ be a multiresolution sequence consisting of closed subspaces of H^s with the property (3.28) whose union is dense in H^s. The following result from [D1] ensures under which conditions norm equivalences hold for the H^s–norm.

Theorem 3.2. *Let* $\{\Phi_j\}_{j=j_0}^{\infty}$ *and* $\{\tilde{\Phi}_j\}_{j=j_0}^{\infty}$ *be uniformly stable, refinable, biorthogonal collections and let* $P_j : H^s \to S(\Phi_j)$ *be defined by (3.74).*
If the Jackson-type estimate

$$\inf_{v_j \in S_j} \|v - v_j\|_{L_2} \lesssim 2^{-sj} \|v\|_{H^s}, \quad v \in H^s,\ 0 < s \le \bar{d}, \tag{3.80}$$

and the Bernstein inequality

$$\|v_j\|_{H^s} \lesssim 2^{sj} \|v_j\|_{L_2}, \quad v_j \in S_j,\ s < \bar{t}, \tag{3.81}$$

hold for

$$S_j = \left\{ \begin{matrix} S(\Phi_j) \\ S(\tilde{\Phi}_j) \end{matrix} \right\} \text{ with order } \bar{d} = \left\{ \begin{matrix} d \\ \tilde{d} \end{matrix} \right\} \text{ and } \bar{t} = \left\{ \begin{matrix} t \\ \tilde{t} \end{matrix} \right\}, \tag{3.82}$$

then for

$$0 < \sigma := \min\{d, t\}, \qquad 0 < \tilde{\sigma} := \min\{\tilde{d}, \tilde{t}\}, \tag{3.83}$$

one has

$$\|v\|_{H^s}^2 \sim \sum_{j=j_0}^{\infty} 2^{2sj} \|(P_j - P_{j-1})v\|_{L_2}^2, \quad s \in (-\tilde{\sigma}, \sigma). \tag{3.84}$$

Recall that we always write $H^s = (H^{-s})'$ for $s < 0$.
The regularity of $\mathcal{S}$ and $\tilde{\mathcal{S}}$ is characterized by

$$t := \sup\{s : S(\Phi_j) \subset H^s,\ j \ge j_0\}, \qquad \tilde{t} := \sup\{s : S(\tilde{\Phi}_j) \subset H^s,\ j \ge j_0\}. \tag{3.85}$$

Recalling the representation (3.78), we can immediately derive the following fact.

Corollary 3.1. *Suppose that the assumptions in Theorem 3.2 hold. Then we have the norm equivalence*

$$\|v\|_{H^s}^2 \sim \sum_{j=j_0-1}^{\infty} 2^{2sj} \|\langle \tilde{\Psi}_j, v\rangle\|_{\ell_2(\nabla_j)}^2, \quad s \in (-\tilde{\sigma}, \sigma). \tag{3.86}$$

In particular for $s = 0$ the Riesz basis property (3.59) of the Ψ, $\tilde{\Psi}$ relative to L_2 is recovered. For many applications it suffices to have (3.84) or (3.86) only for certain $s > 0$ for which one only requires (3.80) and (3.81) for $\{\Phi_j\}_{j=j_0}^{\infty}$. The Jackson estimates (3.80) of order $\tilde{d}$ for $S(\tilde{\Phi}_j)$ imply the cancellation properties (CP) (3.6), for example see [D4].

Remark 3.5.
When the wavelets live on $\Omega \subset \mathbb{R}^n$, (3.80) means that all polynomials up to order $\tilde{d}$ are contained in $S(\tilde{\Phi}_j)$. One also says that $S(\tilde{\Phi}_j)$ is *exact* of order $\tilde{d}$. On account of (3.58), this implies that the wavelets $\psi_{j,k}$ are orthogonal to polynomials up to order $\tilde{d}$ or have $\tilde{d}$'th order *vanishing moments*. By Taylor expansion, this in turn yields (3.6). ∎

Later we will use the following generalization of the discrete norms (3.79). For $s \in \mathbb{R}$ let

$$\|\!|v|\!\|_s := \left(\sum_{j=j_0}^{\infty} 2^{2sj} \|(P_j - P_{j-1})v\|_{L_2}^2 \right)^{1/2} \tag{3.87}$$

which by the relations (3.78) is also equivalent to

$$|\!|v|\!|_s := \left(\sum_{j=j_0-1}^{\infty} 2^{2sj} \|\langle \tilde{\Psi}_j, v\rangle\|_{\ell_2(\nabla_j)}^2 \right)^{1/2}. \tag{3.88}$$

In this notation, (3.84) and (3.86) read

$$\|v\|_{H^s} \sim \|\!|v|\!\|_s \sim |\!|v|\!|_s. \tag{3.89}$$

In terms of such discrete norms, Jackson and Bernstein estimates hold with constants equal to one [K2], which turns out to be useful later in Section 4.2.

Lemma 3.1. *Let $\{\Phi_j\}_{j=j_0}^{\infty}$ and $\{\tilde{\Phi}_j\}_{j=j_0}^{\infty}$ be uniformly stable, refinable, biorthogonal collections and let the P_j be defined by (3.74). Then the estimates*

$$|\!|v - P_j v|\!|_{s'} \le 2^{-(j+1)(s-s')} |\!|v|\!|_s, \qquad v \in H^s, \quad s' \le s \le d, \tag{3.90}$$

and

$$|\!|v_j|\!|_s \le 2^{j(s-s')} |\!|v_j|\!|_{s'}, \qquad v_j \in S(\Phi_j), \quad s' \le s \le d, \tag{3.91}$$

are valid, and correspondingly for the dual side.

The same results hold for the norm $\|\!| \cdot |\!\|$ defined in (3.87).

Reverse Cauchy–Schwarz Inequalities

The biorthogonality condition (3.61) together with direct and inverse estimates implies the following reverse Cauchy–Schwarz inequalities for finite–dimensional spaces [DK2]. It will be one essential ingredient for the discussion of the LBB condition in Section 4.2.

Lemma 3.2. *Let the assumptions in Theorem 3.2 be valid such that the norm equivalence (3.84) holds for $(-\tilde{\sigma}, \sigma)$ with $\sigma, \tilde{\sigma}$ defined by (3.83). Then for any $v \in S(\Phi_j)$ there exists some $\tilde{v}^* = \tilde{v}^*(v) \in S(\tilde{\Phi}_j)$ such that*

$$\|v\|_{H^s} \|\tilde{v}^*\|_{H^{-s}} \lesssim \langle v, \tilde{v}^* \rangle \tag{3.92}$$

for any $0 \le s < \min(\sigma, \tilde{\sigma})$.

The proof of this result given in [DK2] for $s = 1/2$, in terms of the projectors P_j defined in (3.74) and corresponding duals P_j', immediately carries over to more general s. Recalling the representation (3.75) in terms of wavelets, the reverse Cauchy inequality (3.92) attains the following sharp form.

Lemma 3.3. *[K2] Let the assumptions of Lemma 3.1 hold. Then for every $v \in S(\Phi_j)$ there exists some $\tilde{v}^* = \tilde{v}^*(v) \in S(\tilde{\Phi}_j)$ such that*

$$|v|_s \, |\tilde{v}^*|_{-s} \;=\; \langle v, \tilde{v}^* \rangle \tag{3.93}$$

for any $0 \le s \le \min(\sigma, \tilde{\sigma})$.

Proof. Every $v \in S(\Phi_j)$ can be written as

$$v = \sum_{r=j_0-1}^{j-1} 2^{sr} \sum_{k \in \nabla_r} v_{r,k} \psi_{r,k}.$$

Setting now

$$\tilde{v}^* := \sum_{r=j_0-1}^{j-1} 2^{-sr} \sum_{k \in \nabla_r} v_{r,k} \tilde{\psi}_{r,k}$$

with the same coefficients $v_{j,k}$, the definition of $|\cdot|_s$ yields by biorthogonality (3.73)

$$|v|_s \, |\tilde{v}^*|_{-s} \;=\; \sum_{r=j_0-1}^{j-1} \sum_{k \in \nabla_r} |v_{j,k}|^2.$$

Combining this with the observation

$$\langle v, \tilde{v}^* \rangle \;=\; \sum_{r=j_0-1}^{j-1} \sum_{k \in \nabla_r} |v_{j,k}|^2$$

confirms (3.93). □

Remark 3.6.
The previous proof reveals that the identity (3.93) is also true for elements from infinite–dimensional spaces H^s and $(H^s)'$ for which Ψ and $\tilde{\Psi}$ are Riesz bases. ▪

Biorthogonal Wavelets on $\mathbb{R}$
The construction of biorthogonal spline-wavelets on $\mathbb{R}$ from [CDF] for $L_2 = L_2(\mathbb{R})$ employs the multiresolution framework introduced at the beginning of this section. There the $\phi_{j,k}$ are generated through the dilates and translates of a single function $\phi \in L_2$,

$$\phi_{j,k} = 2^{j/2} \phi(2^j \cdot - k). \tag{3.94}$$

This corresponds to the idea of a *uniform* virtual underlying grid, explaining the terminology *uniform refinements*. B–Splines on uniform grids are known to satisfy refinement relations (3.33) in addition to being compactly supported and having L_2–stable integer translates. For computations, they have

the additional advantage that they can be expressed as piecewise polynomials. In the context of variational formulations for second order boundary value problems, a well–used example are the nodal finite elements $\phi_{j,k}$ generated by the cardinal B–Spline of order two, i.e., the piecewise linear continuous function commonly called the 'hat function'. For cardinal B–Splines as generators, a whole class of dual generators $\tilde{\phi}_{j,k}$ (of arbitrary smoothness at the expense of larger supports) can be constructed which are also generated by one single function $\tilde{\phi}$ through translates and dilates. By Fourier techniques, one can construct from $\phi, \tilde{\phi}$ then a pair of biorthogonal wavelets $\psi, \tilde{\psi}$ whose dilates and translates built as in (3.94) constitute Riesz bases for $L_2(\mathbb{R})$.
By taking tensor products of these functions, of course, one can generate biorthogonal wavelet bases for $L_2(\mathbb{R}^n)$.

Biorthogonal Wavelets on Domains
Now some constructions that exist have as a core ingredient tensor products of one-dimensional wavelets on an *interval* derived from the biorthogonal wavelets from [CDF] on $\mathbb{R}$. On finite intervals in $\mathbb{R}$, the corresponding constructions are usually based on keeping the elements of $\Phi_j, \tilde{\Phi}_j$ supported *inside* the interval while modifying those translates overlapping the end points of the interval so as to preserve a desired degree of polynomial exactness. A general detailed construction satisfying all these requirements has been proposed in [DKU]. Here, just the main ideas for constructing a biorthogonal pair $\Phi_j, \tilde{\Phi}_j$ and corresponding wavelets satisfying the above requirements are sketched, where we apply the techniques derived at the beginning of this section.
We start out with those functions from two collections of biorthogonal generators $\Phi_j^{\mathbb{R}}, \tilde{\Phi}_j^{\mathbb{R}}$ for some fixed $j \geq j_0$ living on the whole real line whose support has nonempty intersection with the interval $(0,1)$. In order to treat the boundary effects separately, we assumed that the coarsest resolution level j_0 is large enough so that, in view of (3.31), functions overlapping one end of the interval vanish at the other. One then leaves as many functions from the collection $\Phi_j^{\mathbb{R}}, \tilde{\Phi}_j^{\mathbb{R}}$ living in the interior of the interval untouched and modifies only those near the interval ends. Note that keeping just the restrictions to the interval of those translates overlapping the end points would destroy stability (and also the cardinality of the primal and dual basis functions living on $(0,1)$ since their supports do not have the same size). Therefore, modifications at the end points are necessary; also, just discarding them from the collections (3.29), (3.60) would produce an error near the end points. The basic idea is essentially the same for all constructions of orthogonal and biorthogonal wavelets on $\mathbb{R}$ adapted to an interval. Namely, one takes *fixed* linear combinations of all functions in $\Phi_j^{\mathbb{R}}, \tilde{\Phi}_j^{\mathbb{R}}$ living near the ends of the interval in such a way that monomials up to the exactness order are reproduced there and such that the generator bases have the same cardinality. Because of the boundary modifications, the collections of generators are no longer biorthogonal there. However, one can show in the case of cardinal B–Splines as primal generators

(which is a widely used class for numerical analysis) that biorthogonalization is indeed possible. This yields collections denoted by $\Phi_j^{(0,1)}, \tilde{\Phi}_j^{(0,1)}$ which then satisfy (3.61) on $(0,1)$ and all assumptions required in Proposition 3.2.
For the construction of corresponding wavelets, first an *initial* stable completion $\check{\mathbf{M}}_{j,1}$ is computed by applying Gaussian eliminations to factor $\mathbf{M}_{j,0}$ and then to find a uniformly stable inverse of $\check{\mathbf{M}}_j$. Here we exploit that for cardinal B–Splines as generators the refinement matrices $\mathbf{M}_{j,0}$ are totally positive. Thus, they can be stably decomposed by Gaussian elimination without pivoting. Application of Proposition 3.2 then gives the corresponding biorthogonal wavelets $\Psi_j^{(0,1)}, \tilde{\Psi}_j^{(0,1)}$ on $(0,1)$ which satisfy the requirements in Corollary 3.1. It turns out that these wavelets coincide in the interior of the interval again with those on all of $\mathbb{R}$ from [CDF]. An example of the primal wavelets for $d = 2$ generated by piecewise linear continuous functions is displayed in Figure 3.1 on the left. After constructing these basic versions, one can then perform local transformations near the ends of the interval in order to improve the condition or L_2 stability constants, see [Bu, P] for corresponding results and numerical examples.
We display spectral condition numbers for the FWT for two different constructions of biorthogonal wavelets on the interval computed in [P] in Table 3.1. The first column denotes the finest level on which the spectral condition numbers of the FWT are computed. The next column contains the numbers for the construction of biorthogonal spline-wavelets on the interval from [DKU] for the case $d = 2, \tilde{d} = 4$ while the last column displays the numbers for a scaled version derived in [Bu]. Later in Section 4.1 we will see how the transformation $\mathbf{T}_J$ is used for preconditioning.

j	$\kappa_2(\mathbf{T}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{T}_{\mathrm{B}})$
4	4.743e+00	4.640e+00
5	6.221e+00	6.024e+00
6	8.154e+00	6.860e+00
7	9.473e+00	7.396e+00
8	1.023e+01	7.707e+00
9	1.064e+01	7.876e+00
10	1.086e+01	7.965e+00

j	$\kappa_2(\mathbf{T}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{T}_{\mathrm{B}})$
11	1.097e+01	8.011e+00
12	1.103e+01	8.034e+00
13	1.106e+01	8.046e+00
14	1.107e+01	8.051e+00
15	1.108e+01	8.054e+00
16	1.108e+01	8.056e+00

Table 3.1. Computed spectral condition numbers [P] for the Fast Wavelet Transform for different constructions of biorthogonal wavelets on the interval [Bu, DKU].

Also along these lines, biorthogonal generators and wavelets with homogeneous (Dirichlet) boundary conditions can be constructed. Since the $\Phi_j^{(0,1)}$ are locally near the boundary monomials which all vanish at $0, 1$ except for one, removing the one from $\Phi_j^{(0,1)}$ which corresponds to the constant function produces a collection of generators with homogeneous boundary conditions at $0, 1$. In order for the moment conditions (3.6) still to hold for the Ψ_j,

the dual generators have to have *complementary* boundary conditions. A corresponding construction has been carried out in [DS1] and implemented in [Bu]. Homogeneous boundary conditions of higher order can be generated accordingly.

By taking tensor products of the wavelets on $(0,1)$, in this manner biorthogonal wavelets for Sobolev spaces on $(0,1)^n$ with or without homogeneous boundary conditions are obtained. This construction can be further extended to any other domain or manifold which is the image of a regular parametric mapping of the unit cube. Some results on the construction of wavelets on manifolds are summarized in [D3]. There are essentially two approaches. The first idea is based on domain decomposition and consists of 'glueing' generators across interelement boundaries, see, e.g., [CTU, DS2]. These approaches all have in common that the norm equivalences (3.86) for $H^s = H^s(\Gamma)$ can be shown to hold only for the range $-1/2 < s < 3/2$, due to the fact that duality arguments apply only for this range because of the nature of a modified inner product to which biorthogonality refers. The other approach which overcomes the above limitations on the ranges for which the norm equivalences hold has been developed in [DS3] based on previous characterizations of function spaces as Cartesian products from [CF]. The construction in [DS3] has been optimized and implemented to construct wavelet bases on the sphere in [KS, S], see Figure 3.1.

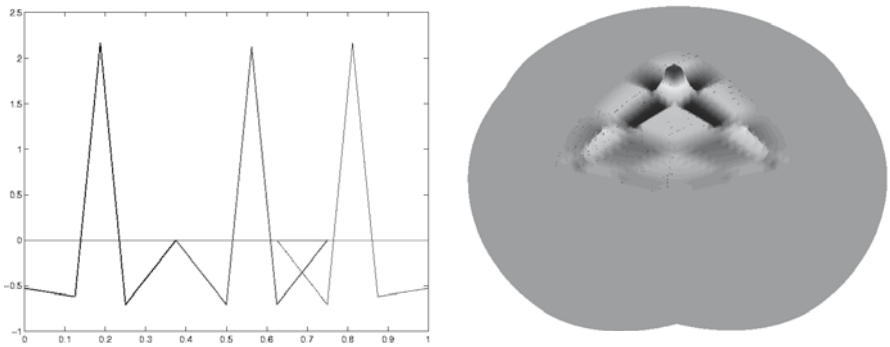


Figure 3.1. Primal wavelets for $d = 2$ on $[0,1]$ (left) and on a sphere (right) from [S].

Of course, there are also other different approaches to constructing wavelet bases with the above properties without using tensor products. On triangles a construction of biorthogonal spline-wavelets has been introduced by [Stv] and implemented in two spatial dimensions with an application to the numerical solution of a linear elliptic boundary value problem in [Kr].

4 Problems in Wavelet Coordinates

4.1 Elliptic Boundary Value Problems

We now consider the wavelet representation of the elliptic boundary value problem from Section 2.2. For $\mathcal{H}$ given by (2.8) or (2.9) let $\Psi_{\mathcal{H}}$ be a wavelet basis with corresponding dual $\tilde{\Psi}_{\mathcal{H}}$ which satisfies the properties (R), (L) and (CP) from Section 3.1. Following the recipe from Section 3.3, expanding $y = \mathbf{y}^T\Psi_{\mathcal{H}}$, $f = \mathbf{f}^T\tilde{\Psi}_{\mathcal{H}}$ and recalling (2.12), the wavelet representation of the elliptic boundary value problem (2.14) is given by

$$\mathbf{A}\mathbf{y} = \mathbf{f} \tag{4.1}$$

where

$$\mathbf{A} := a(\Psi_{\mathcal{H}}, \Psi_{\mathcal{H}}), \qquad \mathbf{f} := \langle \Psi_{\mathcal{H}}, f\rangle. \tag{4.2}$$

Then the mapping property (2.13) and the Riesz basis property (R) yield the following fact.

Proposition 4.1. *The infinite matrix* $\mathbf{A}$ *is a boundedly invertible mapping from* $\ell_2 = \ell_2(I\!I_{\mathcal{H}})$ *into itself, and there exists finite positive constants* $c_{\mathbf{A}} \leq C_{\mathbf{A}}$ *such that*

$$c_{\mathbf{A}}\|\mathbf{v}\| \leq \|\mathbf{A}\mathbf{v}\| \leq C_{\mathbf{A}}\|\mathbf{v}\|, \qquad \mathbf{v} \in \ell_2(I\!I_{\mathcal{H}}). \tag{4.3}$$

Proof. For any $v \in \mathcal{H}$ with coefficient vector $\mathbf{v} \in \ell_2$, we have by the lower estimates in (3.3), (2.13) and the upper inequality in (3.8), respectively,

$$\|\mathbf{v}\| \leq c_{\mathcal{H}}^{-1}\|v\|_{\mathcal{H}} \leq c_{\mathcal{H}}^{-1}c_A^{-1}\|Av\|_{\mathcal{H}'} = c_{\mathcal{H}}^{-1}c_A^{-1}\|(\mathbf{A}\mathbf{v})^T\tilde{\Psi}_{\mathcal{H}}\|_{\mathcal{H}'} \leq c_{\mathcal{H}}^{-2}c_A^{-1}\|\mathbf{A}\mathbf{v}\|$$

where we have used the wavelet representation (3.25) for A. Likewise, the converse estimate

$$\|\mathbf{A}\mathbf{v}\| \leq C_{\mathcal{H}}\|Av\|_{\mathcal{H}'} \leq C_{\mathcal{H}}C_A\|v\|_{\mathcal{H}} \leq C_{\mathcal{H}}^2C_A\|\mathbf{v}\|$$

follows by the lower inequality in (3.8) and the upper estimates in (2.13) and (3.3). The constants appearing in (4.3) are therefore identified as $c_{\mathbf{A}} := c_{\mathcal{H}}^2c_A$ and $C_{\mathbf{A}} := c_{\mathcal{H}}^2C_A$. □

In the present situation where $\mathbf{A}$ is defined via the elliptic bilinear form $a(\cdot,\cdot)$, Proposition 4.1 implies the following result with respect to *preconditioning*. For $I\!I = I\!I_{\mathcal{H}}$ let the symbol Λ denote *any* finite subset of the index set $I\!I$. For the corresponding set of wavelets $\Psi_\Lambda := \{\psi_\lambda : \lambda \in \Lambda\}$ denote by $S_\Lambda := \operatorname{span}\Psi_\Lambda$ the respective finite-dimensional subspace of $\mathcal{H}$. For the wavelet representation of A in terms of Ψ_Λ,

$$\mathbf{A}_\Lambda := a(\Psi_\Lambda, \Psi_\Lambda), \tag{4.4}$$

we obtain the following result.

Proposition 4.2. *If $a(\cdot,\cdot)$ is $\mathcal{H}$-elliptic according to (2.11), the finite matrix $\mathbf{A}_\Lambda$ is symmetric positive definite and its spectral condition number is bounded uniformly in Λ, i.e.,*

$$\kappa_2(\mathbf{A}_\Lambda) \ \leq \frac{C_{\mathbf{A}}}{c_{\mathbf{A}}}, \tag{4.5}$$

where $c_{\mathbf{A}}, C_{\mathbf{A}}$ are the constants from (4.3).

Proof. Clearly, since $\mathbf{A}_\Lambda$ is just a finite section of $\mathbf{A}$, we have $\|\mathbf{A}_\Lambda\| \leq \|\mathbf{A}\|$. On the other hand, by assumption, $a(\cdot,\cdot)$ is $\mathcal{H}$-elliptic which implies that $a(\cdot,\cdot)$ is also elliptic on every finite subspace $S_\Lambda \subset \mathcal{H}$. Thus, we infer $\|\mathbf{A}_\Lambda^{-1}\| \leq \|\mathbf{A}^{-1}\|$, and we have

$$c_{\mathbf{A}}\|\mathbf{v}_\Lambda\| \leq \|\mathbf{A}_\Lambda \mathbf{v}_\Lambda\| \leq C_{\mathbf{A}}\|\mathbf{v}_\Lambda\|, \qquad \mathbf{v}_\Lambda \in S_\Lambda. \tag{4.6}$$

Together with the definition $\kappa_2(\mathbf{A}_\Lambda) := \|\mathbf{A}_\Lambda\|\,\|\mathbf{A}_\Lambda^{-1}\|$ we obtain the claimed estimate. □

In other words, representations of A with respect to properly scaled wavelet bases for $\mathcal{H}$ entail well-conditioned system matrices $\mathbf{A}_\Lambda$ independent of Λ. This in turn means that the convergence speed of an iterative solver applied to the corresponding finite system

$$\mathbf{A}_\Lambda \mathbf{y}_\Lambda = \mathbf{f}_\Lambda \tag{4.7}$$

does not deteriorate as $\Lambda \to \infty$.
In summary, ellipticity implies stability of the Galerkin discretizations for any set $\Lambda \subset I\!I$. This is not the case for finite versions of the saddle point problems discussed in Section 4.2.

Fast Wavelet Transform
Let us briefly summarize how in the situation of uniform refinements, i.e., when $S(\Phi_J) = S(\Psi^J)$, the Fast Wavelet Transformation (FWT) $\mathbf{T}_J$ can be used for preconditioning linear elliptic operators, together with a diagonal scaling induced by the norm equivalence (3.86) [DK1]. Here we recall the notation from Section 3.4 where the wavelet basis is in fact the (unscaled) anchor basis from Section 3.1. Thus, the norm equivalence (3.3) using the scaled wavelet basis Ψ_H is the same as (3.86) in the anchor basis. Recall that the norm equivalence (3.86) implies that every $v \in H^s$ can be expanded uniquely in terms of the Ψ and its expansion coefficients $\mathbf{v}$ satisfy

$$\|v\|_{H^s} \ \sim \ \|\mathbf{D}^s \mathbf{v}\|_{\ell_2}$$

where $\mathbf{D}^s$ is a diagonal matrix with entries $\mathbf{D}^s_{(j,k),(j',k')} = 2^{sj}\delta_{j,j'}\delta_{k,k'}$. For $\mathcal{H} \subset H^1(\Omega)$, the case $s = 1$ is relevant.
In a stable Galerkin scheme for (2.10) with respect to $S(\Psi^J) = S(\Psi_\Lambda)$, we have therefore already identified the diagonal (scaling) matrix $\mathbf{D}_J$ consisting

of the finite portion of the matrix $\mathbf{D} = \mathbf{D}^1$ for which $j_0 - 1 \le j \le J - 1$. The representation of A with respect to the (unscaled) wavelet basis Ψ^J can be expressed in terms of the Fast Wavelet Transform $\mathbf{T}_J$, that is,

$$\langle \Psi^J, A\Psi^J \rangle \;=\; \mathbf{T}_J^T \, \langle \Phi_J, A\Phi_J \rangle \, \mathbf{T}_J, \tag{4.8}$$

where Φ_J is the single–scale basis for $S(\Psi^J)$. Thus, we first set up the operator equation as in Finite Element settings in terms of the single–scale basis Φ_J. Applying the Fast Wavelet Transform $\mathbf{T}_J$ together with $\mathbf{D}_J$ yields that the operator

$$\mathbf{A}_J \;:=\; \mathbf{D}_J^{-1} \, \mathbf{T}_J^T \, \langle \Phi_J, A\Phi_J \rangle \, \mathbf{T}_J \, \mathbf{D}_J^{-1} \tag{4.9}$$

has uniformly bounded condition numbers independent of J. This can be seen by combining the properties of A according to (2.13) with the norm equivalences (3.3) and (3.8).

It is known that the boundary adaptations of the generators and wavelets aggravate the absolute values of the condition numbers. Nevertheless, these constants can be greatly reduced by sophisticated biorthogonalizations of the boundary adapted functions [Bu]. Numerical tests confirm that the absolute constants can be further improved by taking the inverse of the diagonal of $\langle \Psi^J, A\Psi^J \rangle$ instead of $\mathbf{D}_J^{-1}$ for the scaling in (4.9) [Bu, CM, P]. Table 4.2 displays the condition numbers for discretizations of an operator in two spatial dimensions for boundary adapted biorthogonal spline-wavelets in the case $d = 2, \tilde{d} = 4$ with such a scaling.

4.2 Saddle Point Problems Involving Boundary Conditions

As in the previous situation, we first derive an infinite wavelet representation of the saddle point problem introduced in Section 2.3.

For $\mathcal{H} = Y \times Q$ with $Y = H^1(\Omega)$, $Q = (H^{1/2}(\Gamma))'$ let two collections of wavelet bases Ψ_Y, Ψ_Q be available, each satisfying (R), (L) and (CP), with respective duals $\tilde{\Psi}_Y$, $\tilde{\Psi}_Q$. Similar to the previous case, we expand $y = \mathbf{y}^T \Psi_Y$ and $p = \mathbf{p}^T \Psi_Q$ and test with the elements from Ψ_Y, Ψ_Q. Then (2.21) attains the form

$$\mathbf{L} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} := \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \tag{4.10}$$

where

$$\begin{aligned} \mathbf{A} &:= \langle \Psi_Y, A\Psi_Y \rangle \qquad & \mathbf{f} &:= \langle \Psi_Y, f \rangle, \\ \mathbf{B} &:= \langle \Psi_Q, B\Psi_Y \rangle, & \mathbf{g} &:= \langle \Psi_Q, g \rangle. \end{aligned} \tag{4.11}$$

In view of the above assertions, the operator $\mathbf{L}$ is an ℓ_2–automorphism, i.e., for every $(\mathbf{v}, \mathbf{q}) \in \ell_2(I\!I) = \ell_2(I\!I_Y \times I\!I_Q)$ we have

$$c_{\mathbf{L}} \left\| \begin{pmatrix} \mathbf{v} \\ \mathbf{q} \end{pmatrix} \right\| \;\le\; \left\| \mathbf{L} \begin{pmatrix} \mathbf{v} \\ \mathbf{q} \end{pmatrix} \right\| \;\le\; C_{\mathbf{L}} \left\| \begin{pmatrix} \mathbf{v} \\ \mathbf{q} \end{pmatrix} \right\| \tag{4.12}$$

with constants $c_{\mathbf{L}}, C_{\mathbf{L}}$ only depending on $c_{\mathcal{L}}, C_{\mathcal{L}}$ from (2.26) and the constants in the norm equivalences (3.3) and (3.8).
For saddle point problems with an operator $\mathbf{L}$ satisfying (4.12), finite sections are in general not uniformly stable in the sense of (4.6). In fact, for discretizations on uniform grids, the validity of the corresponding mapping property relies on a suitable stability condition, see e.g. [BF, GR]. The relevant facts derived in [DK2] are as follows.
The bilinear form $a(\cdot,\cdot)$ defined in (2.7) is for $c > 0$ elliptic on all of $Y = H^1(\Omega)$ and, hence, also on any finite–dimensional subspace of Y. Let there be two multiresolution analyses $\mathcal{Y}$ of $H^1(\Omega)$ and $\mathcal{Q}$ of Q where the discrete spaces are $Y_j \subset H^1(\Omega)$ and $Q_\Lambda =: Q_\ell \subset (H^{1/2}(\Gamma))'$. With the notation from Section 3.4 and in addition superscripts referring to the domain on which the functions live, these spaces are represented by

$$\begin{aligned} Y_j &= S(\Phi_j^\Omega) = S(\Psi^{j,\Omega}), \qquad & \tilde{Y}_j &= S(\tilde{\Phi}_j^\Omega) = S(\tilde{\Psi}^{j,\Omega}), \\ Q_\ell &= S(\Phi_\ell^\Gamma) = S(\Psi^{\ell,\Gamma}), \qquad & \tilde{Q}_\ell &= S(\tilde{\Phi}_\ell^\Gamma) = S(\tilde{\Psi}^{\ell,\Gamma}). \end{aligned} \tag{4.13}$$

Here the indices j and ℓ refer to mesh sizes on the domain and the boundary,

$$h_\Omega \sim 2^{-j} \qquad \text{and} \qquad h_\Gamma \sim 2^{-\ell}.$$

The discrete inf–sup condition, the *LBB condition*, for the pair Y_j, Q_ℓ requires that there exists a constant $\beta_1 > 0$ *independent* of j and ℓ such that

$$\inf_{q\in Q_\ell} \sup_{v\in Y_j} \frac{b(v,q)}{\|v\|_{H^1(\Omega)}\,\|q\|_{(H^{1/2}(\Gamma))'}} \;\geq\; \beta_1 > 0 \tag{4.14}$$

holds. We have investigated in [DK2] the general case in arbitrary spatial dimensions where the Q_ℓ are *not* trace spaces of Y_j. Employing the reverse Cauchy-Schwarz inequalities from Section 3.4, one can show that (4.14) is satisfied provided that $h_\Gamma(h_\Omega)^{-1} = 2^{j-\ell} \geq c_\Omega > 1$, similar to a condition which was known for bivariate polygons and particular finite elements [Ba1, GG].
It should be mentioned that the obstructions caused by the LBB condition can be avoided by means of stabilization techniques proposed, e.g., in [St] where, however, the location of the boundary of Ω relative to the mesh is somewhat constrained. Another stabilization strategy based on wavelets has been investigated in [Be]. A related approach which systematically avoids restrictions of the LBB type is based on least squares techniques [DKS].
It is particularly interesting that adaptive schemes based on wavelets like the one in Section 5.2 can be designed in such a way that the LBB condition is *automatically* enforced which was first observed in [DDU]. More on this subject can be found in [D4].
In order to get an impression of the value of the constants for the condition numbers for $\mathbf{A}_\Lambda$ in (4.5) and the corresponding ones for the saddle point operator on uniform grids (4.12), we mention an example investigated and

implemented in [P]. In this example, $\Omega = (0,1)^2$ and Γ is one face of its boundary. In Table 4.2 from [P], the spectral condition numbers of $\mathbf{A}$ and $\mathbf{L}$ with respect to two different constructions of wavelets for the case $d = 2$ and $\tilde{d} = 4$ are displayed. We see next to the first column, in which the refinement level j is listed, the spectral condition numbers of $\mathbf{A}$ with the wavelet construction from [DKU] denoted by $\mathbf{A}_{\mathrm{DKU}}$ and with the modification introduced in [Bu] and a further transformation [P] denoted by $\mathbf{A}_{\mathrm{B}}$. The last columns contain the respective numbers for the saddle point matrix $\mathbf{L}$ where $\kappa_2(\mathbf{L}) := \sqrt{\kappa(\mathbf{L}^T\mathbf{L})}$.

j	$\kappa_2(\mathbf{A}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{A}_{\mathrm{B}})$	$\kappa_2(\mathbf{L}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{L}_{\mathrm{B}})$
3	5.195e+02	1.898e+01	1.581e+02	4.147e+01
4	6.271e+02	1.066e+02	1.903e+02	1.050e+02
5	6.522e+02	1.423e+02	1.997e+02	1.399e+02
6	6.830e+02	1.820e+02	2.112e+02	1.806e+02
7	7.037e+02	2.162e+02	2.318e+02	2.145e+02
8	7.205e+02	2.457e+02	2.530e+02	2.431e+02
9	7.336e+02	2.679e+02	2.706e+02	2.652e+02

Table 4.2. Spectral condition numbers of the operators $\mathbf{A}$ and $\mathbf{L}$ for different constructions of biorthogonal wavelets on the interval [P].

4.3 Control Problems: Distributed Control

We now discuss appropriate wavelet formulations for PDE-constrained control problems with distributed control as introduced in Section 2.4. For $\mathcal{V} \in \{H, \mathcal{Z}, \mathcal{U}\}$ let $\Psi_{\mathcal{V}}$ denote a wavelet basis with the properties (R), (L), (CP) for $\mathcal{V}$ with dual basis $\tilde{\Psi}_{\mathcal{V}}$.
Let $\mathcal{Z}$,$\mathcal{U}$ satisfy the embedding (2.29). In terms of wavelet bases and in view of (3.10), the corresponding canonical injections correspond to a multiplication by a diagonal matrix. That is, let $\mathbf{D}_{\mathcal{Z}}, \mathbf{D}_H$ be such that

$$\Psi_{\mathcal{Z}} = \mathbf{D}_{\mathcal{Z}}\Psi_H, \qquad \tilde{\Psi}_H = \mathbf{D}_H\Psi_{\mathcal{U}}. \tag{4.15}$$

Since $\mathcal{Z}$ possibly induces a weaker and $\mathcal{U}$ a stronger topology, the diagonal matrices $\mathbf{D}_{\mathcal{Z}}, \mathbf{D}_H$ are such that their entries are nondecreasing in scale, and there is a finite constant C such that

$$\|\mathbf{D}_{\mathcal{Z}}^{-1}\|, \|\mathbf{D}_H^{-1}\| \le C. \tag{4.16}$$

For instance, for $H = H^\alpha$, $\mathcal{Z} = H^\beta$, or for $H' = H^{-\alpha}$, $\mathcal{U} = H^{-\beta}$, $0 \le \beta \le \alpha$, $\mathbf{D}_{\mathcal{Z}}, \mathbf{D}_H$ have entries $(\mathbf{D}_{\mathcal{Z}})_{\lambda,\lambda} = (\mathbf{D}_H)_{\lambda,\lambda} = (\mathbf{D}^{\alpha-\beta})_{\lambda,\lambda} = 2^{(\alpha-\beta)|\lambda|}$.
We expand y in Ψ_H and u in a wavelet basis $\Psi_{\mathcal{U}}$ for $\mathcal{U} \subset H'$,

$$u = \mathbf{u}^T\Psi_U = (\mathbf{D}_H^{-1}\mathbf{u})^T\Psi_{H'}. \tag{4.17}$$

Following the derivation in Section 4.1, the linear constraints (2.28) attain the form

$$\mathbf{A}\mathbf{y} = \mathbf{f} + \mathbf{D}_H^{-1}\mathbf{u} \tag{4.18}$$

where

$$\mathbf{A} := a(\Psi_H, \Psi_H), \qquad \mathbf{f} := \langle \Psi_H, f\rangle. \tag{4.19}$$

Recall that $\mathbf{A}$ has been assumed to be symmetric. The objective functional (2.33) is stated in terms of the norms $\|\cdot\|_{\mathcal{Z}}$ and $\|\cdot\|_{\mathcal{U}}$. For an exact representation of these norms, corresponding Riesz operators $\mathbf{R}_{\mathcal{Z}}$ and $\mathbf{R}_{\mathcal{U}}$ defined analogously to (3.20) would come into play which may not be explicitly computable if $\mathcal{Z}$,$\mathcal{U}$ are fractional Sobolev spaces. On the other hand, as mentioned before, in many cases such a cost functional serves the purpose of yielding unique solutions while there is some ambiguity in its exact formulation. Hence, in search for a formulation which best supports numerical realizations, it is often sufficient to employ norms which are *equivalent* to $\|\cdot\|_{\mathcal{Z}}$ and $\|\cdot\|_{\mathcal{U}}$. Therefore, in view of the discussion in Section 3.2 we can work with equivalent norms for $\|\cdot\|_{\mathcal{Z}}$, $\|\cdot\|_{\mathcal{U}}$ in terms of the diagonal scaling matrices $\mathbf{D}^s$ induced by the regularity of $\mathcal{Z}$,$\mathcal{U}$, or we can in addition include the Riesz map $\mathbf{R}$ defined in (3.15) to represent $\|\cdot\|_{\mathcal{Z}}$, $\|\cdot\|_{\mathcal{U}}$ by equivalent norms.

In the numerical studies in [Bu], a somewhat better quality of the solution is observed when $\mathbf{R}$ is included. In order to keep track of the appearance of the Riesz maps in the linear systems derived below, here we choose the latter variant.

Moreover, we expand the given observation function $y_* \in \mathcal{Z}$ as

$$y_* = \langle y_*, \tilde{\Psi}_{\mathcal{Z}}\rangle \Psi_{\mathcal{Z}} =: (\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{y}_*)^T \Psi_{\mathcal{Z}} = \mathbf{y}_*^T \Psi_H. \tag{4.20}$$

The way the vector $\mathbf{y}_*$ is defined here, for notational convenience, may by itself actually have infinite norm in ℓ_2. However, its occurrence will always include premultiplication by $\mathbf{D}_{\mathcal{Z}}^{-1}$ which is therefore always well–defined. In view of (3.24), we obtain the relations

$$\|y - y_*\|_{\mathcal{Z}} \sim \|\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y} - \mathbf{y}_*)\| \sim \|\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y} - \mathbf{y}_*\|. \tag{4.21}$$

Note that here $\mathbf{R} = \langle \Psi, \Psi\rangle$ (and not $\mathbf{R}^{-1}$) comes into play since y, y_* have been expanded in a scaled version of the primal wavelet basis Ψ. Hence, equivalent norms for $\|\cdot\|_{\mathcal{Z}}$ may involve $\mathbf{R}$. As for describing equivalent norms for $\|\cdot\|_{\mathcal{U}}$, recall that u is expanded in the basis Ψ_U for $U \subset H'$. Consequently, $\mathbf{R}^{-1}$ is the natural matrix to take into account when considering equivalent norms, i.e., we choose here

$$\|u\|_{\mathcal{U}} \sim \|\mathbf{R}^{-1/2}\mathbf{u}\|. \tag{4.22}$$

Finally, we formulate the following control problem in (infinite) wavelet coordinates.

(DCP) *For given data* $\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{y}_* \in \ell_2(I_{\mathcal{Z}})$, $\mathbf{f} \in \ell_2(I_H)$, *and weight parameter* $\omega > 0$, *minimize the quadratic functional*

$$\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) := \tfrac{1}{2}\,\|\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y}-\mathbf{y}_*)\|^2 + \tfrac{\omega}{2}\,\|\mathbf{R}^{-1/2}\mathbf{u}\|^2 \tag{4.23}$$

over $(\mathbf{y},\mathbf{u}) \in \ell_2(I_H) \times \ell_2(I_H)$ *subject to the linear constraints*

$$\mathbf{A}\mathbf{y} = \mathbf{f} + \mathbf{D}_H^{-1}\mathbf{u}. \tag{4.24}$$

Remark 4.1.
Problem (DCP) can be viewed as (discretized yet still infinite–dimensional) *representation* of the linear–quadratic control problem (2.27) together with (2.28) in wavelet coordinates in the following sense. The functional $\check{\mathbf{J}}(\mathbf{y},\mathbf{u})$ defined in (4.23) is equivalent to the functional $J(y,u)$ from (2.27) in the sense that there exist constants $0 < c_J \le C_J < \infty$ such that

$$c_J\,\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) \;\le\; J(y,u) \;\le\; C_J\,\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) \tag{4.25}$$

holds for any $y = \mathbf{y}^T\Psi_H \in H$, given $y_* = (\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{y}_*)^T\Psi_{\mathcal{Z}} \in \mathcal{Z}$ and any $u = \mathbf{u}^T\Psi_{\mathcal{U}} \in \mathcal{U}$. Moreover, in the case of compatible data $y_* = A^{-1}f$ yielding $J(y,u) \equiv 0$, the respective minimizers coincide, and $\mathbf{y}_* = \mathbf{A}^{-1}\mathbf{f}$ yields $\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) \equiv \mathbf{0}$. In this sense the new functional (4.23) captures the essential features of the model minimization functional. ∎

Once problem (DCP) is posed, we can apply variational principles to derive necessary and sufficient conditions for a unique solution. All control problems considered here are in fact simple in this regard, as we have to minimize a quadratic functional subject to linear constraints, for which the necessary conditions are also sufficient. In principle, there are two ways to derive the optimality conditions for (DCP). In Section 2.4 we have already encountered the technique via the Lagrangian.
We define for (DCP) the *Lagrangian* introducing the *Lagrange multiplier*, *adjoint variable* or *adjoint state* $\mathbf{p}$ as

$$\mathbf{Lagr}(\mathbf{y},\mathbf{p},\mathbf{u}) := \check{\mathbf{J}}(\mathbf{y},\mathbf{u}) + \langle \mathbf{p}, \mathbf{A}\mathbf{y} - \mathbf{f} - \mathbf{D}_H^{-1}\mathbf{u}\rangle. \tag{4.26}$$

Then the KKT conditions $\delta\,\mathbf{Lagr}(\mathbf{w}) = \mathbf{0}$ for $\mathbf{w} = \mathbf{p},\mathbf{y},\mathbf{u}$ are, respectively,

$$\mathbf{A}\mathbf{y} = \quad \mathbf{f} + \mathbf{D}_H^{-1}\mathbf{u}, \tag{4.27a}$$

$$\mathbf{A}^T\mathbf{p} = -\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y}-\mathbf{y}_*), \tag{4.27b}$$

$$\omega\mathbf{R}^{-1}\mathbf{u} = \quad \mathbf{D}_H^{-1}\mathbf{p}. \tag{4.27c}$$

The first system resulting from the variation with respect to the Lagrange multiplier always recovers the original constraints (4.24) and will be referred to as the *primal system* or the *state equation*. Accordingly, we call (4.27b) the *adjoint* or *dual system*, or the *costate equation*. The third equation (4.27c) is sometimes denoted as the *design equation*. Although $\mathbf{A}$ is symmetric, we

continue to write $\mathbf{A}^T$ for the operator of the adjoint system to distinguish it from the primal system.
The coupled system (4.27) is to be solved later. However, in order to derive convergent iterations and deduce complexity estimates, a different formulation will be advantageous. It is based on the fact that $\mathbf{A}$ is, according to Proposition 4.1, a boundedly invertible mapping on ℓ_2. Thus, we can formally invert (4.18) to obtain $\mathbf{y} = \mathbf{A}^{-1}\mathbf{f} + \mathbf{A}^{-1}\mathbf{D}_H^{-1}\mathbf{u}$. Substitution into (4.23) yields a functional depending only on $\mathbf{u}$,

$$\mathbf{J}(\mathbf{u}) := \tfrac{1}{2}\,\|\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}\left(\mathbf{A}^{-1}\mathbf{D}_H^{-1}\mathbf{u} - (\mathbf{y}_* - \mathbf{A}^{-1}\mathbf{f})\right)\|^2 + \tfrac{\omega}{2}\,\|\mathbf{R}^{-1/2}\mathbf{u}\|^2. \quad (4.28)$$

Employing the abbreviations

$$\mathbf{Z} := \quad \mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{A}^{-1}\mathbf{D}_H^{-1}, \quad (4.29a)$$

$$\mathbf{G} := -\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{A}^{-1}\mathbf{f} - \mathbf{y}_*), \quad (4.29b)$$

the functional simplifies to

$$\mathbf{J}(\mathbf{u}) = \tfrac{1}{2}\|\mathbf{Z}\mathbf{u} - \mathbf{G}\|^2 + \tfrac{\omega}{2}\,\|\mathbf{R}^{-1/2}\mathbf{u}\|^2. \quad (4.30)$$

Proposition 4.3. *[K4] The functional* $\mathbf{J}$ *is twice differentiable with first and second variation*

$$\delta\mathbf{J}(\mathbf{u}) = (\mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}^{-1})\mathbf{u} - \mathbf{Z}^T\mathbf{G}, \qquad \delta^2\mathbf{J}(\mathbf{u}) = \mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}^{-1}. \quad (4.31)$$

In particular, $\mathbf{J}$ *is convex so that a unique minimizer exists.*

Setting

$$\mathbf{Q} := \mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}^{-1}, \qquad \mathbf{g} := \mathbf{Z}^T\mathbf{G}, \quad (4.32)$$

the unique minimizer $\mathbf{u}$ of (4.30) is given by solving

$$\delta\mathbf{J}(\mathbf{u}) = \mathbf{0} \quad (4.33)$$

or, equivalently, the system

$$\mathbf{Q}\mathbf{u} = \mathbf{g}. \quad (4.34)$$

By definition (4.32), $\mathbf{Q}$ is a symmetric positive definite (infinite) matrix. Hence, finite versions of (4.34) could be solved by gradient or conjugate gradient iterative schemes. As the convergence speed of any such iteration depends on the spectral condition number of $\mathbf{Q}$, it is important to note the following result.

Proposition 4.4. *The (infinite) matrix* $\mathbf{Q}$ *is uniformly bounded on* ℓ_2*, i.e., there exist constants* $0 < c_{\mathbf{Q}} \le C_{\mathbf{Q}} < \infty$ *such that*

$$c_{\mathbf{Q}}\,\|\mathbf{v}\| \le \|\mathbf{Q}\mathbf{v}\| \le C_{\mathbf{Q}}\,\|\mathbf{v}\|, \qquad \mathbf{v} \in \ell_2. \quad (4.35)$$

The proof follows from (2.13) and (4.16), see [DK3]. Of course, in order to make such iterative schemes for (4.34) practically feasible, the explicit inversion of $\mathbf{A}$ in the definition of $\mathbf{Q}$ has to be avoided and replaced by an iterative solver in turn. This is where the system (4.27) will come into play. In particular, the third equation (4.27c) has the following interpretation which will turn out to be very useful later.

Proposition 4.5. *For a given control vector* $\mathbf{u}$ *if we solve (4.24) for* $\mathbf{y}$ *and (4.27b) for* $\mathbf{p}$ *successively, then the* residual *for (4.34) attains the form*

$$\mathbf{Qu} - \mathbf{g} = \omega \mathbf{R}^{-1}\mathbf{u} - \mathbf{D}_U^{-1}\mathbf{p}. \tag{4.36}$$

Proof. Solving consecutively (4.24) and (4.27b) and recalling the definitions of $\mathbf{Z}$, $\mathbf{g}$ (4.29a), (4.32) we obtain

$$\begin{aligned}
\mathbf{D}_H^{-1}\mathbf{p} &= -\mathbf{D}_H^{-1}(\mathbf{A}^{-T}\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y} - \mathbf{y}_*)) \\
&= -\mathbf{Z}^T\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{A}^{-1}\mathbf{f} + \mathbf{A}^{-1}\mathbf{D}_H^{-1}\mathbf{u} - \mathbf{y}_*) \\
&= \mathbf{Z}^T\mathbf{G} - \mathbf{Z}^T\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{A}^{-1}\mathbf{D}_H^{-1}\mathbf{u} \\
&= \mathbf{g} - \mathbf{Z}^T\mathbf{Z}\mathbf{u}.
\end{aligned}$$

Hence, the residual $\mathbf{Qu} - \mathbf{g}$ attains the form

$$\mathbf{Qu} - \mathbf{g} = (\mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}^{-1})\mathbf{u} - \mathbf{g} = \omega\mathbf{R}^{-1}\mathbf{u} - \mathbf{D}_H^{-1}\mathbf{p},$$

where we have used the definition of $\mathbf{Q}$ from (4.32). □

Having derived the optimality conditions (4.27), the next issue is their efficient numerical solution. In view of the fact that the system (4.27) still involves infinite matrices and vectors, this also raises the question of how to derive computable finite versions. By now we have investigated two scenarios. The first version with respect to *uniform discretizations* is based on choosing finite–dimensional subspaces of the function spaces under consideration. The second version which deals with *adaptive discretizations* is actually based on the infinite system (4.27). In both scenarios, a fully iterative numerical scheme for the solution of (4.27) is designed along the following lines. The basic iteration scheme is a *gradient* or *conjugate gradient iteration* for (4.34) as an *outer iteration* where each application of $\mathbf{Q}$ is in turn realized by solving the primal and the dual system (4.24) and (4.27b) also by a gradient or conjugate gradient method as *inner iterations*.

For *uniform* discretizations for which we wanted to numerically test the role of equivalent norms and the influence of Riesz maps in the cost functional (4.23), we have used in [BK] as central iterative scheme the conjugate gradient (CG) method. Since the interior systems are only solved up to discretization error accuracy, the whole procedure may therefore be viewed as an *inexact conjugate gradient (CG) method*. We stress already at this point that the iteration numbers of such a method do *not* depend on the discretization level

as finite versions of all involved operators are also uniformly well–conditioned in the sense of (4.35). In each step of the outer iteration, the error will be reduced by a fixed factor ρ. Combined with a *nested iteration strategy*, it will be shown that this yields an asymptotically optimal method in the number of arithmetic operations.
Starting from the infinite coupled system (4.27), we have investigated in [DK3] *adaptive schemes* which, given any prescribed accuracy $\varepsilon > 0$, solve (4.27) such that the error for $\mathbf{y}, \mathbf{u}, \mathbf{p}$ is controlled by ε. Here we have used a *gradient scheme* as basic iterative scheme since it somehow simplifies the analysis, see Section 5.2.

4.4 Control Problems: Dirichlet Boundary Control

Having derived a representation in wavelet coordinates for both the saddle point problem from Section 2.3 and the PDE-constrained control problem in the previous section, it is straightforward to also find an appropriate representation of the control problem with Dirichlet boundary control introduced in Section 2.5. In order not to be overburdened with notation, we specifically choose the control space on the boundary as $\mathcal{U} := Q(= (H^{1/2}(\Gamma))')$. For the more general situation covered by (2.37), a diagonal matrix with nondecreasing entries similar to (4.15) would come into play to switch between $\mathcal{U}$ and Q. Thus, the exact wavelet representation of the constraints (2.36) is given by the system (4.10), where we exchange the given Dirichlet boundary term $\mathbf{g}$ by $\mathbf{u}$ in the present situation to express the dependence on the control in the right hand side, i.e.,

$$\mathbf{L}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} := \begin{pmatrix}\mathbf{A} & \mathbf{B}^T\\ \mathbf{B} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ \mathbf{u}\end{pmatrix}. \tag{4.37}$$

The derivation of a representer of the initial objective functional (2.35) is under the embedding condition (2.37) $\|v\|_{\mathcal{Z}} \lesssim \|v\|_Y$ for $v \in Y$ now the same as in the previous section, where all reference to the space H is to be exchanged by reference to Y. We end up with the following minimization problem in wavelet coordinates for the case of Dirichlet boundary control.
(DCP) *For given data* $\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{y}_* \in \ell_2(\mathcal{I}_{\mathcal{Z}})$, $\mathbf{f} \in \ell_2(\mathcal{I}_Y)$, *and weight parameter* $\omega > 0$, *minimize the quadratic functional*

$$\check{\mathbf{J}}(\mathbf{y}, \mathbf{u}) := \tfrac{1}{2}\,\|\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y} - \mathbf{y}_*)\|^2 + \tfrac{\omega}{2}\,\|\mathbf{R}^{-1/2}\mathbf{u}\|^2 \tag{4.38}$$

over $(\mathbf{y}, \mathbf{u}) \in \ell_2(\mathcal{I}_Y) \times \ell_2(\mathcal{I}_Y)$ *subject to the linear constraints (4.37),*

$$\mathbf{L}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ \mathbf{u}\end{pmatrix}.$$

The corresponding Karush-Kuhn-Tucker conditions can be derived by the same variational principles as in the previous section by defining a Lagrangian

in terms of the functional $\check{\mathbf{J}}(\mathbf{y},\mathbf{u})$ and appending the constraints (4.18) with the help of additional Lagrange multipliers $(\mathbf{z},\boldsymbol{\mu})^T$, see [K4]. We obtain in this case a system of coupled saddle point problems

$$\mathbf{L}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ \mathbf{u}\end{pmatrix} \tag{4.39a}$$

$$\mathbf{L}^T\begin{pmatrix}\mathbf{z}\\ \boldsymbol{\mu}\end{pmatrix} = \begin{pmatrix}-\omega\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y}-\mathbf{y}_*)\\ \mathbf{0}\end{pmatrix} \tag{4.39b}$$

$$\mathbf{u} = \boldsymbol{\mu}. \tag{4.39c}$$

Again, the first system appearing here, the *primal system*, is just the constraints (4.18) while (3.9) will be referred to as the *dual* or *adjoint system*. The specific form of the right hand side of the dual system emerges from the particular formulation of the minimization functional (4.38). The (here trivial) equation (4.39c) stems from measuring $\mathbf{u}$ just in ℓ_2, representing measuring the control in its natural trace norm. Instead of replacing $\boldsymbol{\mu}$ by $\mathbf{u}$ in (3.9) and trying to solve the resulting equations, (4.39c) will be essential to devise an inexact gradient scheme. In fact, since $\mathbf{L}$ in (4.18) is an invertible operator, we can rewrite $\check{\mathbf{J}}(\mathbf{y},\mathbf{u})$ by formally inverting (4.18) as a functional of $\mathbf{u}$, that is, $\mathbf{J}(\mathbf{u}) := \check{\mathbf{J}}(\mathbf{y}(\mathbf{u}),\mathbf{u})$ as above. The following result will be very useful for the design of the outer–inner iterative solvers

Proposition 4.6. *The first variation of* $\mathbf{J}$ *satisfies*

$$\delta\mathbf{J}(\mathbf{u}) \;=\; \mathbf{u}-\boldsymbol{\mu}, \tag{4.40}$$

where $(\mathbf{u},\boldsymbol{\mu})$ *are part of the solution of (4.39). Moreover,* $\mathbf{J}$ *is convex so that a unique minimizer exists.*

Hence, equation (4.39c) is just $\delta\mathbf{J}(\mathbf{u}) = \mathbf{0}$. For a unified treatment below of both control problems considered in these notes, it will be useful to rewrite (4.39c) as a condensed equation for the control $\mathbf{u}$ similar to (4.34). We formally invert (4.37) and (4.39b) to obtain

$$\mathbf{Q}\mathbf{u} = \mathbf{g} \tag{4.41}$$

with the abbreviations

$$\mathbf{Q} := \mathbf{Z}^T\mathbf{Z} + \omega\mathbf{I}, \quad \mathbf{g} := \mathbf{Z}^T(\mathbf{y}_* - \mathbf{T}_\square\mathbf{L}^{-1}\mathbf{I}_\square\mathbf{f}) \tag{4.42}$$

and

$$\mathbf{Z} := \mathbf{T}_\square\mathbf{L}^{-1}\mathbf{I}_\square, \qquad \mathbf{I}_\square := \begin{pmatrix}\mathbf{0}\\ \mathbf{I}\end{pmatrix}, \qquad \mathbf{T}_\square := (\mathbf{T}\;\;\mathbf{0}). \tag{4.43}$$

Proposition 4.7. *The vector* $\mathbf{u}$ *as part of the solution vector* $(\mathbf{y},\mathbf{p},\mathbf{z},\boldsymbol{\mu},\mathbf{u})$ *of (4.39) coincides with the unique solution* $\mathbf{u}$ *of the condensed equations (4.41).*

5 Iterative Solution

Each of the four problem classes discussed above finally leads to the problem of solving a system

$$\delta \mathbf{J}(\mathbf{q}) \;=\; \mathbf{0} \tag{5.1}$$

or, equivalently, a linear system

$$\mathbf{M}\mathbf{q} = \mathbf{b}, \tag{5.2}$$

where $\mathbf{M} : \ell_2 \to \ell_2$ is a (possibly infinite) symmetric positive definite matrix satisfying

$$c_{\mathbf{M}}\|\mathbf{v}\| \le \|\mathbf{M}\mathbf{v}\| \le C_{\mathbf{M}}\|\mathbf{v}\|, \quad \mathbf{v} \in \ell_2, \tag{5.3}$$

for some constants $0 < c_{\mathbf{M}} \le C_{\mathbf{M}} < \infty$ and where $\mathbf{b} \in \ell_2$ is some given right hand side.
A simple *gradient method* for solving (5.1) is

$$\mathbf{q}_{k+1} := \mathbf{q}_k - \alpha\, \delta \mathbf{J}(\mathbf{q}_k), \qquad k = 0, 1, 2, \ldots \tag{5.4}$$

with some initial guess $\mathbf{q}_0$. In all of the previously considered situations, it has been asserted that there exists a fixed parameter α, depending on bounds for the second variation of $\mathbf{J}$, such that (5.4) converges and reduces the error in each step by at least a fixed factor $\rho < 1$, i.e.,

$$\|\mathbf{q} - \mathbf{q}_{k+1}\| \le \rho \|\mathbf{q} - \mathbf{q}_k\|, \quad k = 0, 1, 2, \ldots, \tag{5.5}$$

where ρ is determined by $\rho := \|\mathbf{I} - \alpha\mathbf{M}\| < 1$. Hence, the scheme (5.4) is a convergent iteration for the possibly infinite system (5.2). Next we will need to discuss how to reduce the infinite systems to computable finite versions.

5.1 Finite Systems on Uniform Grids

Let us first consider finite-dimensional trial spaces with respect to uniform discretizations. For each of the Hilbert spaces H, in the wavelet setting this means picking the index set of all indices up to some *highest refinement level* J, i.e.,

$$\mathcal{I}_{J,H} := \{\lambda \in \mathcal{I}_H : |\lambda| \le J\} \subset \mathcal{I}_H$$

satisfying $N_{J,H} := \#\mathcal{I}_{J,H} < \infty$. The representation of operators is then built as in Section 3.3 with respect to this truncated index set which corresponds to deleting all rows and columns that refer to indices λ such that $|\lambda| > J$, and correspondingly for functions. There is by construction also a *coarsest level* of resolution denoted by j_0.
Computationally the representation of operators according to (3.25) is in the case of uniform grids always realized as follows. First, the operator is set up in terms of the *generator basis* on the finest level J. This generator basis simply consists of tensor products of B-Splines, or linear combinations of these near

the boundaries. The representation of an operator in the *wavelet basis* is then achieved by applying the Fast Wavelet Transform (FWT) which needs $\mathcal{O}(N_{J,H})$ arithmetic operations and is therefore asymptotically optimal, for example see [D2, DKU, K2] and Section 3.4.
In order not to overburden the notation, in this subsection let the resulting system for $N = N_{J,H}$ unknowns again be denoted by

$$\mathbf{M}\mathbf{q} = \mathbf{b}, \tag{5.6}$$

where now $\mathbf{M} : \mathbb{R}^N \to \mathbb{R}^N$ is a symmetric positive definite matrix satisfying (5.3) on $\mathbb{R}^N$. It will be convenient to abbreviate the residual using an approximation $\tilde{\mathbf{q}}$ to $\mathbf{q}$ for (5.6) as

$$\textsc{Resd}(\tilde{\mathbf{q}}) := \mathbf{M}\tilde{\mathbf{q}} - \mathbf{b}. \tag{5.7}$$

We will employ a basic conjugate gradient method that iteratively computes an approximate solution $\mathbf{q}_K$ to (5.6) with given initial vector $\mathbf{q}_0$ and given tolerance $\varepsilon > 0$ such that

$$\|\mathbf{M}\mathbf{q}_K - \mathbf{b}\| = \|\textsc{Resd}(\mathbf{q}_K)\| \le \varepsilon, \tag{5.8}$$

where K denotes the number of iterations used. Later we specify ε depending on the discretization for which (5.6) is set up. The following CG scheme contains a routine $\textsc{App}(\eta_k, \mathbf{M}, \mathbf{d}_k)$ which, in view of the problem classes discussed above, is to have the property that it approximately computes the product $\mathbf{M}\mathbf{d}_k$ up to a tolerance $\eta_k = \eta_k(\varepsilon)$ depending on ε, i.e., the output $\mathbf{m}_k$ of $\textsc{App}(\eta_k, \mathbf{M}, \mathbf{d}_k)$ satisfies

$$\|\mathbf{m}_k - \mathbf{M}\mathbf{d}_k\| \le \eta_k. \tag{5.9}$$

For the cases where $\mathbf{M} = \mathbf{A}$, this is simply the matrix-vector multiplication $\mathbf{M}\mathbf{d}_k$. For the situations where $\mathbf{M}$ may involve the solution of an additional system, this multiplication will be only approximative.

CG $[\varepsilon, \mathbf{q}_0, \mathbf{M}, \mathbf{b}] \to \mathbf{q}_K$

(I) Set $\mathbf{d}_0 := \mathbf{b} - \mathbf{M}\mathbf{q}_0$ and $\mathbf{r}_0 := -\mathbf{d}_0$. Let $k = 0$.
(II) While $\|\mathbf{r}_k\| > \varepsilon$

$$\begin{aligned}
\mathbf{m}_k &:= \textsc{App}(\eta_k(\varepsilon), \mathbf{M}, \mathbf{d}_k) \\
\alpha_k &:= \frac{(\mathbf{r}_k)^T \mathbf{r}_k}{(\mathbf{d}_k)^T \mathbf{m}_k} \qquad\qquad \mathbf{q}_{k+1} := \mathbf{q}_k + \alpha_k \mathbf{d}_k \\
\mathbf{r}_{k+1} &:= \mathbf{r}_k + \alpha_k \mathbf{m}_k \qquad\qquad \beta_k := \frac{(\mathbf{r}_{k+1})^T \mathbf{r}_{k+1}}{(\mathbf{r}_k)^T \mathbf{r}_k} \\
\mathbf{d}_{k+1} &:= -\mathbf{r}_{k+1} + \beta_k \mathbf{d}_k \\
k &:= k + 1
\end{aligned} \tag{5.10}$$

(III) Set $K := k - 1$.

Briefly in the case $\mathbf{M} = \mathbf{A}$ the final iterate $\mathbf{q}_K$ indeed satisfies (5.8). From the newly computed iterate $\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha_k \mathbf{d}_k$ it follows by applying $\mathbf{M}$ on both sides that $\mathbf{M}\mathbf{q}_{k+1} - \mathbf{b} = \mathbf{M}\mathbf{q}_k - \mathbf{b} + \alpha_k \mathbf{M}\mathbf{d}_k$ which is the same as $\mathrm{RESD}(\mathbf{q}_{k+1}) = \mathrm{RESD}(\mathbf{q}_k) + \alpha_k \mathbf{M}\mathbf{d}_k$. By the initialization for $\mathbf{r}_k$ used above, this in turn is the updating term for $\mathbf{r}_k$, hence, $\mathbf{r}_k = \mathrm{RESD}(\mathbf{q}_k)$. After the stopping criterion based on $\mathbf{r}_k$ is met, the final iterate $\mathbf{q}_K$ satisfies (5.8).
The routine CG computes the *residual* up to the stopping criterion ε. From the residual and in view of (5.3), we can estimate the *error* in the solution as

$$\|\mathbf{q} - \mathbf{q}_K\| = \|\mathbf{M}^{-1}(\mathbf{b} - \mathbf{M}\mathbf{q}_K)\| \le \|\mathbf{M}^{-1}\| \, \|\mathrm{RESD}(\mathbf{q}_K)\| \le \frac{\varepsilon}{c_{\mathbf{M}}}, \tag{5.11}$$

that is, it may deviate from the norm of the residual by a factor proportional to the smallest eigenvalue of $\mathbf{M}$.

Distributed Control
Let us now apply the solution scheme to the situation from Section 4.3 where $\mathbf{Q}$ now involves the inversion of finite-dimensional systems (4.27a) and (4.27b). The material in the remainder of this subsection is essentially contained in [BK].
We begin with a specification of the approximate computation of the right hand side $\mathbf{b}$ which also contains applications of $\mathbf{A}^{-1}$.

$\mathrm{RHS}[\zeta, \mathbf{A}, \mathbf{f}, \mathbf{y}_*] \to \mathbf{b}_\zeta$

(I) CG $[\frac{c_{\mathbf{A}}}{2C} \frac{c_{\mathbf{A}}}{C^2 C_0^2} \zeta, \mathbf{0}, \mathbf{A}, \mathbf{f}] \to \mathbf{b}_1$
(II) CG $[\frac{c_{\mathbf{A}}}{2C} \zeta, \mathbf{0}, \mathbf{A}^T, -\mathbf{D}_{\mathcal{Z}}^{-1} \mathbf{R} \mathbf{D}_{\mathcal{Z}}^{-1} (\mathbf{b}_1 - \mathbf{y}_*)] \to \mathbf{b}_2$
(III) $\mathbf{b}_\zeta := \mathbf{D}_H^{-1} \mathbf{b}_2$.

The tolerances used within the two conjugate gradient methods depend on the constants $c_{\mathbf{A}}, C, C_0$ from (2.13), (4.16) and (3.18), respectively. Since the additional factor $c_{\mathbf{A}}(CC_0)^{-2}$ in the stopping criterion in step (I) in comparison to step (II) is in general smaller than one, this means that in step (II) the primal system needs to be solved more accurately than the adjoint system.

Proposition 5.1. *The result* $\mathbf{b}_\zeta$ *of* $\mathrm{RHS}[\zeta, \mathbf{A}, \mathbf{f}, \mathbf{y}_*]$ *satisfies*

$$\|\mathbf{b}_\zeta - \mathbf{b}\| \le \zeta. \tag{5.12}$$

Proof. Recalling the definition (4.32) of $\mathbf{b}$, step (III) and step (II) yield

$$\begin{aligned} \|\mathbf{b}_\zeta - \mathbf{b}\| &\le \|\mathbf{D}_H^{-1}\| \, \|\mathbf{b}_2 - \mathbf{D}_H \mathbf{b}\| \\ &\le C \|\mathbf{A}^{-T}\| \, \|\mathbf{A}^T \mathbf{b}_2 - \mathbf{D}_{\mathcal{Z}}^{-1} \mathbf{R} \mathbf{D}_{\mathcal{Z}}^{-1} (\mathbf{A}^{-1}\mathbf{f} - \mathbf{b}_1 + \mathbf{b}_1 - \mathbf{y}_*)\| \\ &\le \frac{C}{c_{\mathbf{A}}} \left(\frac{c_{\mathbf{A}}}{2C} \zeta + \|\mathbf{D}_{\mathcal{Z}}^{-1} \mathbf{R} \mathbf{D}_{\mathcal{Z}}^{-1} (\mathbf{A}^{-1}\mathbf{f} - \mathbf{b}_1)\| \right). \end{aligned} \tag{5.13}$$

Employing the upper bounds for $\mathbf{D}_{\mathcal{Z}}^{-1}$ and $\mathbf{R}$, we arrive at

$$\begin{aligned}\|\mathbf{b}_\zeta - \mathbf{b}\| &\le \frac{C}{c_{\mathbf{A}}}\left(\frac{c_{\mathbf{A}}}{2C}\zeta + C^2C_0^2\,\|\mathbf{A}^{-1}\|\,\|\mathbf{f}-\mathbf{A}\mathbf{b}_1\|\right)\\ &\le \frac{C}{c_{\mathbf{A}}}\left(\frac{c_{\mathbf{A}}}{2C}\zeta + \frac{C^2C_0^2}{c_{\mathbf{A}}}\frac{c_{\mathbf{A}}}{2C}\frac{c_{\mathbf{A}}}{C^2C_0^2}\zeta\right) \quad = \zeta. \quad \square\end{aligned} \tag{5.14}$$

Accordingly, an approximation $\mathbf{m}_\eta$ to the matrix-vector product $\mathbf{Qd}$ is the output of the following routine App.

App$[\eta, \mathbf{Q}, \mathbf{d}] \to \mathbf{m}_\eta$

(i) CG $[\frac{c_{\mathbf{A}}}{3C}\frac{c_{\mathbf{A}}}{C^2C_0^2}\eta, \mathbf{0}, \mathbf{A}, \mathbf{f} + \mathbf{D}_H^{-1}\mathbf{d}] \to \mathbf{y}_\eta$
(ii) CG $[\frac{c_{\mathbf{A}}}{3C}\eta, \mathbf{0}, \mathbf{A}^T, -\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_Z^{-1}(\mathbf{y}_\eta - \mathbf{y}_*)] \to \mathbf{p}_\eta$
(iii) $\mathbf{m}_\eta := \mathbf{g}_{\eta/3} + \omega\mathbf{R}^{-1}\mathbf{d} - \mathbf{D}_H^{-1}\mathbf{p}_\eta$.

The choice of the tolerances for the interior application of CG in steps (i) and (ii) will become clear from the following result.

Proposition 5.2. *The result* $\mathbf{m}_\eta$ *of* App$[\eta, \mathbf{Q}, \mathbf{d}]$ *satisfies*

$$\|\mathbf{m}_\eta - \mathbf{Qd}\| \le \eta. \tag{5.15}$$

Proof. Denote by $\mathbf{y_d}$ the exact solution of (4.27a) with $\mathbf{d}$ in place of $\mathbf{u}$ on the right hand side, and by $\mathbf{p_d}$ the exact solution of (4.27b) with $\mathbf{y_d}$ on the right hand side. Then we deduce from step (iii) and (4.36) combined with (3.18) and (4.16)

$$\begin{aligned}\|\mathbf{m}_\eta - \mathbf{Qd}\| &= \|\mathbf{g}_{\eta/3} - \mathbf{g} + \omega\mathbf{R}^{-1}\mathbf{d} - \mathbf{D}_U^{-1}\mathbf{p}_\eta \; - (\mathbf{Qd} - \mathbf{g})\|\\ &\le \tfrac{1}{3}\eta + \|\omega\mathbf{R}^{-1}\mathbf{d} - \mathbf{D}_U^{-1}\mathbf{p}_\eta - (\omega\mathbf{R}^{-1}\mathbf{d} - \mathbf{D}_U^{-1}\mathbf{p_d})\|\\ &\le \tfrac{1}{3}\eta + C\|\mathbf{p_d} - \mathbf{p}_\eta\|.\end{aligned} \tag{5.16}$$

Denote by $\hat{\mathbf{p}}$ the exact solution of (4.27b) with $\mathbf{y}_\eta$ on the right hand side. Then we have $\mathbf{p_d} - \hat{\mathbf{p}} = -\mathbf{A}^{-T}\mathbf{D}_Z^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y_d} - \mathbf{y}_\eta)$. It follows by (2.13), (3.18) and (4.16) that

$$\|\mathbf{p_d} - \hat{\mathbf{p}}\| \le \frac{C^2C_0^2}{c_{\mathbf{A}}}\|\mathbf{y_d} - \mathbf{y}_\eta\| \le \frac{1}{3C}\eta, \tag{5.17}$$

where the last estimate follows by the choice of the threshold in step (i). Finally, the combination(5.16) and (5.17) together with (5.12) and the stopping criterion in step (ii) readily confirms that

$$\begin{aligned}\|\mathbf{m}_\eta - \mathbf{Qd}\| &\le \tfrac{1}{3}\eta + C\left(\|\mathbf{p_d} - \hat{\mathbf{p}}\| + \|\hat{\mathbf{p}} - \mathbf{p}_\eta\|\right)\\ &\le \frac{1}{3}\eta + C\left(\frac{1}{3C}\eta + \frac{1}{3C}\eta\right) = \eta. \quad \square\end{aligned}$$

The effect of perturbed applications of $\mathbf{M}$ in CG and more general Krylov subspace schemes with respect to convergence has been investigated in a numerical linear algebra context for a given linear system (5.6) in several papers, for example see [ES]. Here we have chosen the η_i to be proportional to the outer accuracy ε incorporating a safety factor accounting for the values of β_i and $\|\mathbf{r}_i\|$.

Finally, we can formulate a full nested iteration strategy for finite systems (4.27) on uniform grids which employs outer and inner CG routines as follows. The scheme starts at the coarsest level of resolution j_0 with some initial guess $\mathbf{u}_0^{j_0}$ and successively solves (4.34) with respect to each level j until the norm of the current residual is below the discretization error on that level.

In wavelet coordinates, $\|\cdot\|$ corresponds to the energy norm. If we employ as in [BK] on the primal side for approximation linear combinations of B–splines of order d, the discretization error is for smooth solutions expected to be proportional to $2^{-(d-1)j}$. Then the refinement level is successively increased until on the finest level J a prescribed tolerance proportional to the discretization error $2^{-(d-1)J}$ is met. In the following, superscripts on vectors denote the refinement level on which this term is computed. The given data $\mathbf{y}_*^j$, $\mathbf{f}^j$ are supposed to be accessible on all levels. On the coarsest level, the solution of (4.34) is computed exactly up to double precision by QR decomposition. Subsequently, the results from level j are prolongated onto the next higher level $j+1$. Using wavelets, this is accomplished by simply adding zeros: wavelet coordinates have the character of differences, this prolongation corresponds to the exact representation in higher resolution wavelet coordinates. The resulting *Nested–Iteration–Incomplete–Conjugate–Gradient* Algorithm is the following.

NEICG$[J] \to \mathbf{u}^J$

(I) INITIALIZATION FOR COARSEST LEVEL $j := j_0$
  (1) COMPUTE RIGHT HAND SIDE $\mathbf{g}^{j_0} = (\mathbf{Z}^T\mathbf{G})^{j_0}$ BY QR DECOMPOSITION USING (4.29).
  (2) COMPUTE SOLUTION $\mathbf{u}^{j_0}$ OF (4.34) BY QR DECOMPOSITION.

(II) WHILE $j < J$
  (1) PROLONGATE $\mathbf{u}^j \to \mathbf{u}_0^{j+1}$ BY ADDING ZEROS, SET $j := j+1$.
  (2) COMPUTE RIGHT HAND SIDE USING RHS $[2^{-(d-1)j}, \mathbf{A}, \mathbf{f}^j, \mathbf{y}_*^j] \to \mathbf{g}^j$.
  (3) COMPUTE SOLUTION OF (4.34) USING CG $[2^{-(d-1)j}, \mathbf{u}_0^j, \mathbf{Q}, \mathbf{g}^j] \to \mathbf{u}^j$.

Recall that step (II.3) requires multiple calls of APP$[\eta, \mathbf{Q}, \mathbf{d}]$, which in turn invokes both CG $[\ldots, \mathbf{A}, \ldots]$ as well as CG $[\ldots, \mathbf{A}^T, \ldots]$ in each application. On account of (2.13) and (4.35), finite versions of the system matrices $\mathbf{A}$ and $\mathbf{Q}$ have uniformly bounded condition numbers, entailing that each CG routine employed in the process reduces the error by a fixed rate $\rho < 1$ in each iteration step. Let $N_J \sim 2^{nJ}$ be the total number of unknowns (for

$\mathbf{y}^J, \mathbf{u}^J$ and $\mathbf{p}^J$) on the highest level J. Employing the CG method only on the highest level, one needs $\mathcal{O}(J) = \mathcal{O}(\log \varepsilon)$ iterations to achieve the prescribed discretization error accuracy $\varepsilon_J = 2^{-(d-1)J}$. As each application of $\mathbf{A}$ and $\mathbf{Q}$ requires $\mathcal{O}(N_J)$ operations, the solution of (4.34) by CG only on the finest level requires $\mathcal{O}(J\, N_J)$ arithmetic operations.

Proposition 5.3. *[BK] If the residual (4.36) is computed up to discretization error proportional to $2^{-(d-1)j}$ on each level j and the corresponding solutions are taken as initial guesses for the next higher level,* NEICG *is an asymptotically optimal method in the sense that it provides the solution $\mathbf{u}^J$ up to discretization error on level J in an overall number of $\mathcal{O}(N_J)$ arithmetic operations.*

Proof. In the above notation, nested iteration allows one to get rid of the factor J in the total number of operations. Starting with the exact solution on the coarsest level j_0, in view of the uniformly bounded condition numbers of $\mathbf{A}$ and $\mathbf{Q}$, one only needs a fixed number of iterations to reduce the error up to discretization error accuracy $\varepsilon_j = 2^{-(d-1)j}$ on each subsequent level j, taking the solution from the previous level as initial guess. Thus, on each level, one needs $\mathcal{O}(N_j)$ operations to realize discretization error accuracy. Since the spaces are nested and the number of unknowns on each level grows like $N_j \sim 2^{nj}$, by a geometric series argument the total number of arithmetic operations stays proportional to $\mathcal{O}(N_J)$. □

Numerical Examples

As an illustration of the ingredients for a distributed control problem, we consider the following example taken from [BK] with the Helmholtz operator in (2.6) ($\mathbf{a} = I$, $c = 1$) and homogeneous Dirichlet boundary condition. A non–constant right hand side $f(x) := 1 + 2.3\exp(-15|x - \frac{1}{2}|)$ is chosen, and the target state is set to a constant $y_* \equiv 1$. We first investigate the role the different norms $\|\cdot\|_{\mathcal{Z}}$ and $\|\cdot\|_{\mathcal{U}}$ in (2.27), encoded in the diagonal matrices $\mathbf{D}_{\mathcal{Z}}, \mathbf{D}_H$ from (4.15), have on the solution. We see in Figure 5.1 for the choice $\mathcal{U} = L_2$ and $\mathcal{Z} = H^s(0,1)$ for different values of s varying between 0 and 1 the solution y (left) and the corresponding control u (right) for fixed weight $\omega = 1$. As s is increased, a stronger tendency of y towards the prescribed state $y_* \equiv 1$ can be observed which is, however, deterred from reaching this state by the homogeneous boundary conditions. Extensive studies of this type can be found in [Bu, BK].

As an example displaying the performance of the proposed fully iterative scheme NEICG in two spatial dimensions, Table 5.3 from [BK] is included. This is an example of a control problem for the Helmholtz operator with Neumann boundary conditions. The stopping criterion for the outer iteration (relative to $\|\cdot\|$ which corresponds to the energy norm) on level j is chosen to be proportional to 2^{-j}. The second column displays the final value of the residual of the outer CG scheme on this level, i.e., $\|\mathbf{r}_K^j\| = \|\textsc{Resd}(\mathbf{u}_K^j)\|$.

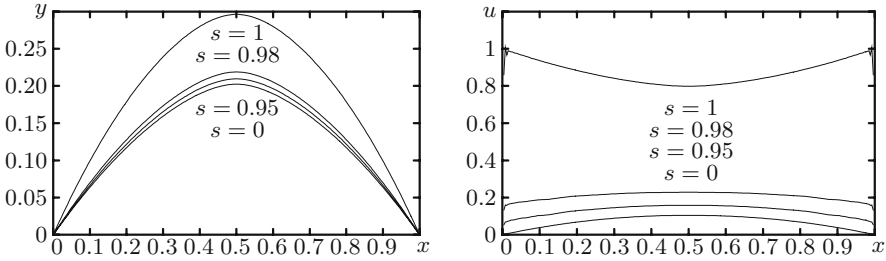


Figure 5.1. Distributed control problem for elliptic problem with Dirichlet boundary conditions, a peak as right hand side f, $y_* \equiv 1$, $\omega = 0$, $\mathcal{U} = L_2$ and varying $\mathcal{Z} = H^s(0,1)$.

The next three columns show the number of outer CG iterations (#O) for $\mathbf{Q}$ according to the APP scheme followed by the maximum number of inner iterations for the primal system (#E), the adjoint system (#A) and the design equation (#R). We clearly see the effect of the uniformly bounded condition numbers of the operators involved. The last columns display different versions of the actual error in the state $\mathbf{y}$ and the control $\mathbf{u}$ when compared to the fine grid solution (R denotes restriction of the fine grid solution to the actual grid, and P prolongation). Here we can see the effect of the constants appearing in (5.11), that is, the error is very well controlled via the residual. More results for up to three spatial dimensions can be found in [Bu, BK].

j	$\|\mathbf{r}_K^j\|$	#O	#E	#A	#R	$\|R(\mathbf{y}^J)-\mathbf{y}^j\|$	$\|\mathbf{y}^J-P(\mathbf{y}^j)\|$	$\|R(\mathbf{u}^J)-\mathbf{u}^j\|$	$\|\mathbf{u}^J-P(\mathbf{u}^j)\|$
3						6.86e-03	1.48e-02	1.27e-04	4.38e-04
4	1.79e-05	5	12	5	8	2.29e-03	7.84e-03	4.77e-05	3.55e-04
5	1.98e-05	5	14	6	9	6.59e-04	3.94e-03	1.03e-05	2.68e-04
6	4.92e-06	7	13	5	9	1.74e-04	1.96e-03	2.86e-06	1.94e-04
7	3.35e-06	7	12	5	9	4.55e-05	9.73e-04	9.65e-07	1.35e-04
8	2.42e-06	7	11	5	10	1.25e-05	4.74e-04	7.59e-07	8.88e-05
9	1.20e-06	8	11	5	10	4.55e-06	2.12e-04	4.33e-07	5.14e-05
10	4.68e-07	9	10	5	9	3.02e-06	3.02e-06	2.91e-07	2.91e-07

Table 5.3. Iteration history for a two-dimensional distributed control problem with Neumann boundary conditions, $\omega = 1$, $\mathcal{Z} = H^1(\Omega)$, $\mathcal{U} = (H^{0.5}(\Omega))'$.

Dirichlet Boundary Control

For the system of saddle point problems (4.39) arising from the control problem with Dirichlet boundary control in Section 2.5, a fully iterative algorithm NEICG can also be designed along the lines above. Again the design equation (4.39c) for $\mathbf{u}$ serves as the equation for which a basic iterative scheme (5.4) can be posed. Of course, the CG method for $\mathbf{A}$ then has to be replaced by a convergent iterative scheme for saddle point operators $\mathbf{L}$ like Uzawa's algorithm. Also the discretization has to be chosen such that the LBB condition is satisfied, see Section 4.2. Details can be found in [K4]. Alternatively,

since $\mathbf{L}$ has a uniformly bounded condition number, the CG scheme can, in principle, also be applied to $\mathbf{L}^T\mathbf{L}$. The performance of wavelet schemes on uniform grids for such systems of saddle point problems arising from optimal control is currently under investigation [P].

Numerical Example
For illustration of the choice of different norms for the Dirichlet boundary control problem, consider the following example taken from [P]. Here we actually have the situation of controlling the system through the control boundary Γ on the right hand side of Figure 5.2 while a prescribed state $y_* \equiv 1$ on the observation boundary Γ_y opposite the control boundary is to be achieved. The right hand side is chosen as constant $f \equiv 1$, and $\omega = 1$. Each layer in Figure 5.2 corresponds to the state y for different values of s when the observation term is measured in $H^s(\Gamma_y)$, that is, the objective functional (2.35) contains a term $\|y - y_*\|^2_{H^s(\Gamma_y)}$ for $s = 0, 1/10, 2/10, 3/10, 4/10, 5/10, 7/10, 9/10$ from bottom to top. We see that as the smoothness index s for the observation increases, the state moves towards the target state at the observation boundary.

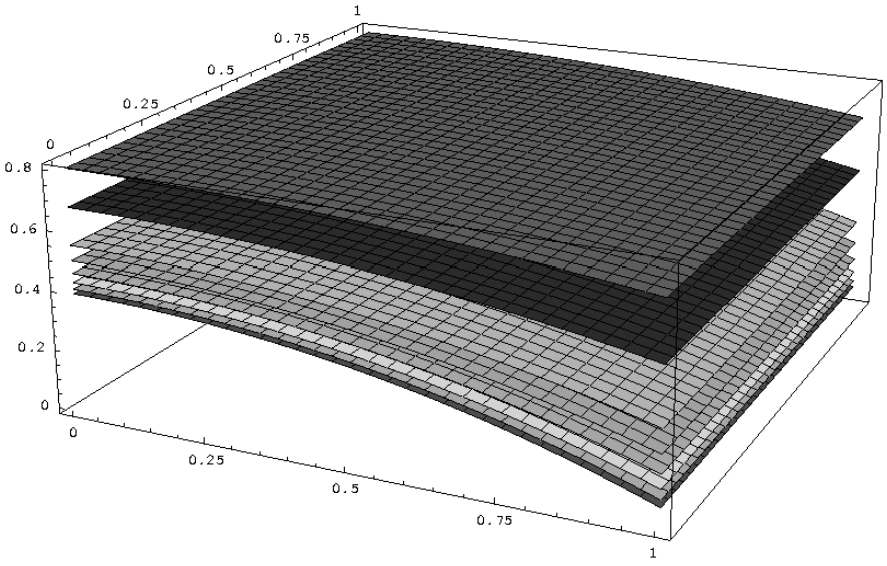


Figure 5.2. State y of the Dirichlet boundary control problem using the objective functional $J(y,u) = \frac{1}{2}\|y - y_*\|^2_{H^s(\Gamma_y)} + \frac{1}{2}\|u\|^2_{H^{1/2}(\Gamma)}$ for $s = 0$, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9 (from bottom to top) on resolution level $J = 5$.

5.2 Adaptive Schemes

In case of the appearance of singularities caused by the data or the domain, a prescribed accuracy may require discretizations with respect to uniform grids to spend a large number of degrees of freedom in areas where the solution is actually smooth. Hence, although the above numerical scheme NEICG is

of optimal linear complexity, the degrees of freedom are not implanted in an optimal way. In these situations, one expects adaptive schemes to work favourably which judiciously place degrees of freedom where singularities occur. Thus, the guiding line for adaptive schemes is to reduce the total number of degrees of freedom when compared to discretizations on a uniform grid. This does not mean that the previous investigations with respect to uniform discretizations are dispensable. In fact, the above results on conditioning carry over to the adaptive case, the solvers are still linear in the number of arithmetic operations and, in particular, one expects to recover the uniform situation when the solutions are smooth. Much on adaptivity for variational problems and the relation to nonlinear approximation can be found in [D4]. The starting point for adaptive wavelet schemes systematically derived for variational problems in [CDD1, CDD2, CDD3] is the infinite formulation in wavelet coordinates as derived for the different problem classes in Section 4. These algorithms have been proven to be optimal in the sense that they match the optimal work/accuracy rate of the wavelet-best N-term approximation, a concept which has been introduced in [CDD1]. The schemes start out with formulating algorithmic ingredients which are then step by step reduced to computable quantities. We follow in this section the material for the distributed control problem from [DK3]. An extension to Dirichlet control problem involving saddle point problems can be found in [K5]. It should be pointed out that the theory is neither confined to symmetric $\mathbf{A}$ nor to the positive definite case.

Algorithmic Ingredients

We start out again with a very simple iterative scheme for the design equation. In view of (4.35) and the fact that $\mathbf{Q}$ is positive definite, there exists a fixed positive parameter α such that in the *Richardson iteration* (which is a special case of a gradient method)

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha(\mathbf{g} - \mathbf{Q}\mathbf{u}^k) \tag{5.18}$$

the error is reduced in each step by at least a factor

$$\rho := \|\mathbf{I} - \alpha\mathbf{Q}\| < 1, \tag{5.19}$$

$$\|\mathbf{u} - \mathbf{u}^{k+1}\| \le \rho\,\|\mathbf{u} - \mathbf{u}^k\|, \quad k = 0, 1, 2, \ldots, \tag{5.20}$$

where $\mathbf{u}$ is the exact solution of (4.34). As the involved system is still infinite, we aim at carrying out this iteration approximately with dynamically updated accuracy tolerances.

The central idea of the wavelet-based adaptive schemes is to start from the infinite system in wavelet coordinates (4.27) and step by step reduce the routines to computable versions of applying the infinite matrix $\mathbf{Q}$ and the evaluation of the right hand side $\mathbf{g}$ of (4.34) involving the inversion of $\mathbf{A}$. The main conceptual tools from [CDD1, CDD2, CDD3] are the following.

We first assume that we have a routine at our disposal with the following property. Later it will be shown how to realize this routine in the concrete case.

RES $[\eta, \mathbf{Q}, \mathbf{g}, \mathbf{v}] \to \mathbf{r}_\eta$

DETERMINES FOR A GIVEN TOLERANCE $\eta > 0$ A FINITELY SUPPORTED SEQUENCE $\mathbf{r}_\eta$ SATISFYING

$$\|\mathbf{g} - \mathbf{Q}\mathbf{v} - \mathbf{r}_\eta\| \le \eta. \tag{5.21}$$

The schemes considered below will also contain the following routine.

COARSE $[\eta, \mathbf{w}] \to \mathbf{w}_\eta$

DETERMINES FOR ANY FINITELY SUPPORTED INPUT VECTOR $\mathbf{w}$ A VECTOR $\mathbf{w}_\eta$ WITH SMALLEST POSSIBLE SUPPORT SUCH THAT

$$\|\mathbf{w} - \mathbf{w}_\eta\| \le \eta. \tag{5.22}$$

This ingredient will eventually play a crucial role in controlling the complexity of the scheme although at this stage its role is not yet apparent. A detailed description of COARSE can be found in [CDD1]. The basic idea is to first sort the entries of $\mathbf{w}$ by size. Then one subtracts squares of their moduli until the sum reaches η^2, starting from the smallest entry. A quasi–sorting based on binary binning can be shown to avoid the logarithmic term in the sorting procedure at the expense of the resulting support size being at most a fixed constant of the minimal size, see [Br].

Next a *perturbed iteration* is designed which converges in the following sense: for every target accuracy ε, the scheme produces after finitely many steps a finitely supported approximate solution with accuracy ε. To obtain a correctly balanced interplay between the routines RES and COARSE, we need the following control parameter. Given (an estimate of) the reduction rate ρ and the step size parameter α from (5.19), let K denote the minimal integer ℓ for which $\rho^{\ell-1}(\alpha\ell + \rho) \le \frac{1}{10}$.

In the following always denoting $\mathbf{u}$ to be the exact solution of (4.34), a perturbed version of (5.18) for a fixed target accuracy $\varepsilon > 0$ is the following.

SOLVE $[\varepsilon, \mathbf{Q}, \mathbf{g}, \overline{\mathbf{q}}^0, \varepsilon_0] \to \overline{\mathbf{q}}_\varepsilon$

(I) GIVEN AN INITIAL GUESS $\overline{\mathbf{q}}^0$ AND AN ERROR BOUND $\|\mathbf{q} - \overline{\mathbf{q}}^0\| \le \varepsilon_0$; SET $j = 0$.

(II) IF $\varepsilon_j \le \varepsilon$, STOP AND SET $\overline{\mathbf{q}}_\varepsilon := \overline{\mathbf{q}}^j$. OTHERWISE SET $\mathbf{v}^0 := \overline{\mathbf{q}}^j$.

(1) FOR $k = 0, \dots, K-1$ COMPUTE RES $[\rho^k \varepsilon_j, \mathbf{Q}, \mathbf{g}, \mathbf{v}^k] \to \mathbf{r}^k$ AND

$$\mathbf{v}^{k+1} := \mathbf{v}^k + \alpha \mathbf{r}^k. \tag{5.23}$$

(2) APPLY COARSE $[\frac{2}{5}\varepsilon_j, \mathbf{v}^K] \to \overline{\mathbf{q}}^{j+1}$; SET $\varepsilon_{j+1} := \frac{1}{2}\varepsilon_j$, $j+1 \to j$ AND GO TO (II).

In the case that no particular initial guess is known, we initialize $\overline{\mathbf{q}}^0 = \mathbf{0}$, set $\varepsilon_0 := c_{\mathbf{Q}}^{-1}\|\mathbf{g}\|$ and briefly write then $\textsc{Solve}\,[\varepsilon, \mathbf{Q}, \mathbf{g}] \to \overline{\mathbf{q}}_\varepsilon$.
In a straightforward manner, perturbation arguments yield the convergence of this algorithm [CDD2, CDD3].

Proposition 5.4. *The iterates* $\overline{\mathbf{q}}^j$ *generated by* $\textsc{Solve}\,[\varepsilon, \mathbf{Q}, \mathbf{g}]$ *satisfy*

$$\|\mathbf{q} - \overline{\mathbf{q}}^j\| \le \varepsilon_j \qquad \text{for any} \quad j \ge 0, \tag{5.24}$$

where $\varepsilon_j = 2^{-j}\varepsilon_0$.

In order to derive appropriate numerical realizations of Solve, recall that (4.34) is equivalent to the KKT conditions (4.27). Although the matrix $\mathbf{A}$ is always assumed to be symmetric here, the distinction between the system matrices for the primal and the dual system, $\mathbf{A}$ and $\mathbf{A}^T$, may be helpful.
The strategy in each step for approximating the residual $\mathbf{g} - \mathbf{Q}\mathbf{u}^k$, that is, realization of the routine Res for the problem (4.34), is based upon the result stated in Proposition 4.5. In turn, this requires solving the two auxiliary systems in (4.27). Since the residual only has to be approximated, these systems will only have to be solved approximately. These approximate solutions, in turn, will be provided again by employing Solve but this time with respect to suitable residual schemes tailored to the systems in (4.27). In our special case, the matrix $\mathbf{A}$ is symmetric positive definite, and the choice of wavelet bases ensures the validity of (2.13). Thus, (5.19) holds for $\mathbf{A}$ and $\mathbf{A}^T$ so that the scheme Solve can indeed be invoked. Although we conceptually use the fact that a gradient iteration for the reduced problem (4.34) reduces the error for $\mathbf{u}$ in each step by a fixed amount, employing (4.27) for the evaluation of the residuals will generate as byproducts approximate solutions to the exact solution triple $(\mathbf{y}, \mathbf{p}, \mathbf{u})$ of (4.27).
Under this hypothesis, next we formulate the ingredients for suitable versions $\textsc{Solve}_{\text{PRM}}$ and $\textsc{Solve}_{\text{ADJ}}$ of Solve for the systems in (4.27). Specifically, this requires identifying residual routines $\textsc{Res}_{\text{PRM}}$ and $\textsc{Res}_{\text{ADJ}}$ for the systems $\textsc{Solve}_{\text{PRM}}$ and $\textsc{Solve}_{\text{ADJ}}$. The main task in both cases is to apply the operators $\mathbf{A}, \mathbf{A}^T$, $\mathbf{D}_H^{-1}$ and $\mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}$. Again we assume for the moment that routines for the application of these operators are available, i.e., that for any $\mathbf{L} \in \{\mathbf{A}, \mathbf{A}^T, \mathbf{D}_H^{-1}, \mathbf{R}^{1/2}\mathbf{D}_{\mathcal{Z}}^{-1}\}$ we have a scheme at our disposal with the following property.

$\textsc{Apply}[\eta, \mathbf{L}, \mathbf{v}] \to \mathbf{w}_\eta$

determines for any finitely supported input vector $\mathbf{v}$ and any tolerance $\eta > 0$ a finitely supported output $\mathbf{w}_\eta$ which satisfies

$$\|\mathbf{L}\mathbf{v} - \mathbf{w}_\eta\| \le \eta. \tag{5.25}$$

The scheme $\textsc{Solve}_{\text{PRM}}$ for the first system in (4.27) is then defined by

$$\textsc{Solve}_{\text{PRM}}\,[\eta, \mathbf{A}, \mathbf{D}_H^{-1}, \mathbf{f}, \mathbf{v}, \overline{\mathbf{y}}^0, \varepsilon_0] := \textsc{Solve}\,[\eta, \mathbf{A}, \mathbf{f} + \mathbf{D}_H^{-1}\mathbf{v}, \overline{\mathbf{y}}^0, \varepsilon_0],$$

where $\overline{\mathbf{y}}^0$ is an initial guess for the solution $\mathbf{y}$ of $\mathbf{A}\mathbf{y} = \mathbf{f} + \mathbf{D}_H^{-1}\mathbf{v}$ with accuracy ε_0. The scheme RES for Step (II) in SOLVE is in this case realized by a new routine RES_{PRM} defined as follows.

$\text{RES}_{\text{PRM}}\,[\eta, \mathbf{A}, \mathbf{D}_H^{-1}, \mathbf{f}, \mathbf{v}, \overline{\mathbf{y}}] \to \mathbf{r}_\eta$

DETERMINES FOR ANY POSITIVE TOLERANCE η, A GIVEN FINITELY SUPPORTED $\mathbf{v}$ AND ANY FINITELY SUPPORTED INPUT $\overline{\mathbf{y}}$ A FINITELY SUPPORTED APPROXIMATE RESIDUAL $\mathbf{r}_\eta$ SATISFYING (5.21), THAT IS,

$$\|\mathbf{f} + \mathbf{D}_H^{-1}\mathbf{v} - \mathbf{A}\overline{\mathbf{y}} - \mathbf{r}_\eta\| \le \eta, \tag{5.26}$$

AS FOLLOWS:

(I) $\text{APPLY}[\frac{1}{3}\eta, \mathbf{A}, \overline{\mathbf{y}}] \to \mathbf{w}_\eta$;
(II) $\text{COARSE}[\frac{1}{3}\eta, \mathbf{f}] \to \mathbf{f}_\eta$;
(III) $\text{APPLY}[\frac{1}{3}\eta, \mathbf{D}_H^{-1}, \mathbf{v}] \to \mathbf{z}_\eta$;
(IV) SET $\mathbf{r}_\eta := \mathbf{f}_\eta + \mathbf{z}_\eta - \mathbf{w}_\eta$.

For RES_{PRM} and the subsequent variants of RES, by the triangle inequality one can show that indeed (5.26) or (5.21) holds.

Similarly, one needs a version of SOLVE for the approximate solution of the second system (4.27b), $\mathbf{A}^T\mathbf{p} = -\mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y} - \mathbf{y}_*)$, which depends on an approximate solution $\overline{\mathbf{y}}$ of the primal system and possibly on some initial guess $\overline{\mathbf{p}}^0$ with accuracy ε_0. Here we set

$$\text{SOLVE}_{\text{ADJ}}\,[\eta, \mathbf{A}, \mathbf{D}_{\mathcal{Z}}^{-1}, \mathbf{y}_*, \overline{\mathbf{y}}, \overline{\mathbf{p}}^0, \varepsilon_0] := \text{SOLVE}\,[\eta, \mathbf{A}^T, \mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\mathbf{y} - \overline{\mathbf{y}}), \overline{\mathbf{p}}^0, \varepsilon_0].$$

As usual we assume that the data $\mathbf{f}, \mathbf{y}_*$ are approximated in a preprocessing step with sufficient accuracy. A suitable residual approximation scheme RES_{ADJ} for Step (II) of this version of SOLVE is the following where the main issue is the approximate evaluation of the right hand side.

$\text{RES}_{\text{ADJ}}\,[\eta, \mathbf{A}, \mathbf{D}_{\mathcal{Z}}^{-1}, \mathbf{y}_*, \overline{\mathbf{y}}, \overline{\mathbf{p}}] \to \mathbf{r}_\eta$

DETERMINES FOR ANY POSITIVE TOLERANCE η, GIVEN FINITELY SUPPORTED DATA $\overline{\mathbf{y}}, \mathbf{y}_*$ AND ANY FINITELY SUPPORTED INPUT $\overline{\mathbf{p}}$ AN APPROXIMATE RESIDUAL $\mathbf{r}_\eta$ SATISFYING (5.21), I.E.,

$$\| - \mathbf{D}_{\mathcal{Z}}^{-1}\mathbf{R}\mathbf{D}_{\mathcal{Z}}^{-1}(\overline{\mathbf{y}} - \mathbf{y}_*) - \mathbf{A}^T\overline{\mathbf{p}} - \mathbf{r}_\eta\| \le \eta, \tag{5.27}$$

AS FOLLOWS:

(I) $\text{APPLY}[\frac{1}{3}\eta, \mathbf{A}^T, \overline{\mathbf{p}}] \to \mathbf{w}_\eta$;
(II) $\text{APPLY}[\frac{1}{6}\eta, \mathbf{D}_{\mathcal{Z}}^{-1}, \overline{\mathbf{y}}] \to \mathbf{z}_\eta$; $\text{COARSE}[\frac{1}{6}\eta, \mathbf{y}_*] \to (\mathbf{y}_*)_\eta$;
SET $\mathbf{d}_\eta := (\mathbf{y}_{\mathcal{Z}})_\eta - \mathbf{z}_\eta$;
$\text{APPLY}[\frac{1}{6}\eta, \mathbf{D}_{\mathcal{Z}}^{-1}, \mathbf{d}_\eta] \to \hat{\mathbf{v}}_\eta$; $\text{APPLY}[\frac{1}{6}\eta, \mathbf{R}, \hat{\mathbf{v}}_\eta] \to \mathbf{v}_\eta$;
(III) SET $\mathbf{r}_\eta := \mathbf{v}_\eta - \mathbf{w}_\eta$.

Finally, we can define the residual scheme for the version of SOLVE applied to (4.34). We shall refer to this specification as $\text{SOLVE}_{\text{DCP}}$ with corresponding residual scheme is RES_{DCP}. Since the scheme is based on Proposition 4.5, it will involve several parameters stemming from the auxiliary systems (4.27).

$\textsc{Res}_{\textsc{dcp}}[\eta, \mathbf{Q}, \mathbf{g}, \tilde{\mathbf{y}}, \delta_y, \tilde{\mathbf{p}}, \delta_p, \mathbf{v}, \delta_v] \to (\mathbf{r}_\eta, \tilde{\mathbf{y}}, \delta_y, \tilde{\mathbf{p}}, \delta_p)$

determines for any approximate solution triple $(\tilde{\mathbf{y}}, \tilde{\mathbf{p}}, \mathbf{v})$ of the system (4.27) satisfying

$$\|\mathbf{y} - \tilde{\mathbf{y}}\| \le \delta_y, \quad \|\mathbf{p} - \tilde{\mathbf{p}}\| \le \delta_p, \quad \|\mathbf{u} - \mathbf{v}\| \le \delta_v, \tag{5.28}$$

an approximate residual $\mathbf{r}_\eta$ such that

$$\|\mathbf{g} - \mathbf{Q}\mathbf{v} - \mathbf{r}_\eta\| \le \eta. \tag{5.29}$$

Moreover, the initial approximations $\tilde{\mathbf{y}}, \tilde{\mathbf{p}}$ are overwritten by new approximations $\tilde{\mathbf{y}}, \tilde{\mathbf{p}}$ satisfying (5.28) with new bounds δ_y and δ_p defined in (5.30) below, as follows:

(i) $\textsc{Solve}_{\textsc{prm}}[\frac{1}{3}c_{\mathbf{A}}\,\eta, \mathbf{A}, \mathbf{D}_H^{-1}, \mathbf{f}, \mathbf{v}, \tilde{\mathbf{y}}, \delta_y] \to \mathbf{y}_\eta$;

(ii) $\textsc{Solve}_{\textsc{adj}}[\frac{1}{3}\eta, \mathbf{A}, \mathbf{D}_{\mathcal{Z}}^{-1}, \mathbf{y}_*, \mathbf{y}_\eta, \tilde{\mathbf{p}}, \delta_p] \to \mathbf{p}_\eta$;

(iii) $\textsc{Apply}[\frac{1}{3}\eta, \mathbf{D}_H^{-1}, \mathbf{p}_\eta] \to \mathbf{q}_\eta$; set $\mathbf{r}_\eta := \mathbf{q}_\eta - \omega\mathbf{v}$;

(iv) set $\xi_y := c_{\mathbf{A}}^{-1}\,\delta_v + \frac{1}{3}c_{\mathbf{A}}\eta$, $\xi_p := c_{\mathbf{A}}^{-2}\,\delta_v + \frac{2}{3}\eta$; replace $\tilde{\mathbf{y}}, \delta_y$ and $\tilde{\mathbf{p}}, \delta_p$ by

$$\begin{aligned} \tilde{\mathbf{y}} &:= \textsc{Coarse}[4\xi_y, \mathbf{y}_\eta], \quad & \delta_y &:= 5\,\xi_y, \\ \tilde{\mathbf{p}} &:= \textsc{Coarse}[4\xi_p, \mathbf{p}_\eta], \quad & \delta_p &:= 5\,\xi_p. \end{aligned} \tag{5.30}$$

Step (iv) already indicates the conditions on the tolerance η and the accuracy bound δ_v under which the new error bounds in (5.30) are actually tighter. The precise relation between η and δ_v in the context of $\textsc{Solve}_{\textsc{dcp}}$ is not apparent yet and emerges as well as the claimed estimates (5.29) and (5.30) from the complexity analysis in [DK3].

Finally, the scheme $\textsc{Solve}_{\textsc{dcp}}$ attains the following form with the error reduction factor ρ from (5.19) and α from (5.18).

$\textsc{Solve}_{\textsc{dcp}}\,[\varepsilon, \mathbf{Q}, \mathbf{g}] \to \overline{\mathbf{u}}_\varepsilon$

(i) Let $\overline{\mathbf{q}}^0 := \mathbf{0}$ and $\varepsilon_0 := c_{\mathbf{A}}^{-1}(\|\mathbf{y}_Z\| + c_{\mathbf{A}}^{-1}\|\mathbf{f}\|)$.
Let $\tilde{\mathbf{y}} := \mathbf{0}$, $\tilde{\mathbf{p}} := \mathbf{0}$ and set $j = 0$.
Define $\delta_y := \delta_{y,0} := c_{\mathbf{A}}^{-1}(\|\mathbf{f}\| + \varepsilon_0)$ and $\delta_p := \delta_{p,0} := c_{\mathbf{A}}^{-1}(\delta_{y,0} + \|\mathbf{y}_Z\|)$.

(ii) If $\varepsilon_j \le \varepsilon$, stop and set $\overline{\mathbf{u}}_\varepsilon := \overline{\mathbf{u}}^j$, $\overline{\mathbf{y}}_\varepsilon = \tilde{\mathbf{y}}$, $\overline{\mathbf{p}}_\varepsilon = \tilde{\mathbf{p}}$.
Otherwise set $\mathbf{v}^0 := \overline{\mathbf{u}}^j$.

(1) For $k = 0, \dots, K-1$, compute
$\textsc{Res}_{\textsc{dcp}}\,[\rho^k\varepsilon_j, \mathbf{Q}, \mathbf{g}, \tilde{\mathbf{y}}, \delta_y, \tilde{\mathbf{p}}, \delta_p, \mathbf{v}^k, \delta_k] \to (\mathbf{r}^k, \tilde{\mathbf{y}}, \delta_y, \tilde{\mathbf{p}}, \delta_p)$,
where $\delta_0 := \varepsilon_j$ and $\delta_k := \rho^{k-1}(\alpha k + \rho)\varepsilon_j$;
set

$$\mathbf{v}^{k+1} := \mathbf{v}^k + \alpha\mathbf{r}^k. \tag{5.31}$$

(2) $\textsc{Coarse}\,[\frac{2}{5}\varepsilon_j, \mathbf{v}^K] \to \overline{\mathbf{u}}^{j+1}$; set $\varepsilon_{j+1} := \frac{1}{2}\varepsilon_j$, $j+1 \to j$ and go to (ii).

By overwriting $\tilde{\mathbf{y}}$, $\tilde{\mathbf{p}}$ at the last stage prior to the termination of $\textsc{Solve}_{\textsc{dcp}}$ one has $\delta_v \le \varepsilon$, $\eta \le \varepsilon$, so that the following fact is an immediate consequence of (5.30).

Proposition 5.5. *The outputs* $\overline{\mathbf{y}}_\varepsilon$ *and* $\overline{\mathbf{p}}_\varepsilon$ *produced by* SOLVE$_{\mathrm{DCP}}$ *in addition to* $\mathbf{u}_\varepsilon$ *are approximations to the exact solutions* $\mathbf{y}, \mathbf{p}$ *of (4.27) satisfying*

$$\|\mathbf{y} - \overline{\mathbf{y}}_\varepsilon\| \leq 5\varepsilon\,(c_{\mathbf{A}}^{-1} + \tfrac{1}{3}c_{\mathbf{A}}), \qquad \|\mathbf{p} - \overline{\mathbf{p}}_\varepsilon\| \leq 5\varepsilon\,(c_{\mathbf{A}}^{-2} + \tfrac{2}{3}).$$

Complexity Analysis

Proposition 5.4 states that the routine SOLVE converges for an arbitrary given accuracy provided that there is a routine RES satisfying the property (5.21). Then we have broken down step by step the necessary ingredients to derive computable versions which satisfy these requirements. What we finally want to show is that the routines are *optimal* in the sense that they provide the optimal work/accuracy rate in terms of best N–term approximation. The complexity analysis given next also reveals the role of the routine COARSE within the algorithms and the particular choices of the thresholds in Step (IV) of RES$_{\mathrm{DCP}}$.

In order to be able to assess the quality of the adaptive algorithm, the notion of *optimality* has to be clarified first in the present context.

Definition 5.1. The scheme SOLVE has an *optimal work/accuracy rate* s if the following holds: Whenever the error of *best* N*–term approximation* satisfies

$$\|\mathbf{q} - \mathbf{q}_N\| := \min_{\#\operatorname{supp}\mathbf{v}\leq N} \|\mathbf{q} - \mathbf{v}\| \lesssim N^{-s},$$

then the solution $\overline{\mathbf{q}}_\varepsilon$ is generated by SOLVE at an expense that also stays proportional to $\varepsilon^{-1/s}$ and in that sense matches the best N–term approximation rate.

Note that this implies that $\#\operatorname{supp}\overline{\mathbf{q}}_\varepsilon$ also stays proportional to $\varepsilon^{-1/s}$. Thus, our benchmark is that whenever the solution of (4.34) can be approximated by N terms at rate s, SOLVE recovers that rate asymptotically. If $\mathbf{q}$ is known, the wavelet-best N–term approximation $\mathbf{q}_N$ of $\mathbf{q}$ is given by picking the N largest terms in modulus from $\mathbf{q}$, of course. However, when $\mathbf{q}$ is the (unknown) solution of (4.34) this information is certainly not available.

Since we are here in the framework of sequence spaces ℓ_2, the formulation of appropriate criteria for complexity will be based on a characterization of sequences which are *sparse* in the following sense. We consider sequences $\mathbf{v}$ for which the best N–term approximation error decays at a particular rate (*Lorentz spaces*). That is, for any given threshold $0 < \eta \leq 1$, the number of terms exceeding that threshold is controlled by some function of this threshold. In particular, for some $0 < \tau < 2$ set

$$\ell_\tau^w := \{\mathbf{v} \in \ell_2 :\ \#\{\lambda \in \mathcal{I} : |v_\lambda| > \eta\} \leq C_{\mathbf{v}}\,\eta^{-\tau},\ \text{for all } 0 < \eta \leq 1\}. \quad (5.32)$$

This determines a strict subspace of ℓ_2 only when $\tau < 2$. Smaller τ's indicate sparser sequences. For a given $\mathbf{v} \in \ell_\tau^w$ let $C_{\mathbf{v}}$ be the smallest constant for which (5.32) holds. Then one has $|\mathbf{v}|_{\ell_\tau^w} := \sup_{n\in\mathbb{N}} n^{1/\tau} v_n^* = C_{\mathbf{v}}^{1/\tau}$, where

$\mathbf{v}^* = (v_n^*)_{n\in\mathbb{N}}$ is a non–decreasing rearrangement of $\mathbf{v}$. Furthermore, $\|\mathbf{v}\|_{\ell_\tau^w} := \|\mathbf{v}\| + |\mathbf{v}|_{\ell_\tau^w}$ is a quasi–norm for ℓ_τ^w. Since the continuous embeddings $\ell_\tau \hookrightarrow \ell_\tau^w \hookrightarrow \ell_{\tau+\varepsilon} \hookrightarrow \ell_2$ hold for $\tau < \tau + \varepsilon < 2$, ℓ_τ^w is 'close' to ℓ_τ and is therefore called *weak* ℓ_τ. The following crucial result connects sequences in ℓ_τ^w to best N–term approximation [CDD1].

Proposition 5.6. *Let positive real numbers s and τ be related by*

$$\tfrac{1}{\tau} = s + \tfrac{1}{2}. \tag{5.33}$$

Then $\mathbf{v} \in \ell_\tau^w$ if and only if $\|\mathbf{v} - \mathbf{v}_N\| \lesssim N^{-s}\,\|\mathbf{v}\|_{\ell_\tau^w}$.

The property that an array of wavelet coefficients $\mathbf{v}$ belongs to ℓ_τ is equivalent to the fact that the expansion $\mathbf{v}^T\Psi_H$ in terms of a wavelet basis Ψ_H for a Hilbert space H belongs to a certain *Besov space* which describes a much weaker regularity measure than a Sobolev space of corresponding order, see, e.g., [Co,DV]. Thus, Proposition 5.6 expresses how much loss of regularity can be compensated by judiciously placing the degrees of freedom in a nonlinear way in order to retain a certain optimal order of error decay.

A key criterion for a scheme SOLVE to exhibit an optimal work/accuracy rate can be formulated through the following property of the respective residual approximation. The routine RES is called τ^**–sparse* for some $0 < \tau^* < 2$ if the following holds: Whenever the solution $\mathbf{q}$ of (4.34) belongs to ℓ_τ^w for some $\tau^* < \tau < 2$, then for any $\mathbf{v}$ with finite support the output $\mathbf{r}_\eta$ of RES $[\eta, \mathbf{Q}, \mathbf{g}, \mathbf{v}]$ satisfies

$$\|\mathbf{r}_\eta\|_{\ell_\tau^w} \lesssim \max\{\|\mathbf{v}\|_{\ell_\tau^w}, \|\mathbf{q}\|_{\ell_\tau^w}\} \quad\text{and}\quad \#\operatorname{supp}\mathbf{r}_\eta \lesssim \eta^{-1/s}\max\{\|\mathbf{v}\|_{\ell_\tau^w}^{1/s}, \|\mathbf{q}\|_{\ell_\tau^w}^{1/s}\}$$

where s and τ are related by (5.33), and the number of floating point operations needed to compute $\mathbf{r}_\eta$ stays proportional to $\#\operatorname{supp}\mathbf{r}_\eta$.

The analysis in [CDD2] then yields the following result.

Theorem 5.1. *If* RES *is τ^*–sparse and if the exact solution $\mathbf{q}$ of (4.34) belongs to ℓ_τ^w for some $\tau > \tau^*$, then for every $\varepsilon > 0$ algorithm* SOLVE $[\varepsilon, \mathbf{Q}, \mathbf{g}]$ *produces after finitely many steps an output $\overline{\mathbf{q}}_\varepsilon$ (which, according to Proposition 5.4, always satisfies $\|\mathbf{q} - \overline{\mathbf{q}}_\varepsilon\| < \varepsilon$) with the following properties: For s and τ related by (5.33), one has*

$$\#\operatorname{supp}\overline{\mathbf{q}}_\varepsilon \lesssim \varepsilon^{-1/s}\|\mathbf{q}\|_{\ell_\tau^w}^{1/s}, \qquad \|\overline{\mathbf{q}}_\varepsilon\|_{\ell_\tau^w} \lesssim \|\mathbf{q}\|_{\ell_\tau^w}, \tag{5.34}$$

and the number of floating point operations needed to compute $\overline{\mathbf{q}}_\varepsilon$ remains proportional to $\#\operatorname{supp}\overline{\mathbf{q}}_\varepsilon$.

Hence, τ^*-sparsity of the routine RES implies asymptotically optimal work/accuracy rates for SOLVE for a certain range of decay rates given by τ^*. We stress that the algorithm itself does *not* require any *a priori* knowledge about the solution such as its actual best N–term approximation rate. Theorem 5.1 also states that controlling the ℓ_τ^w–norm of the quantities generated in the computations is crucial. This finally explains the role of COARSE in Step (II.2) of SOLVE in terms of the following result [CDD1].

Lemma 5.1. *Let* $\mathbf{v} \in \ell_\tau^w$ *and let* $\mathbf{w}$ *be any finitely supported approximation such that* $\|\mathbf{v}-\mathbf{w}\| \le \frac{1}{5}\eta$. *Then the output* $\mathbf{w}_\eta$ *of* COARSE$[\frac{4}{5}\eta, \mathbf{w}]$ *satisfies*

$$\#\operatorname{supp}\mathbf{w}_\eta \lesssim \|\mathbf{v}\|_{\ell_\tau^w}^{1/\tau}\,\eta^{-1/s}, \quad \|\mathbf{v}-\mathbf{w}_\eta\| \lesssim \eta, \quad \text{and} \quad \|\mathbf{w}_\eta\|_{\ell_\tau^w} \lesssim \|\mathbf{v}\|_{\ell_\tau^w}. \tag{5.35}$$

This can be interpreted as follows. If an error bound for a given finitely supported approximation $\mathbf{w}$ is known, a certain coarsening using only knowledge about $\mathbf{w}$ produces a new approximation to (the possibly unknown) $\mathbf{v}$ which gives rise to a slightly larger error but realizes up to a uniform constant the optimal relation between support and accuracy. In the scheme SOLVE, this means that the ℓ_τ^w–norms of the iterates $\mathbf{v}^K$ are controlled by the coarsening step.

It remains to establish that for $\text{SOLVE}_{\text{DCP}}$ the corresponding routine RES_{DCP} is τ^*-sparse. The following results from [DK3] reduce this question to the efficiency of APPLY. We say that APPLY$[\cdot, \mathbf{L}, \cdot]$ is τ^*–*efficient* for some $0 < \tau^* < 2$ if for any finitely supported $\mathbf{v} \in \ell_\tau^w$, for $0 < \tau^* < \tau < 2$, the output $\mathbf{w}_\eta$ of APPLY$[\eta, \mathbf{L}, \mathbf{v}]$ satisfies $\|\mathbf{w}_\eta\|_{\ell_\tau^w} \lesssim \|\mathbf{v}\|_{\ell_\tau^w}$ and $\#\text{supp}\,\mathbf{w}_\eta \lesssim \eta^{-1/s}\|\mathbf{v}\|_{\ell_\tau^w}^{1/s}$ for $\eta \to 0$. Here the constants depend only on τ as $\tau \to \tau^*$ and s, τ satisfy (5.33). Moreover, the number of floating point operations needed to compute $\mathbf{w}_\eta$ is to remain proportional to $\#\text{supp}\,\mathbf{w}_\eta$.

Proposition 5.7. *If the* APPLY *schemes in* RES_{PRM} *and* RES_{ADJ} *are* τ^*–*efficient for some* $\tau^* < 2$, *then* RES_{DCP} *is* τ^*–*sparse whenever there exists a constant* C *such that* $C\eta \ge \max\{\delta_v, \delta_p\}$ *and*

$$\max\{\|\tilde{\mathbf{p}}\|_{\ell_\tau^w}, \|\tilde{\mathbf{y}}\|_{\ell_\tau^w}, \|\mathbf{v}\|_{\ell_\tau^w}\} \le C\left(\|\mathbf{y}\|_{\ell_\tau^w} + \|\mathbf{p}\|_{\ell_\tau^w} + \|\mathbf{u}\|_{\ell_\tau^w}\right),$$

where $\mathbf{v}$ *is the current finitely supported input and* $\tilde{\mathbf{y}}, \tilde{\mathbf{p}}$ *are the initial guesses for the exact solution components* $(\mathbf{y}, \mathbf{p})$.

Theorem 5.2. *If the* APPLY *schemes appearing in* RES_{PRM} *and* RES_{ADJ} *are* τ^**-efficient for some* $\tau^* < 2$ *and the components of the solution* $(\mathbf{y}, \mathbf{p}, \mathbf{u})$ *of (4.27) all belong to the respective space* ℓ_τ^w *for some* $\tau > \tau^*$, *then the approximate solutions* $\mathbf{y}_\varepsilon, \mathbf{p}_\varepsilon, \mathbf{u}_\varepsilon$, *produced by* $\text{SOLVE}_{\text{DCP}}$ *for any target accuracy* ε, *satisfy*

$$\|\mathbf{y}_\varepsilon\|_{\ell_\tau^w} + \|\mathbf{p}_\varepsilon\|_{\ell_\tau^w} + \|\mathbf{u}_\varepsilon\|_{\ell_\tau^w} \lesssim \|\mathbf{y}\|_{\ell_\tau^w} + \|\mathbf{p}\|_{\ell_\tau^w} + \|\mathbf{u}\|_{\ell_\tau^w}, \tag{5.36}$$

and

$$(\#\text{supp}\,\mathbf{y}_\varepsilon) + (\#\text{supp}\,\mathbf{p}_\varepsilon) + (\#\text{supp}\,\mathbf{u}_\varepsilon) \lesssim \left(\|\mathbf{y}\|_{\ell_\tau^w}^{1/s} + \|\mathbf{p}\|_{\ell_\tau^w}^{1/s} + \|\mathbf{u}\|_{\ell_\tau^w}^{1/s}\right)\varepsilon^{-1/s}, \tag{5.37}$$

where the constants only depend on τ *when* τ *approaches* τ^*. *Moreover, the number of floating point operations required during the execution of* $\text{SOLVE}_{\text{DCP}}$ *remains proportional to the right hand side of (5.37).*

Thus, the practical realization of $\text{SOLVE}_{\text{DCP}}$ providing optimal work/accuracy rates for a possibly large range of decay rates of the error of best N–term approximation hinges on the availability of τ^*–efficient schemes APPLY with possibly small τ^* for the involved operators.
For the approximate application of wavelet representations of a wide class of operators, including differential operators, one can indeed devise efficient schemes which is a consequence of the cancellation properties (CP) together with the norm equivalences (3.3) for the relevant function spaces. For the example considered above, the τ^*–efficiency of $\mathbf{A}$ defined in (4.18) can be shown whenever $\mathbf{A}$ is s^*–compressible where τ^* and s^* are related by (5.33). One knows that s^* is larger the higher the 'regularity' of the operator and the order of cancellation properties of the wavelets are. Estimates for s^* in terms of these quantities for spline wavelets and the above differential operator A can be found in [BCDU]. Hence, Theorem 5.2 guarantees asymptotically optimal complexity bounds for $\tau > \tau^*$. This means that the scheme $\text{SOLVE}_{\text{DCP}}$ recovers rates of the error of best N–term approximation of order N^{-s} for $s < s^*$.
When describing the control problem, it has been pointed out that the wavelet framework allows for a flexible choice of norms in the control functional which is reflected by the diagonal matrices $\mathbf{D}_{\mathcal{Z}}$ and $\mathbf{D}_H$ in (DCP), (4.23) together with (4.24). The following result states that multiplication by either $\mathbf{D}_{\mathcal{Z}}^{-1}$ or $\mathbf{D}_H^{-1}$ makes a sequence more compressible, that is, they produce a shift in weak ℓ_τ spaces [DK3].

Proposition 5.8. *For $\beta > 0$, $\mathbf{p} \in \ell_\tau^w$ implies $\mathbf{D}^{-\beta}\mathbf{p} \in \ell_{\tau'}^w$, where $\frac{1}{\tau'} := \frac{1}{\tau} + \frac{\beta}{d}$.*

We can conclude with the following. Whatever the sparsity class of the adjoint variable $\mathbf{p}$ is, the control $\mathbf{u}$ is in view of (4.27c) even sparser. This also means that although the control $\mathbf{u}$ may be accurately recovered with relatively few degrees of freedom, the overall solution complexity is in the case above bounded from below by the less sparse auxiliary variable $\mathbf{p}$.

Acknowledgments

I want to thank Carsten Burstedde and Roland Pabel for their assistance during the preparation of this manuscript. This work has been supported in part by the Deutsche Forschungsgemeinschaft (SFB 611).

References

[Ba1] I. Babuška, The finite element method with Lagrange multipliers, Numer. Math. 20, 1973, 179–192.

[Ba2] I. Babuška, The finite element method with penalty, Math. Comp. 27, 1973, 221–228.

[Br] A. Barinka, Fast Evaluation Tools for Adaptive Wavelet Schemes, PhD. Dissertation, RWTH Aachen, 2004.

[BCDU] A. Barinka, T. Barsch, Ph. Charton, A. Cohen, S. Dahlke, W. Dahmen, K. Urban, Adaptive wavelet schemes for elliptic problems — Implementation and numerical experiments, SIAM J. Sci. Comp., 23 (2001), 910–939.

[Be] S. Bertoluzza, Wavelet stabilization of the Lagrange multiplier method, Numer. Math., 86 (2000), 1–28.

[B] D. Braess, Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics, 2nd ed., Cambridge University Press, Cambridge, 2001.

[BPX] J.H. Bramble, J.E. Pasciak, J. Xu, Parallel multilevel preconditioners, Math. Comp. 55, (1990), 1–22.

[BF] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer, 1991.

[Bu] C. Burstedde, Wavelets Methods for Linear-Quadratic, Elliptic Optimal Control Problems, PhD Thesis, in preparation.

[BK] C. Burstedde, A. Kunoth, Fast iterative solution of elliptic control problems in wavelet discretizations, SFB 611 Preprint No. 127, Universität Bonn, December 2003, submitted for publication.

[CTU] C. Canuto, A. Tabacco, K. Urban, The wavelet element method, part I: Construction and analysis, Appl. Comput. Harm. Anal., 6 (1999), 1–52.

[CDP] J.M. Carnicer, W. Dahmen, J.M. Peña, Local decomposition of refinable spaces, Appl. Comp. Harm. Anal., 3 (1996), 127–153.

[CF] Z. Ciesielski, T. Figiel, Spline bases in classical function spaces on compact C^∞ manifolds: Part I and II, Studia Mathematica (1983), 1–58 and 95–136.

[Co] A. Cohen, Numerical Analysis of Wavelet Methods, Studies in Mathematics and its Applications 32, Elsevier, 2003.

[CDD1] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations – Convergence rates, Math. Comp. 70, 2001, 27–75.

[CDD2] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods II – Beyond the elliptic case, Found. Comput. Math. 2, 2002, 203–245.

[CDD3] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet schemes for nonlinear variational problems, SIAM J. Numer. Anal. 41 (5), 2003, 1785–1823.

[CDF] A. Cohen, I. Daubechies, J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, Comm. Pure Appl. Math. 45 (1992), 485–560.

[CM] A. Cohen, R. Masson, Adaptive wavelet methods for second order elliptic problems, preconditioning and adaptivity, SIAM J. Sci. Comp., 21 (1999), 1006–1026.

[DDU] S. Dahlke, W. Dahmen, K. Urban, Adaptive wavelet methods for saddle point problems — Optimal convergence rates, SIAM J. Numer. Anal. 40 (2002), 1230–1262.

[D1] W. Dahmen, Stability of multiscale transformations, J. Four. Anal. Appl., 2 (1996), 341–361.

[D2] W. Dahmen, Wavelet and multiscale methods for operator equations, Acta Numerica (1997), 55–228.

[D3] W. Dahmen, Wavelet methods for PDEs – Some recent developments, J. Comput. Appl. Math., 128 (2001), 133–185.

[D4] W. Dahmen, Multiscale and wavelet methods for operator equations, in: Multiscale Problems and Methods in Numerical Simulation, C. Canuto (ed.), C.I.M.E. Lecture Notes in Mathematics 1825, Springer Heidelberg 2003, 31–96.

[DK1] W. Dahmen, A. Kunoth, Multilevel preconditioning, Numer. Math., 63 (1992), 315–344.

[DK2] W. Dahmen, A. Kunoth, Appending boundary conditions by Lagrange multipliers: Analysis of the LBB condition, Numer. Math., 88 (2001), 9–42.

[DK3] W. Dahmen, A. Kunoth, Adaptive wavelet methods for linear–quadratic elliptic control problems: Convergence rates, Preprint No. 46, SFB 611, Universität Bonn, December 2002, revised May 2004, to appear in: SIAM J. Contr. Optim.

[DKS] W. Dahmen, A. Kunoth, R. Schneider, Wavelet least squares methods for boundary value problems, SIAM J. Numer. Anal., 39 (2002), 1985–2013.

[DKU] W. Dahmen, A. Kunoth, K. Urban, Biorthogonal spline wavelets on the interval – Stability and moment conditions, Appl. Comput. Harm. Anal., 6 (1999), 132–196.

[DS1] W. Dahmen, R. Schneider, Wavelets with complementary boundary conditions — Function spaces on the cube, Results in Mathematics, 34 (1998), 255–293.

[DS2] W. Dahmen, R. Schneider, Composite wavelet bases for operator equations, Math. Comp., 68 (1999), 1533–1567.

[DS3] W. Dahmen, R. Schneider, Wavelets on manifolds I: Construction and domain decomposition, SIAM J. Math. Anal., 31 (1999), 184–230.

[DSt] W. Dahmen, R. Stevenson, Element–by–element construction of wavelets satisfying stability and moment conditions, SIAM J. Numer. Anal., 37 (1999), 319–325.

[Dau] I. Daubechies, Orthonormal bases of compactly supported wavelets, Comm. Pure Appl. Math., 41 (1988), 909–996.

[DV] Ronald A. DeVore, Nonlinear Approximation, Acta Numerica, 7, (1998), 51-150.

[ES] J. van den Eshof, G.L.G. Sleijpen, Inexact Krylov subspace methods for linear systems, SIAM J. Matr. Anal. Appl. 26 (2004), 125-153.

[GG] V. Girault, R. Glowinski, Error analyis of a fictitious domain method applied to a Dirichlet problem, Japan J. Industr. Appl. Math., 12 (1995), 487–514.

[GR] V. Girault, P.-A. Raviart, Finite Element Methods for Navier–Stokes Equations, Springer, 1986.

[Gr] P. Grisvard, Elliptic Problems in Nonsmooth Domains, Pitman, 1985.

[GL] M.D. Gunzburger, H.C. Lee, Analysis, approximation, and computation of a coupled solid/fluid temperature control problem, Comp. Meth. Appl. Mech. Engrg., 118 (1994), 133–152.

[HM] J. Haslinger, R.A.E. Mäkinen, Introduction to Shape Optimization: Theory, Approximation, and Computation, SIAM, 2003.

[J] S. Jaffard, Wavelet methods for fast resolution of elliptic problems, Siam J. Numer. Anal., 29 (1992), 965-986.

[KP] K. Kunisch, G. Peichl, Shape optimization for mixed boundary value problems based on an embedding domain method, Dyn. Contin. Discrete Impulsive Syst., 4 (1998), 439 – 478.

[K1] A. Kunoth, Multilevel Preconditioning, Verlag Shaker, Aachen 1994.

[K2] A. Kunoth, Wavelet Methods — Elliptic Boundary Value Problems and Control Problems, Advances in Numerical Mathematics, Teubner, 2001.

[K3] A. Kunoth, Wavelet techniques for the fictitious domain—Lagrange multiplier approach, Numer. Algor., 27 (2001), 291–316.

[K4] A. Kunoth, Fast iterative solution of saddle point problems in optimal control based on wavelets, Comput. Optim. Appl., 22 (2002), 225–259.

[K5] A. Kunoth, Adaptive wavelet methods for an elliptic control problem with Dirichlet boundary control, Preprint # 109, SFB 611, Universität Bonn, November 2003, to appear in: Numer. Algor.

[KS] A. Kunoth, J. Sahner, Wavelets on manifolds: An optimized construction, SFB 611 Preprint No. 163, Universität Bonn, July 2004, submitted for publication.

[Kr] J. Krumsdorf, Finite Element Wavelets for the Numerical Solution of Elliptic Partial Differential Equations on Polygonal Domains, Diploma Thesis (in English), Universität Bonn, January 2004.

[Li] J.L. Lions, Optimal Control of Systems Governed by Partial Differential Equations, Springer, Berlin, 1971.

[O] P. Oswald, On discrete norm estimates related to multilevel preconditioners in the finite element method, in: Constructive Theory of Functions, K.G. Ivanov, P. Petrushev, B. Sendov, (eds.), Proc. Int. Conf. Varna 1991, Bulg. Acad. Sci., Sofia (1992), 203–214.

[P] R. Pabel, Wavelet Methods for PDE Constrained Control Problems with Dirichlet Boundary Control, Diploma Thesis, in preparation.

[PT] R. Pinnau, G. Thömmes, Optimal boundary control of glass cooling processes, Math. Methods Appl. Sci. 27 (2004), 1261–1281.

[S] J. Sahner, On the Optimized Construction of Wavelets on Manifolds, Diploma Thesis (in English), Universität Bonn, September 2003.

[St] R. Stenberg, On some techniques for approximating boundary conditions in the finite element method, J. Comp. Appl. Maths., 63 (1995), 139–148.

[Stv] R. Stevenson, Locally supported, piecewise polynomial biorthogonal wavelets on non-uniform meshes, Constr. Approx., 19 (2003), 477-508.

[Sw] W. Sweldens, The lifting scheme: A construction of second generation wavelets, SIAM J. Math. Anal., 29 (1998), 511–546.

[Z] E. Zeidler, Nonlinear Functional Analysis and its Applications; III: Variational Methods and Optimization, Springer, 1985.

On Approximation in Meshless Methods

Jens Markus Melenk

The University of Reading, Department of Mathematics, PO Box 220,
Whiteknights RG6 6AX, United Kingdom
email: j.m.melenk@reading.ac.uk

Abstract We analyze the approximation properties of some meshless methods. Three types of functions systems are discussed: systems of functions that reproduce polynomials, a class of radial basis functions, and functions that are adapted to a differential operator. Additionally, we survey techniques for the enforcement of essential boundary conditions in meshless methods.

1 Introduction

The classical finite element method (FEM) is a well-established tool for numerically solving partial differential equations. New, non-standard methods, that are broadly covered by the term *meshless methods* or *meshfree methods* have recently emerged. A few examples frequently mentioned in this context are the diffuse element method, [87], the *element-free Galerkin* (EFG, [11,13,14]), the X-FEM (extended FEM), [29,84,98], the RKPM (reproducing kernel particle method, [70, 72–75]), the generalized FEM/partition of unity method ([7, 9, 78, 79, 82]), the *hp*-cloud method, [89], the particle partition of unity particle method of [47–51, 96], the finite point method [91], and the method of finite spheres [30]; also the use of radial basis functions, [44, 61, 65, 66, 108, 110] and the older generalized finite difference method of [71] fall into this category. This list is by no means exhaustive, and surveys of such methods include [6, 12, 60]. Two of the reasons given for introducing such methods are:

- The cost of creating good quality meshes can be high. This is particularly true for three-dimensional problems and for problems where the standard FEM requires frequent remeshing such as time-dependent problems and crack propagation problems.
- For some non-standard problems, the standard FEM performs poorly. Here, it is attractive to create custom-tailored methods designed for a particular problem at hand.

A main aim of these notes is to illustrate some of the mechanisms of approximation that underlie meshless methods. In view of the multitude of methods and applications it is impossible to be exhaustive, and a selection had to be made concerning the approximation spaces and the type of approximation results. With respect to the approximation spaces, we have selected three types:

an example of function systems that reproduce polynomials, a class of radial basis functions, and some examples of systems that are tailored to a particular differential operator. The type of approximation results that we obtain are mostly formulated with a view to an application in projection methods for second order elliptic problems. Since the natural setting of such problems is that of the Hilbert space H^1 (or subspaces thereof), most approximation results are formulated in this norm.

1.1 Notation

General Notation
We write $\mathbb{N} = \{1, 2, \dots, \}$ for the positive integers and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ represents the non-negative integers. $\mathbb{R}^+$ stands for the positive real numbers, $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$ for the non-negative real numbers. We will denote by $\mathcal{P}_p$ the space of polynomials of degree p in d variables, that is $\mathcal{P}_p = \operatorname{span}\{\prod_{i=1}^d x_i^{\alpha_i} \,|\, \alpha_i \in \mathbb{N}_0 \text{ with } \sum_{i=1}^d \alpha_i \le p\}$. The Euclidean norm on $\mathbb{R}^d$ will be denoted by $\|\cdot\|_2$. Balls of radius r centred at x_0 are denoted by $B_r(x_0)$.

Spaces and Domains
For domains $\Omega \subset \mathbb{R}^d$, integers $k \in \mathbb{N}_0$ and $q \in [1, \infty]$ the Sobolev spaces $W^{k,q}(\Omega)$ are defined in the usual way (see for example [23, Chap. 1]). Also for values of $k \notin \mathbb{N}_0$ and $q \in [1, \infty)$, the Sobolev spaces $W^{k,q}(\Omega)$ are defined in the usual way, [23]; they can be equipped with the so-called Sobolev-Slobodeckij norm as follows: we write $k = \tilde{k} + \kappa$, where $\tilde{k} \in \mathbb{N}_0$ and $\kappa \in (0, 1)$, and we define

$$\|u\|_{W^{k,q}(\Omega)}^q = \|u\|_{W^{\tilde{k},q}(\Omega)}^q + |u|_{W^{k,q}(\Omega)}^q,$$

where the semi-norm $|\cdot|_{W^{k,q}(\Omega)}$ is given by

$$|u|_{W^{k,q}(\Omega)}^q := \sum_{\substack{\alpha \in \mathbb{N}_0^d \\ |\alpha| = \tilde{k}}} \int_\Omega \int_\Omega \frac{|D^\alpha u(x) - D^\alpha u(y)|^q}{\|x - y\|_2^{d + q\kappa}} \, dx \, dy. \tag{1.1}$$

We remark in passing that an equivalent definition of the fractional order Sobolev spaces $W^{k,q}(\Omega)$ based on the interpolation of spaces using the K-method is possible, [15, 104]. The case $q = 2$ is special in that the spaces $W^{k,2}(\Omega)$ are Hilbert spaces; it is customary to write $H^k(\Omega) = W^{k,2}(\Omega)$.
We denote by $H_0^1(\Omega) = \{u \in H^1(\Omega) \,|\, u|_{\partial\Omega} = 0\}$ the space of functions of $H^1(\Omega)$ that vanish on the boundary of Ω.
For $\xi \in \mathbb{R}^d$ with $\|\xi\|_2 = 1$, $x \in \mathbb{R}^d$, $r > 0$, and $\theta \in (0, \pi)$ we define the cone

$$C(x, \xi, \theta, r) := B_r(x) \cap \{y \in \mathbb{R}^d \,|\, (y - x)^\top \xi > \|y - x\| \cos\theta\}. \tag{1.2}$$

A domain Ω is said to satisfy a cone condition with angle θ and radius r if for each $x \in \Omega$ there exists a $\xi \in \mathbb{R}^d$ with $\|\xi\|_2 = 1$ such that $C(x, \xi, \theta, r) \subset \Omega$.

Notation for Particle Methods
In these notes, the approximation spaces V_N will have the form

$$V_N = \operatorname{span}\{\varphi_i \,|\, i = 1, \ldots, N\};$$

as is customary in FEM, the functions φ_i, $i = 1, \ldots, N$, will be called shape functions. We furthermore introduce the *patches* Ω_i, which are the interior of the supports of the shape functions, and the diameters h_i of the patches by

$$\Omega_i := (\operatorname{supp} \varphi_i)^\circ, \qquad h_i := \operatorname{diam} \Omega_i \leq 1.$$

Remark 1.1.
The assumption $h_i \leq 1$ is made for convenience only and could be replaced by boundedness of the patch diameters. ∎

Frequently, a shape function φ_i will be associated with a *particle* $x_i \in \Omega_i$. The particles are collected in the set

$$X_N := \{x_i \,|\, i = 1, \ldots, N\},$$

which throughout these notes will be assumed to consist of N distinct points $x_i \in \mathbb{R}^d$, $i = 1, \ldots, N$. In the parlance of classical FEM the "connectivity" of the shape functions will be important. We therefore define

$$n(x) := \{i \in \mathbb{N} \,|\, x \in \Omega_i\}, \tag{1.3}$$
$$n(i) := \{j \in \mathbb{N} \,|\, \Omega_j \cap \Omega_i \neq \emptyset\}; \tag{1.4}$$

the notation $n(\cdot)$ is reminiscent of "neighbour."

FEM and Projection Methods
Techniques and terminology of the classical FEM will pervade much of these notes, and we refer to [23, 27, 94] for general reference on the topic. We will, for example, employ the notion of shape-regular affine triangulations $\mathcal{T}$ of a domain Ω. Based on such a triangulation of Ω, one can define the space $S^{p,1}(\mathcal{T}) \subset H^1(\Omega)$ of piecewise polynomials of degree p. We refer to [94] for a precise definition of $S^{p,1}(\mathcal{T})$. We will write $S_0^{p,1}(\mathcal{T})$ for the space $S_0^{p,1}(\mathcal{T}) := S^{p,1}(\mathcal{T}) \cap H_0^1(\Omega)$.
Many of the results of the presentation are obtained with a view to an application in projection methods such as the Galerkin method. An example of such as setting is the following: Let X be a Hilbert space, $a : X \times X \to \mathbb{R}$ be a continuous bilinear form, $l \in X'$ be a continuous linear form, and $u \in X$ solve

$$a(u, v) = l(v) \qquad \forall\, v \in X. \tag{1.5}$$

If $V_N \subset X$ is a subspace, then one can define an approximation $u_N \in V_N$ by:

$$\text{Find } u_N \in V_N \text{ such that} \quad a(u_N, v) = l(v) \qquad \forall\, v \in V_N. \tag{1.6}$$

Once a basis of V_N is chosen, the problem (1.6) represents a linear system of equations that has to be solved. Under suitable assumptions on the bilinear form a, one has existence and uniqueness of u_N together with a quasi-optimality result, that is

$$\|u - u_N\|_X \leq C \inf_{v \in V_N} \|u - v\|_X, \tag{1.7}$$

where the constant $C > 0$ is independent of critical parameters (e.g. N). In this situation it is very important to understand the approximation properties of the space V_N employed so as to be able to be give bounds on the infimum in (1.7).

1.2 The Notion of Optimality

When discussing the approximation properties of a space V_N, it is instructive to have a notion of optimality so as to be able to compare this space V_N with the best possible choice. One notion of optimality that is common in approximation theory is that of n-width (see for example [92]): For a normed space X with norm $\|\cdot\|_X$ and a subset $Y \subset X$ one defines for $n \in \mathbb{N}$

$$d_n := \inf_{\substack{E_n \subset X \\ \dim E_n \leq n}} \sup_{u \in Y} \inf_{v \in E_n} \|u - v\|_X;$$

here, the spaces E_n appearing in the first infimum are arbitrary linear subspaces of dimension n. The quantity d_n thus measures how well functions of the set Y can be approximated from linear spaces E_n of dimension n. Clearly, d_n depends on the error measure $\|\cdot\|_X$ and the set Y. For Sobolev spaces we have [62]:

Theorem 1.1. *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain and $k \geq 1$. Then there exists $C > 0$ such that*

$$\inf_{\substack{V_N \subset H^1(\Omega) \\ \dim V_N \leq N}} \sup_{\substack{u \in H^k(\Omega) \\ \|u\|_{H^k(\Omega)} = 1}} \inf_{v \in V_N} \|u - v\|_{L^2(\Omega)} \geq N^{-(k-1)/d}.$$

The converse of Theorem 1.1 is well-known in classical FEM (see for example [23]):

Theorem 1.2. *Let $\mathcal{T}$ be a quasi-uniform triangulation of a domain $\Omega \subset \mathbb{R}^d$ with maximum element size h. Then for $k \geq 1$ and the classical H^1-conforming space $S^{p,1}(\mathcal{T})$ of piecewise polynomials of degree p we have*

$$\inf_{v \in S^{p,1}(\mathcal{T})} \|u - v\|_{H^1(\Omega)} \leq C N^{-(\min\{p+1,k\}-1)/d} \|u\|_{H^k(\Omega)},$$

where $N = \dim S^{p,1}(\mathcal{T}) \sim h^{-d}$.

Theorems 1.1, 1.2 show that the classical FEM attains already the best possible rate of convergence if the only information available about the function to be approximated is membership in some Sobolev space $H^k(\Omega)$. In this setting, the use of approximation spaces V_N different from the classical FEM spaces is mainly justified by algorithmic considerations.

Remark 1.2.
The approximation results of these notes are obtained with a view to an application in classical projection methods such as the Galerkin scheme (1.6). We will not cover nonlinear approximation techniques, for which we refer to [32]. ∎

2 Polynomial Reproducing Systems

The fist class of approximation spaces V_N that we analyze is one where the space V_N reproduces polynomials of degree p. We will see that the approximation properties of such spaces are very similar to the classical FEM spaces. Such spaces can be constructed in different ways. One possibility is based on the moving least squares technique and will be illustrated in Section 2.3.

2.1 Motivation

Let $\Omega \subset \mathbb{R}^d$ be a domain, let $X_N = \{x_i \,|\, i = 1, \dots, N\}$ be a set of particles, and let $V_N = \operatorname{span}\{\varphi_i \,|\, i = 1, \dots, N\}$ be a space of functions defined on Ω. In this chapter, we will make the following assumptions:

Assumption 2.1 (finite overlap). There exists a constant $M \in \mathbb{N}$ such that for every $x \in \Omega$ the cardinality $n(x)$ of the set $n(x)$ satisfies $1 \le \operatorname{card} n(x) \le M$.

Assumption 2.2 (polynomial reproduction property). $\displaystyle\sum_{i=1}^{N} \pi(x_i)\varphi_i(x) = \pi(x)$ for all $x \in \Omega$ and all $\pi \in \mathcal{P}_p$.

Assumption 2.3 (stability). There exist $C_{\mathrm{stab}} \ge 1$, $r_{\mathrm{stab}} \in \mathbb{N}_0$ such that $\|D^\alpha \varphi_i\|_{L^\infty(\Omega)} \le C_{\mathrm{stab}} h_i^{-|\alpha|}$ for all $i \in \{1, \dots, N\}$ and all $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \le r_{\mathrm{stab}}$.

Assumption 2.4 (local comparability of patches). There exists $C_{\mathrm{comp}} > 0$ such that $C_{\mathrm{comp}}^{-1} h_i \le h_j \le C_{\mathrm{comp}} h_i$ for all $i \in \{1, \dots, N\}$ and $j \in n(i)$.

These assumptions are a generalization of certain properties of the classical FEM. For $p = 1$ and shape-regular affine meshes $\mathcal{T}$, the classical piecewise linear FEM shape functions satisfy the above assumptions. For $p > 1$, the shape functions employed in the FEM are not as standardized; nevertheless, a basis of $S^{p,1}(\mathcal{T})$ satisfying Assumptions 2.1–2.4 can be constructed as the following exercise shows.

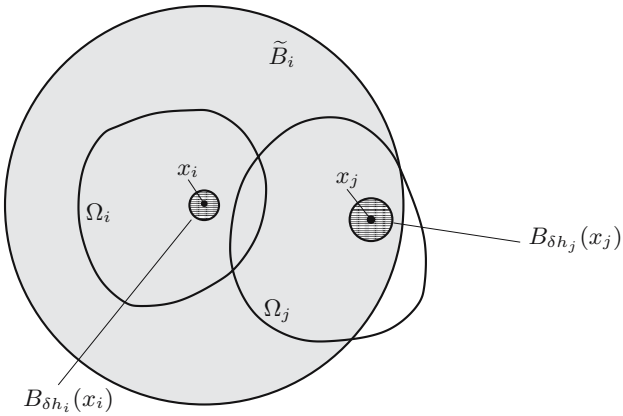


Figure 2.1. Notation of Theorem 2.1.

Exercise 2.1.
Let $\mathcal{T}$ be a mesh on $\Omega = (0,1)$ determined by the points $0 = x_0 < x_1 < \cdots < x_n = 1$. Assume that the element sizes are locally comparable, that is $C^{-1} \le \frac{x_{i+1}-x_i}{x_i - x_{i-1}} \le C$ for $i = 1, \ldots, n-1$. Construct a basis of $S^{p,1}(\mathcal{T}) = \{u \in C([0,1]) \,|\, u|_{(x_i,x_{i+1})} \in \mathcal{P}_p \text{ for } i = 0, \ldots, n-1\}$ such that Assumptions 2.1–2.4 are satisfied. ∎

The construction of shape functions φ_i that satisfy Assumptions 2.1–2.4 will be the topic of Section 2.3.

2.2 Approximation Properties of Systems Reproducing Polynomials

Spaces V_N that satisfy Assumptions 2.1–2.4 inherit the local approximation properties of polynomials:

Theorem 2.1. *Suppose Assumptions 2.1–2.4 hold. Let δ, $C > 0$ be given. Choose for each x_i a ball $\widetilde{B}_i$ with radius $r_i \le Ch_i$ such that $B_{\delta h_j}(x_j) \subset \widetilde{B}_i$ for all $j \in n(i)$ and $\widetilde{B}_i \subset \Omega_i$ (see Figure 2.1).*
Then there exists a linear operator $Q_N : L^1(\mathbb{R}^d) \to V_N$ with the following approximation property: For $u \in H^k(\mathbb{R}^d)$, $k \in \mathbb{N}_0$, with $\sum_{i=1}^N \|u\|^2_{H^k(\widetilde{B}_i)} < \infty$ we have for $s = 0, \ldots, \min\{k, r_{\mathrm{stab}}\}$

$$\|u - Q_N u\|^2_{H^s(\Omega)} \le C \sum_{i=1}^N h_i^{2(\min\{p+1,k\}-s)} \|u\|^2_{H^k(\widetilde{B}_i)}.$$

Remark 2.1.
Theorem 2.1 could be generalized to approximation in the space $W^{k,q}(\Omega)$. Additionally, the proof shows that the balls $\widetilde{B}_i$ could be replaced with other set, e.g. squares, rectangles.

Inspection of the proof also shows that it is sufficient to have u defined on $\cup_{i=1}^N \widetilde{B}_i$ instead of $\mathbb{R}^d$. ∎

Proof of Theorem 2.1. We abbreviate $\mu := \min\{k, p+1\}$ and denote by χ_j the characteristic function of the patch Ω_j, that is $\chi_j(x) = 1$ if $x \in \Omega_j$ and $\chi_j(x) = 0$ if $x \notin \Omega_j$. We note that Assumption 2.1 gives

$$1 \le \sum_{j=1}^N \chi_j(x) \le M \qquad \forall\, x \in \Omega. \tag{2.1}$$

For each patch Ω_i we choose with the aid of the polynomial approximation result Theorem B.1 (and, for the case $\min\{k, p+1\} < \min\{k, r_{\text{stab}}\}$ the inverse estimate Theorem B.2 together with the assumption $h_i \le 1$) a polynomial $\pi_i \in \mathcal{P}_p$ such that

$$\|u - \pi_i\|_{H^s(\widetilde{B}_i)} \le C r_i^{\mu - s} \|u\|_{H^k(\widetilde{B}_i)}, \qquad s = 0, \dots, \min\{k, r_{\text{stab}}\}. \tag{2.2}$$

We then define the desired approximation $Q_N u$ by

$$Q_N u := \sum_{i=1}^N \pi_i(x_i)\varphi_i. \tag{2.3}$$

Note that the map $u \mapsto Q_N u$ is linear since the maps $u|_{\widetilde{B}_i} \mapsto \pi_i$, whose existence is ascertained in Theorem B.1, is linear. By Assumption 2.2 we have for each $i \in \{1, \dots, N\}$

$$\pi_i(x) = \sum_{j=1}^N \pi_i(x_j)\varphi_j(x) \qquad \forall\, x \in \Omega. \tag{2.4}$$

For each $i \in \{1, \dots, N\}$ we can write

$$\begin{aligned} u - Q_N u &= u - \sum_{j=1}^N \pi_j(x_j)\varphi_j \\ &= (u - \pi_i) + \sum_{j=1}^N [\pi_i(x_j) - \pi_j(x_j)]\, \varphi_j =: T_{1,i} + T_{2,i}. \end{aligned}$$

Since the patches Ω_i, $i = 1, \dots, N$, cover Ω by Assumption 2.1, we get for each $s = 0, \dots, \min\{k, r_{\text{stab}}\}$

$$\begin{aligned} \|u - Q_N u\|^2_{H^s(\Omega)} &\le \sum_{i=1}^N \|u - Q_N u\|^2_{H^s(\Omega_i \cap \Omega)} \\ &\le 2 \sum_{i=1}^N \|T_{1,i}\|^2_{H^s(\Omega_i \cap \Omega)} + \|T_{2,i}\|^2_{H^s(\Omega_i \cap \Omega)}. \end{aligned}$$

Using (2.2) we can estimate $\|T_{1,i}\|_{H^s(\Omega_i\cap\Omega)}$ by

$$\|T_{1,i}\|_{H^s(\Omega_i\cap\Omega)} \le Ch_i^{\mu-s}\|u\|_{H^k(\widetilde{B}_i)} \quad s=0,\dots,\min\{k,r_{\mathrm{stab}}\}. \tag{2.5}$$

Hence, $\sum_{i=1}^N \|T_{1,i}\|^2_{H^s(\Omega_i\cap\Omega)}$ can be estimated in the desired fashion. For the term involving the functions $T_{2,i}$, we use Assumptions 2.3 to get for any $\alpha\in\mathbb{N}_0^d$ with $|\alpha|=s\in\{0,\dots,\min\{k,r_{\mathrm{stab}}\}\}$

$$|D^\alpha T_{2,i}(x)| \le C\sum_{j=1}^N |\pi_i(x_j)-\pi_j(x_j)|h_j^{-s}\chi_j(x).$$

Thus, we get for the H^s-semi norm of $T_{2,i}$ on $\Omega_i\cap\Omega$:

$$\begin{aligned}
|T_{2,i}|^2_{H^s(\Omega)} &\le C\int_{\Omega_i\cap\Omega}\left|\sum_{j=1}^N |\pi_i(x_j)-\pi_j(x_j)|h_j^{-s}\chi_j\right|^2\\
&\le CM\int_{\Omega\cap\Omega_i}\sum_{j=1}^N |\pi_i(x_j)-\pi_j(x_j)|^2 h_j^{-2s}\chi_j\\
&\le CM\int_\Omega \sum_{j\in n(i)} |\pi_i(x_j)-\pi_j(x_j)|^2 h_j^{-2s}\chi_j\chi_i,
\end{aligned}\tag{2.6}$$

where we exploited (2.1) in the second bound and, in the last bound, we used the observation that $\chi_j(x)\chi_i(x)\neq 0$ can only happen if $j\in n(i)$. For $j\in n(i)$ we bound $|\pi_i(x_j)-\pi_j(x_j)|\le \|\pi_i-\pi_j\|_{L^\infty(B_{\delta h_j}(x_j))}$, note that $\pi_i-\pi_j\in\mathcal{P}_p$, and use the polynomial inverse estimate Theorem B.2 to get

$$\begin{aligned}
\|\pi_i-\pi_j\|_{L^\infty(B_{\delta h_j}(x_j))} &\le Ch_j^{-d/2}\|\pi_i-\pi_j\|_{L^2(B_{\delta h_j}(x_j))}\\
&\le Ch_j^{-d/2}\left[\|u-\pi_i\|_{L^2(B_{\delta h_j}(x_j))}+\|u-\pi_j\|_{L^2(B_{\delta h_j}(x_j))}\right].
\end{aligned}$$

Using $B_{\delta h_j}(x_j)\subset \widetilde{B}_j\cap\widetilde{B}_i$, we then get from (2.2) and Assumption 2.4

$$\|\pi_i-\pi_j\|_{L^\infty(B_{\delta h_j}(x_j))} \le Ch_j^{-d/2}\left[h_j^\mu\|u\|_{H^\mu(\widetilde{B}_j)}+h_i^\mu\|u\|_{H^\mu(\widetilde{B}_i)}\right].$$

Inserting this in (2.6) and using Assumption 2.4 gives

$$\begin{aligned}
&|T_2|^2_{H^s(\Omega_i\cap\Omega)}\\
&\quad\le CM\int_\Omega\sum_{j=1}^N\left[h_j^{2(\mu-s)-d}\|u\|^2_{H^k(\widetilde{B}_j)}+h_i^{2(\mu-s)-d}\|u\|^2_{H^k(\widetilde{B}_i)}\right]\chi_j\chi_i.
\end{aligned}$$

The sum $\sum_{i=1}^N |T_{1,i}|^2_{H^s(\Omega\cap\Omega_i)}$ can then be bounded by using again (2.1)

$$\sum_{i=1}^N |T_{2,i}|^2_{H^s(\Omega_i\cap\Omega)} \le CM \int_\Omega \sum_{j=1}^N \sum_{i=1}^N h_i^{2(\mu-s)-d} \|u\|^2_{H^k(\widetilde{B}_i)} \chi_i \chi_j$$

$$\le CM^2 \sum_{i=1}^N h_i^{2(\mu-s)-d} \|u\|^2_{H^k(\widetilde{B}_i)} \int_\Omega \chi_i \le CM^2 \sum_{i=1}^N h_i^{2(\mu-s)} \|u\|^2_{H^k(\widetilde{B}_i)}.$$

This concludes the proof of the theorem. □

Theorem 2.1 assumes u to be defined on $\mathbb{R}^d$. An extension result, e.g. Theorem A.1, allows us to treat the case of bounded domains:

Corollary 2.1. *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. Assume that the balls $\widetilde{B}_i$ of Theorem 2.1 satisfy additionally an overlap condition, that is for some $M \in \mathbb{N}$ we have*

$$\sup_{x\in\mathbb{R}^d} \operatorname{card}\{i \in \mathbb{N} \,|\, x \in \widetilde{B}_i\} \le M.$$

Then there exists a linear map $Q_N : L^1(\Omega) \to V_N$ such that for each $k \in \mathbb{N}_0$ there exists $C > 0$ with

$$\|u - Q_N u\|_{H^s(\Omega)} \le h^{\min\{p+1,k\}-s} \|u\|_{H^k(\Omega)}, \quad s = 0, \dots, \min\{p+1, r_{\mathrm{stab}}\},$$

where $h := \max_{i=1,\dots,N} h_i$.

Proof. Let $\widetilde{Q}_N$ be the linear operator of Theorem 2.1 and let $E : L^1(\Omega) \to L^1(\mathbb{R}^d)$ be the extension operator of Theorem A.1. Set $Q_N := \widetilde{Q}_N \circ E$. Then by abbreviating $\mu := \min\{p+1, k\}$ we get from Theorem 2.1 for $s = 0, \dots, \min\{r_{\mathrm{stab}}, k\}$

$$\|u - Q_N u\|^2_{H^s(\Omega)} = \|Eu - \widetilde{Q}_N Eu\|^2_{H^s(\Omega)} \le C \sum_{i=1}^N h_i^{2(\mu-s)} \|Eu\|^2_{H^k(\widetilde{B}_i)}$$

$$\le Ch^{2(\mu-s)} \sum_{i=1}^N \|Eu\|^2_{H^k(\widetilde{B}_i)} \le Ch^{2(\mu-s)} M^2 \|Eu\|^2_{H^k(\mathbb{R}^d)};$$

here, the last step followed from arguments analogous to those employed in the proof of Theorem 2.1. The extension operator E finally has the property $\|Eu\|_{H^k(\mathbb{R}^d)} \le C\|u\|_{H^k(\Omega)}$, which allows us to conclude the proof. □

Approximation of Singular Functions

The diameters of the balls $\widetilde{B}_i$ in Theorem 2.1 play the role of the local mesh size in the classical FEM approximation theorem. In the classical FEM, meshes that are locally refined are important, for example, for the treatment of elliptic boundary value problems in domains with piecewise smooth geometries. The solutions of such problems exhibit singularities (the functions S_{ji} of (6.2) are a typical example), which can be resolved in the classical FEM

by the use of appropriately graded meshes, [8, 93]. In fact, the optimal rate of convergence, as measured in error versus problem size, can be recovered. Meshless methods can mimic this mesh refinement of the classical FEM by an appropriate clustering of particles and a corresponding shrinking of the diameters of the balls $\widetilde{B}_i$. The following two Exercises 2.2, 2.3 illustrate this. To stress the analogy of our approach in Exercises 2.2, 2.3 with the classical FEM situation and to motivate the distribution of the diameters of the balls $\widetilde{B}_i$, we first recall the following example (see for example [94, Sec. 3.3.7]):

Example 2.1.
Let $\Omega = (0,1)$ and $u(x) = x^\alpha$, $\alpha \in (1/2, 1)$. Fix $p \in \mathbb{N}$ and $\beta > \frac{p+1/2}{\alpha - 1/2}$. Consider a mesh $\mathcal{T}$ consisting of N intervals I_i, $i = 0, \dots, N-1$, such that

$$\operatorname{diam} I_0 \le C h^\beta, \qquad \operatorname{diam} I_i \sim h \operatorname{dist}(I_i, 0)^{1-1/\beta}, \quad i = 1, \dots, N-1. \tag{2.7}$$

Then, for some $C > 0$ independent of N we have

$$\inf_{v \in S^{p,1}(\mathcal{T})} \|u - v\|_{H^1(\Omega)} \le C N^{-p},$$

that is the optimal rate of convergence is recovered. A specific mesh $\mathcal{T}$ that satisfies (2.7) is determined by the nodes x_i, $i = 0, \dots, N$, where $x_i = \Phi(\widehat{x}_i)$, $\Phi(x) = x^\beta$, and $\widehat{x}_i = ih$ for $h = 1/N$. ∎

The function Φ of Example 2.1 maps a uniform node distribution to a highly nonuniform one that is suitable for the approximation of the function $x \mapsto x^\alpha$. We use this function Φ to create particle distributions, and we use (2.7) as a guideline for our choice of the diameters of the patches Ω_i and the balls $\widetilde{B}_i$ in the following Exercise 2.2. We will show there that this choice leads to patches that satisfy Assumptions 2.1, 2.4, and we will see that polynomials of degree p have good approximation properties on the balls $\widetilde{B}_i$. The construction of concrete shape functions associated with these patches that satisfy Assumptions 2.2, 2.3 is postponed until Exercise 2.6. Corresponding results exist for two-dimensional problems and are sketched in Exercises 2.2, 2.7.

Exercise 2.2.
Let $\Omega = (0,1)$, $u(x) = x^\alpha$ for some $\alpha \in (1/2, 1)$. Fix $p \in \mathbb{N}_0$ and choose $\beta \ge \frac{p+1/2}{\alpha - 1/2} > 1$. Define

$$\Phi(x) := x^\beta.$$

For $N \in \mathbb{N}$ set $h = 1/N$, $\widehat{x}_i := ih$, $i = 0, \dots, N$, and define the particles $X_N = \{x_i \,|\, i = 0, \dots, N\}$ by $x_i = \Phi(\widehat{x}_i)$. Let $\rho > 0$ be a parameter and choose for each particle x_i

$$\rho_i = \rho \begin{cases} h x_i^{1-1/\beta} & i \ge 1, \\ x_1 & i = 0. \end{cases}$$

Let a shape function φ_i be associated with particle x_i. Assume furthermore that $\Omega_i := (\operatorname{supp} \varphi_i)^\circ = B_{\rho_i}(x_i)$.

(a) Show: For each fixed M there holds $\rho_i \sim h^\beta$ for $i \in \{0, \dots, M\}$ (The constants of the $\sim$-notation depend on ρ, β, M).
(b) Show: There exist λ, λ' (depending only on ρ, β) such that

$$B_{\lambda h}(\widehat{x}_i) \cap \Omega \subset \Phi^{-1}(B_{\rho_i}(x_i) \cap \Omega) \subset B_{\lambda' h}(\widehat{x}_i) \cap \Omega \qquad i = 0, \dots, N.$$

Conclude that Assumption 2.1 is satisfied.
(c) Show: Assumption 2.4 is satisfied.
(d) Let C_{comp} be the constant of Assumption 2.4, whose existence was ascertained in (c). Set $\widetilde{B}_i := B_{\widetilde{\rho}_i}(x_i)$ with $\widetilde{\rho}_i := (1 + (1+\delta)C_{\mathrm{comp}})\rho_i$. Show: $B_{\rho_i}(x_i) \cap B_{\rho_j}(x_j) \neq \emptyset$ implies $B_{\delta\rho_j}(x_j) \subset \widetilde{B}_i$.
(e) Show: The balls $\widetilde{B}_i$, $i = 0, \dots, N$ satisfy an overlap condition, that is there exists $M > 0$ (depending only on ρ, β) such that $\mathrm{card}\{j \,|\, \widetilde{B}_i \cap \widetilde{B}_j \neq \emptyset\} \leq M$ for all $i \in \{0, 1, \dots, N\}$.
(f) Let $I_1 := \{i \in \{0, \dots, N\} \,|\, \mathrm{dist}(\widetilde{B}_i, 0) \geq 2\widetilde{\rho}_i\}$. Show: For $i \in I_1$ the point $\widetilde{x}_i := \inf\{x \,|\, x \in \widetilde{B}_i\}$ satisfies $\widetilde{x}_i \sim x_i$. Furthermore, there exist polynomials $\pi_i \in \mathcal{P}_p$ such that

$$\|u - \pi_i\|_{L^2(\widetilde{B}_i)} + \widetilde{\rho}_i \|(u - \pi_i)'\|_{L^2(\widetilde{B}_i)} \leq C \widetilde{\rho}_i^{p+3/2} \widetilde{x}_i^{\alpha-1-p}.$$

(g) Set $I_2 := \{1, \dots, N\} \setminus I_1$. Show: $I_2 \subset \{1, \dots, M\}$ for some $M > 0$ independent of N. Show: For each $i \in I_2$ one can find a $\pi_i \in \mathcal{P}_1$ such that

$$\|u - \pi_i\|_{L^2(\Omega \cap \widetilde{B}_i)} + \widetilde{\rho}_i \|(u - \pi_i)'\|_{L^2(\Omega \cap \widetilde{B}_i)} \leq C \widetilde{\rho}_0^{\alpha+1/2},$$
$$\|u - \pi_i\|_{L^\infty(\Omega \cap \widetilde{B}_i)} \leq C \widetilde{\rho}_0^{\alpha}.$$

(h) Assume that the shape functions φ_i satisfy Assumptions 2.2, 2.3. (We will see in Exercise 2.6 that such functions can be constructed with the moving least squares procedure if ρ is chosen sufficiently large). By adapting the proof of Theorem 2.1 show that the approximation space $V_N = \mathrm{span}\{\varphi_i \,|\, i = 0, \dots, N\}$ satisfies

$$\inf_{v \in V_N} \|u - v\|_{H^1(\Omega)} \leq Ch^p = CN^{-p}. \qquad \blacksquare$$

A similar idea leads to approximation results in two spatial dimensions:

Exercise 2.3.
Define for $h = 1/n$ the uniform particle distribution $\widehat{X}_n = \{\widehat{x}_{ij} = (ih, jh) \,|\, 0 \leq i, j \leq n\}$. For some $\beta > 1$, let $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ be given by $\Phi(x) = \|x\|_2^{\beta-1} x$. Define the particle distribution $X_n := \{x_{ij} = \Phi(\widehat{x}_{ij}) \,|\, 0 \leq i, j \leq n\}$. Associate with each particle x_{ij} a radius

$$\rho_{ij} = \rho \begin{cases} h\|x_{ij}\|_2^{1-1/\beta} & \text{if } (i,j) \neq (0,0) \\ h^\beta & \text{if } i = j = 0, \end{cases}$$

where $\rho > 0$ is a parameter. The patches Ω_{ij} are taken as $\Omega_{ij} := B_{\rho_{ij}}(x_{ij})$. Set $\Omega := (0, 1/2)^2$.

(a) Proceed as in Exercise 2.2 to show that Assumptions 2.1 and 2.4 hold.
(b) Assume that the shape functions φ_{ij}, $i,j = 0,\dots,n$, that are associated with the nodes x_{ij} satisfy additionally Assumptions 2.2, 2.3. (We will show in Exercise 2.7 that this can be achieved by taking ρ sufficiently large). Consider a function u in polar coordinates (r,φ) of the form $u = r^\alpha \Theta(\varphi)$, where $\alpha > 0$ and $\Theta : (-\varepsilon, \pi/2 + \varepsilon) \to \mathbb{R}$ for some $\varepsilon > 0$ is smooth. Show: If $\beta > \frac{p}{\alpha}$, then

$$\inf_{v \in V_N} \|u - v\|_{H^1(\Omega)} \leq C h^p, \qquad h = \frac{1}{n} \sim \frac{1}{\sqrt{N}},$$

where N denotes the number of particles. Note that this is the optimal rate of convergence. ∎

2.3 Construction of Shape Functions with the Moving Least Squares Procedure

The approximation result Theorem 2.1 hinges on Assumptions 2.1–2.4. In the present section we construct shape functions that satisfy these requirements.

Motivation from Scattered Data Fitting
One approach to construct shape functions φ_i from a collection of particles X_N is based on the so-called moving least squares (MLS) technique that we describe in more detail in this section.
The MLS technique was devised to fit a "smooth" function $x \mapsto If$ to a collection of given scattered data (x_i, f_i), $i = 1,\dots,N$, obtained, for example, from measurements. Here, the points x_i, $i = 1,\dots,N$, are N distinct points and the "smooth" function If that is sought should satisfy $If(x_i) \approx f_i$, $i = 1,\dots,N$. The idea is to define the value $If(x)$ for a given x as a weighted average of the given data f_i. More specifically, one chooses a polynomial degree $p \in \mathbb{N}_0$ and for each $i \in \{1,\dots,N\}$ a weight $w_i(x) \geq 0$ and then defines

$$If(x) := \pi(x), \tag{2.8}$$

where the polynomial $\pi \in \mathcal{P}_p$ is the solution of the minimization problem:

$$\text{Find } \pi \in \mathcal{P}_p \text{ s.t.} \quad \sum_{i=1}^N |f_i - \pi(x_i)|^2 w_i(x) \leq \sum_{i=1}^N |f_i - v(x_i)|^2 w_i(x) \qquad \forall\, v \in \mathcal{P}_p. \tag{2.9}$$

Remark 2.2.
The choice of the weight functions $x \mapsto w_i(x)$ depends, of course, on the application. In practise, the weight function $x \mapsto w_i(x)$ is chosen to have small support or to decay rapidly as $\|x - x_i\| \to \infty$ so as to give the data points x_i close to x more weight than data points far from x. ∎

Under reasonable assumptions on the weight functions w_i, the minimization problem is uniquely solvable. As we will show in Theorem 2.2, this solution If takes the form

$$If(x) = \sum_{i=1}^{N} f_i \varphi_i(x) \tag{2.10}$$

for some functions φ_i. Theorem 2.2 also provides an explicit formula for the functions φ_i. Their differentiability properties are then analyzed in Theorem 2.3. The goal of this section is to show that the functions shape functions φ_i, which are motivated by the above data fitting technique, satisfy the assumptions of the approximation result Theorem 2.1. Indeed, we will discover that Assumption 2.2 is ensured by construction and that Assumption 2.3 can be satisfied if, roughly speaking, each particle has sufficiently many neighbours. Assumptions 2.1, 2.4 have to be checked separately.

Construction of the Shape Functions
The shape functions φ_i appearing in (2.10) are constructed in the following theorem.

Theorem 2.2. *Let particles $X_N = \{x_i \,|\, i = 1, \dots, N\}$ and weight functions $w_i \in C(\mathbb{R}^d)$ with $w_i \geq 0$, $i = 1, \dots, N$ be given. Set $\Omega_i := (\operatorname{supp} w_i)^\circ$. Assume that for each $x \in \Omega$ the set $X(x) := \{x_i \,|\, i \in n(x)\}$ is $\mathcal{P}_p$-unisolvent*[1]*. Then the approximant If of (2.8), (2.9) is well-defined, and there are unique functions φ_i, $i = 1, \dots, N$, depending solely on X_N and the weight functions w_i such that*

$$If(x) = \sum_{i=1}^{N} f_i \varphi_i(x).$$

Moreover, we have the representation formula

$$\varphi_i(x) = w_i(x) \sum_{k=1}^{Q} \lambda_k(x) \pi_k(x_i), \qquad i = 1, \dots, N, \tag{2.11}$$

where $\{\pi_k \,|\, k = 1, \dots, Q\}$ is an arbitrary basis of $\mathcal{P}_p$, and the values $\lambda_k(x)$ are the unique solution of the linear system

$$\sum_{k=1}^{Q} \sum_{i=1}^{N} w_i(x) \pi_k(x_i) \pi_l(x_i) \lambda_k(x) = \pi_l(x), \qquad l = 1, \dots, Q. \tag{2.12}$$

Proof. We follow the presentation of [109]. We fix $x_* \in \Omega$ and seek $\pi \in \mathcal{P}_p$ of (2.8) in the form $\pi = \sum_{l=1}^{Q} \widetilde{\lambda}_l \pi_l$. The minimization problem (2.9) then leads

[1] A set $Y \subset \mathbb{R}^d$ is $\mathcal{P}_p$-unisolvent, if $\pi \in \mathcal{P}_p$ and $\pi(y) = 0$ for all $y \in Y$ implies $\pi \equiv 0$.

to the following system of equations: Find $\widetilde{\lambda}_l$, $l = 1, \dots, Q$, such that

$$\sum_{i=1}^{N} w_i(x_*) \left(f_i - \sum_{l=1}^{Q} \widetilde{\lambda}_l \pi_l(x_i) \right) \pi_k(x_i) = 0, \qquad k = 1, \dots, Q. \tag{2.13}$$

We prove unique solvability of this linear system of equations by proving that the symmetric matrix $\mathbf{G} \in \mathbb{R}^{Q \times Q}$ with entries $\mathbf{G}_{kl} = \sum_{i=1}^{N} w_i(x_*) \pi_l(x_i) \pi_k(x_i)$ is symmetric positive definite: For $\mathbf{a} \in \mathbb{R}^Q$ we compute

$$\mathbf{a}^\top \mathbf{G} \mathbf{a} = \sum_{i=1}^{N} w_i(x_*) \left| \sum_{k=1}^{Q} \mathbf{a}_k \pi_k(x_i) \right|^2 ;$$

in view of the assumption $w_i \geq 0$, we conclude that $\mathbf{G}$ is positive semi-definite. If $\mathbf{G}$ were not positive definite, then there existed a vector $\mathbf{a} \in \mathbb{R}^Q$ with $\mathbf{a} \neq 0$ such that $\mathbf{a}^\top \mathbf{G} \mathbf{a} = 0$. Hence, for the non-trivial polynomial $\widetilde{\pi} = \sum_{k=1}^{Q} \mathbf{a}_k \pi_k$, we would have $\widetilde{\pi}(x_i) = 0$ for all $x_i \in X(x_*)$, since $x_i \in X(x_*)$ implies $x_i \in (\operatorname{supp} w_i)^\circ$, that is by $w_i \in C(\mathbb{R}^d)$ we have $w_i(x_*) > 0$. But then $\widetilde{\pi} = 0$ by our assumption of unisolvence. We have thus arrived at a contradiction and conclude that $\mathbf{G}$ is positive definite.
We now evaluate $If(x_*) = \pi(x_*)$ (writing $w_i = w_i(x_*)$, $\lambda_k = \lambda_k(x_*)$)

$$\pi(x_*) = \sum_{l=1}^{Q} \widetilde{\lambda}_l \pi_l(x_*) \overset{(2.12)}{=} \sum_{i,k,l} \widetilde{\lambda}_l \lambda_k w_i \pi_k(x_i) \pi_l(x_i) \overset{(2.13)}{=} \sum_{i,k} f_i \lambda_k \pi_k(x_i),$$

which leads to the desired representation formula (2.11). □

Exercise 2.4.
Show: For $p = 0$ the functions φ_i are given by

$$\varphi_i(x) = \frac{w_i(x)}{\sum_{j=1}^{N} w_j(x)} = \frac{w_i(x)}{\sum_{j \in n(i)} w_j(x)}. \tag{2.14}$$

These functions are called Shephard functions, [97]. ∎

An important observation is that the functions φ_i constructed by the MLS procedure reproduce polynomials, that is they satisfy Assumption 2.2:

Exercise 2.5.
Show that the functions φ_i satisfy Assumption 2.2, that is

$$\sum_{i=1}^{N} \pi(x_i) \varphi_i(x) = \pi(x) \quad \forall\, x \in \Omega \qquad \forall\, \pi \in \mathcal{P}_p. \qquad \blacksquare \tag{2.15}$$

Remark 2.3.
The representation formula (2.11) shows that the functions φ_i can be evaluated at a point $x \in \Omega$ by solving a $Q \times Q$ system of linear equations. Likewise,

by differentiating the linear system (2.12), it is clear that also the values of derivatives of the functions $x \mapsto \lambda_k(x)$ can be obtained as solutions of linear systems; therefore, derivatives of the functions φ_i can be determined. The question of bounds of the derivatives of the functions φ_i will be discussed in more detail in Theorem 2.3. ∎

The weight functions w_i have to be chosen by the user. A popular form is

$$w_i(x) = w\left(\frac{x - x_i}{\rho_i}\right), \tag{2.16}$$

where the *window function* w is of one of the following types:

1. w is *radial*, that is $w(z) = \widetilde{w}(\|z\|)$ for some $\widetilde{w} : \mathbb{R}_0^+ \to \mathbb{R}_0^+$;
2. w has *tensor product* form, that is $w(z) = \prod_{j=1}^{d} \widetilde{w}_j(z_j)$.

We note that if the window function w is compactly supported, then the parameter ρ_i in (2.16) is a measure for the support size and $\rho_i \sim h_i = \operatorname{diam} \Omega_i$. In this situation, the univariate functions $\widetilde{w}$ or $\widetilde{w}_j$ are often taken to be compactly supported splines, e.g. the symmetric part of the classical piecewise cubic C^2 B-spline given by

$$w(r) = \begin{cases} 4 - 6r^2 + 3r^3 & \text{for } 0 \le r \le 1, \\ (2 - r)^3 & \text{for } 1 < r \le 2, \\ 0 & \text{for } r > 2. \end{cases}$$

Remark 2.4.
If the window function is a radial function and has compact support, then the norm $\|\cdot\|$ on $\mathbb{R}^d$ can be still be chosen. For example, the patches Ω_i can be balls (or, more generally, ellipsoids) if $\|\cdot\|$ is taken as the Euclidean norm; the patches Ω_i can be cubes if $\|\cdot\|_{l^\infty}$ is chosen. ∎

Regularity of the Shape Functions
Our analysis of the differentiability properties of the functions φ_i in Theorem 2.3 below will be based on the assumption that the weight functions w_i are determined by a window function w via (2.16). This window function w will be required to satisfy

Assumption 2.5. The window function $w \in C^k(\mathbb{R}^d)$ satisfies $w(x) \ge 0$ for all $x \in \mathbb{R}^d$, and $(\operatorname{supp} w)^\circ = B_1(0)$.

Remark 2.5.
We take $B_1(0)$ as the unit ball with respect to the Euclidean norm. This is not essential, however, and results analogous to Theorem 2.3 below hold if we replace the Euclidean norm with another norm on $\mathbb{R}^d$. ∎

The formula (2.14) for the special case $p = 0$ suggests that $\varphi_i \in C^k$ if the weights w_i are determined by a window function w satisfying Assumptions 2.5. Roughly speaking, if for every $x \in \Omega$ the number of particles in the vicinity of x, that is $\operatorname{card} n(x)$, is sufficiently large, then the shape functions φ_i are indeed as smooth as the window function. In order to prove this result in Theorem 2.3 below, we introduce the fill distance function h by

$$h(x) := \operatorname{dist}(x, X_N) \tag{2.17}$$

and can now formulate:

Theorem 2.3. *Let Ω satisfy a cone condition with angle θ and radius r. Let $\alpha \in (0,1)$, $X_N = \{x_i \,|\, i = 1, \dots, N\} \subset \mathbb{R}^d$ and $\{\rho_i \,|, i = 1, \dots, N\} \subset \mathbb{R}^+$. Set*

$$\hat{\rho}_i := \min\{\rho_i, r\}, \qquad i = 1, \dots, N,$$

and assume the covering condition

$$\Omega \subset \cup_{i=1}^N B_{\alpha\hat{\rho}_i}(x_i). \tag{2.18}$$

Let w satisfy Assumption 2.5, define the weight functions $w_i(x) := w(\frac{x-x_i}{\rho_i})$ with corresponding patches $\Omega_i = (\operatorname{supp} w_i)^\circ = B_{\rho_i}(x_i)$. Suppose that Assumption 2.4 is valid. Let $p \in \mathbb{N}_0$.
Then there exist $\delta > 0$ and $C > 0$ (depending only on θ, r, α, p, k, C_{comp}) such that if

$$\sup_{x \in B_{\hat{\rho}_i}(x_i) \cap \Omega} h(x) \le \delta \hat{\rho}_i \qquad \forall\, x_i \in X_N, \tag{2.19}$$

then the functions φ_i of (2.11) satisfy $\varphi_i \in C^k(\mathbb{R}^k)$, $\operatorname{supp} \varphi_i \subset \overline{B_{\rho_i}(x_i)}$, and

$$\|D^\alpha \varphi_i\|_{L^\infty(\Omega)} \le C \rho_i^{-|\alpha|} \qquad \forall\, \alpha \in \mathbb{N}_0^d, \quad |\alpha| \le k. \tag{2.20}$$

Before proving Theorem 2.3 it is instructive to check that the assumptions of Theorem 2.3 can be satisfied in simple circumstances.

Example 2.2.
The assumption (2.19) is often formulated in a simpler, global way. If we define the *fill distance* $\overline{h} := \sup_{x \in \Omega} h(x)$ and use constant $\rho_i = \rho$ for all $i \in \{1, \dots, N\}$, then (2.19) merely requires that $\overline{h}$ be sufficiently small compared to ρ, the size of the supports of the patches Ω_i. ∎

We have seen Exercises 2.2, 2.3 two examples of highly nonuniform particle distributions and greatly varying patches sizes that are suitable for the approximation of singularity functions. The following two exercises show that the assumptions of Theorem 2.3 can be fulfilled in such circumstances as well.

Exercise 2.6.
In Exercise 2.2 we constructed particles and patch sizes that were appropriate for the approximation of the singular function $x \mapsto x^\alpha$. We assumed, however,

that the shape functions φ_i satisfied Assumptions 2.2 and 2.3. Show that by choosing ρ in Exercise 2.2 sufficiently large, the hypotheses of Theorem 2.3 are satisfied. Conclude that the shape functions obtained by the MLS technique yield the optimal approximation result of Exercise 2.2.
Hint: Show that the fill distance function h satisfies

$$h(x) \le C \left[h x^{1-1/\beta} + h^\beta \right]$$

for a constant $C > 0$ independent of ρ and N. ∎

Exercise 2.7.
Assume the hypotheses of Exercise 2.3. Show: If ρ is chosen sufficiently large, then the hypotheses of Theorem 2.3 are satisfied.
Hint: Show that the fill distance function h satisfies $h(x) \le C\big[h\|x\|_2^{1-1/\beta} + h^\beta\big]$ for a constant $C > 0$ independent of ρ and N. ∎

Proof of Theorem 2.3. The proof is broken up into several steps.
First step: We notice that the representation formula (2.11) is independent of the choice of the basis of $\mathcal{P}_p$. In particular, we may chose for each $x_* \in \Omega$ a different basis. We will exploit this observation as follows: First, we fix a basis $\{\widetilde{\pi}_k \,|\, k = 1, \dots, Q\}$ of $\mathcal{P}_p$; then, for each fixed $x_* \in \Omega$, we define the basis $\{\pi_k \,|\, k = 1, \dots, Q\}$ by

$$\pi_k(x) := \widetilde{\pi}_k \left(\frac{x - x_*}{\rho_*} \right),$$

where, for some arbitrary (but fixed) $i_* \in n(x_*)$ we set

$$\rho_* := \rho_{i_*}.$$

(Note that the covering condition (2.18) guarantees that $n(x_*) \neq \emptyset$). Since $2\rho_i = h_i = \operatorname{diam} \Omega_i = \operatorname{diam} B_{\rho_i}(x_i)$, Assumption 2.4 guarantees that

$$\rho_* C_{\text{comp}}^{-1} \le \rho_j \le \rho_* C_{\text{comp}} \qquad \forall\, j \in n(x_*). \tag{2.21}$$

We next define the matrix $\mathbf{G}(x_*) \in \mathbb{R}^{Q \times Q}$ with entries

$$\begin{aligned} \mathbf{G}_{kl}(x_*) &:= \sum_{i=1}^{N} w_i(x_*) \pi_k(x_i) \pi_l(x_i) \\ &= \sum_{i \in n(x_*)} w \left(\frac{x_* - x_i}{\rho_i} \right) \widetilde{\pi}_k \left(\frac{x_* - x_i}{\rho_*} \right) \widetilde{\pi}_l \left(\frac{x_* - x_i}{\rho_*} \right). \end{aligned}$$

By Theorem 2.2 the function value $\varphi_i(x_*)$ is given by

$$\varphi_i(x_*) = w_i(x_*) \sum_{k=1}^{Q} \lambda_k(x_*) \pi_k(x_i), \tag{2.22}$$

where the vector $\lambda(x_*) = (\lambda_1(x_*), \dots, \lambda_Q(x_*))^\top \in \mathbb{R}^Q$ is the solution of the linear system

$$\mathbf{G}(x_*)\lambda(x_*) = \left(\widetilde{\pi}_1(0) \cdots \widetilde{\pi}_Q(0)\right)^\top. \tag{2.23}$$

In order to get bounds on the derivatives of φ_i, we need to get bounds on the derivatives of the function λ. In this direction, we first notice that the product rule together with (2.21) gives

$$|D^\alpha \mathbf{G}(x_*)| \le C_\alpha {\rho_*}^{-|\alpha|} \qquad \forall\, \alpha \in \mathbb{N}_0^d, \quad |\alpha| \le k, \tag{2.24}$$

where the constant C_α depends only on α, the function w, and the choice of basis $\{\widetilde{\pi}_l \,|\, l = 1, \dots, Q\}$. The analogous bound

$$|D^\alpha \mathbf{G}^{-1}(x_*)| \le C_\alpha {\rho_*}^{-|\alpha|} \qquad \forall\, \alpha \in \mathbb{N}_0^d, \quad |\alpha| \le k, \tag{2.25}$$

holds by Cramer's rule, provided that we can show the existence of $\underline{C} > 0$ such that

$$\inf_{x_* \in \Omega} |\det \mathbf{G}(x_*)| \ge \underline{C} > 0. \tag{2.26}$$

From (2.25) follows a bound similar to (2.25) for the derivatives of the solution λ_l, $l = 1, \dots, Q$ of (2.23); the product rule applied to (2.22) together with (2.21) then gives the desired bound (2.20) for the shape functions φ_i. We are thus left with establishing (2.26).

Second step: To see (2.26) we prove a lower bound on the smallest eigenvalue of the symmetric matrix $\mathbf{G}(x_*)$. To that end, let $\mathbf{a} \in \mathbb{R}^Q$ be arbitrary but fixed. We define the polynomial

$$\pi := \sum_{k=1}^{Q} \mathbf{a}_k \pi_k$$

and observe

$$\mathbf{a}^\top \mathbf{G}(x_*)\mathbf{a} = \sum_{i,k,l} w_i(x_*)\mathbf{a}_k\mathbf{a}_l\pi_k(x_i)\pi_l(x_i) = \sum_{i=1}^{N} w_i(x_*)|\pi(x_i)|^2. \tag{2.27}$$

We wish to exploit that Assumption 2.5 gives us the existence of $C_{\min} > 0$ such that

$$\min\{w(x) \,|\, x \in B_\alpha(0)\} = C_{\min} > 0. \tag{2.28}$$

To do so, we define $\eta < 1/2$ by

$$\eta := \frac{1}{2}\frac{\alpha}{C_{\mathrm{comp}}} \le \frac{1}{2}\alpha < \frac{1}{2}, \tag{2.29}$$

where we used $C_{\mathrm{comp}} \ge 1$. Next, we choose δ appearing in (2.19) according to the definition (2.33) below; in particular, therefore, $\delta < \eta$ so that there exists an index $i \in \mathbb{N}$ such that $x_* \in B_{\eta\hat{\rho}_i}(x_i)$. We fix this index and define

$$\widetilde{n}(x_*) := \{j \in \mathbb{N} \,|\, x_j \in X_N \cap B_{\eta\hat{\rho}_i}(x_*)\}. \tag{2.30}$$

Our goal in this second step is to show

$$\mathbf{a}^\top \mathbf{G}(x_*)\mathbf{a} \geq \sum_{j\in\widetilde{n}(x_*)} w_j(x_*)|\pi(x_j)|^2 \geq C_{\min} \sum_{j\in\widetilde{n}(x_*)} |\pi(x_j)|^2. \tag{2.31}$$

The first bound in (2.31) is obvious since $w_j \geq 0$ for all j. To see the second estimate, in view of (2.28), it suffices to see $\|x_j - x_*\|_2 < \alpha\hat{\rho}_j$ for $j \in \widetilde{n}(x_*)$. Let therefore $j \in \widetilde{n}(x_*)$. Then

$$\|x_j - x_i\|_2 \leq \|x_j - x_*\|_2 + \|x_i - x_*\|_2 < 2\eta\hat{\rho}_i \leq \hat{\rho}_i,$$

where in the last step, we used $\eta \leq 1/2$. Hence, $x_j \in B_{\rho_i}(x_i)$, and thus $j \in n(i)$. We conclude with Assumption 2.4

$$\hat{\rho}_i \leq C_{\mathrm{comp}}\hat{\rho}_j \qquad \forall\ j \in \widetilde{n}(x_*).$$

Together with the definition of η in (2.29), we arrive at the desired bound $\|x_j - x_*\|_2 < \eta\hat{\rho}_i \leq \eta C_{\mathrm{comp}}\hat{\rho}_j \leq \frac{1}{2}\alpha\hat{\rho}_j \leq \alpha\hat{\rho}_j$.

Third step: To get further, we apply Lemma 2.1. Our choice of δ above is precisely the choice of Lemma 2.1 so that we can find $C > 0$ depending only on Ω, η, and p such that

$$\|\pi\|_{L^\infty(B_{\hat{\rho}_i}(x_*))} \leq C \max\{|\pi(x_j)| \,|\, j \in \widetilde{n}(x_*)\}.$$

Thus, we get from (2.31)

$$\mathbf{a}^\top \mathbf{G}(x_*)\mathbf{a} \geq C\|\pi\|^2_{L^\infty(B_{\hat{\rho}_i}(x_*))}.$$

In view of (2.21), we get from Bernstein's estimate Lemma B.1 the existence of $C > 0$ (depending only on p, C_{comp} and the parameter r of the cone condition) such that $\|\pi\|_{L^\infty(B_{\rho_*}(x_*))} \leq C\|\pi\|_{L^\infty(B_{\hat{\rho}_i}(x_*))}$. Thus, we get

$$\mathbf{a}^\top \mathbf{G}(x_*)\mathbf{a} \geq C\|\pi\|^2_{L^\infty(B_{\rho_*}(x_*))}. \tag{2.32}$$

To control the smallest eigenvalue of $\mathbf{G}(x_*)$, we are therefore left with estimating $\sum_{k=1}^Q |\mathbf{a}_k|^2$ by $\|\pi\|^2_{L^\infty(B_{\rho_*}(x_*))}$. We achieve this by a scaling argument: We define the function $\overline{\pi}(x) := \pi((x - x_*)/\rho_*)$ on $B_1(0)$ and note

$$\overline{\pi}(x) = \sum_{k=1}^Q \mathbf{a}_k \widetilde{\pi}_k(x).$$

We observe $\|\overline{\pi}\|_{L^\infty(B_1(0))} = \|\pi\|_{L^\infty(B_{\rho_*}(x_*))}$. By the equivalence of norms on finite dimensional space, we then get the existence of $C > 0$ (depending solely on p and the choice of the basis $\{\widetilde{\pi}_k \,|\, k = 1, \ldots, Q\}$) such that

$$C^{-1} \sum_{k=1}^Q |\mathbf{a}_k|^2 \leq \|\overline{\pi}\|^2_{L^\infty(B_1(0))} \leq C \sum_{k=1}^Q |\mathbf{a}_k|^2.$$

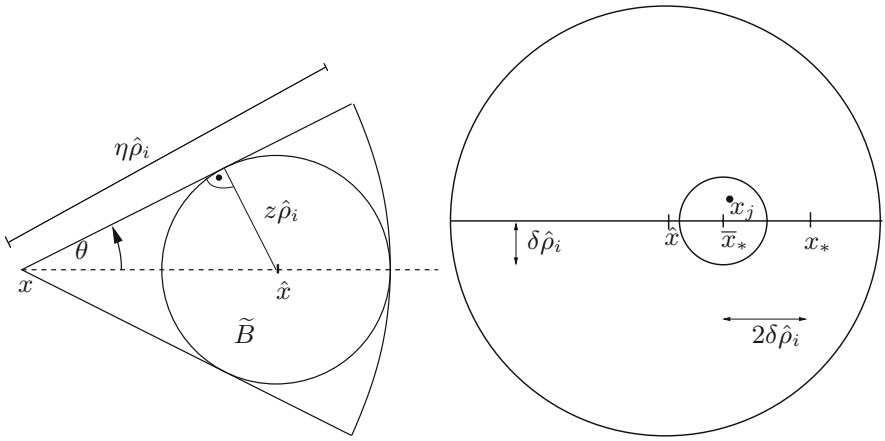


Figure 2.2. Notation for Lemma 2.1. Left: Ball $\widetilde{B}$. Right: Location of $\hat{x}$, x_*, x_j.

This establishes the desired lower bound on the eigenvalues of $\mathbf{G}(x_*)$. Finally, we note that this bound holds in fact uniformly in $x_* \in \Omega$, thus completing the proof of (2.26). □

The following lemma allows us to bound the L^∞-norm of a polynomial in terms of values in discrete points:

Lemma 2.1. *Let $X_N = \{x_i \,|\, i = 1, \ldots, N\} \subset \mathbb{R}^d$ and $\{\rho_i \,|\, i = 1, \ldots, N\} \subset \mathbb{R}^+$. Let $\Omega \subset \mathbb{R}^d$ satisfy an interior cone condition with angle θ and radius $r > 0$. Define*

$$\hat{\rho}_i := \min\{\rho_i, r\}, \qquad i = 1, \ldots, N.$$

Let $\eta \in (0, 1]$ and $p \in \mathbb{N}_0$. Set

$$\delta := \eta \frac{\sin\theta}{1 + \sin\theta} \min\left\{\frac{1}{3}, \frac{1}{36p^2}\right\}. \tag{2.33}$$

Then the following holds: If $\Omega \subset \cup_{i=1}^N B_{\hat{\rho}_i}(x_i)$ and if for all $i \in \{1, \ldots, N\}$

$$\sup_{y \in B_{\hat{\rho}_i}(x_i) \cap \Omega} h(y) \le \delta\hat{\rho}_i, \tag{2.34}$$

then for each $x \in \Omega$ and any $x_i \in X_N \cap B_{\hat{\rho}_i}(x)$ and all $\pi \in \mathcal{P}_p$

$$\begin{aligned}\|\pi\|_{L^\infty(B_{\hat{\rho}_i}(x))} &\le \|\pi\|_{L^\infty(B_{2\hat{\rho}_i}(x_i))} \\ &\le 2\left(\frac{4(1+\sin\theta)}{\eta\sin\theta}\right)^p \max\{|\pi(x_j)| \,|\, x_j \in X_N \cap B_{\eta\hat{\rho}_i}(x)\}.\end{aligned}$$

Proof. The proof follows the arguments of [109] and proceeds in several steps. We fix $x \in \Omega \cap B_{\hat{\rho}_i}(x_i)$ and $\pi \in \mathcal{P}_p$. We also define

$$z := \eta \frac{\sin\theta}{1 + \sin\theta}$$

and note that δ, z are chosen such that

$$3\delta \le z.$$

First step: By the cone condition, there exists a cone $C_1 = C(x, \xi, \theta, \eta\hat{\rho}_i) \subset \Omega$. Elementary geometric considerations (see Figure 2.2) then show the existence of a ball $\widetilde{B} = B_{z\hat{\rho}_i}(\hat{x})$, where $\hat{x} = x + \frac{\eta}{1+\sin\theta}\xi$ with the following properties:

$$\widetilde{B} \subset C_1 \subset \Omega \cap B_{\eta\hat{\rho}_i}(x) \cap B_{2\hat{\rho}_i}(x_i). \tag{2.35}$$

Second step: From Lemma B.1, we get

$$\|\pi\|_{L^\infty(B_{2\hat{\rho}_i}(x_i))} \le \left(\frac{4}{z}\right)^p \|\pi\|_{L^\infty(\widetilde{B})}. \tag{2.36}$$

It therefore suffices to bound $\|\pi\|_{L^\infty(\widetilde{B})}$ in terms of the values of π in the discrete set $X_N \cap B_{\eta\hat{\rho}_i}(x)$. Towards this goal, we construct in this second step an $x_j \in X_N \cap B_{\eta\hat{\rho}_i}(x)$ that will be seen in the fourth step to have the property that $|\pi(x_j)|$ is comparable to $\|\pi\|_{L^\infty(\widetilde{B})}$. Choose $x_* \in \overline{\widetilde{B}}$ such that

$$\|\pi\|_{L^\infty(\widetilde{B})} = |\pi(x_*)|.$$

We claim the existence of $x_j \in X_N \cap \widetilde{B} \cap \overline{B_{3\delta\hat{\rho}_i}(x_*)}$. To see this, we recall that $\hat{x}$ is the centre of $\widetilde{B}$ and define the auxiliary point

$$\overline{x}_* := \begin{cases} x_* + 2\delta\hat{\rho}_i \frac{1}{\|\hat{x}-x_*\|_2}(\hat{x} - x_*) & \text{if } x_* \neq \hat{x}, \\ x_* & \text{if } x_* = \hat{x}. \end{cases}$$

Since $3\delta \le z$, elementary considerations show $\|\overline{x}_* - \hat{x}\|_2 < (z - \delta)\hat{\rho}_i$; hence $\overline{B_{\delta\hat{\rho}_i}(\overline{x}_*)} \subset \widetilde{B}$. The assumption (2.34) then implies the existence of an $x_j \in X_N \cap \overline{B_{\delta\hat{\rho}_i}(\overline{x}_*)} \subset X_N \cap \widetilde{B}$. By the triangle inequality we furthermore get $x_j \in \overline{B_{3\delta\hat{\rho}_i}(x_*)}$.

Third step: Let x_j be the point constructed in the second step and set

$$\zeta := \frac{1}{\|x_j - x_*\|_2}(x_j - x_*) \qquad \text{if } x_j \neq x_*.$$

If $x_j = x_*$, then choose an arbitrary $\zeta \in \mathbb{R}^d$ with $\|\zeta\|_2 = 1$. We claim:

$$\{x_* + t\zeta \,|\, t \in [0, \tfrac{1}{3}z\hat{\rho}_i]\} \subset \overline{\widetilde{B}}.$$

To see this, we first note that the case $x_* = \hat{x}$ is trivial. We therefore assume that $x_* \neq \hat{x}$. From the second step we recall

$$\|\overline{x}_* - x_j\|_2 \le \delta\hat{\rho}_i, \qquad \|\overline{x}_* - x_*\|_2 = 2\delta\hat{\rho}_i, \tag{2.37}$$

so that we can conclude

$$\|x_j - x_*\|_2 \geq \delta\hat{\rho}_i. \tag{2.38}$$

In order to see that $x_* + t\zeta \in \widetilde{B}$ for $t \in [0, \frac{1}{3}z\hat{\rho}_i]$ we write

$$x_j = \overline{x}_* + (x_j - \overline{x}_*) = x_* + \frac{2\delta\hat{\rho}_i}{\|\hat{x} - x_*\|_2}(\hat{x} - x_*) + (x_j - \overline{x}_*).$$

and compute

$$\|x_* + t\zeta - \hat{x}\|_2 \leq \left| \|x_* - \hat{x}\|_2 - \frac{2\delta\hat{\rho}_i}{\|x_j - x_*\|_2} t \right| + \frac{\|x_j - \overline{x}_*\|_2}{\|x_j - x_*\|_2} t.$$

Requiring

$$\left| \|x_* - \hat{x}\|_2 - \frac{2\delta\hat{\rho}_i}{\|x_j - x_*\|_2} t \right| + \frac{\|x_j - \overline{x}_*\|_2}{\|x_j - x_*\|_2} t \leq z\hat{\rho}_i$$

is equivalent to the following two inequalities:

$$\|x_* - \hat{x}\|_2 - z\hat{\rho}_i \leq \frac{2\delta\hat{\rho}_i - \|x_j - \overline{x}_*\|_2}{\|x_j - x_*\|_2} t \qquad \text{and}$$

$$t \leq (\|x_* - \hat{x}\|_2 + z\hat{\rho}_i) \frac{\|x_j - x_*\|_2}{2\delta\hat{\rho}_i + \|x_j - \overline{x}_*\|_2},$$

which are indeed both satisfied for $t \in [0, \frac{1}{3}z\hat{\rho}_i]$ in view of $\|x_* - \hat{x}\|_2 \leq z\hat{\rho}_i$ and (2.37), (2.38).

Fourth step: We now turn to estimating $|\pi(x_*)|$ in terms of $|\pi(x_j)|$. To that end, we define with the vector ζ of the fourth step the polynomial

$$p(t) := \pi(x_* + t\zeta), \qquad t \in [0, \tfrac{1}{3}z\hat{\rho}_i],$$

and note that $x_j = x_* + \tau\zeta$ for some τ with $0 \leq \tau \leq 3\delta\hat{\rho}_i$ since $x_j \in \overline{B_{3\delta\hat{\rho}_i}(x_*)}$. Additionally, we have (for $p \geq 1$) in view of the definition of δ that $\tau \leq \frac{1}{3}z$. Using Markov's inequality (see for example [33, Chap. 4, Thm. 1.4]), we can bound

$$|\pi(x_*) - \pi(x_j)| = |p(\|x_* - x_j\|_2) - p(0)| = \left| \int_0^\tau p'(t)\, dt \right|$$

$$\leq \tau \|p'\|_{L^\infty(0, \frac{1}{3}z\hat{\rho}_i)} \leq \frac{2\tau p^2}{\frac{1}{3}z\hat{\rho}_i} \|p\|_{L^\infty(0, \frac{1}{3}z\hat{\rho}_i)} \leq \frac{18\delta}{z} p^2 \|\pi\|_{L^\infty(\widetilde{B})}.$$

Recalling now that $|\pi(x_*)| = \|\pi\|_{L^\infty(\widetilde{B})}$, we get

$$\|\pi\|_{L^\infty(\widetilde{B})} \leq \frac{1}{1 - 18p^2\delta/z} |\pi(x_j)|.$$

This estimate is also trivially true for $p = 0$. We therefore conclude, since $x_j \in X_N \cap \widetilde{B} \subset X_N \cap B_{\eta\hat{\rho}_i}(x)$

$$\|\pi\|_{L^\infty(B_{2\hat{\rho}_i}(x_i))} \leq \left(\frac{4}{z}\right)^p \frac{1}{1 - 18p^2\delta/z} \max\{x_j \,|\, x_j \in X_N \cap B_{\eta\hat{\rho}_i}(x)\}.$$

Using $\delta \leq \frac{1}{36p^2} z$ and the definition of z, we arrive at the desired bound. □

Exercise 2.8.
Assumption 2.5 requires the function w to be k-times continuously differentiable. Consider what assumptions (e.g. on the definition of $n(x)$) need to be changed if w is in $C^{k-1,1}$. ∎

2.4 Bibliographical Remarks

The construction of the Q_N in the proof of Theorem 2.1 that is based on point evaluations of locally approximating polynomials is just one possible technique; variations of such constructions can be found in [1, 6]. The proof of the stability result Theorem 2.3 follows in essence [109]. Variants can be found, for example, in [1, 41, 55].
The moving least squares technique originates from scattered data approximation. Early references include [45, 97]. It is, however, just one way of generating shape functions that reproduce polynomials. Alternatives include the reproducing kernel particle methods (RKPM), [70, 72–75].
One reason for introducing meshless methods is to alleviate the costly meshing. Completely regular meshes on the other hand are very simple to generate and have many advantages. With this in mind, the web-splines (weighted extended B-splines) were introduced in [57]. The computational domain is covered with a regular mesh on which standard splines can be defined easily. Appropriate adjustments near the boundary are made to be able to handle essential boundary conditions.

3 Approximation Properties of Radial Basis Functions

A second class of shape functions that can be motivated from scattered data interpolation are radial basis functions (RBFs). In scattered data *interpolation* the basic problem is as follows: given a norm $\|\cdot\|$ on $\mathbb{R}^d$, a function $\Phi : \mathbb{R}_0^+ \to \mathbb{R}$, distinct points $X_N = \{x_i \,|\, i = 1, \ldots, N\} \subset \mathbb{R}^d$ and function values f_i, $i = 1, \ldots, N$, the goal is to find If of the form

$$If(\cdot) = \sum_{j=1}^{N} u_j \Phi(\|\cdot - x_j\|) \quad \text{such that} \quad If(x_i) = f_i \qquad i = 1, \ldots, N. \tag{3.1}$$

The problem (3.1) represents a linear system of equations. Clearly, existence and uniqueness of If depends on the function Φ. An important class for which this can be established is that of *positive definite* functions Φ:

Definition 3.1. A continuous function $\Phi : \mathbb{R}_0^+ \to \mathbb{R}$ is *positive definite*, if for any set $X = \{x_1, \ldots, x_M\}$ of M distinct points the Gram matrix $\mathbf{G} \in \mathbb{R}^{M \times M}$ with entries $\mathbf{G}_{ij} = \Phi(\|x_i - x_j\|)$ is symmetric positive definite.

Proposition 3.1. *If Φ is positive definite, then the interpolation problem (3.1) is uniquely solvable.*

Proof. Exercise. □

Example 3.1.
Classically, the norm $\|\cdot\|$ on $\mathbb{R}^d$ is taken to be the Euclidean norm $\|\cdot\|_2$. Popular examples of radial basis functions Φ are the Gaussians ($\Phi(r) = e^{-r^2}$), Hardy's multiquadrics $\Phi(r) = \sqrt{1+r^2}$, and the inverse multiquadrics $\Phi(r) = (1+r^2)^{-1/2}$. It is also a widely used practise to employ scaled versions, that is, to use the function $\widetilde{\Phi}(r) = \Phi(r/h)$ with a suitable scaling parameter $h > 0$. These RBFs can be used for scattered data interpolation in any dimension. Another class is obtained by taking the fundamental solution of the iterated Laplacian Δ^m. For $2m \geq d$, these RBFs are given by $\Phi(r) = r^{2m-d} \ln r$ if d is even and $\Phi(r) = r^{2m-d}$ if d is odd. The function Φ in the special case $m = d = 2$ is called the thin-plate spline since in the Kirchhoff plate model, which is a biharmonic equation, the deflection of an infinite plate under a point load coincides with Φ (up to scaling). ∎

The functions of Example 3.1 do not have bounded support. As was shown in [106, 107] it is possible to construct RBFs that have compact support:

Example 3.2.
A class of RBFs $\Phi_{d',k}$, $k \in \mathbb{N}_0$ for applications in spatial dimension $d \leq d'$ are the compactly supported RBFs of H. Wendland, [106, 107]. A few examples of this class are:

function	smoothness	for problems in $\mathbb{R}^d$
$\Phi_{1,0}(r) = (1-r)_+$	C^0	$d = 1$
$\Phi_{1,1}(r) = (1-r)_+^3(3r+1)$	C^2	$d = 1$
$\Phi_{1,2}(r) = (1-r)_+^5(8r^2+5r+1)$	C^4	$d = 1$
$\Phi_{3,0}(r) = (1-r)_+^2$	C^0	$d \leq 3$
$\Phi_{3,1}(r) = (1-r)_+^4(4r+1)$	C^2	$d \leq 3$
$\Phi_{3,2}(r) = (1-r)_+^6(35r^2+18r+3)$	C^4	$d \leq 3$

With the exception of $\Phi_{1,0}$, $\Phi_{3,0}$, the functions $\Phi_{k,d'}$ satisfy Assumption 3.1 below (see [107] and Exercise 3.1) and hence are positive definite. As in Example 3.1 scaled version $\Phi_{k,d}(r/\rho)$ for a scaling parameter $\rho > 0$ are frequently employed as well. ∎

3.1 Analysis of a Class of RBFs

We consider the following class of RBF functions $x \mapsto \Phi(\|x\|_2)$:

Assumption 3.1. The Fourier transform[2] ψ of the function $x \mapsto \Phi(\|x\|_2)$ satisfies for some $\tau > d/2$ and $C > 0$

$$C^{-1}(1+\|\xi\|_2^2)^{-\tau} \leq \psi(\xi) \leq C(1+\|\xi\|_2^2)^{-\tau} \qquad \forall\, \xi \in \mathbb{R}^d.$$

[2] $\hat{f}(\xi) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} f(x) e^{-\mathrm{i}x\cdot\xi}\, dx$ denotes the Fourier transform $\hat{f}$ of a function f. The inversion formula takes the form $f(x) = \int_{\mathbb{R}^d} \hat{f}(\xi) e^{\mathrm{i}x\cdot\xi}\, d\xi$.

The set of RBFs that satisfy Assumption 3.1 is not empty:

Exercise 3.1.
Check that the compactly supported RBF $\Phi_{1,1}$ of Example 3.2 for $d = 1$ satisfies Assumption 3.1 with $\tau = 2$. ∎

The strict positivity of ψ stipulated in Assumption 3.1 allows us to define an inner product $\langle \cdot, \rangle_\Phi$ and the corresponding Hilbert space H_Φ, which is called the "native space":

$$\langle f, g\rangle_\Phi := \int_{\mathbb{R}^d} \frac{1}{\psi} \hat{f}(\xi)\overline{\hat{g}(\xi)}\, d\xi, \qquad H_\Phi := \{f \,|\, \|f\|_\Phi^2 := \langle f, f\rangle_\Phi < \infty\}. \quad (3.2)$$

We have

Proposition 3.2. *Let Φ satisfy Assumption 3.1. Then*

1. $H_\Phi \subset C(\mathbb{R}^d)$.
2. $H_\Phi = H^\tau(\mathbb{R}^d)$ *with equivalent norms.*
3. $\Phi \in H_\Phi$.
4. Φ *is positive definite.*

Proof. The second assertion is just one of several equivalent definitions of the Sobolev spaces $H^\tau(\mathbb{R}^d)$. The other assertions are left as an exercise. □

Theorem 3.1. *Let Assumption 3.1 be valid. Then for distinct points $X_N = \{x_i \,|\, i = 1, \ldots, N\}$ and $f \in H_\Phi$ the scattered interpolation problem:*

$$\begin{aligned} &\textit{Find } If \in V_N := \operatorname{span}\{\Phi(\|\cdot - x_i\|_2) \,|\, i = 1, \ldots, N\} \\ &\textit{such that} \quad If(x_i) = f(x_i) \quad i = 1, \ldots, N, \end{aligned}$$

has a unique solution, which satisfies

$$\langle f - If, v\rangle_\Phi = 0 \quad \forall\ v \in V_N \quad (3.3)$$

and

$$\|f - If\|_\Phi = \min_{v \in V_N} \|f - v\|_\Phi. \quad (3.4)$$

Proof. Existence and unique follows from the fact that $x \mapsto \Phi(\|x\|_2)$ is positive definite. The orthogonality relation can be seen as follows: The function $v_k = \Phi(\|\cdot - x_k\|_2)$ satisfies $v_k \in V_N$ and $\widehat{v_k}(\xi) = \psi(\xi)e^{\mathbf{i}x_k\xi}$. Next,

$$\langle f - If, v_k\rangle_\Phi = \int_{\mathbb{R}^d} \frac{1}{\psi} \left(\hat{f} - \widehat{If}\right) \psi e^{\mathbf{i}x_k\xi}\, d\xi = f(x_k) - If(x_k) = 0,$$

where the last step follows from the interpolation property. Hence, (3.3) is true. This orthogonality relation implies the best approximation result (3.4) in the $\|\cdot\|_\Phi$-norm in the standard way (see for example the proof of Céa's Lemma in [23, Thm. 2.8.1]). □

Corollary 3.1 (stability of scattered data interpolation). *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain (or $\Omega = \mathbb{R}^d$). Let $X_N = \{x_i \,|\, i = 1, \dots, N\} \subset \Omega$ and suppose Assumption 3.1. Then for all $f \in H^\tau(\Omega)$*

$$\|f - If\|_{H^\tau(\Omega)} \le C \|f\|_{H^\tau(\Omega)}.$$

Proof. We will only treat the case of Ω being a Lipschitz domain. Let $E : H^\tau(\Omega) \to H^\tau(\mathbb{R}^d)$ be the universal extension operator of Theorem A.1. Since $X_N \subset \Omega$, we have $Ef(x_i) = f(x_i)$, $i = 1, \dots, N$. By Proposition 3.2, the interpolant If exists and is unique. Since $H^\tau(\mathbb{R}^d) = H_\Phi$, we have $Ef \in H_\Phi$. By Proposition 3.2 and Theorem 3.1 we arrive at

$$\begin{aligned}
\|Ef - If\|^2_{H^\tau(\mathbb{R}^d)} &\le C \langle Ef - If, Ef - If \rangle_\Phi = C \langle Ef - If, Ef \rangle_\Phi \\
&\le C \|Ef - If\|_\Phi \|Ef\|_\Phi \le C \|Ef - If\|_{H^\tau(\mathbb{R}^d)} \|Ef\|_{H^\tau(\mathbb{R}^d)} \\
&\le C \|Ef - If\|_{H^\tau(\mathbb{R}^d)} \|f\|_{H^\tau(\Omega)}.
\end{aligned}$$

We conclude $\|Ef - If\|_{H^\tau(\mathbb{R}^d)} \le C \|f\|_{H^\tau(\Omega)}$. Since $Ef = f$ on Ω and trivially $\|Ef - If\|_{H^\tau(\Omega)} \le C \|Ef - If\|_{H^\tau(\mathbb{R}^d)}$, the proof is complete. □

This stability result is the key to approximation results for the scattered data interpolant If:

Corollary 3.2. *Let Assumption 3.1 be satisfied and let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. Define the* fill distance

$$h := \sup_{x \in \Omega} \min_{i=1,\dots,N} \|x - x_i\|_2. \tag{3.5}$$

Then there exists $C > 0$ such that for $f \in H^\tau(\Omega)$ there holds

$$\|f - If\|_{H^s(\Omega)} \le C h^{\tau - s} \|f\|_{H^\tau(\Omega)}, \qquad 0 \le s \le \tau.$$

Proof. We proceed in two steps.
First step: By Theorem 3.1, the linear operator $\mathrm{Id} - I : H^\tau(\Omega) \to V_N \subset H^\tau(\Omega)$ satisfies $\|\mathrm{Id} - I\|_{H^\tau(\Omega) \to H^\tau(\Omega)} \le C$. If we can show the claim for $s = 0$,

$$\|\mathrm{Id} - I\|_{H^\tau(\Omega) \to L^2(\Omega)} \le C h^\tau,$$

then the desired bound $\|\mathrm{Id} - I\|_{H^\tau(\Omega) \to H^s(\Omega)} \le C h^{\tau - s}$ for any $s \in [0, \tau]$ follows by interpolation. We are thus left with showing the special case $s = 0$.
Second step: Choose $p \in \mathbb{N}_0$ such that $\tau \le p$. By Lemma 2.1 there exist C, $\hat{C} > 0$ depending only on Ω such that for $\rho = Ch$ we have for all balls $B_\rho(x)$, $x \in \Omega$:

$$\|\pi\|_{L^\infty(B_\rho(x))} \le \hat{C} \max_{x_i \in B_\rho(x)} |\pi(x_i)| \qquad \forall\, \pi \in \mathcal{P}_p. \tag{3.6}$$

We cover $\Omega \subset \bigcup_{x \in \Omega} \overline{B_\rho(x)}$. By the Besicovitch covering theorem, Theorem A.3, we can extract from the cover $\mathcal{B} = \{\overline{B_\rho(x)} \,|\, x \in \Omega\}$ a subcover

$\mathcal{B}_j$, $i = j, \dots, M$, with the following properties: $\Omega \subset \cup_{j=1}^{M} \cup_{\overline{B} \in \mathcal{B}_j} \overline{B}$ and each collection $\mathcal{B}_j$ consists of countably many disjoint balls.
We set $z := f - If$ and assume for notational convenience, as we may using the extension operator of Theorem A.1, that z is defined on $\mathbb{R}^d$ with $\|z\|_{H^\tau(\mathbb{R}^d)} \leq C\|z\|_{H^\tau(\Omega)}$. For each ball $\overline{B}$ of $\cup_{j=1}^{M} \mathcal{B}_j$ we select $Q \in \mathcal{P}_p$ as given by the polynomial approximation result Theorem B.1. We can then bound with the triangle inequality and the polynomial inverse estimate of Theorem B.2

$$\|z\|_{L^2(B)} \leq \|z - Q\|_{L^2(B)} + \|Q\|_{L^2(B)} \leq C \left\{ \rho^\tau \|z\|_{H^\tau(B)} + \rho^{d/2} \|Q\|_{L^\infty(B)} \right\}.$$

Our choice of the balls B in $\mathcal{B}$ guarantees (3.6). Hence, we can estimate

$$\|Q\|_{L^\infty(B)} \leq \hat{C} \max\{|Q(x_i)| \,|\, x_i \in \overline{B}\} = C \sup\{|Q(x_i)| \,|\, x_i \in B\}.$$

Since z vanishes in the interpolation points x_i, we get

$$\begin{aligned} \|Q\|_{L^\infty(B)} &\leq \hat{C} \sup\{|Q(x_i) - z(z_i)| \,|\, x_i \in B\} \\ &\leq \hat{C} \|z - Q\|_{L^\infty(B)} \leq C\rho^{\tau - d/2} \|z\|_{H^\tau(B)}, \end{aligned}$$

where we used again the approximation properties in L^∞ ascertained in Theorem B.1. Using the fact that $\Omega \subset \cup_{j=1}^{M} \cup_{\overline{B} \in \mathcal{B}_j} \overline{B}$ and that for each $j \in \{1, \dots, M\}$ the balls of the collection $\mathcal{B}_j$ are pairwise disjoint, we get

$$\|z\|_{L^2(\Omega)}^2 \leq \sum_{j=1}^{M} \sum_{\overline{B} \in \mathcal{B}_j} \|z\|_{L^2(B)}^2 \leq C\rho^{2\tau} \sum_{j=1}^{M} \sum_{\overline{B} \in \mathcal{B}_j} \|z\|_{H^\tau(B)}^2 \leq C\rho^{2\tau} \sum_{j=1}^{M} \|z\|_{H^\tau(\Omega)}^2.$$

This concludes the proof in view of the stability result Corollary 3.1. □

It is of interest to consider functions $f \in H^k(\Omega)$ with $k < \tau$. Since in this case the function f may not be continuous, we cannot define the scattered data interpolant; nevertheless, the space $V_N = \operatorname{span}\{\Phi(\| \cdot - x_i\|_2) \,|\, i = 1, \dots, N\}$ can still have good approximation properties. Indeed, we have the following:

Proposition 3.3. *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. Assume that Φ satisfies Assumption 3.1. Let X_N be a particle distribution with fill distance h given by (3.5). Set $V_N := \operatorname{span}\{\Phi(\| \cdot - x_i\|_2) \,|\, x_i \in X_N\}$. Then for $0 \leq k \leq \tau$ and real numbers $0 \leq s_1 \leq \cdots \leq s_m = k$, we have for some $C > 0$ independent of h and f:*

$$\inf_{v \in V_N} \sum_{j=1}^{m} h^{s_j} \|f - v\|_{H^{s_j}(\Omega)} \leq Ch^k \|f\|_{H^k(\Omega)}.$$

Proof. We will prove the following, weaker statement:

$$\inf_{v \in V_N} \|f - v\|_{H^s(\Omega)} \leq Ch^{k-s} \|f\|_{H^k(\Omega)}, \qquad 0 \leq s \leq k. \tag{3.7}$$

The statement of the proposition then follows from (3.7) and a result on simultaneous approximation in Sobolev space, [22]. To see (3.7), fix s and let $\Pi : H^s(\Omega) \to V_N$ be the $H^s(\Omega)$-orthogonal projection. Then by Corollary 3.2

$$\|\operatorname{Id} - \Pi\|_{H^s(\Omega)\to H^s(\Omega)} = 1, \qquad \|\operatorname{Id} - \Pi\|_{H^\tau(\Omega)\to H^s(\Omega)} \le Ch^{\tau-s}.$$

Since the space $H^k(\Omega)$ can be obtained by interpolation between $H^s(\Omega)$ and $H^\tau(\Omega)$ we arrive at the desired bound.

3.2 Bibliographical Remarks

The presentation here follows [86]. The presentation is restricted to positive definite RBFs for simplicity. A very important, more general class of functions is that of conditionally positive RBFs: For given $p \in \mathbb{N}_0$, norm $\|\cdot\|$ on $\mathbb{R}^d$, a function $\Phi(\|\cdot\|)$ is called conditionally positive definite if for any set $X_M = \{x_1, \dots, x_M\}$ of distinct points, the matrix $\mathbf{G} \in \mathbb{R}^{M\times M}$ defined by $\mathbf{G}_{ij} = \Phi(\|x_i - x_j\|)$ is positive definite on the set subspace $\{\mathbf{a} \in \mathbb{R}^M \mid \sum_{k=1}^M \mathbf{a}_k \pi(x_k) = 0 \quad \forall\, \pi \in \mathcal{P}_p\}$. The interpolation problem (3.1) is then replaced with the problem of finding If of the form $\sum_{j=1}^N u_j \Phi(\|\cdot - x_j\|) + \pi$ for a $\pi \in \mathcal{P}_p$ such that $If(x_i) = f_i$ for $i = 1, \dots, N$. For a detailed survey of RBF functions we refer to [25, 26, 61, 110].

The approximation theory for RBFs can be traced back to the work of Duchon, [35, 36], where in particular the RBFs Φ that are fundamental solutions of the iterated Laplacian are analyzed.

The approximation result Proposition 3.3 is just one example of a setting where the function f to be approximated is not in the native space H_Φ. We refer to [24] and the reference there for a more detailed discussion.

It should be noted that even for the compactly supported radial basis functions of Example 3.2 the Gram matrix $\mathbf{G}$ of the interpolation problem or the stiffness matrix, if they are used as shape functions in Galerkin methods, is not sparse. Multiresolution analysis ideas have been proposed and employed in the context of radial basis functions. For example, if for each level $l \in \{0, \dots, L\}$ a collection of points $x_{i,l}$, $i = 1, \dots, N_l$, is given or constructed, one can approximate from the space $\operatorname{span}\{\Phi(\|(\cdot - x_{i,l})/h_l\|_2) \,|\, i = 1, \dots, N_l, l = 0, \dots, L\}$, where the scaling parameters h_l are additional, suitably chosen parameters. We refer [61] and the references there for more details.

4 Partition of Unity Method and Generalized FEM

The approximation properties of the spaces discussed in Sections 2, 3 ultimately rely on the local approximation properties of polynomials. The Partition of Unity Method/generalized FEM [7, 9, 78, 79, 82, 101–103] is a generalization of the classical FEM and the above approaches in that it allows the creation of special approximation spaces that are tailored to a particular problem. As we will see in Theorem 4.1, one can construct, starting from local approximation spaces V_i, a global approximation space V by means of a partition of unity, where the global space V inherits the approximation properties from the local spaces V_i. As we will illustrate in Section 5, the approximation properties of the local spaces V_i need not rely on those of polynomials.

4.1 Approximation Theory

Theorem 4.1. *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain and let $\{\psi_i \,|\, i = 1, \dots, N\}$ be a collection of $W^{1,\infty}(\Omega)$ functions. Set $\Omega_i := (\operatorname{supp} \psi_i)^\circ \subset \Omega$, $h_i := \operatorname{diam} \Omega_i$, and assume*

$$\|\psi_i\|_{L^\infty(\Omega)} \le C_\infty, \qquad \|\nabla \psi_i\|_{L^\infty(\Omega)} \le \frac{C_G}{h_i} \qquad i = 1, \dots, N,$$

$$\sum_{i=1}^{N} \psi_i \equiv 1 \quad \text{on } \Omega, \qquad \sup_{x \in \Omega} \operatorname{card}\{i \in \mathbb{N} \,|\, x \in \Omega_i\} \le M.$$

Assume that each Ω_i, $i = 1, \dots, N$, is a Lipschitz domain as well. For each $i \in \{1, \dots, N\}$ let $V_i \subset H^1(\Omega_i)$ be given and set

$$V := \sum_{i=1}^{N} \psi_i V_i = \left\{ \sum_{i=1}^{N} \psi_i v_i \,|\, v_i \in V_i \right\}. \tag{4.1}$$

Then $V \subset H^1(\Omega)$.
Assume that for a given $u \in H^1(\Omega)$ the spaces V_i have a local approximation property*, that is there exist $v_i \in V_i$ such that*

$$\|u - v_i\|_{L^2(\Omega_i)} =: \varepsilon_1(i), \qquad \|\nabla(u - v_i)\|_{L^2(\Omega_i)} =: \varepsilon_2(i). \tag{4.2}$$

Then the approximant $v := \sum_{i=1}^{N} \psi_i v_i \in V$ satisfies

$$\|u - v\|^2_{L^2(\Omega)} \le M C_\infty^2 \sum_{i=1}^{N} |\varepsilon_1(i)|^2, \tag{4.3}$$

$$\|\nabla(u - v)\|^2_{L^2(\Omega)} \le 2M \sum_{i=1}^{N} \left[\left(\frac{C_G}{h_i} \right)^2 |\varepsilon_1(i)|^2 + C_\infty^2 |\varepsilon_2(i)|^2 \right]. \tag{4.4}$$

Proof. The assumption that the patches Ω_i be Lipschitz domain is required to ensure that $V \subset H^1(\Omega)$ as we now show: By the extension result Theorem A.1, there exist extension operators $E_i : H^1(\Omega_i) \to H^1(\mathbb{R}^d)$. For each $i \in \{1, \dots, N\}$ we choose $v_i \in V_i$. We then check that $\psi_i(E_i v_i) \in H^1(\Omega)$ as the product of a Lipschitz continuous function and an $H^1(\Omega)$-function. Hence, $\sum_{i=1}^N \psi_i E_i v_i \in H^1(\Omega)$. By the support properties of the functions ψ_i we get $\sum_{i=1}^N \psi_i v_i = \sum_{i=1}^N \psi_i E_i v_i$. In this way, we see that $V = \sum_{i=1}^N \psi_i V_i \subset H^1(\Omega)$. We will now prove (4.4) and leave (4.3) as an exercise. Using $\sum_{i=1}^N \psi_i \equiv 1$ on Ω we can write with the product rule

$$\nabla(u - \sum_{i=1}^N \psi_i v_i) = \nabla \sum_{i=1}^N \psi_i (u - v_i) = \sum_{i=1}^N (u - v_i)\nabla\psi_i + \psi_i \nabla(u - v_i).$$

This allows us to bound the error $e := u - \sum_{i=1}^N \psi_i v_i$ by

$$\int_\Omega |e|^2 \, dx \le 2 \int_\Omega \left| \sum_{i=1}^N (u - v_i)\nabla\psi_i \right|^2 + \left| \sum_{i=1}^N \psi_i \nabla(u - v_i) \right|^2 dx. \tag{4.5}$$

The assumption $\sup_{x\in\Omega} \operatorname{card}\{i \,|\, x \in \Omega_i\} \le M$ implies that for each fixed $x \in \Omega$ each of the sums consists of at most M terms. Hence, exploiting the bound $(\sum_{j=1}^M |a_j|)^2 \le M \sum_{j=1}^M |a_j|^2$, which is valid for any finite sequence $(a_j)_{j=1}^M$, and using the bounds on the functions ψ_i, $\nabla\psi_i$, we arrive at

$$\begin{aligned}
\left| \sum_{i=1}^N (u - v_i)(x)\nabla\psi_i(x) \right|^2 &\le M \sum_{i=1}^N |\nabla\psi_i(x)|^2 \, |(u - v_i)(x)|^2 \\
&\le M C_G^2 \sum_{i=1}^N \frac{1}{h_i^2} |(u - v_i)(x)|^2, \\
\left| \sum_{i=1}^N \psi_i(x)\nabla(u - v_i)(x) \right|^2 &\le M \sum_{i=1}^N |\psi_i(x)|^2 \, |\nabla(u - v_i)(x)|^2 \\
&\le M C_\infty^2 \sum_{i=1}^N |\nabla(u - v_i)(x)|^2.
\end{aligned}$$

Inserting these bounds in (4.5) then gives the desired estimate. □

Remark 4.1.
Theorem 4.1 is formulated for L^2-based spaces—an extension to spaces $W^{k,q}$, $1 \le q < \infty$ is possible. If the partition of unity is smoother, that is $\psi_i \in W^{k,\infty}(\Omega)$ and the local spaces V_i satisfy $V_i \subset H^k(\Omega_i)$, then again $V \subset H^k(\Omega)$ and analogous approximation results in H^k can be obtained. Thus, applications requiring subspaces of $H^k(\Omega)$ instead of $H^1(\Omega)$ as approximation spaces can easily be constructed. ∎

A prominent example of a partition of unity satisfying the assumptions of Theorem 4.1 consists of the standard basis of a FEM space:

Example 4.1.
Let $\mathcal{T}$ be a shape-regular mesh on a domain $\Omega \subset \mathbb{R}^d$. Let $\{x_i \,|\, i = 1, \ldots, N\}$ be the vertices of $\mathcal{T}$ and let $\{\psi_i \,|\, i = 1, \ldots, N\}$ be the standard piecewise linear basis of $S^{1,1}(\mathcal{T})$. Then $\{\psi_i \,|\, i = 1, \ldots, N\}$ is a partition of unity satisfying the assumptions of Theorem 4.1. ∎

Remark 4.2.
Partitions of unity are systems of functions that reproduce polynomials of degree $p = 0$. Hence, one can obtain a partition of unity with the Shephard construction of Exercise 2.4 from a collection of particles $X_N = \{x_i \,|\, i = 1, \ldots, N\}$ and corresponding weight functions w_i, $i = 1, \ldots, N$. As discussed in Section 2.3, the regularity of the shape functions obtained in this way is determined by the regularity of the weight functions w_i.
Of particular note in the Shephard construction is the case when each patch Ω_i contains an open subset Ω_i' such that $\Omega_i' \cap \Omega_j = \emptyset$ for $j \neq i$. Then $\psi_i \equiv 1$ on Ω_i'. Such a partition of unity is employed in the particle partition of unity method of [96]. ∎

For practical implementations, it is important to identify a basis of the space V. It appears natural to base it on bases $\mathcal{B}_i = \{b_{i,j} \,|\, j = 1, \ldots, \dim V_i\}$, $i = 1, \ldots, N$, and consider the set $\mathcal{B} = \{\psi_i b_{i,j} \,|\, i = 1, \ldots, N, j = 1, \ldots, \dim V_i\}$. In general $\mathcal{B}$ is *not* a basis of V as the following exercise shows:

Exercise 4.1.
Let $\Omega = (0, 1)$ and $0 = x_0 < x_1 < \cdots x_N = 1$ be a partition of Ω. let ψ_i, $i = 0, \ldots, N$, be the standard piecewise linear hat function associated with node x_i. Let $V_i = \mathcal{P}_p = \operatorname{span}\{b_j \,|\, j = 0, \ldots, p\}$ for each $i = 0, \ldots, p$. Show by a dimension argument that $\{\psi_i b_j \,|\, i = 0, \ldots, N, j = 0, \ldots, p\}$ is not a basis of $V = \sum_{i=0}^{N} \psi_i V_i$. ∎

If the partition of unity is suitably chosen, then the set $\mathcal{B}$ is a basis of V:

Exercise 4.2.
Let the partition of unity $\{\psi_i \,|\, i = 1, \ldots, N\}$ be such that for each i there exists an open set Ω_i' with $\Omega_i' \cap \operatorname{supp} \psi_j = \emptyset$ for all $j \neq i$. Show: The set $\mathcal{B}$ is a basis of V. This fact is exploited in the particle partition of unity of [96]. ∎

4.2 Example: Polynomial Local Approximation Spaces

There are several ways to employ the approximation result Theorem 4.1 in a numerical scheme. One way is to use polynomials as local approximation spaces V_i; the partition of unity method could, for example, be obtained from a collection of particles and the partition of unity is based on the Shepard function of Exercise 2.4. This is approach is pursued in a series of papers

by Griebel and Schweitzer [47–51] and collected in the monograph [96]. The approximation properties of this method are comparable to the classical FEM as is shown in the following Exercises 4.3, 4.4.

Exercise 4.3.
Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. For each patch Ω_i choose a polynomial degree $p_i \in \mathbb{N}_0$ and set $V_i := \mathcal{P}_{p_i}$. For each $i \in \{1, \dots, N\}$ let $\widetilde{B}_i$ be a ball of diameter $\operatorname{diam} \widetilde{B}_i \le C h_i$ such that $\Omega_i \subset \widetilde{B}_i$. Assume additionally that the balls $\widetilde{B}_i$ satisfy an overlap condition, that is

$$\sup_{x \in \mathbb{R}^d} \{ i \,|\, x \in \widetilde{B}_i \} \le M. \tag{4.6}$$

Show: Under the hypotheses of Theorem 4.1 on the functions ψ_i there holds

$$\inf_{v \in V_N} \|u - v\|^2_{H^1(\Omega)} \le C \sum_{i=1}^N h_i^{2(\min\{p_i+1,k\}-1)} \|u\|^2_{H^k(\widetilde{B}_i)}.$$

In particular, if $p_i = p$ for all i and if we set $h := \max h_i$, then

$$\inf_{v \in V_N} \|u - v\|^2_{H^1(\Omega)} \le C h^{2\min\{p,k-1\}} \|u\|^2_{H^k(\Omega)}.$$

∎

The size $\operatorname{diam} \widetilde{B}_i$ of the ball $\widetilde{B}_i$ in Exercise 4.3 plays the role of the local mesh size in the classical FEM. Graded meshes can also be simulated as illustrated in the following exercise.

Exercise 4.4.
Continue Exercise 4.3 for the approximation of singularity functions of the form $u(r, \varphi) = r^\alpha \Theta(\varphi)$ as discussed in Exercise 2.3. Let $\Omega = (0, 1/2)^2$, let X_N be the particle distribution given in Exercise 2.3 with $\beta > p/\alpha$. Let the patches Ω_i be such that $x_i \in \Omega_i \subset \widetilde{B}_i$, where $\widetilde{B}_i = B_{\rho_i}(x_i)$ with ρ_i given in Exercise 2.3. Let $V_i = \mathcal{P}_p$ as in the preceding exercise. Show: (4.6) holds, and the approximation space V satisfies

$$\inf_{v \in V} \|u - v\|_{H^1(\Omega)} \le C N^{-p},$$

that is the optimal rate of convergence is achieved. ∎

5 Examples of Operator Adapted Approximation Spaces

Theorem 4.1 allows us to construct approximation spaces V where the global space V inherits the approximation properties of the local spaces V_i. These spaces can be custom tailored to the approximation of a function u. We illustrate this with a few examples.

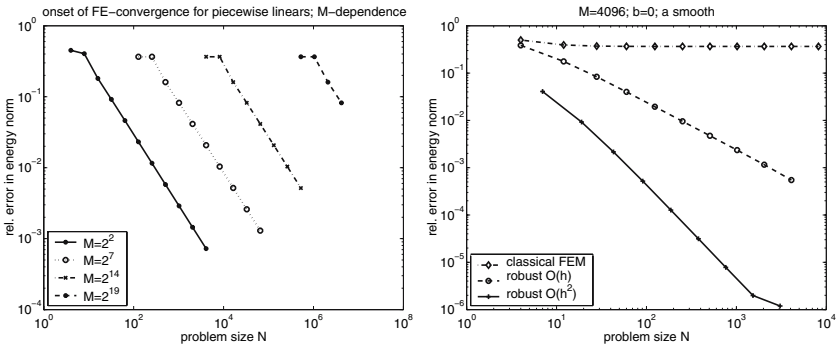


Figure 5.1. Left: Convergence of the classical FEM. Right: Convergence of the PUM.

5.1 A One-Dimensional Example

We consider the following one-dimensional model problem:

$$Lu := -(a(Mx)u')' + b(x)u = f \quad \text{on } \Omega = (0,1), \quad u(0) = u(1) = 0, \tag{5.1}$$

where $M \in \mathbb{N}$ and $a \in L^\infty(\mathbb{R})$ is 1-periodic. Additionally, we assume ellipticity, that is $0 < \underline{a} \le a(x)$ for all $x \in \mathbb{R}$ and $0 \le b(x) \le \|b\|_{L^\infty(\Omega)}$ for all $x \in \Omega$. If M is large, then the coefficient $a(M\cdot)$ is highly oscillatory and so is the solution u. The standard FEM performs poorly in the situation, namely, convergence is only observed under the assumption of scale resolution, that is if the mesh size h is sufficiently small to resolve all scales. The following example illustrates this.

Example 5.1.
We consider the case $a = \frac{1}{2+\cos(2\pi x)}$, $b \equiv 0$, and $f \equiv 1$. In the left graph in Figure 5.1 we show the convergence behaviour of the classical FEM based on the space $S_0^{1,1}(\mathcal{T})$ on uniform meshes. The error measure is relative error in the energy norm, that is

$$\frac{\|u - u_N\|_E}{\|u\|_E} = \sqrt{\frac{\int_\Omega a(Mx)|(u-u_N)'|^2\,dx}{\int_\Omega a(Mx)|u'|^2\,dx}}. \tag{5.2}$$

The solution u can be computed analytically and it can be checked that $\|u'\|_{L^2(\Omega)} \sim M$ and $\|u''\|_{L^2(\Omega)} = O(M^2)$. The classical FEM convergence analysis then gives

$$\frac{\|(u-u_N)'\|_{L^2(\Omega)}}{\|u'\|_{L^2(\Omega)}} \le C \min\left\{1, \frac{h\|u''\|_{L^2(\Omega)}}{\|u'\|_{L^2(\Omega)}}\right\} \le C\min\{1, hM\}. \tag{5.3}$$

We clearly observe in Figure 5.1 the expected asymptotic first order convergence; nevertheless, the asymptotic convergence behaviour is not observed until $h \approx 1/M$, that is, until scale resolution is reached. Note that this is in agreement with (5.3). ∎

It is possible to design local approximation spaces that have good approximation properties for the solution of (5.1).

Lemma 5.1. *Let $I = (x_0, x_0 + h)$ and $\gamma < 1$. Let $h^2 \frac{\|b\|_{L^\infty(I)}}{\underline{a}} \le \gamma < 1$. Let $\mathcal{B} = \{u_0, u_1\}$ be a* fundamental system *for L, that is $Lu_0 = Lu_1 = 0$ on I and u_0, u_1 are linearly independent. Then there exists a $C > 0$ depending only on $\underline{a}$, $\|a\|_{L^\infty(I)}$, $\|b\|_{L^\infty(I)}$, γ, such that for a solution $u \in H^1(I)$ of $Lu = f$ there holds*

$$\inf_{v \in V} \|u - v\|_{L^\infty(I)} + h\|(u - v)'\|_{L^\infty(I)} \le Ch^2 \|f\|_{L^\infty(I)},$$

where $V := \operatorname{span} \mathcal{B}$.

Proof. Since $f \in L^\infty(I)$ and $u \in H^1(I)$ we get that u and au' are continuous. We then choose $v \in V$ such that $v(x_0) = u(x_0)$ and $(av')(x_0) = (au')(x_0)$. The error $e := u - v$ then satisfies $e(x_0) = 0$ and $(ae')(x_0)$ together with $Le = f$. The differential equation $Le = f$ gives us $-(ae')' = f - be$ so that

$$|e(x)| \le \left| \int_{x_0}^{x} e'(t)\, dt \right| \le h\|e'\|_{L^\infty(I)},$$

$$|e'(x)| \le \frac{1}{\underline{a}} \, |(ae')(x)| \le \frac{1}{\underline{a}} \left| \int_{x_0}^{x} f - be \, dt \right| \le \frac{1}{\underline{a}} h \|f\|_{L^\infty(I)} + \frac{\|b\|_{L^\infty(I)}}{\underline{a}} h \|e\|_{L^\infty(I)}.$$

Combining these two estimates, we arrive at

$$\|e'\|_{L^\infty(I)} \le \frac{\|f\|_{L^\infty(I)}}{\underline{a}} h + h^2 \frac{\|b\|_{L^\infty(I)}}{\underline{a}} \|e'\|_{L^\infty(I)} \le \frac{\|f\|_{L^\infty(I)}}{\underline{a}} h + \gamma \|e'\|_{L^\infty(I)},$$

which allows us to conclude $\|e'\|_{L^\infty(I)} \le h\|f\|_{L^\infty(I)}/(\underline{a}(1 - \gamma))$. □

Remark 5.1.
It should be noted that the approximation spaces constructed in Lemma 5.1 merely require a and b to be L^∞—no further regularity is required. ▪

Extensions of the approximation result Lemma 5.1 are obtained in the following exercise.

Exercise 5.1.

(a) Construct a one-dimensional space $V_0 = \operatorname{span}\{u_0\}$ such that $u_0(x_0) = 0$ and V_0 satisfies, for $u(x_0) = 0$ and $Lu = f$,

$$\inf_{v \in V_0} \|u - v\|_{L^\infty(I)} + h\|(u - v)'\|_{L^\infty(I)} \le Ch^2 \|f\|_{L^\infty(I)}.$$

(b) Let u_2 be such that $Lu_2 = 1$. Let u_0, u_1 be defined in Lemma 5.1. Set $V_2 := \operatorname{span}\{u_0, u_1, u_2\}$. Show:

$$\inf_{v \in V_2} \|u - v\|_{L^\infty(I)} + h\|(u - v)'\|_{L^\infty(I)} \le Ch^3 \|f'\|_{L^\infty(I)}.$$

(c) Construct a two-dimensional space $V_{0,2} = \text{span}\{u_0, u_1\}$ such that $v(x_0) = 0$ for $v \in V_{0,2}$ and $V_{0,2}$ satisfies, for $u(x_0) = 0$ and $Lu = f$,

$$\inf_{v \in V_{0,2}} \|u - v\|_{L^\infty(I)} + h\|(u - v)'\|_{L^\infty(I)} \leq Ch^3 \|f'\|_{L^\infty(I)}.$$

■

Example 5.2.
We use the partition of unity method (PUM) with a partition of unity given by the piecewise linear functions on a uniform mesh with mesh size h for the approximation of the solution of (5.1) where $a = 1/(2 + \cos(2\pi x))$, $b \equiv 0$, and $f(x) = x$. We choose $M = 4096$. In the first experiment the local approximation spaces are taken as the spaces V constructed in Lemma 5.1 for the internal nodes and the space V_0 constructed in Exercise 5.1 for the two nodes at the boundary of Ω. In view of Lemma 5.1 and Theorem 4.1 we expect convergence $O(h)$ in the energy norm (cf. (5.2)), where the constant in the $O(h)$ convergence is *independent* M. The convergence behaviour of this projection method is depicted in the graph labelled "robust $O(h)$" in the right picture of Figure 5.1. Since the problem size $N \sim 1/h$, the expected convergence $O(h)$ is indeed confirmed numerically.
In the second experiment, the local spaces for the internal nodes are taken as the spaces V_2 of Exercise 5.1 and the spaces $V_{0,2}$ of Exercise 5.1 for the boundary nodes. In view of Exercise 5.1 and Theorem 4.1 we expect a convergence $O(h^2)$ in the energy norm. This expectation is confirmed by the graph labelled "robust $O(h^2)$" in the right picture of Figure 5.1. Again, the constant hidden in the $O(h^2)$ convergence result is independent of M. For more details on this one-dimensional problem, we refer to [82]. ■

Exercise 5.2.
The approximation properties of the space V constructed in Lemma 5.1 can also be understood by transforming the problem. Consider the case $b \equiv 0$. Then

$$V = \text{span}\left\{1, \int_{x_0}^{x} \frac{1}{a(t)}\, dt\right\}.$$

Let $f \in L^2(I)$ and define the change of variable $\widetilde{x} := \int_{x_0}^{x} \frac{1}{a(t)}\, dt$. Show: The function $\widetilde{u}(\widetilde{x}) := u(x)$ is in H^2 (*hint:* write down a differential equation satisfied by $\widetilde{u}$). Hence it can be approximated well from $\mathcal{P}_1$. Infer from that approximation results for u for the approximation from V. ■

Remark 5.2.
The construction in Lemma 5.1 exploits in a crucial way the fact that a one-dimensional problem is considered: the solution space of homogeneous linear second order differential equations is two-dimensional. Nevertheless analogous approximation results can be shown for quasi one-dimensional cases. Exercise 5.2 illustrates an old, but powerful tool of numerical mathematics,

namely, the use of suitable transformations. This device is also the reasons for the results of [7]. In [7] problems of the form

$$-\partial_x \left(a(x)\partial_x u\right) - \partial_y \left(a(x)\partial_y u\right)) = f \qquad \text{on } (0,1)^2$$

are considered; the coefficient $a \geq \underline{a} > 0$ depends on the single variable x but may be merely bounded and measurable. For such problems, it is shown that local approximation space of the form

$$V := \operatorname{span}\left\{1, \int_{x_0}^{x} \frac{1}{a(t)}\, dt, y\right\}$$

can lead to the optimal rate $O(h)$. ∎

5.2 Laplace's Equation

We consider the two-dimensional case $\Omega \subset \mathbb{R}^2$ and solutions to Laplace's equation

$$-\Delta u = 0 \qquad \text{on } \Omega. \tag{5.4}$$

It seems reasonable to try to approximate the solutions to a differential equation with systems of functions that likewise solve the differential equation. For the Laplace equation one such system is that of harmonic polynomials:

$$\mathcal{HP}_p := \operatorname{span}\{\operatorname{Re} z^n, \operatorname{Im} z^n \mid n = 0, \ldots, p\}, \tag{5.5}$$

where $z = x + \mathbf{i}y \in \mathbb{C}$. Note that $\dim \mathcal{HP}_p = 2p + 1$. We have exponential convergence if the function u to be approximated is harmonic on set that strictly contains the domain of interest:

Theorem 5.1. *Let $\Omega \subset \mathbb{R}^2$ be a simply connected domain and let $\Omega' \subset\subset \Omega$ be a compact subset. Let $k \in \mathbb{N}_0$. Let u satisfy $-\Delta u = 0$ on Ω. Then there exist C, $b > 0$ such that for all $p \in \mathbb{N}_0$*

$$\inf_{v \in \mathcal{HP}_p} \|u - v\|_{W^{k,\infty}(\Omega')} \leq Ce^{-bp}.$$

Proof. This result is due to Szegö. We refer to [80] for a proof. □

Example 5.3.
We consider the approximation of the solution u of (5.4), where $\Omega = (0,1)^2$. The exact solution is given by

$$u(x,y) = \operatorname{Re}\left(\frac{1}{a^2 + z^2} + \frac{1}{a^2 - z^2}\right), \qquad a = 1.05.$$

Ω is partitioned into n^2 square of equal size, and the partition of unity is taken as the standard bilinear hat functions associated with this mesh. This partition of unity is fixed and the local approximation spaces V_i are taken as

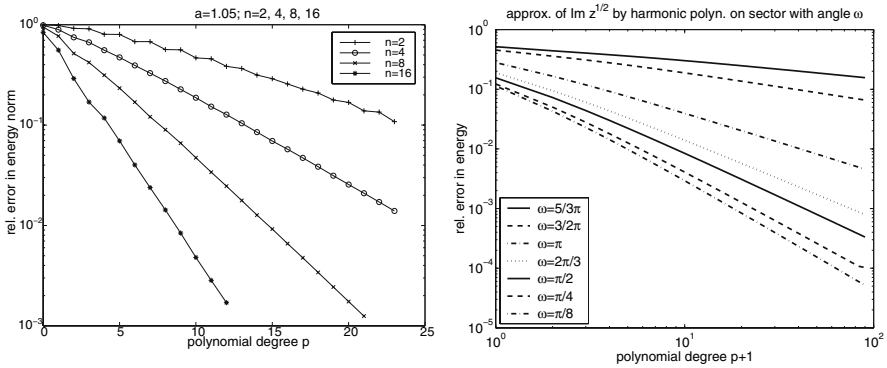


Figure 5.2. Left: Exponential convergence of Example 5.3. Right: Algebraic convergence of Example 5.4.

$\mathcal{HP}_p$ for different values of $p \in \mathbb{N}_0$. The numerical results in the left graph in Figure 5.2 present the result of the minimization problem

$$\min\Big\{ \frac{\|\nabla(u-v)\|_{L^2(\Omega)}}{\|\nabla u\|_{L^2(\Omega)}} \,\Big|\, v \in V := \sum_{i=1}^{(n+1)^2} \varphi_i V_i \Big\}$$

in dependence on the polynomial degree p. ∎

Algebraic convergence results are also available:

Theorem 5.2. *Let $\Omega \subset \mathbb{R}^2$ be star-shaped with respect to a ball and let Ω satisfy an exterior cone condition with angle $\lambda\pi$. Let $k \geq 1$ and let $u \in H^k(\Omega)$ satisfy (5.4). Then there exists $C > 0$ and harmonic polynomials $u_p \in \mathcal{HP}_p$ such that*

$$\|u - u_p\|_{H^j(\Omega)} \leq C \left(\frac{\ln(p+2)}{p+2} \right)^{\lambda(k-j)}, \qquad j = 0, 1.$$

Proof. See [80]. □

Example 5.4.
Theorem 5.2 can be sharpened in the following situation (see [80] for a more detailed discussion of this effect): Define the sector $S_\omega = \{(r\cos\varphi, r\sin\varphi) \,|\, 0 < r < 1, 0 < \varphi < \omega\}$ and let $u(x,y) = \operatorname{Re} z^\alpha$ or $u(x,y) = \operatorname{Im} z^\alpha$ for some $\alpha > 0$. Then we have with $\lambda = 2 - \frac{\omega}{\pi}$ and any $\varepsilon > 0$

$$\inf_{v \in \mathcal{HP}_p} \|u - v\|_{H^1(S_\omega)} \leq C_\varepsilon p^{-\lambda\alpha+\varepsilon},$$

where C_ε depends on α, ω, and ε. Figure 5.2 illustrates this convergence behaviour by plotting for different values of ω the result of the minimization

problem

$$\min\Big\{ \frac{\|\nabla(u_{1/2}-v)\|^2_{L^2(S_\omega)}}{\|\nabla u_{1/2}\|^2_{L^2(S_\omega)}} \,\Big|\, v \in \mathcal{HP}_p \Big\}, \qquad u_{1/2} = \operatorname{Im} z^{1/2},$$

in dependence on the polynomial degree p. It is noteworthy that in this particular example λ may be bigger than 1—this cannot be expected in the situation of Theorem 5.2. ∎

Remark 5.3.
The harmonic polynomials system is just one possible choice. Near corners, the solution of (5.4) has singularities, which are known. The corresponding singularity functions could be used as approximation systems. We will describe the idea of augmenting a standard FEM space with such singularity function in more detail in Section 6. ∎

5.3 Helmholtz Equation

We consider for two-dimensional problems the Helmholtz equation

$$-\Delta u - k^2 u = 0 \qquad \text{on } \Omega \subset \mathbb{R}^2, \tag{5.6}$$

and we discuss the following two choices of local approximation systems:

1. Systems of *plane waves*, $W(p)$, given by

$$W(p) := \operatorname{span}\Big\{ e^{\mathbf{i}k\omega_n\cdot(x,y)} \,|\, n = 0,\dots,p-1 \Big\}, \tag{5.7}$$

   where the vectors ω_n are given by $\omega_n := (\cos\frac{2\pi n}{p}, \sin\frac{2\pi n}{p})^\top$.
2. *Generalized harmonic polynomials* given by

$$V(p) := \operatorname{span}\{ J_n(kr)\sin(n\varphi), J_n(kr)\cos(n\varphi) \,|\, n = 0,\dots,p \}, \tag{5.8}$$

   where we employed polar coordinates (r,φ) in the definition of $V(p)$; the functions J_n are the first kind Bessel function.

We note that $\dim V(p) = O(p)$, $\dim W(p) = O(p)$. These spaces have the following approximation properties:

Theorem 5.3. *Let $\Omega \subset \mathbb{R}^2$ be a simply connected domain, $\Omega' \subset\subset \Omega$ be a compact subset. Let u solve (5.6). Then there exist C, $b > 0$ such that for all $p \in \mathbb{N}$, $p \geq 2$:*

$$\inf_{v\in V(p)} \|u-v\|_{H^1(\Omega')} \leq Ce^{-bp}, \qquad \inf_{v\in W(p)} \|u-v\|_{H^1(\Omega')} \leq Ce^{-bp/\ln p}. \tag{5.9}$$

Proof. The first estimate is proved in [80]. The second one can be proved using the arguments detailed in Section C.2. □

Theorem 5.4. *Let $\Omega \subset \mathbb{R}^2$ be star-shaped with respect to a ball. Let Ω satisfy an exterior cone condition with angle $\lambda\pi$. Let $u \in H^k(\Omega)$, $k \geq 1$, solve (5.6). Then there exists $C > 0$ such that*

$$\inf_{v \in V(p)} \|u - v\|_{H^1(\Omega)} \leq C \left(\frac{\ln(p+2)}{p+2} \right)^{\lambda(k-1)}, \tag{5.10}$$

$$\inf_{v \in W(p)} \|u - v\|_{H^1(\Omega)} \leq C \left(\frac{\ln^2(p+2)}{p+2} \right)^{\lambda(k-1)}. \tag{5.11}$$

Proof. (5.10) is proved in [80]. See Section C.2 for the proof of (5.11). □

Example 5.5.
The function

$$u(x, y) = e^{\mathbf{i}k(\cos\theta, \sin\theta)}, \qquad \theta = \frac{\pi}{16},$$

is a solution of (5.6). Let $\Omega = (0, 1)$, and let g be defined on $\partial\Omega$ by $g := \partial_n u + \mathbf{i}ku$. Then u solves

$$-\Delta u - k^2 u = 0 \quad \text{on } \Omega, \qquad \partial_n u + \mathbf{i}ku = g \quad \text{on } \partial\Omega. \tag{5.12}$$

Let Ω be partitioned into $n \times n$ squares of equal size. We take as the partition of unity ψ_i, $i = 1, \ldots, (n+1)^2$, the standard bilinear hat functions associated with the $(n+1)^2$ nodes. The approximation space V is then constructed as in Theorem 4.1 with local spaces taken either as $V(p)$ (with p ranging from 1 to 15) or as $W(p)$ (with $p \in \{2, 6, 10, 14, 18, 22, 26, 30, 34, 38\}$). Contrary to our exposition so far, all spaces are taken as spaces over the field $\mathbb{C}$ instead of $\mathbb{R}$. The numerical approximation u_N is obtained as the standard Galerkin approximation for problem (5.12), viz.,

$$\text{find } u_N \in V \text{ s.t. } \int_\Omega (\nabla u_N \cdot \nabla \overline{v} - k^2 u \overline{v}) + \mathbf{i}k \int_{\partial\Omega} u\overline{v} = \int_\Omega f\overline{v} + \int_{\partial\Omega} g\overline{v} \quad \forall\, v \in V.$$

Theorem 5.3 suggests that an exponential rate of convergence could be achieved. The numerical results for $k = 32$ are displayed in Figure 5.3. Indeed, we observe for fixed n an exponential convergence in $p \sim N$ for the relative error $\|u - u_N\|_{H^1(\Omega)} / |u|_{H^1(\Omega)}$. We refer to [79] for more details. ∎

5.4 Linear Elasticity

In two-dimensional linear elasticity and in the absence of body forces, the displacement field (u, v) satisfies the following system of equations:

$$\partial_x \sigma_x + \partial_y \tau_{xy} = 0, \qquad \partial_x \tau_{xy} + \partial_y \sigma_y = 0; \tag{5.13}$$

here, the stresses σ_x, σ_y, and τ_{xy} are defined by

$$\sigma_x = \lambda(\partial_x u + \partial_y v) + 2\mu\partial_x u, \;\; \sigma_y = \lambda(\partial_x u + \partial_y v) + 2\mu\partial_y v, \;\; \tau_{xy} = \mu(\partial_y u + \partial_x v).$$

The material constants λ, μ are called the Lamé constants.

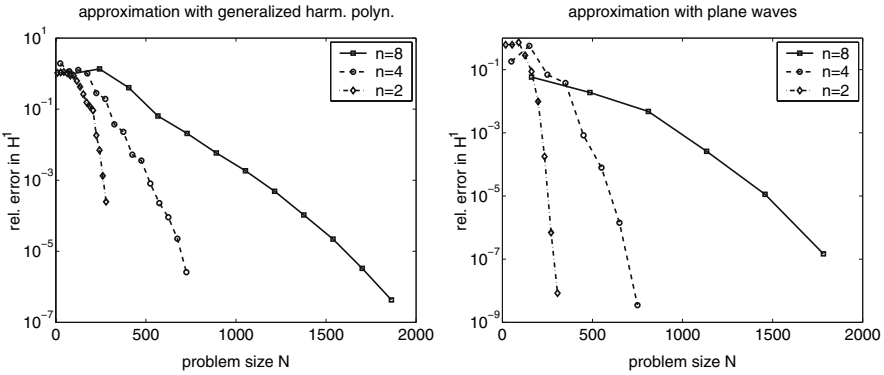


Figure 5.3. Operator adapted methods for Helmholtz equation; see Example 5.5. Local approximation space $V(p)$ (left) and $W(p)$ (right).

Remark 5.4.
The above system is written for the so-called plane strain case. For plane stress, λ should be replaced with $\lambda^* = 2\lambda\mu/(\lambda + 2\mu)$. ∎

Let Ω be simply connected. By [85], the displacement field (u, v) can then be expressed in terms of two holomorphic functions φ, ψ, namely,

$$2\mu\,[u(x,y) + \mathbf{i}v(x,y)] = \kappa\varphi(z) - z\overline{\varphi'(z)} - \overline{\psi}(z); \tag{5.14}$$

here, we set $\kappa = (\lambda + 3\mu)/(\lambda + \mu)$. This representation is unique if we require additionally $\varphi(z_0) = 0$ for an arbitrarily chosen point z_0. This representation suggests to use as an approximation space for the approximation of the complex function $u + \mathbf{i}v$ the space

$$V_p^{\text{elast}} := \operatorname{span}\{\kappa\pi(z) - z\overline{\pi'(z)} - \overline{\rho(z)} \,|\, \pi, \rho \in \mathcal{H}_p\}, \tag{5.15}$$

where $\mathcal{H}_p$ denote the space of (complex) polynomials of degree p. An approximation result analogous to Theorem 5.2 can indeed be obtained:

Theorem 5.5. *Let $\Omega \subset \mathbb{R}^2$ be star-shaped with respect to a ball. Let Ω satisfy an exterior cone condition with angle $\hat{\lambda}\pi$. Let $m \in \mathbb{N}$, $s \in [0, 1)$ and assume that the displacement field $(u, v) \in H^{m+s}(\Omega)$ satisfies the homogeneous elasticity equations (5.13). Then the function $\mathbf{u} := u + \mathbf{i}v$ can be approximated from V_p^{elast} such that*

$$\inf_{\mathbf{u}_{ap} \in V_p^{\text{elast}}} \|\mathbf{u} - \mathbf{u}_{ap}\|_{H^1(\Omega)} \le C \left(\frac{\ln(p+2)}{p+2}\right)^{\hat{\lambda}(m+s-1)} \|\mathbf{u}\|_{H^{m+s}(\Omega)}.$$

Proof. See Section C.3. □

Remark 5.5.
The proof of Theorem 5.5 shows that the improved rate of convergence for the typical singularity functions that we observed in Example 5.4 are also obtained for the elasticity equations. ∎

5.5 Further Examples

The Laplace equation and the Helmholtz equation are merely two examples of elliptic equations for which special approximation systems can be constructed. A more general theory by S. Bergman [16–18] and I.N. Vekua [105] is in fact available: For two-dimensional elliptic equations of the form

$$-\Delta u + a(x,y)\partial_x u + b(x,y)\partial_y u + c(x,y)u = 0, \tag{5.16}$$

where the functions a, b, c are real analytic on Ω, there exists a linear operator ReV that maps functions holomorphic on Ω onto solutions of solution of (5.16). Essentially, this operator is a bijection and bicontinuous in Sobolev norms. That is: regularity assertions for u can be translated into regularity assertions for the corresponding holomorphic functions; this function may then be approximation by (complex) polynomials; the image of (complex) polynomials under ReV then yields a good approximation space. In some cases, the operator ReV can be computed explicitly (e.g. in the case of the Helmholtz equations, where the space $V(p)$ is precisely the image of complex polynomials under the map ReV); we refer to Appendix C and [80] for more details on this. The representation theory of Bergman and Vekua is, due to its close link with complex analysis, largely a two-dimensional theory. Some extensions to three dimensions have been done in [28].

5.6 Local Approximation Spaces Obtained Numerically

In the above examples the local approximation spaces were given in closed form. They can, however, be obtained numerically as well. For example, while the form of the singularity functions of linear elasticity is known, the precise exponents have to be determined as solutions of small auxiliary problems. More in the spirit of domain decomposition is the following approach for problems of the form $Lu = 0$: For each patch Ω_i, one chooses a finite dimensional space $V_{i,\partial\Omega_i} = \operatorname{span}\{\tilde{b}_{i,j} \mid j = 1, \dots, N_i\}$ of functions that are defined on $\partial\Omega_i$. The space V_i is then obtained by (numerically) solving boundary value problems

$$Lb_{i,j} = 0 \quad \text{on } \Omega_i, \qquad b_{i,j}|_{\partial\Omega_i} = \tilde{b}_{i,j}.$$

The total computation is therefore done in two steps: first, many local problems are solved (which can be done completely in parallel), and in a second step a global problems is solved. Conceptually, this is the approach taken for example in [5] and [37, 58, 59] for calculations of very heterogeneous media.

Remark 5.6.
The functions $b_{i,j}$ were computed above as solutions of Dirichlet problems. The approximation space V_i could be determined by solving other boundary value problems, e.g. by solving Neumann problems. It has also been observed that it is advantageous to define them as solutions of boundary value problems defined on Ω_i', where $\Omega_i \subset\subset \Omega_i'$. We refer, for example, to [5] for more details on this. ∎

Another example of a method where the approximation spaces are determined numerically in a preprocessing step is the generalized FEM of [77, 95] for problems with periodic microstructures.

5.7 Bibliographical Remarks

Approximation systems that are tailored to the differential operator are used by engineers, where such methods are known, among others, under the name of Trefftz methods, see for example [56,63,64]. In the context of the partition of unity method/generalized FEM special approximation systems have been used in [69] for Helmholtz problems and in [34, 90] for elasticity and crack problems. The "method of particular solutions" [43], [19] (see, in particular, the references in [19]) is closely related to the ideas presented here.
We have seen the poor performance of the classical FEM in Section 5.1. Indeed, it was already shown in [10] that the classical FEM can perform arbitrarily poorly. On the other hand, the constructions in [81] show that for reasonable classes of right-hand sides, it is in principle possible to construct good approximation spaces. Such approximation spaces have to be adapted to a particular problem at hand.

6 Augmenting Classical FEM Spaces

The partition of unity method/generalized FEM can be viewed as a framework for incorporating information about the problem into the approximation space. The simplest such technique is to augment a standard finite element space with special functions.

6.1 Singular Functions

The power of augmenting a classical FEM space with special functions can be seen in the following model problem: Let $\Omega \subset \mathbb{R}^2$ be a polygon and consider

$$-\Delta u = f \quad \text{on } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega. \tag{6.1}$$

If we denote by A_j, $j = 1, \dots, J$, the vertices of Ω and by $\omega_j \in (0, 2\pi)$ the internal angle of Ω at A_j, then it is well-known that the classical FEM-space $S_0^{1,1}(\mathcal{T})$ that is based on a quasi-uniform mesh $\mathcal{T}$ of mesh size h performs poorly if $\max_{j=1,\dots,J} \omega_j > \pi$; namely, the rate of convergence is

$$\inf_{v \in S_0^{1,1}(\mathcal{T})} \|u - v\|_{H^1(\Omega)} \le Ch^\alpha, \qquad \alpha = \min_{j=1,\dots,J} \frac{\pi}{\omega_j} < 1.$$

This is indeed observed in practise. By augmenting this FEM space by a few suitably chosen singularity functions, however, we recover the optimal rate of convergence. To this end, it is important to note the following regularity assertion for the solution u of (6.1):

Lemma 6.1. *Let $\Omega \subset \mathbb{R}^2$ be a polygon with vertices A_j, $j = 1, \dots, J$, and internal angles ω_j, $j = 1, \dots, J$. Define for each vertex A_j the singularity functions $S_{j,i}$, $i = 1, 2, \dots,$ by*

$$S_{j,i}(r_j, \varphi_j) := \begin{cases} r_j^{i\pi/\omega_j} \sin(i\frac{\pi}{\omega_j}\varphi_j) & \text{if } i\pi/\omega_j \notin \mathbb{N} \\ r_j^{i\pi/\omega_j} \left[\ln r_j \sin(i\frac{\pi}{\omega_j}\varphi_j) + \varphi_j \cos(i\frac{\pi}{\omega_j}\varphi_j)\right] & \text{if } i\pi/\omega_j \in \mathbb{N} \end{cases} \tag{6.2}$$

where (r_j, φ_j) represent polar coordinates with origin A_j such that the two edges of Ω meeting at A_j fall on the lines $\varphi_j = 0$ and $\varphi_j = \omega_j$.
Let $f \in H^{-1+k}(\Omega)$, $k > 0$ and $k \notin \mathbb{N}$. Then the solution u of (6.1) can be written in the form

$$u = \sum_{j=1}^{J} \sum_{\substack{i \in \mathbb{N} \\ i\frac{\pi}{\omega_j} < k}} a_{ij} S_{ij} + u_0, \tag{6.3}$$

for some numbers $a_{ij} \in \mathbb{R}$ and $u_0 \in H^{1+k}(\Omega)$.

Proof. Such decompositions can be found, for example, in [52, 53]. □

This regularity assertion allows us to design approximation spaces that recover the optimal rate of convergence (in terms of "error vs. problem size"):

Exercise 6.1.
Fix a cut-off function $\chi_j \in C_0^\infty(\mathbb{R}^2)$ for each corner A_j such that $\chi_j \equiv 1$ in a neighbourhood of A_j and such that $\chi_j \equiv 0$ in a neighbourhood of the vertices A_i, $i \neq j$.

(a) Show: The decomposition (6.3) can take the form

$$u = \sum_{j=1}^{J} \sum_{\substack{i \in \mathbb{N} \\ i\frac{\pi}{\omega_j} < k}} a_{ij} \chi_j S_{ij} + \tilde{u}_0,$$

where $\tilde{u}_0 \in H^{1+k}(\Omega) \cap H_0^1(\Omega)$. Additionally, $\chi_j S_{i,j} \in H_0^1(\Omega)$.

(b) Show: The space

$$V_N := S_0^{p,1}(\mathcal{T}) \oplus \operatorname{span}\{\chi_j S_{j,i} \,|\, j = 1, \dots, J, i\frac{\pi}{\omega_j} < k\} \subset H_0^1(\Omega)$$

satisfies

$$\inf_{\in V_N} \|u - v\|_{H^1(\Omega)} \leq C h^{\min\{p,k\}}. \tag{6.4}$$

Note that $\dim V_N \sim \dim S_0^{p,1}(\mathcal{T})$. ∎

The purpose of the cut-off functions χ_j is to localize the singularity functions. This could also be achieved with the aid classical FEM functions:

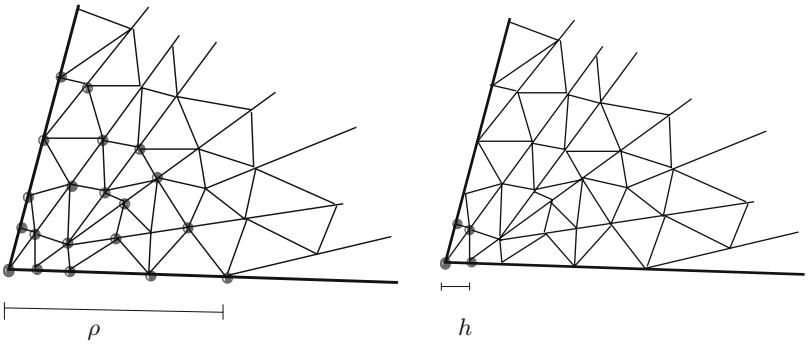


Figure 6.1. Nodes marked • are augmented with singularity function. Left: $O(h^{-2})$ nodes are augmented to ensure optimal rate of convergence. Right: augmenting very few nodes often suffices in practise.

Exercise 6.2.
Let $\mathcal{T}$ be a quasi-uniform mesh on the polygon $\Omega \subset \mathbb{R}^2$. Let $\{\psi_i \,|\, i = 1, \ldots, N_1\}$ be set of the classical piecewise linear hat functions associated with $\mathcal{T}$ and $S^{1,1}(\mathcal{T}) = \operatorname{span}\{\psi_i \,|\, i = 1, \ldots, N_1\}$. Fix $\rho > 0$ and define, for each $j \in \{1, \ldots, J\}$, the set $I_j := \{i \,|\, \operatorname{supp} \psi_i \subset B_\rho(A_j)\}$. Define

$$V_N := S_0^{p,1}(\mathcal{T}) \oplus \operatorname{span}\{\psi_i S_{j,m} \,|\, m\frac{\pi}{\omega_j} < k, \quad i \in I_j, \quad j = 1, \ldots, J\}.$$

Show: Also for this choice of approximation space the approximation property (6.4) holds. Note: $V_N \subset H_0^1(\Omega)$ and $\dim V_N \sim \dim S_0^{p,1}(\mathcal{T})$. ∎

The above construction involves only classical FEM functions and the singularity functions $S_{j,i}$. Of course, since $\rho > 0$ is fixed, a rather large number of nodes is affected (see the left picture in Figure 6.1, where the nodes that require multiplication with singularity functions are denoted •), namely, $O(h^{-2})$ nodes. A variety of practitioners have therefore looked at further simplifications:

Example 6.1.
In practise, a) only the strongest singularity functions are added (typically only $S_{j,1}$), b) only those singularity functions at re-entrant corners (that is for corners A_j where $\pi/\omega_j < 1$) and c) $\rho \sim h$ is chosen (see the right picture in Figure 6.1). While the choice $\rho \sim h$ does not improve the rate of convergence, the constant is greatly improved so that in many cases good engineering accuracy is reached. ∎

6.2 Crack Propagation Problems

Crack propagation problems have been put forward as an example where augmenting a standard FEM space with special functions is advantageous. In

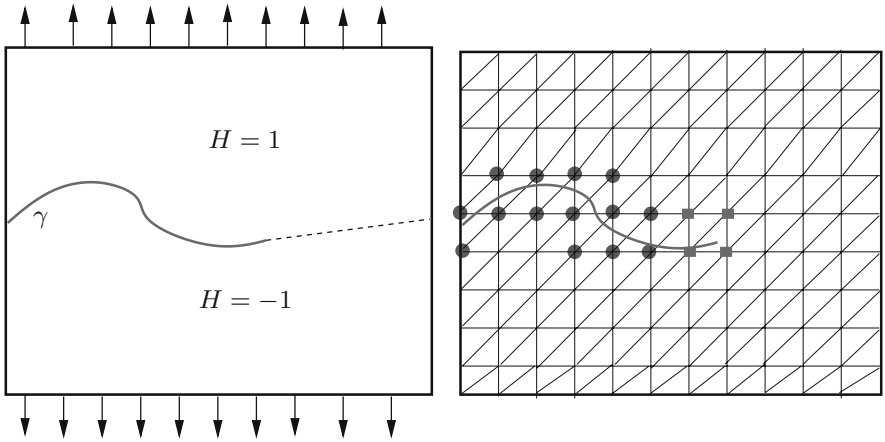


Figure 6.2. Left: Crack problem. Right: Classical FEM mesh. Nodes • are enriched with discontinuity functions; nodes marked ▪ are enriched with singularity function.

many 2D crack problems, the crack is modelled as a curve γ (see Figure 6.2). A linear elasticity problem is solved on $\Omega \setminus \gamma$; then the so-called stress intensity factors are extracted from the FEM solution; from these stress intensity factors the crack propagation is determined according to some engineering model; finally, the crack is extended, and the next iteration of this loop is performed. Performing such a crack propagation analysis is costly since the domain $\Omega \setminus \gamma$ on which the elasticity equations have to be solved, changes in each iteration step thus requiring (at least local) remeshing. Additionally, since the solution exhibits a strong singularity at the crack tip, a strongly refined mesh is required near the crack tip to resolve this singularity and guarantee reliable results. The technique of augmenting a standard FEM space by a few special functions to overcome these two difficulties seems very attractive and has been proposed, for example, under the name X-FEM (extended FEM) in [29,84,98] and in the context of the generalized FEM. We will only sketch the key ideas of the X-FEM applied to crack propagation problems. For that, we will not consider the elasticity equation but the simpler scalar case of

$$-\Delta u = 0 \quad \text{on } \Omega \setminus \gamma, \qquad \partial_n u = 0 \quad \text{on } \gamma^+ \text{ and on } \gamma^- \tag{6.5}$$

together with further boundary conditions on $\partial\Omega$. Here, γ^+ and γ^- denote the upper and lower part of the curve γ (see Figure 6.2). If γ is sufficiently smooth, then an expansion analogous to that of Lemma 6.1 can be obtained, namely, near the crack tip (located at the origin), the solution u of (6.5) takes the form

$$u = \sum_{n=0}^{\infty} S_n r^{n/2} \cos\left(\frac{n}{2}\varphi\right);$$

here, the coefficient $S_1 \in \mathbb{R}$ of the first singularity function is called, in analogy to the elasticity case, the stress intensity factor. The solution u need not be continuous across the curve γ. It is, away from the crack tip, only piecewise smooth. The idea of the X-FEM is to employ a standard FEM space V_{FE} on Ω. This space ignores the crack γ but takes care of the geometry of Ω and the boundary conditions on $\partial\Omega$. The crack γ is then accounted for as follows: nodes near γ but far from the crack tip are collected in the set I_H, nodes near the crack tip are collected in the set I_{CT} (see Figure 6.2 where these sets are denoted $\bullet$ and $\blacksquare$). One defines the discontinuity function

$$H(x) := \begin{cases} 1 & \text{if } x \text{ is above } \gamma \\ -1 & \text{if } x \text{ is below } \gamma \end{cases}$$

and takes as approximation space

$$V_N := V_{FE} \oplus \operatorname{span}\{H\psi \,|\, i \in I_H\} \oplus \operatorname{span}\{\psi_i r^{1/2} \cos\frac{1}{2}\varphi \,|\, i \in I_{CT}\}.$$

This approximation space is chosen so as to account for the expected solution behaviour near the crack tip. Near the crack but away from the crack tip, the space V_N contains discontinuous functions, reflecting the fact that the solution sought may jump across the crack γ.

Remark 6.1.
Some extensions of this choice would be: a) add more singularity functions, b) use higher order discontinuity functions, e.g. $H(x)\pi(x)$, where $\pi \in \mathcal{P}_p$ (the above construction corresponds to $p = 0$). ∎

We will not analyze the approximation properties of the space V_N defined above. The following exercise, however, gives an indication of what can be expected away from the crack tip.

Exercise 6.3.
Let $\Omega = (-1, 1)$ and consider a uniform mesh $\mathcal{T}$ of mesh size $h = 2/(2N+1)$ with nodes $x_i = -1 + ih$, $i = 0, \ldots, 2N+1$. Let $S^{p,1}(\mathcal{T}) \subset C(\Omega)$ be a standard FEM space on the mesh $\mathcal{T}$ and consider the approximation of a function u that is smooth on $[-1, 0) \cup (0, 1]$ but has a jump discontinuity at 0. What convergence rate (in L^2) can be expected? Augment the nodes x_N, x_{N+1} with the Heaviside function $H(x) = \operatorname{sign} x$, that is consider $S^{p,1}(\mathcal{T}) \oplus \operatorname{span}\{\psi_N(x)H(x), \psi_{N+1}(x)H(x)\}$, where ψ_i is the standard hat function associated with nodes x_i. What convergence rate can be expected? Consider $S^{p,1}(\mathcal{T}) \oplus \operatorname{span}\{\psi_N(x)H(x)x^j, \psi_{N+1}H(x)x^j \,|\, j = 0, \ldots, p-1\}$. ∎

Remark 6.2.
If only very few close neighbours of the crack tip are enriched with the singularity function, then the rate of convergence cannot be expected to be good. Nevertheless, as already pointed out in Example 6.1, good engineering accuracy can be reached. ∎

6.3 Further Examples: The Generalized FEM

The generalized FEM in the form [101–103] is very similar to the X-FEM. The versatility of the generalized FEM is demonstrated in [101–103] by calculations on complicated domains, for example, domains with many holes or cracks. A classical FEM is augmented by special functions the reflect the proper behaviour of the solution near these features. Related earlier work on the generalized FEM for elasticity and crack problems can be found in [34,90].

6.4 Bibliographical Remarks

The idea of augmenting classical FEM spaces with special functions adapted to a problem has a long history. For problems with singularities (e.g. corner singularities) it can be found in [20, 42].
The bilinear form a in all the above examples involves an integration over Ω. In practise, this integration is replaced by numerical quadrature. Based on modern adaptive quadrature techniques (possibly including adaptive order control for higher efficiency) it is possible to perform the integration in a completely black box fashion where the user merely needs to provide information whether a point $x \in \mathbb{R}^d$ is in Ω. The "pixelation" technique of [102] can be viewed as an example of such an approach. For geometries whose boundary is piecewise smooth or piecewise affine, it can be much more efficient to deviate from the black box approach by employing local meshing near the boundary, [48,96,101,103]. Note that this local mesh near the boundary need not be regular since it is only used for quadrature purposes. The structure of the shape functions also greatly affects the cost of the quadrature. Consider as an example the particle partition of unity method of [96]. There, the shape functions whose support is contained in Ω are constructed such that they are piecewise smooth, where the regions of smoothness are axis parallel boxes. Clearly, this choice greatly simplifies the design of appropriate quadrature rules. We finally mention that the use of numerical quadrature entails errors; some ideas for their control are discussed in [101].

7 Enforcement of Essential Boundary Conditions

In many applications, essential boundary conditions have to be enforced. As a model problem we consider the classical Poisson problem: Find $u \in H_0^1(\Omega)$ such that

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v \, dx = F(v) := \int_\Omega f v, \; dx \quad \forall \; v \in H_0^1(\Omega). \tag{7.1}$$

The ideas how to enforce essential boundary conditions in meshless mehods are essentially the same ones as in the classical FEM. They can be split into two categories:

- *Conforming methods*: The approximation space V_N is chosen as a subspace of $H^1_0(\Omega)$, that is $V_N \subset H^1_0(\Omega)$. This can be achieved by:
  - using cut-off functions;
  - combining the classical FEM near the boundary with particle methods in the interior;
  - creating $H^1_0(\Omega)$-conforming spaces in the framework of the partition of unity method by properly selecting the local approximation spaces V_i near the boundary.
- *Non-conforming methods*: In these methods, the variational formulation is changed. These methods include:
  - Lagrange multiplier methods;
  - collocation of boundary conditions;
  - penalty methods;
  - Nitsche's method.

7.1 Conforming Methods

For the model case (7.1) the approximation space V_N has to be chosen to satisfy $V_N \subset H^1_0(\Omega)$.

A Simple Approach
The simplest approach is to select from a given set $\mathcal{B} = \{\varphi_i \,|\, i = 1, \dots, N\}$ of shape functions only those that satisfy $\varphi_i \in H^1_0(\Omega)$, that is to take

$$V_{N,0} := \operatorname{span}\{\varphi_i \,|\, (\operatorname{supp} \varphi_i)^\circ \subset \Omega\}. \tag{7.2}$$

Good approximation properties cannot be expected of $V_{N,0}$, however, even if the function to be approximated is smooth:

Exercise 7.1.
Let $V_N := \operatorname{span}\{\varphi_i \,|\, i = 0, \dots, N\}$ be the space of piecewise linear functions associated with the mesh given by the points $x_i = -\frac{h}{2} + ih$, $i = 0, \dots, N$, $h = 1/(N-1)$. Consider for $\Omega = (0,1)$ the subspace $V_{N,0} \subset V_N$ given by $V_{N,0} = \operatorname{span}\{\varphi_i \,|\, (\operatorname{supp} \varphi_i)^\circ \subset \Omega\}$. Show that for the smooth function $u(x) = x(1-x) \in H^1_0(\Omega)$ we have

$$\inf_{v \in V_{N,0}} \|u - v\|_{H^1(\Omega)} \geq C\sqrt{h}.$$

∎

Cut-Off Function Methods
In cut-off function methods, the essential boundary conditions are enforced by multiplying an approximation space V_N by a weight function w, where w vanishes on $\partial\Omega$ and satisfies $w \sim \operatorname{dist}(\cdot, \partial\Omega)$. If w is sufficiently smooth and $V_N \subset H^1(\Omega)$, then we obtain an $H^1_0(\Omega)$-conforming subspace $V_{w,N}$ by setting $V_{w,N} := wV_N \subset H^1_0(\Omega)$. These ideas can be traced back to [67, 83] and were revived in [57]. Concerning the approximation properties of the space $V_{w,N}$ we follow [57].

Lemma 7.1. *Let $k \geq 2$, and let $w \in W^{k,\infty}(\Omega)$ be such that $w \sim \operatorname{dist}(\cdot, \partial\Omega)$. Then there exists $C > 0$ such that for any compact subset $\Omega' \subset\subset \Omega$ we have for functions u, v satisfying $u = vw$*

$$\|v\|_{H^k(\Omega)} \leq C\delta^{-1} \left[\|u\|_{H^k(\Omega)} + \|v\|_{H^{k-1}(\Omega')} \right], \qquad \|v\|_{H^{k-1}(\Omega)} \leq C\|u\|_{H^k(\Omega)},$$

where $\delta = \operatorname{dist}(\Omega', \partial\Omega)$.

Proof. The proof follows from Hardy's inequality. The details can be found in [57, Thm. 6.1]. □

Lemma 7.1 can be employed to recover the optimal rate of convergence if $u \in H^k(\Omega) \cap H_0^1(\Omega)$:

Exercise 7.2.
Let $\Omega \subset \mathbb{R}^d$ have a smooth boundary. Assume the setting of Exercise 4.3. Suppose that $p_i = p \geq k - 1 \geq 1$ for all i and that $h_i \sim h$ for all i. Show, using Lemma 7.1, that the space $V_{w,N} = wV_N$ satisfies

$$\inf_{v \in V_{w,N}} \|u - v\|_{H^1(\Omega)} \leq Ch^{k-1}\|u\|_{H^k(\Omega)} \qquad \forall\, u \in H^k(\Omega) \cap H_0^1(\Omega);$$

here V_N is chosen as in Exercise 4.3. ∎

Remark 7.1.
The existence of a weight function w with the above regularity properties is closely related to the smoothness of $\partial\Omega$: the "natural" choice $w(x) := \operatorname{dist}(x, \partial\Omega)$ is only smooth if $\partial\Omega$ is. ∎

Combination with the Classical FEM
A technique proposed, e.g. in [68], is to combine shape functions of the classical FEM with general particle methods. In the vicinity of the boundary $\partial\Omega$, a standard mesh is defined and a standard FE space is employed. This space guarantees optimal approximation properties and gives the flexibility of the classical FEM to handle boundary conditions. For the approximation in the interior of Ω, any system can be used, e.g. systems $V_{N,0}$ of the form (7.2). These ideas can be shaped into several forms. In order to illustrate what can be expected, we present the following example:

Example 7.1.
Let $\Omega \subset \mathbb{R}^2$ be a polygon, and let $2 \leq k \leq p$. Let $V_N \subset H^1(\Omega)$ be an approximation space with the property

$$\inf_{v \in V_N} \|u - v\|_{L^2(\Omega)} + h\|u - v\|_{H^1(\Omega)} \leq Ch^k\|u\|_{H^k(\Omega)}. \tag{7.3}$$

Let $S_h := \{x \in \Omega \mid \operatorname{dist}(x, \partial\Omega) < h\}$ be a tubular neighbourhood of $\partial\Omega$. Let $\mathcal{T}$ be an affine, quasi-uniform triangulation of mesh size $O(h)$ of a set $\Omega' \subset \Omega$ that satisfies $S_h \subset \Omega'$. Let $S^{p,1}(\mathcal{T})$ be the standard finite element space of

piecewise polynomials of degree p on the mesh $\mathcal{T}$ and set $S_0^{p,1}(\mathcal{T}) = S^{p,1}(\mathcal{T}) \cap H_0^1(\Omega')$. Note that by extending functions of $S_0^{p,1}(\mathcal{T})$ by zero outside of Ω', we may think of $S_0^{p,1}(\mathcal{T})$ as a subset of $H_0^1(\Omega)$. Let $\{\psi_i \,|\, i \in I_{\partial\Omega}\} \subset S^{1,1}(\mathcal{T})$ be the standard piecewise linear hat functions associated with the nodes on $\partial\Omega$ and set

$$\omega := \sum_{i \in I_{\partial\Omega}} \psi_i.$$

Again, by the support properties of the piecewise linear hat functions ψ_i, we may think of ω as being defined on Ω. We observe:

$$\begin{aligned} &\omega \equiv 1 \quad \text{on } \partial\Omega, \qquad \omega \equiv 0 \quad \text{on } \Omega \setminus \Omega', \\ &\omega \in W^{1,\infty}(\Omega), \qquad \|\nabla\omega\|_{L^\infty(\Omega)} \le Ch^{-1}. \end{aligned}$$

We select as the approximation space

$$V_{p,N} := (1-\omega)V_N \oplus S_0^{p,1}(\mathcal{T}) \subset H_0^1(\Omega).$$

We claim that for $u \in H^k(\Omega) \cap H_0^1(\Omega)$

$$\inf_{v \in V_{p,N}} \|u - v\|_{H^1(\Omega)} \le Ch^{k-1}\|u\|_{H^k(\Omega)}. \tag{7.4}$$

(7.4) is shown using the same ideas as in the proof of Theorem 4.1. Let $u_N \in V_N$ be an approximation of u from V_N such that

$$\|u - u_N\|_{L^2(\Omega)} + h\|u - u_N\|_{H^1(\Omega)} \le Ch^k\|u\|_{H^k(\Omega)}.$$

We will take the approximant to u from $V_{p,N}$ of the form $(1-\omega)u_N + v$, where $v \in S_0^{p,1}(\mathcal{T})$ will be determined below. The error can be written as $u - (1-\omega)u_N - v = (1-\omega)(u-u_N) + (\omega u - v)$. For the first term, we calculate

$$\|(1-\omega)(u-u_N)\|_{L^2(\Omega)} + h\|(1-\omega)(u-u_N)\|_{H^1(\Omega)} \le Ch^k\|u\|_{H^k(\Omega)},$$

which has the desired form (7.4). We now turn to the definition of $v \in S_0^{p,1}(\mathcal{T})$, which approximates ωu. We select $I_{p-1}u \in S^{p-1,1}(\mathcal{T})$ by a standard FEM interpolation procedure. Then, $(I_{p-1}u)|_{\partial\Omega} = 0$ and

$$\|u - I_{p-1}u\|_{L^2(K)} + h\|\nabla(u - I_{p-1}u)\|_{L^2(K)} \le Ch^k|u|_{H^k(K)} \qquad \forall\, K \in \mathcal{T}.$$

Here, we exploited the assumption $p \ge k$. As the product of a piecewise linear function and a piecewise polynomial of degree $p-1$, the function $\omega I_{p-1}u$ satisfies $\omega I_{p-1}u \in S_0^{p,1}(\mathcal{T})$. We conclude using the support properties of ω and $\|\nabla\omega\|_{L^\infty(\Omega)} \le Ch^{-1}$

$$\|\omega u - \omega I_{p-1}u\|_{L^2(\Omega)} + h|\omega u - \omega I_{p-1}u|_{H^1(\Omega)} \le Ch^k.$$

Thus taking $v := \omega I_{p-1}u$ gives an approximation $(1-\omega)u_N + \omega I_{p-1}u \in V_{p,N}$ that realizes the desired bound (7.4). ∎

Local Approximation Spaces Satisfying Essential Boundary Conditions

The previous idea of combining the classical FEM in a strip near the boundary with general approximation spaces V_N in the interior of Ω can be viewed as a variant of the partition of unity method where the local approximation spaces V_i for the patches Ω_i near the boundary are chosen such that they conform to the boundary conditions. A more general approach is the outlined in the following exercise.

Exercise 7.3.
Assume the hypotheses of Theorem 4.1. Suppose additionally: if $\Gamma_{i,D} := \partial\Omega_i \cap \partial\Omega \neq \emptyset$, then $V_i \subset H^1_D(\Omega_i) := \{u \in H^1(\Omega_i) \,|\, u|_{\Gamma_{i,D}} = 0\}$. Show: The space V of Theorem 4.1 satisfies $V \subset H^1_0(\Omega)$, and the approximation result of Theorem 4.1 is still valid. ∎

Local approximation spaces V_i that satisfy the correct boundary conditions can be derived in different ways. They can be determined analytically or numerically.

Example 7.2.
Let u solve Laplace's equation and assume that u vanishes on a straight line. Extending u by reflection across this line yields a function (again denoted u) that is anti-symmetric with respect to this line and again solves Laplace's equation. It is shown in [78] that harmonic polynomials that are anti-symmetric with respect to this line (and hence vanish on it), can approximate the function u at the same rate as the full space $\mathcal{HP}_p$ of harmonic polynomials. ∎

As discussed in Section 5.6, local approximation spaces V_i can also be computed numerically. If these spaces are computed using the standard FEM, then it is easy to enforce essential boundary conditions.

7.2 Non-Conforming Methods: Lagrange Multiplier Methods and Collocation Techniques

The essential boundary condition could also be enforced in a weak sense. The simplest such approach is to collocate the boundary condition in a (finite) set of points $Y \subset \partial\Omega$ as was proposed, for example, in [2,54,111]. Such methods are, however, difficult to analyze even in the setting of the classical FEM.
Early references to the Lagrange Multiplier Method are [3,4]. One introduces a bilinear form $b : H^1(\Omega) \times H^{-1/2}(\partial\Omega)$ by

$$b(v,\mu) := \langle \gamma_0 v, \mu \rangle_{H^{1/2}(\partial\Omega)\times H^{-1/2}(\partial\Omega)},$$

where $\gamma_0 : H^1(\Omega) \to H^{1/2}(\partial\Omega)$ is the trace operator $\gamma_0 v = v|_{\partial\Omega}$. One then considers the problem: Find $(u,\lambda) \in H^1(\Omega) \times H^{-1/2}(\partial\Omega)$ such that

$$\begin{aligned} a(u,v) + b(v,\lambda) &= F(v) && \forall\, v \in H^1(\Omega), \\ b(u,\mu) &= 0 && \forall\, \mu \in H^{-1/2}(\partial\Omega). \end{aligned} \tag{7.5}$$

The function u of the pair (u, λ) solving (7.5) is in fact an element of $H_0^1(\Omega)$ and also a solution of the original problem (7.1). A natural discretization of (7.5) is to take subspaces $V_N \subset H^1(\Omega)$, $M_N \subset H^{-1/2}(\partial\Omega)$ and then consider the problem: Find $(u_N, \lambda_N) \in V_N \times M_N$ such that

$$\begin{aligned} a(u_N, v) + b(v, \lambda_N) &= F(v) \qquad \forall\, v \in V_N, \\ b(u_N, \mu) &= 0 \qquad \forall\, \mu \in M_N. \end{aligned} \tag{7.6}$$

We mention in passing that $\langle v, \mu \rangle_{H^{1/2}(\partial\Omega) \times H^{-1/2}(\partial\Omega)} = \int_{\partial\Omega} v\mu \, ds$ if $\mu \in L^2(\partial\Omega)$ so that the discrete problem (7.6) represents a linear system of equations that can be set up for any reasonable choice of space M_N (e.g. a space of piecewise constant functions). One challenge in the Lagrange multiplier method is that the spaces V_N and M_N cannot be chosen independently. As is well-known the so-called "inf-sup" condition, or Babuška-Brezzi condition, needs to be satisfied: If

$$\inf_{\mu \in M_N} \sup_{v \in V_N} \frac{b(v, \mu)}{\|v\|_{H^1(\Omega)} \|\mu\|_{H^{-1/2}(\partial\Omega)}} \geq \gamma_N > 0, \tag{7.7}$$

then the error $u - u_N$ satisfies (see for example [94, Thm. 5.13])

$$\|u - u_N\|_{H^1(\Omega)} \leq C \left(1 + \frac{1}{\gamma_N}\right) \inf_{(v,\mu) \in V_N \times M_N} \|u - v\|_{H^1(\Omega)} + \|\lambda - \mu\|_{H^{-1/2}(\partial\Omega)}.$$

This bound suggests that the inf-sup constant γ_N should be bounded away from zero uniformly in the discretization parameter N to guarantee good performance. The condition $\gamma_N > 0$ is indeed necessary as the following exercise shows.

Exercise 7.4.
Show: $\gamma_N = 0$ implies that the matrix representing the linear system (7.6) is not invertible. ∎

In the classical FEM, various combinations of spaces V_N and M_N are known to be "stable" in the sense that (7.6) holds for a constant independent of the mesh size; we refer to [100] for a more detailed discussion and appropriate references. In the context of the classical FEM, a key ingredient in the stability proofs for pairs V_N, M_N are inverse estimates. To the knowledge of the author, such estimates are not available for meshless methods, and an analysis is therefore hard. We will encounter a similar difficulty in our analysis of Nitsche's method below; the appropriate inverse estimate is therefore stipulated as Assumption 7.1.

7.3 Non-Conforming Methods: Penalty Method

In the conforming FEM, one would have to choose $V_N \subset H_0^1(\Omega)$. In the penalty method, the essential boundary conditions are weakened by changing

the problem: Taking $V_N \subset H^1(\Omega)$ and $\psi \geq 1$ the problem is to find $u_N \in V_N$ such that

$$a_\psi(u_N, v) := a(u_N, v) + \int_{\partial\Omega} \psi u_N v \, ds = F(v) \qquad \forall\, v \in V_N. \tag{7.8}$$

We recognize this as the Galerkin approximation to the following problem:

$$\text{Find } u_\psi \in H^1(\Omega) \text{ s.t. } a_\psi(u_\psi, v) = F(v) \qquad \forall\, v \in H^1(\Omega). \tag{7.9}$$

The strong form of this problem is:

$$-\Delta u_\psi = f \qquad \text{on } \Omega, \qquad \partial_n u_\psi + \psi u = 0 \qquad \text{on } \partial\Omega. \tag{7.10}$$

One sees that, if $\psi \to \infty$, then $u_\psi \to u$, where u is the solution of (7.1). We will make this more precise below.

Theorem 7.1 (penalty method). *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. Let $k \geq 2$. Assume $u \in H^k(\Omega)$ is the solution of (7.1). Let $\xi \in H^{k-1}(\Omega)$ solve*

$$-\Delta \xi + \xi = 0 \quad \text{on } \Omega, \qquad \xi|_{\partial\Omega} = \partial_n u \quad \text{on } \partial\Omega. \tag{7.11}$$

Assume that the approximation space $V_N \subset H^1(\Omega)$ satisfies:

$$\inf_{v \in V_N} \|u - v\|_{L^2(\Omega)} + h\|\nabla(u - v)\|_{L^2(\Omega)} \leq Ch^k, \tag{7.12}$$

$$\inf_{v \in V_N} \|\xi - v\|_{L^2(\Omega)} + h\|\nabla(\xi - v)\|_{L^2(\Omega)} \leq Ch^{k-1}. \tag{7.13}$$

Then there holds for a $C > 0$ independent of ψ and h

$$\|u - u_N\|_{H^1(\Omega)} \leq C \left\{ \psi^{-1} + \psi^{-1/2} h^{k-3/2} + \psi^{1/2} h^{k-1/2} + h^{k-1} \right\}.$$

Setting $\psi = h^\sigma$ with the optimal value $\sigma = \frac{2k-1}{3}$ gives

$$\|u - u_N\|_{H^1(\Omega)} \leq h^\sigma, \qquad \sigma = \frac{2k-1}{3}.$$

Remark 7.2.
The regularity assumption $\xi \in H^{k-1}(\Omega)$ is satisfied, for example, if $\partial\Omega$ is smooth. ∎

Proof of Theorem 7.1. The proof follows the exposition of [3, Thm. 7.2.2]. From the Lax-Milgram Lemma (see for example [23, Thm. 2.7.7]) we have upon equipping the space $H^1(\Omega)$ with the norm $\|\cdot\|_\psi := \sqrt{a_\psi(\cdot,\cdot)}$, which is equivalent to the standard $\|\cdot\|_{H^1(\Omega)}$ norm,

$$\|u_\psi - u_N\|_\psi = \inf_{v \in V_N} \|u_\psi - v\|_\psi.$$

We now write

$$u = u_\psi + \frac{1}{\psi}\xi + \zeta.$$

The function ζ satisfies

$$\begin{aligned} a_\psi(\zeta, v) &= \underbrace{a(u,v)}_{=\int_\Omega fv\,dx + \int_{\partial\Omega} \partial_n uv\,ds} + \psi \underbrace{\int_{\partial\Omega} uv\,ds}_{=0} - \underbrace{a_\psi(u_\psi, v)}_{=\int_\Omega fv\,dx} - \frac{1}{\psi} a_\psi(\xi, v) \\ &= \int_{\partial\Omega} \partial_n uv\,ds - \frac{1}{\psi} a(\xi, v) - \underbrace{\int_{\partial\Omega} \xi v\,ds}_{=\int_{\partial\Omega} \partial_n uv\,ds} = -\frac{1}{\psi}\int_\Omega \nabla\xi\cdot\nabla v\,dx. \end{aligned}$$

Hence, the Lax-Milgram Lemma gives us

$$\|\zeta\|_\psi \le \frac{1}{\psi}\|\xi\|_{H^1(\Omega)}. \tag{7.14}$$

The function u_N is the Galerkin approximation to u_ψ, so we get $\|u_\psi - u_N\|_\psi = \inf_{v\in V_N}\|u_\psi - v\|_\psi$. Thus:

$$\|u_\psi - u_N\|_\psi = \inf_{v\in V_N}\|u_\psi - v\|_\psi \le \inf_{v\in V_N}\|u - v\|_\psi + \frac{1}{\psi}\inf_{v\in V_N}\|\xi - v\|_\psi + \|\zeta\|_\psi.$$

Using the bound $\|z\|^2_{L^2(\partial\Omega)} \le C\|z\|_{L^2(\Omega)}\|z\|_{H^1(\Omega)}$ (see for example Theorem A.2), we can bound with our assumptions on the approximation properties of V_N

$$\|u_\psi - u_N\|_\psi \le C\big\{h^{k-1} + \psi^{1/2}h^{k-1/2} + \psi^{-1/2}h^{k-3/2} + \psi^{-1}\big\}.$$

Choosing $\psi = h^{-\sigma}$ gives

$$\|u_\psi - u_N\|_\psi \le Ch^{\min\{\sigma, \sigma/2+k-3/2, -\sigma/2+k-1/2, k-1\}}.$$

The optimal rate of convergence is obtained for $\sigma = \frac{2k-1}{3}$. We get

$$\begin{aligned} \|u - u_N\|_{H^1(\Omega)} &\le \|u_\psi - u_N\|_{H^1(\Omega)} + \frac{1}{\psi}\|\xi\|_{H^1(\Omega)} + \|\zeta\|_{H^1(\Omega)} \\ &\le \|u_\psi - u_N\|_\psi + C\psi^{-1}, \end{aligned}$$

which gives the desired bound. □

Remark 7.3.
In the case $k = 2$, we see that the choice $\sigma = (2k-1)/3$ leads to the optimal rate of convergence. For $k > 2$, the penalty method leads to suboptimal rates. ∎

7.4 Non-Conforming Methods: Nitsche's Method

Nitsche's method was introduced in [88]; a good account that relates it to various forms of Lagrange Multiplier Methods can be found in [100]. Like the penalty method, Nitsche's method alters the variational formulation albeit in a more subtle way. For definiteness' sake, we consider again the model problem (7.1).

For simplicity, we will assume that the approximation space V_N satisfies $V_N \subset H^2(\Omega)$, although weaker assumptions suffice[3]. We need to identify the shape functions φ_i that are near the boundary. Hence, upon recalling the definition of patches, $\Omega_i = (\operatorname{supp} \varphi_i)^\circ$, we define

$$I_{\partial\Omega} := \{i \in \mathbb{N} \,|\, \Omega_i \cap \partial\Omega \neq \emptyset\}. \tag{7.15}$$

For $i \in I_{\partial\Omega}$ we set

$$\Gamma_i := \Omega_i \cap \partial\Omega, \qquad \widetilde{h}_i := \operatorname{diam} \Gamma_i. \tag{7.16}$$

For a penalty parameter $\gamma > 0$ define

$$a_N(u,v) := a(u,v) - \int_{\partial\Omega} \partial_n u v \, ds - \int_{\partial\Omega} u \partial_n v \, ds + \gamma \sum_{i \in I_{\partial\Omega}} \widetilde{h}_i^{-1} \int_{\Gamma_i} uv \, ds. \tag{7.17}$$

One variant of Nitsche's method can then be formulated as:

$$\text{Find} \quad u_N \in V_N \;\; \text{s.t.} \;\; a_N(u_N, v) = F(v) \qquad \forall\, v \in V_N. \tag{7.18}$$

In contrast to the penalty method, Nitsche's method is consistent if the exact solution is sufficiently regular:

Lemma 7.2 (consistency of Nitsche's method). *Let Ω be a Lipschitz domain. If for some $\varepsilon > 0$ the solution u of (7.1) satisfies $u \in H^{3/2+\varepsilon}(\Omega)$, then $a_N(u,v) = F(v)$ for all $v \in V_N$.*

Proof. By the trace theorem, the assumption $u \in H^{3/2+\varepsilon}(\Omega)$ guarantees that $\partial_n u$ is well-defined and $\partial_n u \in L^2(\partial\Omega)$. Since also the Gauss-Green theorem holds, the result now follows by inspection. □

The consistency result Lemma 7.2 will allow us to obtain quasi-optimality results in appropriate norms. In order to perform this analysis, we introduce a few discrete norms on the space $H^{3/2+\varepsilon}(\Omega)$:

$$\|u\|_{1/2,h}^2 := \sum_{i \in I_{\partial\Omega}} \widetilde{h}_i^{-1} \|u\|_{L^2(\Gamma_i)}^2, \tag{7.19}$$

$$\|\partial_n u\|_{-1/2,h}^2 := \sum_{i \in I_{\partial\Omega}} \widetilde{h}_i \|\partial_n u\|_{L^2(\Gamma_i)}^2, \tag{7.20}$$

$$\|u\|_{1,h}^2 := \|\nabla u\|_{L^2(\Omega)}^2 + \|u\|_{1/2,h}^2 + \|\partial_n u\|_{-1/2,h}^2. \tag{7.21}$$

[3] One has to be able to define the conormal derivative $\partial_n u$ for $u \in V_N$ as an element of $H^{-1/2}(\partial\Omega)$ in a meaningful way. In view of practical computations, one would like $\partial_n u \in L^2(\partial\Omega)$. For example, $V_N \subset H^s(\Omega)$ for some $s > 3/2$ suffices.

Central to the analysis of Nitsche's method is an inverse assumption:

Assumption 7.1 (inverse assumption). There exists $C_{\mathrm{inv}} > 0$ such that

$$\|\partial_n u\|_{-1/2,h} \le C_{\mathrm{inv}} \|\nabla u\|_{L^2(\Omega)} \qquad \forall\, u \in V_N.$$

In the case of the classical FEM, this inverse assumption can be proved:

Exercise 7.5.
Let $\mathcal{T}$ be a shape-regular triangulation of a polygon in $\mathbb{R}^2$. For the space of piecewise linears $S^{1,1}(\mathcal{T})$, let $\mathcal{E}_{\partial\Omega}$ be the set of edges that lie on $\partial\Omega$ and let h_e be the length of edge $e \in \mathcal{E}_{\partial\Omega}$. Show: There exists $C > 0$ depending solely on the shape-regularity constant of $\mathcal{T}$ such that upon setting

$$\|\partial_n u\|^2_{-1/2,h} := \sum_{e \in \mathcal{E}_{\partial\Omega}} h_e \|\partial_n u\|^2_{L^2(e)}$$

we have $\|\partial_n u\|_{-1/2,h} \le C_{\mathrm{inv}} \|\nabla u\|_{L^2(\Omega)}$ for all $u \in S^{1,1}(\mathcal{T})$ for some suitable $C_{\mathrm{inv}} > 0$. ∎

If the inverse Assumption 7.1 is satisfied, then the bilinear form a_N is coercive on V_N provided that the parameter γ is chosen sufficiently large:

Lemma 7.3. *If Assumption 7.1 is satisfied, then we have for* $\gamma > 2C^2_{\mathrm{inv}}$

$$\min\left\{\frac{1}{4}, \frac{1}{4C_{\mathrm{inv}}}, \gamma - 2C^2_{\mathrm{inv}}\right\} \|u\|^2_{1,h} \le a_N(u,u) \qquad \forall\, u \in V_N, \tag{7.22}$$

$$|a_N(u,v)| \le (1+\gamma)\|u\|_{1,h}\|v\|_{1,h} \qquad \forall\, u, v \in H^{3/2+\varepsilon}(\Omega). \tag{7.23}$$

Proof. Using the fact that $\partial\Omega \subset \cup_{i \in I_{\partial\Omega}} \Gamma_i$, we can estimate with the Cauchy-Schwarz inequality

$$\left| \int_{\partial\Omega} \partial_n u u \, ds \right| \le \|\partial_n u\|_{-1/2,h} \|u\|_{1/2,h}.$$

Using next the bound $2|ab| \le \epsilon a^2 + \frac{1}{\epsilon} b^2$, which is valid for all $\epsilon > 0$, we get

$$\begin{aligned} a_N(u,u) &\ge \|\nabla u\|^2_{L^2(\Omega)} - 2\|\partial_n u\|_{-1/2,h}\|u\|_{1/2,h} + \gamma \|u\|^2_{1/2,h} \\ &\ge \|\nabla u\|^2_{L^2(\Omega)} - \epsilon \|\partial_n u\|^2_{-1/2,h} - \epsilon^{-1}\|u\|^2_{1/2,h} + \gamma \|u\|^2_{1/2,h} \\ &\ge (1 - \epsilon C^2_{\mathrm{inv}}) \|\nabla u\|^2_{L^2(\Omega)} + (\gamma - \epsilon^{-1}) \|u\|^2_{1/2,h}, \end{aligned}$$

where we appealed to the inverse assumption. Choosing now $\epsilon = (2C^2_{\mathrm{inv}})^{-1}$ gives the desired bound (7.22).
The bound (7.23) follows from the trace theorem. □

Remark 7.4.
Lemma 7.3 shows that the problem (7.18) is well-defined and leads to a symmetric positive definite stiffness matrix, provided that the parameter γ is chosen sufficiently large. A good estimate on C_{inv} is required for that. Determining C_{inv} can be formulated as an eigenvalue problem, and a numerical scheme that works well has been proposed in [51, 96]. ∎

The consistency result Lemma 7.2 allows us to get quasi-optimality of the Nitsche method:

Lemma 7.4. *Set* $\underline{a} := \min\{\frac{1}{4}, \frac{1}{4C_{\text{inv}}}, \gamma - 2C_{\text{inv}}^2\}$. *Assume that the solution* u *of (7.1) satisfies* $u \in H^{3/2+\varepsilon}(\Omega)$ *for some* $\varepsilon > 0$. *Then*

$$\|u - u_N\|_{1,h} \leq \left(1 + \frac{1+\gamma}{\underline{a}}\right) \inf_{v \in V_N} \|u - v\|_{1,h}.$$

Proof. The proof is the same as the proof of Céa's lemma, for which we refer, for example, to [23, Thm. 2.8.1]. □

Theorem 7.2 (Convergence of Nitsche's method). *Let the solution* u *of (7.1) satisfy* $u \in H^k(\Omega)$ *for some* $k \geq 2$. *Assume:*

(a) the constant $\underline{a}$ *of Lemma 7.4 is positive;*
(b) the sets Γ_i, $i \in I_{\partial\Omega}$ *satisfy an overlap condition;*
(c) $h_i \sim h$ *for all* $i \in I_{\partial\Omega}$,
(d) $\inf_{v \in V_N} \|u - v\|_{L^2(\Omega)} + h\|u - v\|_{H^1(\Omega)} + h^2\|u - v\|_{H^2(\Omega)} \leq Ch^k\|u\|_{H^k(\Omega)}$.

Then

$$\|u - u_N\|_{H^1(\Omega)} \leq Ch^{k-1}.$$

Proof. By the quasi-optimality result Lemma 7.4 it suffices to bound the expression $\inf_{v \in V_N} \|u - v\|_{1,h}$. Using $h_i \sim h$ for all $i \in I_{\partial\Omega}$ and the overlap condition on the sets Γ_i gives us for arbitrary $v \in V_N$

$$\|u - v\|_{1,h}^2 \leq \|u - v\|_{H^1(\Omega)}^2 + Ch\|\partial_n(u - v)\|_{L^2(\partial\Omega)}^2 + h^{-1}\|\partial_n(u - v)\|_{L^2(\partial\Omega)}^2.$$

The trace Theorem A.2 applied to $z \in H^2(\Omega)$ gives in view of $\nabla z \in H^1(\Omega)$

$$\|u - v\|_{1,h}^2 \leq C\big\{\|u - v\|_{H^1(\Omega)}^2 + h\|u - v\|_{H^1(\Omega)}\|u - v\|_{H^2(\Omega)} + \frac{1}{h}\|u - v\|_{L^2(\Omega)}\|u - v\|_{H^1(\Omega)}\big\}.$$

The assumptions on the approximation properties of V_N allow us to conclude the argument. □

We required $k \geq 2$ in the proof of Theorem 7.2 for convenience only. The follow exercise shows that $k > 3/2$ is in fact sufficient:

Exercise 7.6.
Use Theorem A.2 to show that the approximation result of Theorem 7.2 is true for $k \in (3/2, 2)$ provided

$$\inf_{v \in V_N} \|u - v\|_{L^2(\Omega)} + h\|u - v\|_{H^1(\Omega)} + h^k\|u - v\|_{H^k(\Omega)} \leq Ch^k\|u\|_{H^k(\Omega)}.$$

■

Remark 7.5.
The approximation properties of V_N stated in Theorem 7.2 required simultaneous approximation properties of V_N in three norms. Such results were established in Theorem 2.1 and Proposition 3.3. ■

A Results from Analysis

Theorem A.1 (universal extension operator). *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. Then there exists a linear operator $E : L^1(\Omega) \to L^1(\mathbb{R}^d)$ with the following properties:*

(i) $(Eu)|_\Omega = u$ *for all* $u \in L^1(\Omega)$.
(ii) For each $k \in \mathbb{N}_0$, $p \in [1,\infty]$, *there exists* $C > 0$ *such that*

$$\|Eu\|_{W^{k,p}(\mathbb{R}^d)} \le C\|u\|_{W^{k,p}(\Omega)} \text{ for all } u \in W^{k,p}(\Omega).$$

Proof. See [99, Chap. VI.3]. □

Theorem A.2 (multiplicative trace theorem). *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain, $s \in (1/2, 1]$. Then there exists a constant $C > 0$ such that for all $u \in H^s(\Omega)$ the trace $\gamma_0 u = u|_{\partial\Omega}$ satisfies*

$$\|\gamma_0 u\|_{L^2(\partial\Omega)} \le C\|u\|_{L^2(\Omega)}^{1-1/(2s)} \|u\|_{H^s(\Omega)}^{1/(2s)}.$$

Proof. The case $s = 1$ is well-known (see for example [23, Prop. 1.6.3]). For the case $s \in (1/2, 1)$, a proof that is based on elementary techniques can be found in Exercise A.1. A short proof resting on the theory of interpolation spaces is as follows. From [104, Thm. 2.9.3], we can infer the trace theorem

$$\|\gamma_0 u\|_{L^2(\partial\Omega)} \le C\|u\|_{B^{1/2}_{2,1}(\Omega)}, \tag{A.1}$$

where the Besov space $B^{1/2}_{2,1}(\Omega) = (L^2(\Omega), H^1(\Omega))_{1/2,1}$; here, the K-method of interpolation, [15, 104] is employed. For $s \in (1/2, 1]$, the reiteration theorem then allows us to recognize $B^{1/2}_{2,1}$ as an interpolation space between $L^2(\Omega)$ and $H^s(\Omega)$, namely, $B^{1/2}_{2,1}(\Omega) = (L^2(\Omega), H^s(\Omega))_{\theta,1}$, where $\theta = 1/(2s)$. Inserting into (A.1) the interpolation inequality $\|u\|_{B^{1/2}_{2,1}(\Omega)} \le C_\theta \|u\|_{L^2(\Omega)}^{1-\theta} \|u\|_{H^s(\Omega)}^{\theta}$ then gives the desired result. □

Exercise A.1 (alternative proof of Theorem A.2).
The present exercise illustrates a very useful device of analysis, namely, how scaling arguments can lead to multiplicative bounds.
For simplicity, consider the case $\Omega = (0,1)^d$. Write $\Gamma := \mathbb{R}^{d-1} \times \{0\}$. Using the extension operator of Theorem A.1, we may assume $u \in H^s(\mathbb{R}^d)$. Proceed in several steps:

(a) Starting from the estimate $\|v\|_{L^2(\partial\Omega)} \le C\|v\|_{H^s(\Omega)}$ for all $v \in H^s(\Omega)$, show that

$$\|v\|_{L^2(\Gamma)} \le C\left[\|v\|_{L^2(\mathbb{R}^d)} + |v|_{H^s(\mathbb{R}^d)}\right] \qquad \forall\, v \in C_0^\infty(\mathbb{R}^d) \tag{A.2}$$

where we recall that $|\cdot|_{H^s(\mathbb{R}^d)}$ is defined as the Slobodeckij norm (1.1).

(b) By scaling (that is considering the function $\tilde{u}(x) := u(Rx)$) show that (A.2) has actually the form

$$\|v\|^2_{L^2(\Gamma)} \le C\left[R\|v\|^2_{L^2(\mathbb{R}^d)} + R^{1-2s}|v|^2_{H^s(\mathbb{R}^d)}\right] \qquad \forall\, v \in C_0^\infty(\mathbb{R}^d) \quad \text{(A.3)}$$

for *arbitrary* $R > 0$.

(c) Choose R in (A.3) suitably to obtain $\|v\|^2_{L^2(\Gamma)} \le C\|v\|^{2-1/s}_{L^2(\mathbb{R}^d)}|v|^{1/s}_{H^s(\mathbb{R}^d)}$. ∎

The following theorem shows that it is possible to cover arbitrary bounded sets by balls that satisfy a finite overlap property:

Theorem A.3 (Besicovitch covering theorem). *Let $d \in \mathbb{N}$. Then there exists a constant $M_d > 0$ (depending solely on d) with the following property: Let $\mathcal{B}$ be a collection of nondegenerate closed balls in $\mathbb{R}^d$ with*

$$\sup\{\operatorname{diam} B \mid B \in \mathcal{B}\} < \infty.$$

Let A be the set of centres of the balls of $\mathcal{B}$. Then there exist countable collections $\mathcal{B}_1, \dots, \mathcal{B}_{M_d} \subset \mathcal{B}$ such that each $\mathcal{B}_i$, $i = 1, \dots, M_d$, is a collection of disjoint *balls and*

$$A \subset \bigcup_{i=1}^{M_d} \bigcup_{B \in \mathcal{B}_i} B.$$

Proof. See, for example, [112, Thm. 1.3.5] or [40, Sec. 1.5.2]. □

B Properties of Polynomials

Theorem B.1 (polynomial approximation). *Let $B \subset \mathbb{R}^d$ be a ball of diameter $h \le 1$. Then for each polynomial degree $p \in \mathbb{N}_0$ there exists a linear operator $Q_p : L^1(B) \to \mathcal{P}_p$ with the following properties:*

$$Q_p u = u \qquad \forall\, u \in \mathcal{P}_p, \tag{B.1}$$

$$\|u - Q_p u\|_{W^{s,q}(B)} \le C_{p,q,k} h^{(\min\{p+1,k\}-s)_+} \|u\|_{W^{k,q}(B)}, \quad 0 \le s \le k. \tag{B.2}$$

Here, the notation $(\cdot)_+$ represents the function $x \mapsto (x)_+ = \max\{x, 0\}$. The constant $C_{p,q,k}$ depends only on $p \in \mathbb{N}_0$, $q \in [1, \infty)$, d, and $k \ge 0$. The bound (B.2) also holds for $q = \infty$ if k and s are restricted to integer values s, $k \in \mathbb{N}_0$.
If $q \in (1, \infty)$ and $k > d/q$ or if $q = 1$ and $k \ge d$, then additionally

$$\|u - Q_p u\|_{L^\infty(B)} \le \tilde{C}_{p,q,k} h^{\min\{p+1,k\}-d/q} \|u\|_{W^{k,q}(B)}, \tag{B.3}$$

where $\tilde{C}_{p,q,k}$ depends only on p, q, d, and k.

Proof. The L^∞-bound (B.3) will be treated in the following Exercise B.1. We elaborate the arguments of [23, Chap. 4] in order to show the statements (B.1), (B.2). We proceed in several steps.
First step: Let $F : B_1(0) \to B$ be an affine bijection. We define $u \mapsto Q_p u$ by $(Q_p u) \circ F := \widehat{Q}_p(u \circ F)$, where $\widehat{Q}_p : L^1(B_1(0)) \to \mathcal{P}_p$ is defined as in [23, Chap. 4]. From [23, Prop. 4.3.8 and Cor. 4.1.15] we have

$$\widehat{Q}_p u = u \qquad \forall\, u \in \mathcal{P}_p, \tag{B.4}$$

$$\|\widehat{Q}_p u\|_{W^{m,\infty}(B_1(0))} \le C_m \|u\|_{L^1(B_1(0))} \qquad \text{for any } m \in \mathbb{N}_0. \tag{B.5}$$

(B.4) implies (B.1). We therefore turn to the proof of (B.2). We set $\mu := \min\{p+1, k\}$, let $v \in \mathcal{P}_p$ be arbitrary, and calculate for $s \in [0, \mu]$ using (B.4) and the stability result (B.5)

$$\begin{aligned}
&\|u - \widehat{Q}_p u\|_{W^{s,q}(B_1(0))} \le \|u - \widehat{Q}_p u\|_{W^{\mu,q}(B_1(0))} \\
&\le \|u - v\|_{W^{\mu,q}(B_1(0))} + \|\widehat{Q}_p(u-v)\|_{W^{\mu,q}(B_1(0))} \\
&\le \|u - v\|_{W^{\mu,q}(B_1(0))} + C\|(u-v)\|_{L^1(B_1(0))} \le C\|u - v\|_{W^{\mu,q}(B_1(0))}.
\end{aligned} \tag{B.6}$$

Second step: In order to employ scaling arguments, we have to replace the full norm on the right-hand side of (B.6) by a semi-norm. The technique for doing this can be traced back to [21, 31] and is based on a compactness argument: From Rellich's theorem, [39, Chap. 5.7], we have that the embedding $W^{k,q}(B_1(0)) \subset\subset W^{k-1,q}(B_1(0))$ is compact for $k \in \mathbb{N}$; for $k = \tilde{k} + s$ with $\tilde{k} \in \mathbb{N}_0$ and $s \in (0,1)$ we have $W^{k,q}(B_1(0)) \subset\subset W^{\tilde{k},q}(B_1(0))$, [104, Sec. 1.16.4, Thm. 2]. Reasoning in the same way by contradiction as in the classical proof of the Poincaré inequality (see for example [39, Sec. 5.8.1]), we can infer for $p \in \mathbb{N}_0$ with $p \ge k-1$

$$\inf_{v \in \mathcal{P}_p} \|u - v\|_{W^{k,q}(B_1(0))} \le C|u|_{W^{k,q}(B_1(0))} \qquad \forall\, u \in W^{k,q}(B_1(0)). \tag{B.7}$$

Third step: Since $v \in \mathcal{P}_p$ in (B.6) is arbitrary and $\mu \le p+1$, we get for $s \in [0, \mu]$

$$\|u - \widehat{Q}_p u\|_{W^{s,q}(B_1(0))} \le C \inf_{v \in \mathcal{P}_p} \|u - v\|_{W^{\mu,q}(B_1(0))} \le C|u|_{W^{\mu,q}(B_1(0))}.$$

By transforming to B and observing how the semi-norms $|\cdot|_{W^{s,q}}$, $|\cdot|_{W^{\mu,q}}$ scale (cf. (1.1)) we obtain the desired bound (B.2) for $s \in [0, \mu]$.
Fourth step: It remains to see the bound for $\min\{p+1, k\} < s \le k$. This can only happen for $p+1 < k$. But then $p+1 < s$ and an easy calculation shows that $|Q_p|_{W^{s,q}(B)} = 0$. We conclude for the semi norm

$$|u - Q_p u|_{W^{s,q}(B)} \le |u|_{W^{s,q}(B)} + |Q_p u|_{W^{s,q}(B)} = |u|_{W^{s,q}(B)} \le C\|u\|_{W^{k,q}(B)}.$$

This allows us to obtain the desired bound (B.2) for the case $\min\{p+1, k\} < s \le k$. □

Exercise B.1.
Show (B.3) by proving the following two results.

(a) Show the following generalization of (B.7) for $p+1<k$ and $\Omega := B_1(0)$:

$$\inf_{v\in\mathcal{P}_p} \|u-v\|_{W^{k,q}(\Omega)} \le C|u|_{W^{p+1,q}(\Omega)} + \sum_{\substack{j\in\mathbb{N}\\ p+2\le j<k}} |u|_{W^{j,q}(\Omega)} + |u|_{W^{k,q}(\Omega)}.$$

(b) The parameter k in the statement of Theorem B.1 is such that the Sobolev embedding theorem $W^{k,q}(B_1(0)) \subset L^\infty(B_1(0))$ holds. By proceeding as in the proof of Theorem B.1 show the estimate (B.3). ∎

Theorem B.2 (polynomial inverse estimates). *Let $p \in \mathbb{N}_0$, $d \in \mathbb{N}$, $k \in \mathbb{N}$. Then there exists a constant $C > 0$ depending only on p, d, and there exists a constant C_k depending only on d, p, k such that for any ball $B \subset \mathbb{R}^d$ of radius $h \le 1$ there holds for all $\pi \in \mathcal{P}_p$:*

$$\|\pi\|_{L^\infty(B)} \le Ch^{-d/2}\|\pi\|_{L^2(B)},$$
$$\|\pi\|_{H^k(B)} \le C_k h^{-k}\|\pi\|_{L^2(B)}.$$

Proof. For $h = 1$ this estimate follows from the equivalence of norm of the finite dimensional space $\mathcal{P}_p$. The general case $h \ne 1$ follows by a scaling argument (see also [23, Lemma 4.5.3]).

Lemma B.1. *Let $B_1 \subset B_2 \subset \mathbb{R}^d$ be two balls of radius r_1, r_2, respectively. Then*

$$\|\pi\|_{L^\infty(B_2)} \le \left(\frac{2r_2}{r_1}\right)^p \|\pi\|_{L^\infty(B_1)} \qquad \forall\ \pi \in \mathcal{P}_p. \tag{B.8}$$

Proof. To show this, we employ the following one-dimensional Bernstein estimate for $r \ge 1$, [33, Chap. 4, Thm. 2.2]:

$$\|\pi\|_{L^\infty(-r,r)} \le r^p \|\pi\|_{L^\infty(-1,1)} \qquad \forall\ \pi \in \mathcal{P}_p. \tag{B.9}$$

Let $B_1 = B_{r_1}(x_1)$, $B_2 = B_{r_2}(x_2)$. Let $y \in B_{r_2}(x_2)\setminus\{x_1\}$ be arbitrary; let l be the line passing through the points y and x_1. Then the length of $l \cap B_1$ is $2r_1$ and the length of $l \cap B_{r_2}(x_2)$ is bounded by $\operatorname{diam} B_{r_2}(x_2)$. Since the restriction of π to l can be viewed as a univariate polynomial, the one-dimensional result (B.9) implies

$$|\pi(y)| \le \|\pi\|_{L^\infty(l)} \le \left(\frac{\operatorname{diam} B_{r_2}(x_2)}{r_1}\right)^p \|\pi\|_{L^\infty(l\cap B_1)} \le \left(\frac{2r_2}{r_1}\right)^p \|\pi\|_{L^\infty(B_1)}.$$

Since $y \in B_{r_2}(x_2)$ was arbitrary, the desired bound (B.8) follows. □

C Approximation with Adapted Function Systems

In this appendix, we prove Theorems 5.3, 5.4, and 5.5. These results are restricted to two-dimensional problems and make use of complex variables. We will identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$ where appropriate without explicit mention.

C.1 The Theory of Bergman and Vekua

We consider equations of the form

$$-\Delta u + a\partial_x u + b\partial_y u + cu = 0 \qquad \text{on } \Omega \subset \mathbb{R}^2, \tag{C.1}$$

where the constants a, b, c are real. The theory of S. Bergman [16] and I.N. Vekua [105] asserts the existence of a bijection between (suitably normalized) holomorphic functions and the solutions of (C.1). This bijection is even bicontinuous in Sobolev norms:

Lemma C.1. *Let $\Omega \subset \mathbb{C}$ be a simply connected Lipschitz domain. Fix $z_0 \in \Omega$. Let $\mathcal{H} := \{\varphi \,|\, \varphi$ holomorphic on Ω and $\varphi(z_0)$ real$\}$. Then there exists a linear map* ReV *with the following properties:*

1. $\mathrm{ReV}(\varphi)$ *solves (C.1) for every $\varphi \in \mathcal{H}$.*
2. *For every solution u of (C.1) there exists a unique $\varphi \in \mathcal{H}$ such that $\mathrm{ReV}(\varphi) = u$.*
3. $\|\mathrm{ReV}(\varphi)\|_{H^k(\Omega)} \le C\|\varphi\|_{H^k(\Omega)}$ *for all $\varphi \in \mathcal{H}$ and $k \ge 0$.*
4. *If $u \in H^k(\Omega)$, $k \ge 1$, solves (C.1), then the corresponding $\varphi = \mathrm{ReV}^{-1}(u) \in \mathcal{H}$ is likewise in $H^k(\Omega)$ and $\|\varphi\|_{H^k(\Omega)} \le C\|u\|_{H^k(\Omega)}$.*

In the last two estimates, the constant C depends on k, Ω, and the differential operator.

Proof. See [80]. Corresponding bicontinuity results in Hölder spaces have been obtained in [38]. □

Remark C.1.
The case of Laplace's equation is particularly simple. Then ReV reduces to the operator Re, that is taking the real part of a holomorphic function. Lemma C.1 can be generalized to the case of real analytic coefficients a, b, c; we refer to [80] and [16, 105] for the precise statements. ▪

An important observation is that the operator ReV can also be computed for Helmholtz's equation. For $z_0 = 0$ and writing (x, y) in polar coordinates, it is shown in [80] that

$$\mathrm{ReV}[z^n] = n! \left(\frac{2}{k}\right)^n \cos(n\varphi) J_n(kr), \tag{C.2a}$$

$$\mathrm{ReV}[\mathbf{i}z^n] = -n! \left(\frac{2}{k}\right)^n \sin(n\varphi) J_n(kr); \tag{C.2b}$$

here and in the remainder of this section (r, φ) denotes polar coordinates, that is $x = r\cos\varphi$, $y = r\sin\varphi$; the functions J_n are the first kind Bessel functions.

C.2 Proof of Theorems 5.3, 5.4

The approximation properties of the spaces $V(p)$ of (5.8) are proved in [80]. The purpose of the present section is to show how the approximation properties of $W(p)$ (see (5.7)) can be inferred from those of $V(p)$. To that end, we need to approximate the functions $e^{\mathbf{i}n\varphi}J_n(kr)$ from $W(p)$:

Lemma C.2. *Let the spaces $W(p)$ be defined by (5.7). Then there exists $C > 0$ independent of $n \in \mathbb{N}_0$ and $p \in \mathbb{N}$ and there exists, for each $n \in \mathbb{N}_0$, a function $v \in W(p)$ such that for all $R \geq 1$, $(x, y) \in \mathbb{R}^2$, $k \geq 0$ we have*

$$|e^{\mathbf{i}n\varphi}J_n(kr) - v(x,y)| \leq Ce^{nR}e^{ke^R(|x|+|y|)}e^{-pR/e},$$
$$|\nabla(e^{\mathbf{i}n\varphi}J_n(kr) - v(x,y))| \leq Ce^{nR}(1+ke^R)e^{ke^R(|x|+|y|)}e^{-pR/e}.$$

Proof. Given n and p, we will construct the function $v \in W(p)$ explicitly.
First step: We start by deriving an integral representation for $e^{\mathbf{i}n\varphi}J_n(kr)$. From [46, 8.411] we have for $z \in \mathbb{C}$ the integral representation

$$J_n(z) = \frac{1}{\pi}\int_{-\pi}^{\pi} e^{-n\mathbf{i}\theta + \mathbf{i}z\sin\theta}\,d\theta. \tag{C.3}$$

Next, we recall $x = r\cos\varphi$, $y = r\sin\varphi$, and we get using the periodicity of the integrand in (C.3)

$$\begin{aligned}\pi e^{\mathbf{i}n\pi/2}e^{\mathbf{i}n\varphi}J_n(kr) &= e^{\mathbf{i}n\pi/2}e^{\mathbf{i}n\varphi}\int_{-\pi}^{\pi} e^{-\mathbf{i}n(\theta+\varphi+\pi/2)+\mathbf{i}kr\sin(\theta+\varphi+\pi/2)}\,d\theta\\ &= \int_{-\pi}^{\pi} e^{-\mathbf{i}n\theta+\mathbf{i}kr\{\cos\theta\cos\varphi-\sin\theta\sin\varphi\}}\,d\theta = \int_{-\pi}^{\pi} e^{-\mathbf{i}n\theta+\mathbf{i}k\{x\cos\theta-y\sin\theta\}}\,d\theta\\ &= \int_{-\pi}^{\pi} e^{-\mathbf{i}n\theta+\mathbf{i}k\{x\cos\theta+y\sin\theta\}}\,d\theta. \end{aligned}\tag{C.4}$$

By differentiating under the integral sign with respect to x and y, we obtain a similar expression for the gradient of $e^{\mathbf{i}n\varphi}J_n(kr)$.
Second step: For $\rho > 0$ we define the strip $S_\rho := \{z \in \mathbb{C} \,|\, |\operatorname{Im} z| < \rho\}$. We claim that the Fourier coefficients g_ν of periodic functions g that are holomorphic on a strip S_R decay exponentially. For $\rho < R$ the expression $g_\rho := \sup_{z\in S_\rho}|g(z)|$ is finite and an m-fold integration by parts gives for $\nu \neq 0$

$$g_\nu = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-\mathbf{i}\nu\theta}g(\theta)\,d\theta = \frac{1}{2\pi}\left(\frac{1}{\mathbf{i}\nu}\right)^m\int_{-\pi}^{\pi} e^{-\mathbf{i}\nu\theta}g^{(m)}(\theta)\,d\theta.$$

Using the Cauchy integral representation formula we get for $\nu \neq 0$

$$|g_\nu| = \left| \frac{1}{2\pi} \frac{m!}{2\pi \mathbf{i}} \left(\frac{1}{\mathbf{i}\nu} \right)^m \oint_{|t|=\rho} \int_{-\pi}^{\pi} e^{\mathbf{i}\nu\theta} \frac{g(\theta+t)}{(t)^{m+1}} \, d\theta \, dt \right| \leq C \frac{m!}{(\rho|\nu|)^m} g_\rho.$$

The parameter $m \in \mathbb{N}_0$ is at our disposal. We choose it as $\lfloor |\nu|\rho/e \rfloor$ and get, using the generous bound $m! \leq m^m$,

$$\frac{m!}{(\rho|\nu|)^m} \leq \left(\frac{m}{\rho|\nu|} \right)^m \leq \left(\frac{|\nu|\rho/e}{|\nu|\rho} \right)^{|\nu|\rho/e - 1} = e e^{-|\nu|\rho/e}.$$

Thus, we arrive at

$$|g_\nu| \leq e e^{-\rho|\nu|/e} g_\rho \qquad \forall\, \nu \in \mathbb{Z}$$

and conclude

$$\sum_{|\nu| \geq p} |g_\nu| = \frac{2e}{1 - e^{-\rho/e}} e^{-p\rho/e} g_\rho. \tag{C.5}$$

Third step: For $p \in \mathbb{N}$ and $\theta_j := -\pi + \frac{2\pi}{p} j$, $j = 0, \ldots, p-1$, we denote by T_p the trapezoidal rule for integration on the interval $(-\pi, \pi)$, that is

$$T_p f := \frac{2\pi}{p} \sum_{j=0}^{p-1} f(\theta_j).$$

The rule T_p is exact for trigonometric polynomials of degree $p-1$, that is

$$T_p f = \int_{-\pi}^{\pi} f(\theta) \, d\theta \qquad \forall\, f \in \mathcal{T}_p := \operatorname{span}\{ e^{j\theta}, e^{-j\theta} \,|\, j = 0, \ldots, p-1 \}.$$

Hence, if the periodic function g has the Fourier representation $g(\theta) = \sum_{\nu \in \mathbb{Z}} g_\nu e^{\mathbf{i}\nu\theta}$, we can bound

$$\left| \int_{-\pi}^{\pi} g(\theta) \, d\theta - T_p g \right| \leq 4\pi \inf_{v \in \mathcal{T}_p} \|g - v\|_{L^\infty(-\pi,\pi)} \leq 4\pi \sum_{|\nu| \geq p} |g_\nu|. \tag{C.6}$$

Fourth step: We observe that an approximation of $e^{\mathbf{i}n\varphi} J_n(kr)$ from $W(p)$ can be obtained by applying the trapezoidal rule to the integral (C.4). We set

$$g(\theta) := \frac{1}{\pi} e^{-\mathbf{i}n\pi/2} e^{-\mathbf{i}n\theta + \mathbf{i}k\{x\cos\theta + y \sin\theta\}}$$

and note $v := T_p g \in W(p)$. It therefore remains to get bounds on the error $e^{\mathbf{i}n\varphi} J_n(kr) - v$. The function g is entire, and we can bound for any $R > 0$

$$\sup_{z \in S_R} |g(z)| \leq e^{nR} e^{k e^R (|x| + |y|)}. \tag{C.7}$$

Hence, we get by combining (C.5), (C.6), (C.7)

$$|e^{\mathbf{i}n\varphi}J_n(kr) - v| \le 4\pi \sum_{|\nu|\ge p} |g_\nu| \le Ce^{nR}e^{ke^R(|x|+|y|)}e^{-pR/e},$$

where the constant $C > 0$ is independent of n, p, $R \ge 1$, and x, y.
Fifth step: The bound for the gradient $\nabla(e^{\mathbf{i}n\varphi}J_n(kr)-v)$ is obtained similarly: By differentiating under the integral sign, we have the representation formula $\partial_x e^{\mathbf{i}n\varphi}J_n(kr) = \int_{-\pi}^{\pi} \partial_x g(\theta)\, d\theta$; by linearity of the operator T_p we have $\partial_x v = T_p(\partial_x g)$. Reasoning as above then gives the desired bound. □

Proof of Theorems 5.3 and 5.4. It only remains to prove the approximation properties of the space $W(p)$. We will only show Theorem 5.4 and leave the proof of Theorem 5.3 to the reader. Let Ω be star-shaped with respect to the ball $B_\rho(0)$. The real and imaginary parts $u_1 := \operatorname{Re} u$ and $u_2 := \operatorname{Im} u$ of the complex-valued solution u of the Helmholtz equation also solve the Helmholtz equation. Additionally, $\|u_1\|_{H^k(\Omega)} + \|u_2\|_{H^k(\Omega)} \le C\|u\|_{H^k(\Omega)}$.
From the approximation properties of $V(p)$ detailed in (5.10) and the observation (C.2) we have the existence of holomorphic polynomials $P_j \in \mathcal{H}_N$ of degree N such that

$$\|u_j - \operatorname{ReV} P_j\|_{H^1(\Omega)} \le C\left(\frac{\ln N}{N}\right)^{\lambda(k-1)}. \tag{C.8}$$

Lemma C.1 asserts that ReV is bicontinuous in Sobolev spaces, so we get

$$\|P_j\|_{H^1(\Omega)} \le C\|\operatorname{ReV} P_j\|_{H^1(\Omega)} \le C \tag{C.9}$$

for some $C > 0$ that is independent of N. We now approximate $\operatorname{ReV} P_j$ from $W(p)$. To that end, we write the polynomial P_j as $P_j(z) = \sum_{n=0}^{N} a_{n,j} z^n$. Cauchy's integral representation then gives

$$a_{n,j} = \frac{1}{2\pi\mathbf{i}} \oint_{|t|=\rho/2} \frac{P_j(t)}{(t)^{n+1}}\, dt.$$

The bound (C.9) and Lemma C.7 then imply

$$\|P_j\|_{L^\infty(B_{\rho/2}(0))} \le \frac{1}{\sqrt{\pi}\operatorname{dist}(\partial B_{\rho/2}(0), \partial B_\rho(0))}\|P_j\|_{L^2(B_\rho(0))} \le C$$

for some $C > 0$ independent of N. From this, we infer for the coefficients $a_{n,j}$ of the polynomial P_j

$$|a_{n,j}| \le C\frac{1}{(\rho/2)^n}\|P_j\|_{L^2(B_\rho(0))} \le C\frac{1}{(\rho/2)^n}.$$

In view of (C.2) and Lemma C.2, we can approximate for $p \geq N$

$$\begin{aligned}&\inf_{v\in W(p)} \|\operatorname{ReV} P_j - v\|_{H^1(\Omega)} \\ &\leq C\sum_{n=0}^{N} |a_{n,j}| n! \left(\frac{2}{k}\right)^n e^{nR}(1+ke^R)e^{ke^R \operatorname{diam}\Omega} e^{-pR/e}.\end{aligned}$$

Here, the constant $C > 0$ is independent of the parameters R and N, both of which we will now choose. We estimate

$$\sum_{n=0}^{N} |a_{n,j}| n! e^{nR} \left(\frac{2}{k}\right)^n \leq CN! e^{(\gamma+R)N}$$

for suitable C, $\gamma > 0$ independent of N, R. Choosing now (ignoring the complications do to rounding $p/\ln p$ to the nearest integer)

$$N = \frac{p}{\ln p} \tag{C.10}$$

we can bound $\ln N! \leq N \ln N = \frac{p}{\ln p} \ln(p/\ln p) \leq p$ to arrive at

$$\sum_{n=0}^{N} |a_{n,j}| n! e^{nR} \left(\frac{2}{n}\right)^n \leq Ce^{\gamma' p}$$

for some $C, \gamma' > 0$ independent of p and R. Hence, choosing $R > 0$ sufficiently large allows us to estimate

$$\inf_{v\in W(p)} \|\operatorname{ReV} P_j - v\|_{H^1(\Omega)} \leq Ce^{-bp}, \tag{C.11}$$

for some appropriate $b > 0$ independent of p. The triangle inequality

$$\|u_j - v\|_{H^1(\Omega)} \leq \|u_j - \operatorname{ReV} P_j\|_{H^1(\Omega)} + \|\operatorname{ReV} P_j - v\|_{H^1(\Omega)}$$

and making use of (C.8), (C.10), (C.11) allows us to conclude the proof. □

C.3 Two-Dimensional Elasticity

For complex-valued functions, we use the standard abbreviations $\partial_z = \frac{1}{2}(\partial_x - \mathbf{i}\partial_y)$, $\partial_{\overline{z}} = \frac{1}{2}(\partial_x + \mathbf{i}\partial_y)$. As discussed in (5.14), the displacement field (u, v) can be expressed on simply connected domains in terms of two holomorphic function φ, ψ. We can then check that

$$2\mu\partial_{\overline{z}}^m(u + \mathbf{i}v) = -z\overline{\varphi^{(m+1)}} - \overline{\psi^{(m)}}, \tag{C.12a}$$

$$\sigma_x + \sigma_y = 2\operatorname{Re}\varphi', \tag{C.12b}$$

$$2\mu\partial_z(u + \mathbf{i}v) = (\kappa+1)\operatorname{Re}\varphi' + \mathbf{i}(\kappa-1)\operatorname{Im}\varphi', \tag{C.12c}$$

where the stresses σ_x, σ_y are defined in Section 5.4. It will be convenient to combine the components of the displacement field (u, v) into the complex-valued function

$$\mathbf{u}(x, y) := u(x, y) + \mathbf{i}v(x, y).$$

The next lemma shows that the functions φ, ψ appearing in the representation formula (5.14) inherit regularity from the displacement field $\mathbf{u}$:

Lemma C.3. *Let $\Omega \subset \mathbb{R}^2$ be star-shaped with respect to a ball $B_\rho(z_0)$. Let the displacement field $\mathbf{u} = u + \mathbf{i}v \in H^k(\Omega)$ for some $k \in \mathbb{N}$. Let $z_0 \in \Omega$. Let φ, ψ be the holomorphic functions appearing in the representation formula (5.14), which are uniquely determined by stipulating $\varphi(z_0) = 0$. Then*

$$\|\varphi\|_{H^k(\Omega)} + \|\psi\|_{H^{k-1}(\Omega)} \leq C\|\mathbf{u}\|_{H^k(\Omega)},$$

where $C > 0$ depends only on the Lamé constants, upper bounds on $\operatorname{diam}\Omega$*, and lower bounds on ρ.*

Proof. We will only show the case $k = 1$ and leave the case $k > 1$ to the reader. Equation (C.12b) implies that $\operatorname{Re}\varphi' \in L^2(\Omega)$ with $\|\operatorname{Re}\varphi'\|_{L^2(\Omega)} \leq C\|\mathbf{u}\|_{H^1(\Omega)}$. Equation (C.12c) then shows that also $\operatorname{Im}\varphi' \in L^2(\Omega)$ with $\|\operatorname{Im}\varphi'\|_{L^2(\Omega)} \leq C\|\mathbf{u}\|_{H^1(\Omega)}$. The condition $\varphi(z_0) = 0$ then allows us to infer from Lemma C.8 that $\|\varphi\|_{L^2(\Omega)} \leq C\|\varphi'\|_{L^2(\Omega)}$ for a constant $C > 0$ that depends only on upper bounds on $\operatorname{diam}\Omega$ and lower bounds on ρ. Finally, we use once more the representation formula (5.14) to get the desired L^2 estimate for ψ. □

Lemma C.4. *Let $\Omega \subset \mathbb{C}$ be a domain and define for $\varepsilon > 0$ the set $\Omega_\varepsilon = \{z \in \Omega \,|\, B_\varepsilon(z) \subset \Omega\}$. If f, g are holomorphic on Ω and satisfy $f \in H^s(\Omega)$, $z\overline{f'} + \overline{g} \in H^s(\Omega)$ for some $s \in [0, 1]$, then*

$$\|z\overline{f'} + \overline{g}\|_{H^1(\Omega_\varepsilon)} \leq C\varepsilon^{s-1}\left\{\|f\|_{H^s(\Omega)} + \|z\overline{f'} + \overline{g}\|_{H^s(\Omega)}\right\}.$$

Proof. The case $s = 1$ is trivial and the case $s = 0$ is very similar to the case $s \in (0, 1)$. We have to bound the $L^2(\Omega_\varepsilon)$-norms of

$$\partial_z(z\overline{f'} + \overline{g}) = \overline{f'}, \qquad \partial_{\overline{z}}(z\overline{f'} + \overline{g}) = z\overline{f''} + \overline{g'}.$$

By an interior estimate for holomorphic functions, [80, Lemma 2.4], we have for each $s' \in [0, 1]$ a constant $C_{s'} > 0$ such that for all $f \in H^{s'}(\Omega)$ that are holomorphic on Ω

$$\|f'\|_{L^2(\Omega_\varepsilon)} \leq C\varepsilon^{s'-1}|f|_{H^{s'}(\Omega)}. \tag{C.13}$$

For the bound on $z\overline{f''} + \overline{g'}$ we use Cauchy's integral representation formula to get for $z \in \Omega_\varepsilon$

$$\begin{aligned}\overline{z\overline{f''} + \overline{g'}} = {} & \frac{1}{2\pi\mathbf{i}} \oint_{|t-z|=\varepsilon} \frac{(\overline{z} - \overline{t})f'(t)}{(t-z)^2}\, dt \\ & + \frac{1}{2\pi\mathbf{i}} \oint_{|t-z|=\varepsilon} \frac{\overline{t}f'(t) + g(t) - (\overline{z}f'(z) + g(z))}{(t-z)^2}\, dt.\end{aligned} \tag{C.14}$$

For the second term, we used additionally $\oint_{|z-t|=\varepsilon} \frac{1}{(z-t)^2}\,dt = 0$. For the first integral in (C.14), we observe that $|t-z| = \varepsilon$ implies $\overline{z}-\overline{t} = \frac{\varepsilon^2}{z-t}$ and recognize the first integral to be

$$\frac{1}{2\pi\mathbf{i}} \oint_{|t-z|=\varepsilon} \frac{(\overline{z}-\overline{t})f'(t)}{(z-t)^2}\,dt = \frac{\varepsilon^2}{2!}\frac{2!}{2\pi\mathbf{i}} \oint_{|t-z|=\varepsilon} \frac{f'(t)}{(t-z)^3}\,dt = \frac{\varepsilon^2}{2!} f'''(z).$$

Together with bounds on the second integral, we arrive at

$$\left|\overline{z\overline{f''} + \overline{g'}}\right|^2 \le C\varepsilon^4 |f'''(z)|^2 + C\varepsilon^{+2s} \sup_{t\in\partial B_\varepsilon(z)} \frac{|\overline{t}f'(t) + g(t) - (\overline{z}f'(z) + g(z)|^2}{|z-t|^{2+2s}}.$$

Upon integrating in $z \in \Omega_\varepsilon$, we can bound $\varepsilon^2 \|f'''\|_{L^2(\Omega_\varepsilon)} \le C\varepsilon^{s-1}|f|_{H^s(\Omega)}$ if we note $\Omega_\varepsilon \subset \Omega_{\varepsilon/3} \subset \Omega_{2\varepsilon/3} \subset \Omega$ and use (C.13) repeatedly, namely, twice with $s' = 0$ and once with $s' = s$. For the second term involving the supremum, we use the interior estimate (C.22) to bound the supremum and then integrate in the z-variable to obtain the desired result. □

Lemma C.5. *Let Ω be star-shaped with respect to the ball $B_\rho(0)$. Let $m \in \mathbb{N}$, $s \in [0,1)$. Let the displacement field (u,v) be in $H^{m+s}(\Omega)$. Define the function*

$$g(t) := 2\mu\left(u((1-t)z) + \mathbf{i}v((1-t)z)\right).$$

Then for $t \in (0,1/2)$

$$\|g^{(m+1)}(t)\|_{L^2(\Omega)} + \|g^{(m)}(t)\|_{H^1(\Omega)} \le Ct^{-(1-s)}\|\mathbf{u}\|_{H^{m+s}(\Omega)}. \tag{C.15}$$

Proof. We will only show the bound on $g^{(m)}$, the other one being handled similarly. Using the representation formula (5.14) for $\mathbf{u} = u + \mathbf{i}v$, we write

$$\begin{aligned}
g(t) &= -(1-t)z\overline{\varphi'((1-t)z)} - \overline{\psi((1-t)z))} + \kappa\varphi((1-t)z),\\
g^{(m)}(t) &= \left[-(1-t)z\overline{\varphi^{(m+1)}((1-t)z)} - \overline{\psi^{(m)}((1-t)z))}\right]\overline{(-z)^m}\\
&\quad + mz\overline{(-z)^{m-1}\varphi^{(m)}((1-t)z)} + \kappa(-z)^m\varphi^{(m)}((1-t)z),\\
\partial_z g^{(m)}(t) &= -(1-t)\overline{(-z)^m\varphi^{(m+1)}((1-t)z)} + m\overline{(-z)^{m-1}\varphi^{(m)}((1-t)z)}\\
&\quad + \kappa\frac{d}{dz}\left[(-z)^m\varphi^{(m)}((1-t)z)\right],\\
\partial_{\overline{z}} g^{(m)}(t) &= (1-t)\left[-(1-t)z\overline{\varphi^{(m+2)}((1-t)z)} - \overline{\psi^{(m+1)}((1-t)z)}\right]\overline{(-z)^m}\\
&\quad - m\left[-(1-t)z\overline{\varphi^{(m+1)}((1-t)z)} - \overline{\psi^{(m)}((1-t)z)}\right]\overline{(-z)^{m-1}}\\
&\quad + mz\overline{\frac{d}{dz}\left[(-z)^{m-1}\varphi^{(m)}((1-t)z)\right]}.
\end{aligned}$$

The estimate (C.15) follows from the change of variables $\zeta = (1-t)z$, the observations (C.12), and Lemma C.3. An additional ingredient to the proof is the fact that there exists $C > 0$ such that $B_{Ct}(z) \subset \Omega$ for all $z \in (1-t)\Omega$ so that Lemma C.4 can be employed. □

Lemma C.6. *Assume the hypotheses of Lemma C.5. Let T_m be the Taylor polynomial of g about the point $t_0 = \varepsilon$ that is evaluated at $t = 0$, that is*

$$T_m = \sum_{\nu=0}^{m} g^{(\nu)}(\varepsilon)\frac{(-\varepsilon)^\nu}{\nu!}.$$

Then T_m is defined on $\frac{1}{1-\varepsilon}\Omega$ and

$$\|T_m\|_{L^2(\frac{1}{1-\varepsilon}\Omega)} \le C\|\mathbf{u}\|_{H^m(\Omega)}, \tag{C.16}$$

$$\|T_m\|_{H^1(\frac{1}{1-\varepsilon/2}\Omega)} \le C\varepsilon^{-1}\|\mathbf{u}\|_{H^m(\Omega)}, \tag{C.17}$$

$$\|g(0) - T_m\|_{L^2(\Omega)} + \varepsilon\|g(0) - T_m\|_{H^1(\Omega)} \le C\varepsilon^{m+s}\|\mathbf{u}\|_{H^{m+s}(\Omega)}. \tag{C.18}$$

Proof. The bound (C.16) follows from the change of variables $\zeta = (1-\varepsilon)z$, an inspection of the definition of the terms $g^{(j)}$, $j = 0, \dots, m$, equation (C.12), and Lemma C.3. The proof of (C.17) follows along the same lines. Estimating $\partial_z g^{(m)}(t)$, however, requires additionally to use Lemma C.4 and the observation that $\frac{1}{1-\varepsilon/2}\Omega \subset \{z \in \frac{1}{1-\varepsilon}\Omega \,|\, B_{\varepsilon'}(z) \subset \frac{1}{1-\varepsilon}\Omega\}$ for some $\varepsilon' \sim \varepsilon$. In the bound (C.18), we will only show the $H^1(\Omega)$-estimate. We will also exclude the case $m = 1$, $s = 0$, which we leave to the reader. We choose $\delta \in (0, 1/2)$ such that $2(m-1) - 2(1-s) + 2\delta > 0$ and recall the Taylor formula

$$g(0) - T_m = -\frac{1}{m!}(-\varepsilon)^m g^{(m)}(\varepsilon) - \frac{1}{(m-1)!}\int_\varepsilon^0 g^{(m)}(t)(-t)^{m-1}\,dt.$$

The first term can be bounded by $\varepsilon^{m+s-1}\left[\|u\|_{H^{m+s}(\Omega)} + \|v\|_{H^{m+s}(\Omega)}\right]$ by Lemma C.5. For the integral, we estimate

$$\left\|\int_\varepsilon^0 g^{(m)}(t)t^{m-1}\,dt\right\|^2_{H^1(\Omega)} \le \int_0^\varepsilon \|g^{(m)}(t)\|^2_{H^1(\Omega)} t^{2(1-s-\delta)}dt \int_0^\varepsilon |t^{-(1-s)+\delta+m-1}|^2 dt,$$

which can again be estimated in the desired fashion using Lemma C.5. □

Proof of Theorem 5.5. Without loss of generality, we assume that Ω is star-shaped with respect to the ball $B_\rho(0)$. For a parameter $\varepsilon > 0$ sufficiently small, which will be chosen below in dependence on the polynomial degree p, we define g and T_m as in Lemmas C.5, C.6. Then T_m is defined on $\frac{1}{1-\varepsilon}\Omega$ and, since $g(0) = \mathbf{u}$, we get from Lemma C.6

$$\|\mathbf{u} - T_m\|_{H^j(\Omega)} \le C\varepsilon^{m+s-j}\|\mathbf{u}\|_{H^{m+s}(\Omega)}, \qquad j = 0, 1. \tag{C.19}$$

From the representation formulae for the $g^{(j)}$, $j = 0, \dots, m$, in the proof of Lemma C.5, we observe that T_m has the form $T_m = \kappa\varphi_1 - z\overline{\varphi_1'} - \overline{\psi_1}$, where φ_1, ψ_1 are functions holomorphic on $\frac{1}{1-\varepsilon}\Omega$ and $\varphi_1(0) = \varphi(0) = 0$. Lemma C.3

(together with the observation that the constant appearing in Lemma C.3 can be made independent of $\varepsilon \in (0,1/2)$) and Lemma C.6 then imply

$$\begin{aligned}\|\varphi_1\|_{H^1(\frac{1}{1-\varepsilon/2}\Omega)} + \|\psi_1\|_{L^2(\frac{1}{1-\varepsilon/2}\Omega)} &\le C\|T_m\|_{H^1(\frac{1}{1-\varepsilon/2}\Omega)} \\ &\le C\varepsilon^{-1}\|\mathbf{u}\|_{H^m(\Omega)}. \end{aligned} \tag{C.20}$$

Since φ_1, ψ_1 are holomorphic on $\frac{1}{1-\varepsilon}\Omega$, they can be approximated on Ω by (complex) polynomials at an exponential rate. Namely, by Szegö's approximation result (see [80, Thm. 2.6]) there exist complex polynomials φ_{ap}, $\psi_{ap} \in \mathcal{H}_p$ of degree p such that

$$\|\varphi_1 - \varphi_{ap}\|_{W^{j,\infty}(\Omega)} \le Ch^{-\alpha}(1+h)^{-p}\|\varphi_1\|_{L^2(\mathrm{Int}(L_4h))}, \quad j = 0,1,2, \tag{C.21a}$$

$$\|\psi_1 - \psi_{ap}\|_{W^{j,\infty}(\Omega)} \le Ch^{-\alpha}(1+h)^{-p}\|\varphi_1\|_{L^2(\mathrm{Int}(L_4h))}, \quad j = 0,1,2; \tag{C.21b}$$

here, $L_h = \{\varphi_\Omega(z) \,|\, |z| = 1+h\}$, where $\varphi_\Omega : \mathbb{C}\setminus B_1(0) \to \mathbb{C}\setminus\Omega$ is the unique conformal map with $\varphi_\Omega(\infty) = \infty$ and $\varphi_\Omega'(\infty) > 0$. The constants $C, \alpha > 0$ are independent of h and p. By geometric considerations (see [80, Lemma 2.3]), we can ascertain the existence of $D > 0$ such that for $h^{\hat\lambda} = D\varepsilon$ we have $\mathrm{Int}\, L_{4h} \subset \frac{1}{1-\varepsilon/2}\Omega$. Hence, combining (C.19), (C.20), (C.21), we can conclude for $j \in \{0,1\}$

$$\begin{aligned}&\|\mathbf{u} - (-z\overline{\varphi_{ap}'} - \overline{\psi_{ap}} + \kappa\varphi_{ap})\|_{H^j(\Omega)} \\ &\quad\le C\varepsilon^{m+s-j}\|\mathbf{u}\|_{H^{m+s}(\Omega)} + \varepsilon^{-\hat\lambda\alpha}(1+(D\varepsilon)^{1/\hat\lambda})^{-p}\varepsilon^{-1}\|\mathbf{u}\|_{H^m(\Omega)}.\end{aligned}$$

Choosing

$$\varepsilon = K\left(\frac{\ln(p+2)}{p+2}\right)^{\hat\lambda}$$

for sufficiently large K gives the desired bound stated in Theorem 5.5. □

Lemma C.7 (interior estimates for holomorphic functions). *Let $\Omega \subset \mathbb{C}$ be a domain. Define for $\varepsilon > 0$ the set $\Omega_\varepsilon := \{z \in \Omega \,|\, B_\varepsilon(z) \subset \Omega\}$. Then for any function f that is holomorphic on Ω*

$$\|f\|_{L^\infty(\Omega_\varepsilon)} \le \frac{1}{\sqrt{\pi}\varepsilon}\|f\|_{L^2(\Omega)}. \tag{C.22}$$

Proof. The proof can be found, for example, in [76]. For the reader's convenience, we reproduce it here: For fixed $z \in \Omega_\varepsilon$ we use Cauchy's integral representation theorem to write for any $r \in (0,\varepsilon)$

$$|f(z)| = \left|\frac{1}{2\pi\mathbf{i}}\oint_{|t|=r}\frac{f(z+t)}{-t}\,dt\right| = \frac{1}{2\pi}\left|\int_{\partial B_1(0)} f(z+rt)\,|dt|\right|.$$

Multiplying this equality by r and integrating over r from 0 to ε gives, if we note that the right-hand side integral is then an area integral in polar

coordinates,

$$\frac{1}{2}\varepsilon^2|f(z)| = \int_0^\varepsilon r|f(z)|\,dr = \frac{1}{2\pi}\int_0^\varepsilon \left|\int_{\partial B_1(0)} f(z+rt)\,|dt|\right| r\,dr$$
$$\leq \frac{\varepsilon}{2\sqrt{\pi}}\left(\int_0^\varepsilon \int_{\partial B_1(0)} |f(z+rt)|^2\,|dt|r\,dr\right)^{1/2} = \frac{\varepsilon}{2\sqrt{\pi}}\|f\|_{L^2(B_\varepsilon(z))}.$$

Since $z \in \Omega_\varepsilon$ was arbitrary, the proof is complete. □

Lemma C.8. *Let $\Omega \subset \mathbb{C}^2$ be star-shaped with respect to 0 and assume that $B_\rho(0) \subset \Omega$. Then for $f \in H^1(\Omega)$ holomorphic on Ω we have*

$$\|f - f(0)\|_{L^2(\Omega)} \leq \sqrt{2}\operatorname{diam}\Omega\left[\frac{1}{\pi} + \left(\frac{2\operatorname{diam}\Omega}{\rho}\right)^2\right]^{1/2}\|f'\|_{L^2(\Omega)}. \quad \text{(C.23)}$$

Proof. We define $\delta := \rho/(2\operatorname{diam}\Omega) < 1$. Since Ω is star-shaped with respect to 0, we can write for $z \in \Omega$ by integrating on the line connecting 0 and z

$$f(z) - f(0) = \int_{t=0}^1 zf'(tz)\,dt = \int_{t=0}^\delta zf'(tz)\,dt + \int_{t=\delta}^1 zf'(tz)\,dt.$$

For the first integral, we note that $t \in (0,\delta)$ and $z \in \Omega$ implies $|tz| \leq \rho/2$. Hence, Lemma C.7 implies

$$\left|\int_{t=0}^\delta zf'(tz)\,dt\right| \leq \frac{\delta\operatorname{diam}\Omega}{\sqrt{\pi}\rho/2}\|f'\|_{L^2(B_\rho(0))} \leq \frac{1}{\sqrt{\pi}}\|f'\|_{L^2(\Omega)}.$$

Thus,

$$\|f - f(0)\|^2_{L^2(\Omega)} \leq 2\frac{\operatorname{area}(\Omega)}{\pi}\|f'\|^2_{L^2(\Omega)} + 2\int_\Omega \left|\int_{t=\delta}^1 zf'(tz)\,dt\right|^2.$$

The second term is treated as follows: First, the Cauchy-Schwarz inequality is applied to the inner integral; then the order of integration is switched, and finally a change of variables $\zeta := tz$ is performed. This leads to

$$\int_\Omega \left|\int_{t=\delta}^1 zf'(tz)\,dt\right|^2 \leq \left(\frac{\operatorname{diam}\Omega}{\delta}\right)^2\|f'\|^2_{L^2(\Omega)}.$$

Combining the above estimates leads to (C.23). □

References

1. M. Armentano. Error estimates in Sobolev spaces for moving least square approximations. *SIAM J. Numer. Anal.*, 39:38–51, 2001.
2. S.N. Atluri and S. Shen. The meshless local Petrov-Galerkin (MLPG) method: a simple & less-costly alternative to the finite element and boundary element methods. *CMES Comput. Model. Eng. Sci.*, 3(1):11–51, 2002.
3. A.K. Aziz and I.M. Babuška, editors. *Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations.* Academic Press, New York, 1972.
4. I. Babuška. The finite element method with Lagrange multipliers. *Numer. Math.*, 20:179–192, 1973.
5. I. Babuška, B. Andersson, P.J. Smith, and K. Levin. Damage analysis of fiber composites. I: Statistical analysis of fiber scale. *Comput. Meth. Appl. Mech. Engrg.*, 172:27–77, 1999.
6. I. Babuška, U. Banerjee, and J. Osborn. Survey of meshless and generalized finite element methods: a unified approach. In *Acta Numerica 2003*, pages 1–125. Cambridge University Press, 2003.
7. I. Babuška, G. Caloz, and J. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31:945–981, 1994.
8. I. Babuška, R.B. Kellogg, and J. Pitkäranta. Direct and inverse error estimates for finite elements with mesh refinements. *Numer. Math.*, 33:447–471, 1979.
9. I. Babuška and J. M. Melenk. The partition of unity method. *Internat. J. Numer. Meths. Engrg.*, 40:727–758, 1997.
10. I. Babuška and J. Osborn. Can a finite element method perform arbitrarily badly? *Math. Comput.*, 69:443–462, 2000.
11. T. Belytschko, L. Gu, and Y.Y. Lu. Fracture and crack growth by element-free Galerkin methods. *Modelling Simul. Mater. Sci. Eng.*, 2:519–534, 1994.
12. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Comput. Meth. Appl. Mech. Engrg.*, 139:3–47, 1996.
13. T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *Internat. J. Numer. Meths. Engrg.*, 37:229–256, 1994.
14. T. Belytschko, Y.Y. Lu, and L. Gu. A new implementation of the element-free Galerkin method. *Comput. Meth. Appl. Mech. Engrg.*, 113:397–414, 1994.
15. J. Bergh and J. Löfström. *Interpolation Spaces.* Springer Verlag, 1976.
16. S. Bergman. *Integral operators in the theory of linear partial differential equations.* Springer Verlag, 1961.
17. S. Bergman and J. Herriot. Application of the method of the kernel function for solving boundary value problems. *Numer. Math.*, 3:209–225, 1961.
18. S. Bergman and J. Herriot. Numerical solution of boundary-value problems by the method of integral operators. *Numer. Math.*, 7:42–65, 1965.
19. T. Betcke and N.L. Trefethen. Reviving the method of particular solutions. *SIAM Review*, to appear.
20. H. Blum and M-Dobrowolski. On finite element methods for elliptic equations on domains with corners. *Computing*, 28:53–61, 1982.
21. J. Bramble and S.R. Hilbert. Estimation of linear functionals on sobolev spaces with application to fourier transforms and spline interpolation. *SIAM J. Numer. Anal.*, 7:112–124, 1970.

22. J. Bramble and R. Scott. Simultaneous approximation in scales of Banach spaces. *Math. Comput.*, 32:947–954, 1978.
23. S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods.* Springer Verlag, 1994.
24. Rob Brownlee and Will Light. Approximation orders for interpolation by surface splines to rough functions. *IMA J. Numer. Anal.*, 24(2):179–192, 2004.
25. M. Buhmann. Radial basis functions. In *Acta Numerica 2000*, pages 1–38. Cambridge University Press, 2000.
26. M. D. Buhmann. *Radial basis functions: theory and implementations*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 2003.
27. P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland Publishing Company, 1976.
28. D. Colton. Bergman operators for elliptic equations in three independent variables. *Bull. Amer. Math. Soc.*, 77(5):752–756, 1971.
29. C. Daux, N. Moës, J. Dolbrow, N. Sukumar, and T. Belytschko. Arbitrary cracks and holes with the extended finite element method. *Internat. J. Numer. Meths. Engrg.*, 48(12):1741–1760, 2000.
30. S. De and K.J. Bathe. The method of finite spheres. *Computational Mechanics*, 25:329–345, 2000.
31. J. Deny and J.L. Lions. Les espaces du type de Beppo Levi. *Ann. Inst. Fourier, Grenoble*, 5:305–370, 1955.
32. R.A. DeVore. Nonlinear approximation. In *Acta Numerica 1998*, pages 51–150. Cambridge University Press, 1998.
33. R.A. DeVore and G.G. Lorentz. *Constructive Approximation.* Springer Verlag, 1993.
34. C. A. Duarte, I. Babuška, and J. T. Oden. Generalized finite element methods for three-dimensional structural mechanics problems. *Comput. & Structures*, 77(2):215–232, 2000.
35. J. Duchon. Splines minimizing rotation-invariant seminorms in Sobolev norms. In W. Schempp and K. Zeller, editors, *Constructived Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics*, pages 85–100. Springer Verlag, 1976.
36. J. Duchon. Sur l'erreur d'interpolation des fonctions de plusieurs variables par les D^m-splines. *RAIRO Anal. Numérique*, 12(4):325–334, 1978.
37. Y. Efendiev, T. Hou, and X.-H. Wu. Convergence of a nonconforming multiscale finite element method. *SIAM J. Numer. Anal.*, 37(3):888–910, 2000.
38. Stanley C. Eisenstat. On the rate of convergence of the Bergman-Vekua method for the numerical solution of elliptic boundary value problems. *SIAM J. Numer. Anal.*, 11:654–680, 1974.
39. L.C. Evans. *Partial Differential Equations.* American Mathematical Society, 1998.
40. L.C. Evans and R.F. Gariepy. *Measure Theory and Fine Properties of Functions.* CRC Press, 1992.
41. S. Fernández-Méndez, P. Díez, and A. Huerta. Convergence of finite elements enriched with meshless methods. *Numer. Math.*, 96:43–59, 2003.
42. G. Fix, S. Gulati, and G. I. Wakoff. On the use of singular functions with the finite element method. *J. Comput. Phys.*, 13:209–228, 1973.
43. L. Fox, P. Henrici, and C. Moler. Approximations and bounds for eigenvalues of elliptic operators. *SIAM J. Numer. Anal.*, 4:89–102, 1967.

44. C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. Comp. Math.*, 8:381–399, 1998.
45. R. Franke. Scattered data interpolation: test of some methods. *Math. Comput.*, 38:181–200, 1982.
46. I.S. Gradshteyn and I.M. Ryzhik. *Table of Integrals, Series, and Products, corrected and enlarged edition.* Academic Press, New York, 1980.
47. M. Griebel and M.A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic pdes. *SIAM J. Sci. Stat. Comp.*, 22:853–890, 2000.
48. M. Griebel and M.A. Schweitzer. A particle-partition of unity method—part II: Efficient cover construction and reliable integration. *SIAM J. Sci. Stat. Comp.*, 23:1655–1682, 2002.
49. M. Griebel and M.A. Schweitzer. A particle-partition of unity method—part III: A multilevel solver. *SIAM J. Sci. Stat. Comp.*, 24(2):377–409, 2002.
50. M. Griebel and M.A. Schweitzer. A particle-partition of unity method—part IV: Parallelization. In M. Griebel and M.A. Schweitzer, editors, *Meshfree Methods for Partial Differential Equations*, volume 26 of *Lecture Notes in Computational Science and Engineering*, pages 161–192. Springer, 2002.
51. M. Griebel and M.A. Schweitzer. A particle-partition of ynity method—part V: Boundary conditions. In S. Hildebrandt and H. Karcher, editors, *Geometric Analysis and Nonlinear Partial Differential Equations*, pages 517–540. Springer, 2002.
52. P. Grisvard. *Elliptic Problems in Nonsmooth Domains.* Pitman, 1985.
53. P. Grisvard. *Singularities in Boundary Value Problems.* Springer Verlag/Masson, 1992.
54. D. Hagen. Element-free Galerkin methods in combination with finite element approaches. *Comput. Meth. Appl. Mech. Engrg.*, 139:237–262, 1996.
55. W. Han and X. Meng. error analysis of the reproducing kernel particle method. *Comput. Meth. Appl. Mech. Engrg.*, 190:6157–6181, 2001.
56. I. Herrera. *Boundary Methods: An Algebraic Theory.* Pitman, Boston, 1984.
57. K. Höllig, U. Reif, and J. Wipper. Weighted extended b-spline approximation of Dirichlet problems. *SIAM J. Numer. Anal.*, 39(2):442–462, 2001.
58. T. Hou. Numerical approximations to multiscale solutions in partial differential equations. In J. Blowey, A. Craig, and T. Shardlow, editors, *Frontiers in numerical analysis (Durham, 2002)*, pages 241–301. Springer, 2003.
59. T. Hou, X.-H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Math. Comput.*, 68(227):913–943, 1999.
60. A. Huerta, T. Belytschko, S. Fernández-Méndez, and T. Rabczuk. Meshfree methods. In R. de Borst, T.J.R. Hughes, and E. Stein, editors, *Encyclopedia of Computational Mechanics.* Elsevier, to appear.
61. A. Iske. *Multiresolution Methods in Scattered Data Modelling.* Number 37 in Lecture Notes in Computational Science and Engineering. Springer Verlag, 2004.
62. J.W. Jerome. On n-widths in Sobolev spaces and applications to elliptic boundary value problems. *Journal of Mathematical Analysis and Applications*, 29:201–215, 1970.
63. J. Jirousek and A. Venkatesh. Hybrid-Trefftz plane elasticity elements with p-method capabilities. *Internat. J. Numer. Meths. Engrg.*, 35:1443–1472, 1992.

64. J. Jirousek and A.P. Zielinski. Survey of Trefftz-type element formulations. *Computers and Structures*, 63(2):225–242, 1997.
65. E.J. Kansa. Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics–I surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, 19(8/9):127–145, 1990.
66. E.J. Kansa. Multiquadrics–a scattered data approximation scheme with applications to computational fluid-dynamics–II solutions to parabolic, hyperbolic, and elliptic partial differential equations. *Computers and Mathematics with Applications*, 19(8/9):147–161, 1990.
67. I.V. Kantorovich and V.I. Krylov. *Approximate Methods of Higher Analysis*. Interscience Publishers, 1958.
68. Y. Krongaus and T. Belytschko. Enforcement of essential boundary conditions in meshless approximation using finite elments. *Comput. Meth. Appl. Mech. Engrg.*, 131:133–145, 1996.
69. O. Laghrouche and P. Bettes. Solving short wave problems using special finite elements; towards an adaptive approach. In J. Whiteman, editor, *Mathematics of Finite Elements and Applications X*, pages 181–195. Elsevier, 2000.
70. S. Li, H. Lu, W. Han, W. K. Liu, and D. C. Simkins. Reproducing kernel element method. II. Globally conforming I^m/C^n hierarchies. *Comput. Methods Appl. Mech. Engrg.*, 193(12-14):953–987, 2004.
71. T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures*, 11:83–95, 1980.
72. W. K. Liu, J. Adee, and S. Jun. Reproducing kernel particle methods for elastic and plastic problems. In D.J. Benson and R.A. Asaro, editors, *Advanced Computational Methods for Material Modeling*, pages 175–190. AMD 180 and PVP 268, ASME, 1993.
73. W. K. Liu, W. Han, H. Lu, S. Li, and J. Cao. Reproducing kernel element method. I. Theoretical formulation. *Comput. Methods Appl. Mech. Engrg.*, 193(12-14):933–951, 2004.
74. W. K. Liu and S. Li. Reproducing kernel particle hierarchical partition of unity I: Formulation and theory. *Internat. J. Numer. Meths. Engrg.*, 45:251–288, 1999.
75. W.K. Liu and S. Li. Reproducing kernel particle hierarchical partition of unity II: Applications. *Internat. J. Numer. Meths. Engrg.*, 45:289–317, 1999.
76. A. I. Markushevich. *Theory of functions of a complex variable*. Chelsea Publishing Company, N.Y., 1965.
77. A.-M. Matache, I. Babuška, and C. Schwab. Generalized p-FEM in homogenization. *Numer. Math.*, 86:319–375, 2000.
78. J. M. Melenk. Finite element methods with harmonic shape functions for solving Laplace's equation. Master's thesis, University of Maryland, 1992.
79. J. M. Melenk. *On Generalized Finite Element Methods*. PhD thesis, University of Maryland, 1995.
80. J.M. Melenk. Operator adapted spectral element methods. I: Harmonic and generalized harmonic polynomials. *Numer. Math.*, 84(1):35–69, 1999.
81. J.M. Melenk. On n-widths for elliptic problems. *J. Math. Anal. Appl.*, 274:272–289, 2000.

82. J.M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Comput. Meth. Appl. Mech. Engrg.*, 139:289–314, 1996.
83. S.G. Mikhlin. *Numerical Perforamnce of Variational Methods*. Nordhoff, 1971.
84. N. Moës, J. Dolbrow, and T. Belytschko. A finite element method for crack growth without remeshing. *Internat. J. Numer. Meths. Engrg.*, 46(1):131–150, 1999.
85. N. I. Muskhelishvili. *Some Basic Problems of the Mathematical Theory of Elasticity*. P. Noordhoff, Groningen, 1963.
86. F. Narcowich, J. Ward, and H. Wendland. Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. *Math. Comput.*, (to appear).
87. B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
88. J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Univ. Hamburg*, 36:9–15, 1970/71.
89. J. T. Oden and A. Duarte. *hp* clouds – a meshless method. *Num. Meths. Part. Diff. Eqns.*, 12:673–705, 1996.
90. J.T. Oden and C.A. Duarte. Clouds, cracks and fem's. In B.D. Reddy, editor, *Recent developments in computational and applied mechanics. A volume in honour of John B. Martin. Barcelona: CIMNE*, pages 302–321, 1997.
91. E. Oñate, F. Perazzo, and J. Miguel. A finite point method for elasticity problems. *Computers & Structures*, 79:2143–2149, 2001.
92. A. Pinkus. *n-widths in approximation theory*. Springer Verlag, 1984.
93. G. Raugel. Résolution numérique par une méthode d'élements finis du problème de Dirichlet pour le Laplacien dans un polygone. *C. R. Acad. Sci. Paris*, 286:791–794, 1978.
94. C. Schwab. *p- and hp-Finite Element Methods*. Oxford University Press, 1998.
95. C. Schwab and A.M. Matache. Generalized FEM for homogenization problems. In *Multiscale and multiresolution methods*, volume 20 of *Lect. Notes Comput. Sci. Eng.*, pages 197–237. Springer, Berlin, 2002.
96. M.A. Schweitzer. *A parallel multilevel partition of unity method for elliptic partial differential equations*, volume 29 of *Lecture Notes in Computational Science and Engineering*. Springer, 2003.
97. D. Shephard. A two-dimensional function for irregularly spaced data. In *AMC National Conference*, pages 517–524, 1968.
98. F.L. Stazi, E. Budyn, J. Chessa, and T. Belytschko. An extended finite element with higher-order elements for curved cracks. *Computational Mechanics*, 31:38–48, 2003.
99. E.M. Stein. *Singular integrals and differentiability properties of functions*. Princeton University Press, 1970.
100. R. Stenberg. On some techniques for approximating boundary conditions in the finite element method. *J. Comput. Appl. Math.*, 63:139–148, 1995.
101. T. Strouboulis, K. Copps, and I. Babuška. The design and analysis of the generalized finite element method. *Comput. Meth. Appl. Mech. Engrg.*, 181:43–69, 2000.

102. T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method: an example of its implementation and illustration of its performance. *Internat. J. Numer. Meths. Engrg.*, 47:1401–1417, 2000.
103. T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Comput. Meth. Appl. Mech. Engrg.*, 190:4081–4193, 2001.
104. H. Triebel. *Interpolation Theory, Function Spaces, Differential Operators.* Johann Ambrosius Barth, 2 edition, 1995.
105. I. N. Vekua. *New Methods for Solving Elliptic Equations.* North Holland, 1967.
106. H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4:258–272, 1995.
107. H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *J. Approx. Theory*, pages 361–368, 1998.
108. H. Wendland. Meshless Galerkin methods using radial basis functions. *Math. Comput.*, 68:1521–1531, 1999.
109. H. Wendland. Local polynomial reproduction and moving least squares approximation. *IMA J. Numer. Anal.*, 21:285–300, 2001.
110. H. Wendland. *Scattered Data Approximation.* Cambridge University Press, 2004.
111. T. Zhu and S.N. Atluri. A modified collocation method and a penalty formulation for enforcing essential bounadry conditions. *Comp. Mech.*, 21:165–178, 1998.
112. W.P. Ziemer. *Weakly Differentiable Functions.* Springer Verlag, 1989.

Theory and Applications of Smoothed Particle Hydrodynamics

Joseph J. Monaghan

School of Mathematical Sciences, Monash University, Australia
email: joe.monaghan@sci.monash.edu.au

1 Introduction

Many problems in fluid dynamics involve more than one material, and more than one phase. An example is the eruption of a volcano where the magma contains gas which bubbles out as the magma reaches the surface. The result is a hot gas containing liquid rock which rapidly cools to form small pieces of solid rock. Such problems are not easily solved using finite difference methods because more than one material may move through a cell and, in general, each phase or material requires a different resolution. Related problems occur in astrophysics where the particle method Smoothed Particle Hydrodynamics (SPH) was developed to solve these problems (Gingold and Monaghan (1977), Lucy (1977), for a review see Monaghan (1992)).
The basic idea behind SPH is to replace the fluid by a set of points which follow the motion of the fluid and carry information about the properties of the fluid. For the mathematician these points are just interpolation points, but to the physicist and engineer it is naturally to think of them as real material particles. Whatever viewpoint is adopted it is necessary to assign properties to the particles and derive equations which will describe how these properties change. The simplest such property is mass, and in most problems the mass of each particle will remain constant. In addition we need to know the velocity, density and position of the particles (and possibly other quantities) and how these change with time. The equations which determine these changes are the equations of fluid dynamics.
The simplest set of equations are the acceleration and density equations for an ideal gas without dissipation. These are the Euler equations. The acceleration equation is

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla P, \tag{1.1}$$

where $\mathbf{v}$ is the velocity, ρ is the density, and P is the pressure. In this equation the time derivative is the derivative following the motion

$$\frac{d\mathbf{v}}{dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}. \tag{1.2}$$

In general P is a function of ρ and the thermal energy, but in the case where there is no dissipation the pressure can be taken as a function of ρ alone.

The density (continuity) equation is

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}. \tag{1.3}$$

In these equations the rates of change of physical quantities are determined by spatial derivatives. The key step in any numerical method for the solution of these equations is to approximate these derivatives by using information from a finite number of points. In finite difference methods the points are the vertices of a mesh. In the SPH method the interpolating points are particles which move with the flow. In the next section we will study the details of this particle interpolation method which is the characteristic feature of SPH.

2 Integral and Summation Interpolants

Suppose we wish to interpolate some property A which is a function of the spatial coordinates. A could be a scalar, a vector or a tensor quantity. We begin by writing the equality

$$A(\mathbf{r}) = \int A(\mathbf{r}')\delta(\mathbf{r} - \mathbf{r}')\, \mathbf{dr}', \tag{2.1}$$

where $\delta(\mathbf{r})$ denotes the Dirac delta function and $\mathbf{dr}'$ is an element of volume in the space being considered. The Dirac delta function has the property that it vanishes everywhere except where $\mathbf{r}$ vanishes where it becomes infinite in such a way that

$$\int \delta(\mathbf{r} - \mathbf{r}')\, \mathbf{dr}' = 1. \tag{2.2}$$

As a result the integral relation for A is an identity because the only non-zero contribution comes from the point where $\mathbf{r}' = \mathbf{r}$.
A delta function can be thought of as a limit of a well behaved function $W(\mathbf{r}, h)$ which has the following properties:

$$\lim_{h \to 0} W(\mathbf{r}, h) = \delta(\mathbf{r}), \tag{2.3}$$

and is normalised so that

$$\int W(\mathbf{r})\, \mathbf{dr}' = 1. \tag{2.4}$$

One example in one dimension is the Gaussian

$$W(x, h) = \frac{1}{h\sqrt{\pi}} e^{-x^2/h^2}, \tag{2.5}$$

which is a C^∞ function. Another example in one dimension is the spline defined as follows. If $q = |x|/h$ then

$$W(x,h) = \begin{cases} \frac{1}{h}(\frac{2}{3} - q^2 + \frac{1}{2}q^3), & \text{for } 0 \le q \le 1, \\ \frac{1}{6h}(2-q)^3, & \text{for } 1 \le q \le 2, \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

This function has continuous second derivatives and compact support. Using such a function we can replace (2.4) by the integral interpolant

$$A(\mathbf{r})_I = \int A(\mathbf{r}')W(\mathbf{r}-\mathbf{r}',h)\,\mathbf{dr}'. \tag{2.7}$$

In the following we will refer to the function W as the kernel.
Suppose now we divide the volume of fluid into a set of small volume elements. The element a will have a mass m_a, density ρ_a, and position $\mathbf{r}_a$. We denote the value of A at particle a by A_a. We can approximate the integral in the following way. First write the integral as

$$\int \frac{A(\mathbf{r}')}{\rho(\mathbf{r}')}\rho(\mathbf{r}')\,\mathbf{dr}'. \tag{2.8}$$

An element of mass is $\rho\mathbf{dr}'$. We can therefore approximate the integral by a summation over the mass elements. This gives us the summation interpolant

$$A_s(\mathbf{r}) = \sum_b m_b \frac{A_b}{\rho_b} W(\mathbf{r}-\mathbf{r}_b,h), \tag{2.9}$$

where the summation is over all the particles but, in practice, is only over near neighbours because W falls off rapidly with distance. Typically, h is close to the particle spacing, and the kernel W is effectively zero beyond a distance $2h$. In practice we choose kernels which have compact support i.e. they vanish at a finite distance. We will discuss the various types of kernels later.
As an example of the use of kernel estimation suppose A is the density ρ. The interpolation formula then gives the following estimate for the density at a point $\mathbf{r}$

$$\rho(\mathbf{r}) = \sum_b m_b W(\mathbf{r}-\mathbf{r}_b,h), \tag{2.10}$$

which shows how the mass of a set of particles is smoothed to produce the estimated density. The reader who is familiar with the technique of estimating probability densities from sample points (Parzen (1962)) will see that our formula for the density is the same with m_b replaced by $1/N$, where N is the number of sample points.
If h is constant we can integrate the density estimate to give

$$\int \rho(\mathbf{r})\,\mathbf{dr} = \sum_b m_b = M, \tag{2.11}$$

which shows that mass is conserved exactly (in the probability case the kernel estimate ensures the total probability is 1). If we allow h to vary, the integral is no longer exactly M but the errors are small because the particles carry their mass unchanged.

We not only want to estimate functions we also want to estimate gradients. The SPH formulation allows us to do this with ease. If we take W to be a differentiable function then we can differentiate our estimate of A exactly. For example

$$\frac{\partial A}{\partial x} = \sum_b m_b \frac{A_b}{\rho_b} \frac{\partial W}{\partial x}. \tag{2.12}$$

However, the straightforward SPH forms of spatial derivatives are not necessarily the most accurate. The following exercise involves the simple forms of the divergence and curl. We show below how better forms of these quantities can be constructed. The reader is urged to investigate the form of the terms in the case where the kernel is a Gaussian. In particular, note that the contribution to the divergence of the velocity from a particle is negative when it is moving towards the particle of interest. Correspondingly, the contribution to the density is positive when the particle is approaching.

Exercise 2.1. Show by taking the divergence of the SPH interpolation formula for $\mathbf{v}$ that

$$\nabla \cdot \mathbf{v} = \sum_b \frac{m_b}{\rho_b} \mathbf{v}_b \cdot \nabla W \tag{2.13}$$

and that

$$\nabla \times \mathbf{v} = \sum_b \frac{m_b}{\rho_b} \mathbf{v}_b \times \nabla W. \tag{2.14}$$

Show also that the contribution of particle b to $\nabla \cdot \mathbf{v}$ in the case of a Gaussian kernel is

$$-\frac{2m_b}{h^2} \mathbf{v}_b \cdot (\mathbf{r} - \mathbf{r}_b) W, \tag{2.15}$$

which is negative if particle b is moving towards $\mathbf{r}$. ∎

In the previous exercise the simplest form of the divergence and curl were worked out in the SPH formulation. However, in general these estimates do not vanish exactly when the velocity field is zero. To guarantee that they do vanish we use the trick of subtracting a quantity which would be zero if the SPH interpolation was perfect. Thus we write

$$\nabla \cdot \mathbf{v} = \nabla \cdot \mathbf{v} - \mathbf{v} \cdot \nabla 1, \tag{2.16}$$

and we use the SPH approximation of 1, namely

$$1 = \sum_b \frac{m_b}{\rho_b} W(\mathbf{r} - \mathbf{r}_b, h), \tag{2.17}$$

with gradient

$$\nabla 1 = \sum_b \frac{m_b}{\rho_b} \nabla W. \tag{2.18}$$

This gradient would be zero if the SPH interpolation was exact. Returning to our new expression for $\nabla \cdot \mathbf{v}$ we write the SPH form as

$$\nabla \cdot \mathbf{v} = \sum_b \frac{m_b}{\rho_b} (\mathbf{v}_b - \mathbf{v}) \cdot \nabla W. \tag{2.19}$$

For later reference we evaluate this expression at the position of particle a. We then find

$$(\nabla \cdot \mathbf{v})_a = \sum_b \frac{m_b}{\rho_b} \mathbf{v}_{ba} \cdot \nabla_a W_{ab}, \tag{2.20}$$

where $\mathbf{v}_{ba} = \mathbf{v}_b - \mathbf{v}_a$, the gradients ∇_a is taken with respect to the coordinates of particle a, and $W_{ab} = W(\mathbf{r}_a - \mathbf{r}_b, h)$.
In a similar way we get

$$(\nabla \times \mathbf{v})_a = \sum_b \frac{m_b}{\rho_b} (\mathbf{v}_b - \mathbf{v}_a) \times \nabla_a W_{ab}. \tag{2.21}$$

Exercise 2.2. Show that the contribution of particle b to $(\nabla \cdot \mathbf{v})_a$ in the case of a Gaussian kernel is

$$\frac{2m_b}{h^2} \mathbf{v}_{ab} \cdot \mathbf{r}_{ab} W_{ab}, \tag{2.22}$$

which is negative if particles a and b are moving towards each other. Show also that the contribution of particle b to $\nabla \times \mathbf{v}$ for particle a is proportional to the relative angular momentum of the two particles. ∎

Another way of getting more accurate formulae is to use another trick. This trick involves putting ρ inside expressions and compensating by adding or subtracting a term. For example we can write $\nabla \cdot \mathbf{v}$ another way by noting

$$\rho \nabla \cdot \mathbf{v} = \nabla \cdot (\rho \mathbf{v}) - \mathbf{v} \cdot \nabla \rho. \tag{2.23}$$

If we write the right hand side in SPH form we find

$$(\nabla \cdot \mathbf{v}_a) = -\frac{1}{\rho_a} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{2.24}$$

If this is compared with (2.21) it will be seen that one has ρ inside and one outside. Both expressions vanish, as they should, when the velocity is constant. However, when the system involves two or more fluids with large density ratios, the expression for $\nabla \cdot \mathbf{v}$ with ρ inside the summation is more accurate. The reason being that near an interface the summation for $\nabla \cdot \mathbf{v}$ for one type of fluid SPH particle involves contributions from the other fluid. If we imagine the other fluid changed for a fluid with exactly the same velocity

field, and exactly the same particle positions, but different density, we would still want the same estimate of $\nabla \cdot \mathbf{v}$. However, with (2.24) the mass elements will be changed and the estimate will be different. On the other hand, if (2.21) is used the ratio of mass to density will be invariant. In practice it turns out that either (2.21) or (2.24) can be used for density ratios less than about 2, but for larger density ratios it is better to use (2.21).
These two tricks, one where the $\nabla 1$ is subtracted, and one where ρ is put into expressions and then compensated by another term, are used frequently.

2.1 Errors in the Integral Interpolant

Let's go back to the integral interpolant in one dimension. We have

$$A_I(x) = \int A(x')W(x - x', h)\, dx', \tag{2.25}$$

and we assume h is constant. We expand $A(x')$ in a Taylor series about x to get

$$A_I(x) = \int [A(x) + (x' - x)\frac{dA(x)}{dx} + \tfrac{1}{2}(x' - x)^2 \frac{d^2A(x)}{dx^2} + \ldots]W(x' - x, h)\, dx'. \tag{2.26}$$

We now assume that $W(q, h)$ is an even function of q. In three dimensions this means the kernels are spherical. If $W(q, h)$ is an even function of q this means that the terms with odd powers of $x - x'$ will vanish giving

$$A_I(x) = \; A(x) + \frac{1}{2}\frac{d^2A(x)}{dx^2} \int (x' - x)^2 W(x' - x, h)\, dx' + \ldots. \tag{2.27}$$

Keeping in mind the example of the Gaussian kernel it is easy to see that

$$\int (x' - x)^2 W(x' - x, h)\, dx' = \sigma h^2, \tag{2.28}$$

where σ is a constant. We therefore write

$$A_I(x) = \; A(x) + \frac{\sigma h^2}{2}\frac{d^2A(x)}{dx^2} + \ldots, \tag{2.29}$$

which shows that the integral interpolant gives at least second order interpolation. The interpolation is better if σ is zero. Then higher order terms must be included in the expansion. The third order term vanishes because of symmetry leaving a possible fourth order term. An example of a higher order kernel is

$$W(x, h) = \frac{1}{h\sqrt{\pi}} \left(\frac{3}{2} - \frac{x^2}{h^2}\right) e^{-x^2/h^2}. \tag{2.30}$$

For this kernel, the integral interpolant is accurate to $O(h^4)$. However, this example shows the general result that to achieve higher order interpolation the kernel must change sign. This may have unwanted side effects. For example the density might become negative near a very strong shock.

2.2 Errors in the Summation Interpolant

If the particles are equi-spaced then we can easily estimate the errors in the summation interpolant. However, in general, the particles in an SPH calculation will be disordered. Before considering this case we calculate some examples in one dimension.

Exercise 2.3. Estimate the accuracy of interpolation for particles separated by a constant distance Δx on an infinite one dimensional line. Assume that A is a constant K and assign to each SPH particle a mass $m = \rho\Delta x$. Starting with the summation interpolant for any SPH point on the line

$$A(x_a) = \sum_{-\infty}^{\infty} A_b \Delta x W(x_a - x_b, h), \tag{2.31}$$

show that if the kernel is a Gaussian then

$$A(x_a) = \frac{K\Delta x}{h\sqrt{\pi}} \left[1 + 2e^{-q^2} + 2e^{-4q^2} + 2e^{-9q^2} + \ldots\right], \tag{2.32}$$

where $q = \Delta x/h$. Show that

$$A(x_a) = \begin{cases} 1.00010K & \text{if } q = 1, \\ 0.99999K & \text{if } q = 1.5, \\ 1.170K & \text{if } q = 0.5. \end{cases}$$

■

The previous exercise shows that if $h > \Delta x$ the interpolation over equi-spaced particles is very good, but if $h < \Delta x$ the accuracy is poor. In this last case the kernels do not overlap sufficiently to give high accuracy.

The reader will note that in the case of equi-spaced particles the results show that the integral interpolant with a Gaussian kernel is approximated very accurately by the summation interpolant. This result is well known and reflects the fact that integrals of Gaussian functions over an infinite interval can be represented by a finite summation with errors which are $\sim \exp(-(\pi h/\Delta x)^2)$.

Therefore, although a constant is not interpolated exactly, the errors are exponentially small. The reader will be able to confirm that the interpolation of a linear function of x has the same high accuracy. It is often said that functions should be represented is such a way that completeness is satisfied. The Gaussian kernels are not complete since they never interpolate any function, even a constant, exactly. Nevertheless they give accurate interpolation. This reminds us that what we are looking for is accuracy, and although completeness might be one route to accuracy, it is not the only route.

The following exercise is one way of showing that the dominant error depends on the Fourier Transform of the kernel.

Exercise 2.4. Estimate the accuracy of interpolation of a linear function $A(x) = \beta + \alpha x$ for particles separated by a constant distance Δ on an infinite one dimensional line as in the previous two problems and with a Gaussian kernel. Begin with the Poisson summation formula

$$\sum_{j=-\infty}^{\infty} f(j) = \int_{\infty}^{\infty} f(j)\, dj + 2\sum_{r=1}^{\infty} \int_{-\infty}^{\infty} f(j) \cos(2\pi j r)\, dj, \tag{2.33}$$

where, in the integrals, j becomes a continuous variable. The integral interpolant is approximated according to

$$\frac{\Delta}{h\sqrt{\pi}} \sum_{j=-\infty}^{\infty} [\beta + \alpha \Delta j] e^{-(\Delta(y-j)/h)^2} \tag{2.34}$$

where we have written $x = y\Delta$. Show from the Poisson summation formula, when x is on one of the points (so that y is an integer), that the summation interpolant gives

$$(\beta + \alpha \Delta y)\left(1 + \frac{2\Delta}{h\sqrt{\pi}} \int_{-\infty}^{\infty} \cos(2\pi q) e^{-(\Delta q/h)^2}\, dq + \ldots\right). \tag{2.35}$$

Work out the integral and show the error is exponentially small, and is smaller as h/Δ increases. Repeat the calculation for arbitrary (non-integer) y. ∎

If the kernel does not have the rapid decrease and smoothness of the Gaussian kernel the accuracy is less unless h/Δ is much larger than for the Gaussian. The following exercise illustrates this.

Exercise 2.5. Show that in one dimension the kernel $\exp(-|x|/h)$ when normalized is a possible kernel. Show also that a constant function K is interpolated by the function

$$\frac{K\Delta x}{2h} \frac{(1 + e^{-q})}{(1 - e^{-q})}. \tag{2.36}$$

∎

Shoenberg (1946) showed that interpolation accuracy could be related to the properties of the Fourier Transform of the interpolating kernel. Smoothness then shows up as rapid decrease of the Fourier Transform for large k and the order of accuracy shows up in the expansion of the Fourier Transform in powers of k. In particular, if the Fourier transform has a zero of order m at $k = 0$, then the kernel has continuous derivatives up to the $(m-2)$th.
Shoenberg was concerned with interpolation when the data was noisy. For that reason he wasn't interested in the standard interpolation formula such as those due to Everett or Bessel but rather interpolation with smoothing. In Shoenberg's formalism the interpolation is written in the form

$$f(x) = \sum_j f_j L(x - x_j), \tag{2.37}$$

which has the same as our SPH interpolation. If the points are equi-separated with spacing Δ, as in a table, then the Bessel formula which interpolates quadratic functions exactly is given by the following (where $q = |x|/h$)

$$L(x) = \begin{cases} (1-q)(1+\frac{1}{4}q), & \text{for } 0 \le q \le 1 \\ \frac{1}{4}(1-q)(2-q), & \text{for } 1 \le q \le 2. \\ 0, & \text{otherwise.} \end{cases} \tag{2.38}$$

The first derivative of this function is not continuous everywhere. When the data is noisy it is an advantage to have smoother interpolating kernels. Shoenberg (1946) constructed a set of basic smoothing functions which he called Cardinal Splines. They can be defined by their Fourier transform. Thus, the spline with continuous $(n-2)$ derivatives, $M_n(x)$, (which is an even function of x) is given by

$$M_n(x) = \int_{-\infty}^{\infty} \left(\frac{\sin \pi k \Delta}{\pi k \Delta} \right)^n \cos(2\pi k x)\, dk. \tag{2.39}$$

These spline kernels all interpolate with errors of $O(h^2)$, but they are smoother as n increases. The M_0 spline gives nearest grid point interpolation. The M_2 spline is:

$$M_2(x) = \begin{cases} 1-q, & \text{for } 0 \le q \le 1, \\ 0, & \text{for } q \ge 1. \end{cases} \tag{2.40}$$

In this, and the following expressions, q denotes $|x|/\Delta$. M_2 gives linear interpolation but its first derivative is discontinuous.

Exercise 2.6. Work out the Fourier transform defining M_2 and show it agrees with the linear function just defined. ∎

A commonly used kernel is the M_4 kernel (commonly called the cubic spline because it is a piecewise cubic polynomial). It has the form:

$$M_4(x) = \begin{cases} \frac{1}{6}(2-q)^3 - \frac{2}{3}(1-q)^3, & \text{for } 0 \le q \le 1, \\ \frac{1}{6}(2-q)^3, & \text{for } 1 \le q \le 2, \\ 0, & \text{for } q > 2. \end{cases} \tag{2.41}$$

The SPH kernel associated with $M_4(x)$ is $W(x,h) = \frac{1}{h} M_4(x)$ where now $q = |x|/h$. These kernels have been used for SPH interpolation because they are less sensitive to particle disorder.

For reference we end with the formula for M_n which can be determined from the Fourier Transform. Because the expressions are lengthy we use the notation

$$X_j = n/2 - q - j,$$

then

$$(n-1)!M_n(x) = \begin{cases} 0, & \text{for } q > n/2, \\ X_0^{n-1}, & \text{for } n/2-1 \le q \le n/2 \\ X_0^{n-1} - C(n,1)X_1^{n-1}, & \text{for } n/2-2 \le q \le n/2-1 \\ X_0^{n-1} - C(n,1)X_1^{n-1} & \\ \quad +C(n,2)X_2^{n-1}, & \text{for } n/2-3 \le q \le n/2-2 \end{cases} \tag{2.42}$$

and so on where $C(n,k) = n!/(k!(n-k)!)$. The sub-ranges are continued until we reach $0 \le q \le 1$ if n is even or $-1/2 \le q \le 1/2$ if n is odd (though in the latter case since $q \ge 0$ we use the domain $0 \le q \le 1/2$).
The kernel proportional to $e^{-|x|/h}$ is an example of a kernel that requires the contribution of many neighbours to give high accuracy. For example, if we are estimating a constant, and we decide to add particle contributions until the error is less that 0.001 of the first term, and we take $h = \Delta x$, then we must add contributions from 7 particles on each side of the particle of interest since $e^{-7} \sim 0.001$. This makes this kernel inefficient. The Fourier transform of this kernel is

$$\int_{-\infty}^{\infty} \frac{e^{-|x|/h}}{2h} e^{2\pi kx}\, dx = \frac{1}{1+(2\pi kh)^2}, \tag{2.43}$$

which decreases as $\sim 1/k^2$ for large k. By comparison the Fourier transform of the Gaussian is

$$e^{(-\pi kh)^2}, \tag{2.44}$$

which decreases much faster. From Schoenberg's analysis we expect that the Gaussian kernel would be much more efficient than the exponential kernel and this is found in practice.

2.3 Errors when the Particles are Disordered

During the course of an SPH calculation the particles become disordered. The exact form of this disorder depends on the dynamics. When Bob Gingold and I first ran SPH calculations we thought that the disorder could be described by a probability distribution proportional to the mass density, and that the errors could be estimated in the same way as a Monte Carlo estimate. In particular we expected that the errors arising from fluctuations would be $\sim 1/\sqrt{N}$. However, the errors were much smaller than this estimate would suggest. The reason for the smaller errors is that the probability estimates allow fluctuations which are inconsistent with the dynamics. Because the disorder depends on the dynamics it is not possible to make traditional error estimates like those used for finite differences or finite elements. For that reason estimates of SPH calculations have had to depend on comparisons with known solutions, experiments, or by studying how the error varies

with particle number for particular calculations (see for example Cleary and Monaghan (1999)). These comparisons show that it is possible to achieve very accurate results with SPH.
It is clear that the errors depend on the type of disorder. Niedereiter (1978) showed that if the points are quasi-ordered then the error of an integration in d dimensions varies with the number of particles as $\ln(N^d)/N$ which is not very satisfactory in 1 dimension, but in 3 dimensions it means the error varies as $h^3 \ln|h|$ which is efficient. There is a large industry working out multi dimensional integration algorithms using quasi-disordered numbers. As a simple example suppose points x_j in $0 \leq x_j \leq 1$, are computed by recurrence from

$$x_{j+1} = x_j + \alpha, \quad \mathrm{mod}(1) \tag{2.45}$$

where α is an irrational number and $x_0 = 0$. If we wish to evaluate the integral

$$\int_0^1 F(x)\,dx, \tag{2.46}$$

then we can approximate it by using the points x_j. Thus

$$\int_0^1 F(x)\,dx = \frac{1}{N}\sum_{j=1}^{N} F(x_j) + \epsilon, \tag{2.47}$$

where ϵ is the error. We can estimate this error by using the Fourier expansion of $F(x)$. We can write

$$F(x) = \sum_{n=-\infty}^{\infty} C_n e^{2\pi nix}, \tag{2.48}$$

where

$$C_0 = \int_0^1 F(x)\,dx. \tag{2.49}$$

Then we can write the integral as

$$\frac{1}{N}\sum_{j=0}^{N}\sum_{n=-\infty}^{\infty} C_n e^{2\pi nij\alpha}, \tag{2.50}$$

where we can replace x_j by $j\alpha$ because the mod(1) operation has no effect on the complex exponentials. We can write this summation as

$$C_0 + \frac{2}{N}\sum_{n=1}^{\infty} C_n \frac{\sin(\pi\alpha n(N+1))}{\sin(\pi\alpha n)}\cos(\pi nN\alpha). \tag{2.51}$$

This shows that the error in this case is $\propto 1/N$ but the constant depends on how rapidly the Fourier coefficients decrease and how small $\sin(\pi n\alpha) \neq 0$ becomes (for a detailed discussion see Davis and Rabinowitz, (1967)).

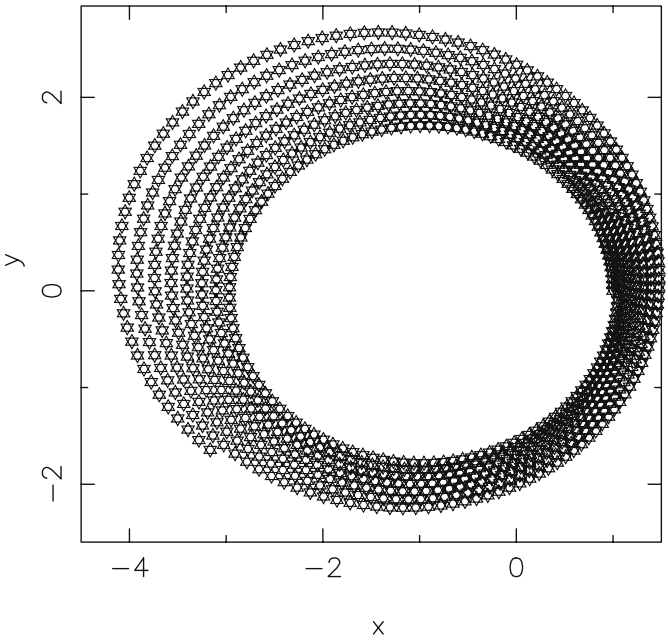


Figure 2.1. The Kepler problem integrated using a 4th order Runge Kutta method. Note the orbit is not closed and spirals out.

Another reason for the accuracy of SPH despite the disorder is that it is possible to set up SPH calculations so that they conserve important quantities like momentum and energy. The importance of this conservation shows up in very simple problems. Suppose, for example, that we wish to integrate the equations for a binary star system with the stars treated as points and we are offered either a second order reversible symplectic integrator, or a standard 4th order Runge Kutta integrator. If we use the Runge Kutta scheme the orbit, instead of being an ellipse, will spiral in or spiral out with the effect being more extreme as the eccentricity gets closer to 1. The problem arises because the standard 4th order Runge Kutta does not conserve angular momentum. On the other hand, the symplectic Verlet integrator, which is a second order integrator, gives much better results because it conserves angular momentum exactly, and conserves energy better than the Runge Kutta method. In addition it is easy to ensure that the numerical orbit is reversible. In Figure 2.1 we show the orbit for the Kepler problem with eccentricity 0.5 integrated with the Runge Kutta method. The orbit spirals out and the axis of symmetry rotates. In Figure 2.2 the orbit calculated using the symplectic Verlet method is shown. This example shows very clearly that high order does not necessarily mean that important properties of the dynamics will be retained. We will show later that the SPH equations for non-dissipative flow can be derived from a particle Lagrangian which preserves many of the invariants of

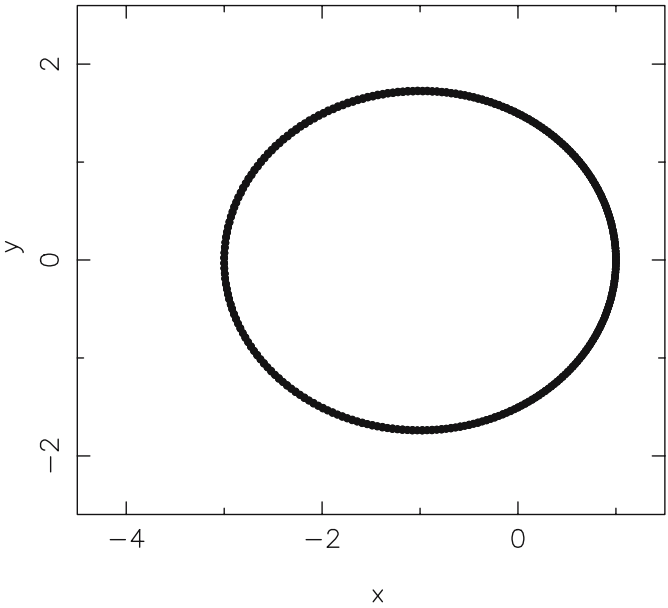


Figure 2.2. The Kepler problem integrated with the symplectic Verlet method using the same step size as for the Runge Kutta example shown in Figure 2.1. Note that the orbit is closed to high accuracy.

the original system. These include not only the additive invariants of energy, and momentum but also the integral invariants such as Liouville's theorem and Poincare invariants which involve integrations in phase space. In addition, with appropriate time integration, the reversibility of the dynamical system can be preserved. It seems, but has not been proven, that in this case it is the good approximation of the Lagrangian that is the key reason for the robustness and accuracy of the SPH equations.

3 Euler Equations

In the previous sections we showed how the spatial gradients could be estimated from information at the particle positions. In this section we will study how the equations of non-dissipative fluid dynamics (the Euler equations) can be approximated by SPH using expressions for the spatial derivatives.
The simplest set of equations we need to solve are the acceleration and continuity equations for an ideal gas without dissipation. These are the Euler equations. For the present we assume there are no body forces. The acceleration equation in the absence of gravity or other body forces is then

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla P, \tag{3.1}$$

where $\mathbf{v}$ is the velocity, ρ is the density, P is the pressure and $\mathbf{g}$ is the body force per unit mass. In this equation the time derivative is the derivative following the motion

$$\frac{d\mathbf{v}}{dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}. \tag{3.2}$$

The characteristics of this equation are the trajectories of the fluid elements. These trajectories will be approximated by the trajectories of the SPH particles.

In general P is a function of ρ and the thermal energy, but in the case where there is no dissipation the pressure can be taken as a function of ρ and the entropy per unit mass s which remains constant. In some cases we will assume the entropy is the same for all particles, but in general each particle could have a different entropy which does not change with time.

The density (continuity) equation is

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}, \tag{3.3}$$

and to move the particles we solve the equation

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}. \tag{3.4}$$

In the following we will work out SPH forms of these equations which will determine how the position and density at each particle changes with time.

3.1 The SPH Continuity Equation

As shown in the previous section we can estimate the density at particle a by the summation

$$\rho_a = \sum_b m_b W_{ab}, \tag{3.5}$$

where W_{ab} denotes $W(\mathbf{r}_a - \mathbf{r}_b, h)$ and m_b is the mass of particle b. If we take the time derivative of (3.5) we find

$$\frac{d\rho_a}{dt} = \sum_b m_b \frac{dW_{ab}}{dt} = \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}, \tag{3.6}$$

where $\mathbf{v}_{ab} = \mathbf{v}_a - \mathbf{v}_b$. We can decode the RHS of (3.6) by writing it first as

$$\mathbf{v}_a \cdot \sum_b m_b \nabla_a W_{ab} - \sum_b m_b \mathbf{v}_b \cdot \nabla_a W_{ab}. \tag{3.7}$$

which can be interpreted as the SPH expression for

$$\mathbf{v}_a \cdot (\nabla \rho)_a - (\nabla \cdot (\rho \mathbf{v}))_a = -(\rho \nabla \cdot \mathbf{v})_a. \tag{3.8}$$

Our form of the rate of change of density (3.6) is therefore what we would have arrived at by using the expression (2.24) for $\nabla \cdot \mathbf{v}$ in the continuity equation. Another form of the continuity equation follows from (2.21)

$$\frac{d\rho_a}{dt} = \rho_a \sum_b \frac{m_b}{\rho_b} \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{3.9}$$

This equation is more accurate when there are fluids with very different densities in contact.

3.2 The SPH Acceleration Equation

We can convert the acceleration equation for our ideal fluid into SPH form by writing

$$(\nabla P)_a = \sum_b m_b \frac{P_b}{\rho_b} \nabla_a W_{ab}. \tag{3.10}$$

Our first, crude, SPH form of the acceleration equation is then

$$\frac{d\mathbf{v}_a}{dt} = -\frac{1}{\rho_a} \sum_b m_b \frac{P_b}{\rho_b} \nabla_a W_{ab}. \tag{3.11}$$

However, this equation doesn't conserve linear or angular momentum exactly since the force on particle a due to b is not equal and opposite to the force on b due to a or

$$\frac{m_a m_b P_b}{\rho_a \rho_b} \neq \frac{m_a m_b P_a}{\rho_a \rho_b}. \tag{3.12}$$

To write the acceleration equation in a form which conserves linear and angular momentum we make the force term symmetric by noting that

$$\frac{\nabla P}{\rho} = \nabla \left(\frac{P}{\rho} \right) + \frac{P}{\rho^2} \nabla \rho. \tag{3.13}$$

Using the SPH interpolation rules we can write the first term on the right hand side as

$$\nabla \left(\frac{P}{\rho} \right)_a = \sum_b \frac{P_b}{\rho_b^2} \nabla_a W_{ab}, \tag{3.14}$$

and the second term as

$$\frac{P_a}{\rho_a^2} (\nabla \rho)_a = \frac{P_a}{\rho_a^2} \sum_b m_b \nabla_a W_{ab}, \tag{3.15}$$

combining these we get the acceleration equation

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} \right) \nabla_a W_{ab}, \tag{3.16}$$

Assuming that the kernel W_{ab} is a function of $|\mathbf{r}_a - \mathbf{r}_b|$ we can write its gradient in the following form

$$\nabla_a W_{ab} = \mathbf{r}_{ab} F_{ab}, \tag{3.17}$$

where F_{ab} is a scalar function of $|\mathbf{r}_a - \mathrm{r}_b|$ and $F_{ab} \leq 0$. The force/mass on a due to b is then

$$m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} \right) \mathbf{r}_{ab} F_{ab}, \tag{3.18}$$

which shows that the force on a due to b is now equal and opposite to the force on b due to a.

Exercise 3.1. Show that the linear momentum $\sum_a m_a \mathbf{v}_a$ and angular momentum $\sum_a m_a \mathbf{r}_a \times \mathbf{v}_a$ are conserved if the symmetric form of the acceleration equation is used. ∎

3.3 The Thermal Energy Equation

We get the thermal energy equation from the first law of thermodynamics

$$T\, ds = du + P\, dv \tag{3.19}$$

$$= du - \frac{P}{\rho^2} d\rho \tag{3.20}$$

where s is the entropy, and we have assumed in the last equation that all quantities are per/unit mass. If there is no source of heat we deduce

$$\frac{du}{dt} = \frac{P}{\rho^2} \frac{d\rho}{dt} = -\frac{P}{\rho^2} \nabla \cdot \mathbf{v}. \tag{3.21}$$

We can write this equation in various ways. For example

$$\frac{du}{dt} = \frac{P}{\rho^2} \left(\nabla \cdot (\rho \mathbf{v}) - \mathbf{v} \cdot \nabla \rho \right), \tag{3.22}$$

and, in SPH form for any particle a, this equation becomes

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{3.23}$$

Alternatively, we could make use of one of the forms of $\nabla \cdot \mathbf{v}$ from the previous section and deduce the thermal energy equation

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a} \sum_b \frac{m_b}{\rho_b} \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{3.24}$$

A good general principle when writing SPH equations is to be consistent. For example, if we use a particular expression for $\nabla \cdot \mathbf{v}$ in the continuity equation, we should use the same form in the energy equation.

3.4 Dispersion of Sound Waves

We now have a set of ordinary differential equations for the motion of a fluid when there is no dissipation. We will discuss how to integrate the equations in a later section. For the present it is useful to consider how the SPH formulation affects the dispersion relation for small amplitude waves in a gas in one dimension. We assume the initial density $\bar{\rho}$ is constant and the domain is infinite. The SPH particles have equal mass and are initially equi-spaced with spacing Δx. For convenience we assume that the equation of state is $P = K\rho^2$. We assume the sound waves have sufficiently small amplitude then we can write the position of particle a as

$$x_a = \bar{x}_a + Xe^{i(k\bar{x}_a - \omega t)}, \tag{3.25}$$

where $\bar{x}_a$ is the unperturbed position of particle a and the wave has frequency ω and wave number k. The velocity can be written as

$$v_a = Ve^{i(k\bar{x}_a - \omega t)}, \tag{3.26}$$

and the density

$$\rho_a = \bar{\rho}_a + De^{i(k\bar{x}_a - \omega t)}. \tag{3.27}$$

Because $P = K\rho^2$ we do not need to consider the continuity equation to obtain the dispersion relation for the SPH system. The first order perturbation to the acceleration equation gives

$$-i\omega v_a = -2mK \sum_b (\delta x_a - \delta x_b) \frac{d^2 W_{ab}}{d\bar{x}_a^2}, \tag{3.28}$$

where

$$\delta x_a = Xe^{i(k\bar{x}_a - \omega t)}. \tag{3.29}$$

Substituting for v_a we get

$$-i\omega V = -2mKX \sum_b \left[1 - e^{ik(\bar{x}_b - \bar{x}_a)}\right] \frac{d^2 W_{ab}}{d\bar{x}_a^2}. \tag{3.30}$$

From the equation for the change in position $dx_a/dt = v_a$ we get

$$-i\omega X = V. \tag{3.31}$$

Substituting this result into the previous equation we get the dispersion relation

$$\omega^2 = 2mK \sum_b \left[1 - e^{ik(\bar{x}_b - \bar{x}_a)}\right] \frac{d^2 W_{ab}}{d\bar{x}_a^2}, \tag{3.32}$$

Because the particles are equi-spaced and the line is infinite we can shift the origin in the summation to $\bar{x}_a$ and measure lengths from this point. We can then write (3.32) as

$$\omega^2 = 2mK \sum_b \left[1 - e^{ik\bar{x}_b}\right] \frac{d^2W(\bar{x}_b, h)}{d\bar{x}_b^2}. \tag{3.33}$$

If the wavelength is much large than the particle spacing we can replace the summation by an integration according to

$$\sum_{b=-\infty}^{\infty} \left[1 - e^{ik\bar{x}_b}\right] \frac{d^2W(\bar{x}_b, h)}{d\bar{x}_b^2} \simeq \frac{1}{\Delta x} \int_{-\infty}^{\infty} \left[1 - e^{ikb\Delta x}\right] \frac{\partial^2 W}{\partial b^2}\, db, \tag{3.34}$$

where we have used the fact that $\bar{x}_b = b\Delta x$ and, for convenience, b is used to denote both the discrete and the continuous variable. Integrating by parts twice we find

$$\sum_{b=-\infty}^{\infty} \left[1 - e^{ik\bar{x}_b}\right] \frac{d^2W(\bar{x}_b, h)}{d\bar{x}_b^2} \simeq k^2 \int_{-\infty}^{\infty} We^{ikb\Delta x}\, db. \tag{3.35}$$

Since we have assumed $P = K\rho^2$ the speed of sound c_s is equal to $2K\rho = 2Km/\Delta x$. We can therefore write the dispersion relation as

$$\omega^2 \simeq c_s^2 k^2 \tilde{W}, \tag{3.36}$$

where $\tilde{W}$ denotes the Fourier transform of W

$$\int_{-\infty}^{\infty} We^{ikb\Delta x}\, db. \tag{3.37}$$

If the kernel is a Gaussian we can evaluate the integral to get

$$\omega^2 = c_s^2 k^2 e^{-(kh/2)^2}. \tag{3.38}$$

If the $kh \ll 2$ the dispersion relation is a close approximation to the exact form $\omega^2 = c_s^2 k^2$, but as k increases the frequency of the wave calculated using SPH drops below the correct value. The largest allowed value of k is $\pi/\Delta x$ and for this k the error is a maximum. However, it is not the error that is a concern for short wave lengths but rather whether or not the method remains stable. To determine the stability we evaluate the dispersion relation numerically.

Because the term $\frac{d^2W(\bar{x}_b,h)}{dx_b^2}$ in the original dispersion relation is an even function of the coordinates we can write it as

$$\omega^2 = 2mK \sum_b \left[1 - \cos k\bar{x}_b\right] \frac{d^2W(\bar{x}_b, h)}{d\bar{x}_b^2}. \tag{3.39}$$

If the system is unstable the fastest growing mode is usually the one with the shortest wavelength (largest k) and this shows up as clumping. We therefore evaluate the dispersion relation for $k = \pi/\Delta x$ and get

$$\omega^2 = 8mK \sum_{j=1}^{\infty} \frac{d^2 W(\bar{x}_j, h)}{d\bar{x}_j^2}, \tag{3.40}$$

where the summation is over odd values of j. For the Gaussian kernel we get

$$\omega^2 = \frac{8mK}{h^3\sqrt{\pi}} \sum_{j=1}^{\infty} \left(-2 + \frac{4\bar{x}_j^2}{h^2} \right) e^{-\bar{x}_j^2/h^2}. \tag{3.41}$$

Evaluating the right hand side for $\Delta x \leq h \leq 2\Delta x$, the usual range for SPH calculations we find it is positive, showing that ω is real and the method stable with a Gaussian kernel. In practice the time evolution is approximated by discrete steps and the stability then depends on the scheme used. We will discuss time stepping schemes later.

Exercise 3.2. If the equation of state is $P = K\rho^\gamma$ show that the dispersion relation is

$$\omega^2 = \frac{c_s^2}{\gamma} \left(B + (\gamma - 2)A^2 \right), \tag{3.42}$$

where

$$B = \Delta x \sum_b \left(1 - \cos\left(k\bar{x}_b\right)\right) \frac{d^2 W(\bar{x}_b, h)}{d\bar{x}_b^2}, \tag{3.43}$$

and

$$A = -\Delta x \sum_b \sin\left(k\bar{x}_b\right) \frac{dW(\bar{x}_b, h)}{d\bar{x}_b}. \tag{3.44}$$

In the long wave length limit replace summations by integrations and show that

$$B = k^2 \quad \text{and} \quad A = k, \tag{3.45}$$

where we have used the approximation $\tilde{W} = 1$. Show that the dispersion relation becomes

$$\omega^2 = c_s^2 k^2, \tag{3.46}$$

in agreement with the exact value. As before the deviation from the correct value is a maximum for $k = \pi/\Delta x$. For this k, $A = 0$ and $B \geq 0$ so that SPH is stable for arbitrary γ. This result is also true for the spline kernel. ∎

4 Tests of the SPH Euler Equations

The usual tests in computational gas dynamics involve systems with rigid or periodic boundaries. In this section we consider tests where the system is a

finite region of gas held together by a simple force. In this sense, they are like model stars with the gravitational force replaced by another force which is easy to calculate. These systems are therefore called Toy Stars (Monaghan and Price (2004)). The force we consider is such that for any two elements of mass the force between them proportional to their separation and along the line of their centres. This force is the simplest many-body force. It was discovered by Newton who pointed out that if two particles attract each other with a linear force then they move as if attracted to the centre of mass of the pair (see Chandrasekhar (1995) for a modern interpretation of Newton's Principia and, in particular, Newton's proposition LXIV which discusses this force).

If there are N particles attracting each other with a force proportional to the separation, and directed along the line joining pairs of particles, then each particle moves as if independent of the others. The force is a linear force towards the centre of mass of the N particles. In the case of two particles the trajectories are Lissajous figures. A gaseous system with this force has a number of attractive features for testing algorithms for fluid dynamics. The modes of oscillation can be calculated easily, and there is a nonlinear solution where the velocity is a linear function of the coordinates. This solution can be calculated very accurately by integrating a small number (2 in the case of one dimension) of ordinary differential equations.

The simplest version of the Toy star assumes the pressure P is given in terms of the density ρ by $P = K\rho^2$ where K is a constant. This makes the problem analogous to the problem of shallow water motion in paraboloidal basins. There is an extensive literature on this problem including the early papers of Goldsbrough (1930) and the general analysis by Holm (1991) which contains many further references.

4.1 The Force Law in One Dimension

Newton proposed the linear force law in the Principia but for our purposes the modern discussion by Chandrasekha (1995) is clearer. Suppose for example that we have an isolated group of N particles in one dimension interacting with linear forces so that the potential energy is

$$\Phi = \frac{1}{4}\nu \sum_{j=1}^{N}\sum_{k=1}^{N} m_j m_k (x_j - x_k)^2, \tag{4.1}$$

The equation of motion of the j^{th} particle is then

$$m_j \frac{d^2 x_j}{dt^2} = -\nu m_j \sum_k m_k (x_j - x_k). \tag{4.2}$$

However, the centre of mass

$$\frac{\sum_k m_k x_k}{\sum_k m_k}, \tag{4.3}$$

can be chosen as the origin so the equation of motion becomes

$$\frac{d^2x_j}{dt^2} = -\nu M x_j, \tag{4.4}$$

where M is the total mass. The potential can then be written

$$\Phi = \frac{1}{2}\nu M \sum_j m_j x_j^2, \tag{4.5}$$

The motion of the N-body system is therefore identical to the independent motion of each particle in a harmonic potential. In the following we replace $M\nu$ by Ω^2.

4.2 The Equations of Motion

The system is one dimensional with velocity v, density ρ, and pressure P. The acceleration equation is

$$\frac{dv}{dt} = -\frac{1}{\rho}\frac{\partial P}{\partial x} - \Omega^2 x. \tag{4.6}$$

We assume the equation of state is

$$P = K\rho^2, \tag{4.7}$$

which makes our equations identical in form to those for the shallow water equations with density replacing the water depth. The acceleration equation is then

$$\frac{dv}{dt} = -2K\frac{\partial \rho}{\partial x} - \Omega^2 x. \tag{4.8}$$

The simplest case to consider is the static model. The next simplest case is to study the small oscillations about the static model. However, one of the attractive features of the toy star is that we can find an exact stable nonlinear oscillation. We now consider each of these in turn.

Exercise 4.1. Show that the static model has density

$$\rho = \rho_0 - \frac{\Omega^2 x^2}{4K}. \tag{4.9}$$

so that the radius x_e of the static model is

$$x_e^2 = \frac{4K\rho_0}{\Omega^2}. \tag{4.10}$$

and the mass M is $4\rho_0 x_e/3$. ∎

If M, K and Ω are specified then ρ_0 and therefore x_e can be calculated. To simplify the following equations we use x_e as the unit of length, and we use $1/\Omega$ as the unit of time. The acceleration equation then becomes

$$\frac{dv}{dt} = -\frac{1}{2}\frac{\partial \rho}{\partial x} - x, \tag{4.11}$$

and then the static density $\bar{\rho}$ is

$$\bar{\rho} = 1 - x^2, \tag{4.12}$$

while $P = \rho^2/4$ and $M = 4/3$.

4.3 Oscillations

We now consider small oscillations of our toy star. We assume v is small and we write the density in the form

$$\rho = \bar{\rho} + \eta. \tag{4.13}$$

If we retain only quantities which are linear in the perturbations the acceleration equation becomes

$$\frac{\partial v}{\partial t} = -\frac{1}{2}\frac{\partial \eta}{\partial x}, \tag{4.14}$$

and the continuity equation becomes

$$\frac{\partial \eta}{\partial t} = -\frac{\partial(\bar{\rho} v)}{\partial x}. \tag{4.15}$$

We let the time variation be $e^{i\omega t}$ and, by combining the equations, the equation for v becomes

$$(1 - x^2)\frac{d^2 v}{dx^2} - 4x\frac{dv}{dx} + 2(\omega^2 - 1)v = 0. \tag{4.16}$$

The solutions of this equation are the Gegenbauer polynomials $G_n(x)$. The solution for $G_n(x)$ requires that

$$2(\omega^2 - 1) = n^2 + 3n, \tag{4.17}$$

or

$$\omega^2 = \frac{(n+1)(n+2)}{2}. \tag{4.18}$$

Typical examples of Gegenbauer polynomials are $G_0(x) = 1$, $G_1(x) = x$, $G_2(x) = 3(5x^2 - 1)/2$, and $G_3(x) = 5(7x^3 - 3x)/2$. Note that the Gegenbauer polynomials rise rapidly near the edge of the Toy Star. The standard normalization is

$$\int_{-1}^{1} G_n^2(x)(1 - x^2)\, dx = 2\frac{(n+1)(n+2)}{2n+3}. \tag{4.19}$$

Other properties of G_n can be found in books on special functions e.g. Abramowitz and Stegun which is available on the Web). The equation for the density perturbation is

$$(1-x^2)\frac{d^2\eta}{dx^2} - 2x\frac{d\eta}{dx} + 2\omega^2\eta = 0. \tag{4.20}$$

The solution to this equation are Legendre polynomials $P_m(x)$. We note that

$$\frac{d}{dx}P_{m+1}(x) = G_m(x). \tag{4.21}$$

We compare the perturbation solution with the SPH calculation below.

4.4 SPH Results for Small Oscillations

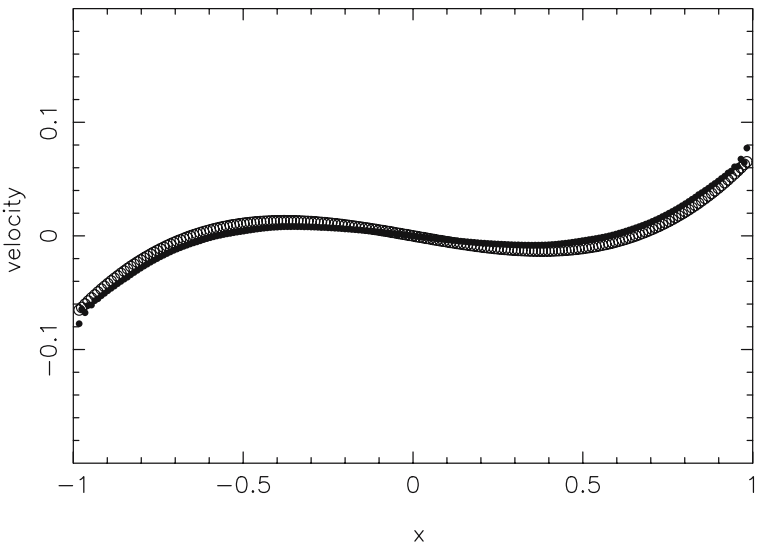


Figure 4.1. The velocity field for the toy star oscillating with the velocity field in the 3rd mode after 4 periods. The SPH results are shown by the filled symbols and the exact result by the circles.

To simulate the high order oscillations we need enough particles to ensure the resolution length is much smaller than the wave lengths of the modes. In the present case we use 400 particles. The toy star can be set up with the particles in the static position with initial velocity $v = 0.01C_sG_n(x)$ and solution $v = 0.01C_sG_n(x)\cos(\omega t)$, where C_s is the speed of sound (with value

$1/\sqrt{2}$ in our scaled units). Correspondingly the expected density perturbation is $\eta = 0.02 C_s \omega P_{n+1} \sin(\omega t)$. A comparison between the SPH and the exact solution is shown in Figure 4.1 for the 3rd mode. Results for other modes are given by Monaghan and Price (2004).

Exercise 4.2. Show that there is an exact nonlinear solution in the form

$$v = A(t)x, \tag{4.22}$$

with

$$\rho = H(t) - C(t)x^2, \tag{4.23}$$

so that the time dependent radius of the toy star is $\sqrt{h/C}$, and the mass M is given by

$$M = 2\int_0^{\sqrt{H/C}} \rho \, dx = \frac{4}{3}\left(\frac{H^3}{C}\right)^{1/2} \tag{4.24}$$

where the equations determining A, H, C are

$$\dot{A} + A^2 = C - 1, \tag{4.25}$$

and

$$\dot{H} = -AH, \quad \text{and} \quad \dot{C} = -3AC. \tag{4.26}$$

Deduce from the last two equations that $H^3 \propto C$, which guarantees the conservation of mass. The solution of the original equations of motion therefore reduce to the solution of the two first order autonomous equations

$$\dot{A} = C - 1 - A^2 \tag{4.27}$$

$$\dot{C} = -3AC, \tag{4.28}$$

after which H can be found from C. The equations for A and C can be solved for given initial conditions on v and ρ. ∎

Exercise 4.3. Generalize the theory to the case where $P = K\rho^\gamma$ again assuming

$$\rho^{\gamma-1} = H(t) - C(t)x^2, \tag{4.29}$$

and $v = A(t)x$. Show that

$$\dot{A} + A^2 = \frac{2K\gamma}{\gamma - 1}C - 1, \tag{4.30}$$

and from the continuity equation

$$\dot{H} = -AH(\gamma - 1), \tag{4.31}$$

and

$$\dot{C} = -AC(\gamma + 1). \tag{4.32}$$

From these last two equations deduce that $H^{\gamma+1} \propto C^{\gamma-1}$ which guarantees the conservation of mass. ∎

The Toy stars considered here can be extended to 2 and 3 dimensions. In the 3 dimensional case there are no incompressible flow problems which are related to the Toy star. Current work on the 2 dimensional case shows again that the SPH algorithm is very stable and gives accurate results.

5 Lagrangian SPH

We will now discuss how to derive SPH equation from a Lagrangian (for the theory and application to mechanics see Landau and Lifshitz, Mechanics or the Feynman Lectures Vol II). The advantage of using a Lagrangian is that conservation laws are built into the equations from the beginning and this was the motivation for using Lagrangians in early papers on SPH (Gingold and Monaghan (1978), (1979), (1982)). In fact, the form of the pressure force we used previously, (3.18), was discovered by deriving the equations from a Lagrangian. For more complicated problems the Lagrangian leads to robust equations. In the case of relativistic flow a Lagrangian was used by Price and Monaghan (2001), and in the case of elasticity, Bonet and Lok (1999) showed the advantages of using a Lagrangian. A recent study of MHD simulations using SPH is based on equations derived from a Lagrangian (Price and Monaghan (2004)). Another nice feature of the Lagrangian is that other symmetries can be exploited and, if desired, the whole system of equations can be written in Hamiltonian form.

5.1 The Lagrangian

The Lagrangian L for the non-dissipative motion of a fluid is (Eckart (1960))

$$L = \int \rho \left(\frac{1}{2} v^2 - u(\rho, s) \right) \mathbf{dr}, \tag{5.1}$$

where v is the velocity, u the thermal energy per unit mass, ρ the density and s is the entropy. We assume the entropy of each element of fluid remains constant though each particle can have a different entropy.
The SPH form of this Lagrangian is

$$L = \sum_b m_b \left(\frac{1}{2} v_b^2 - u(\rho_b, s_b) \right) \tag{5.2}$$

From Lagrange's equations for particle a

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \mathbf{v}_a} \right) - \frac{\partial L}{\partial \mathbf{r}_a} = 0, \tag{5.3}$$

we find

$$\frac{d\mathbf{v}_a}{dt} = - \sum_b m_b \left(\frac{\partial u}{\partial \rho} \right)_s \frac{\partial \rho_b}{\partial \mathbf{r}_a}. \tag{5.4}$$

From the first law of thermodynamics we find

$$\left(\frac{\partial u}{\partial \rho}\right) = \frac{P}{\rho^2}, \tag{5.5}$$

where P_a is the pressure at particle a and, from the SPH summation for the density (but assuming h is constant),

$$\frac{\partial \rho_b}{\partial \mathbf{r}_a} = \sum_c m_c \nabla_b W_{bc}(\delta_{ab} - \delta_{ac}), \tag{5.6}$$

where δ_{ab} is a Kronecker delta, and ∇_a denotes the gradient taken with respect to the coordinates of particle a.
Substituting these results into Lagrange's equation and noting that

$$\frac{\partial L}{\partial \mathbf{v}_b} = m_b \mathbf{v}_b, \tag{5.7}$$

we get

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right) \nabla_a W_{ab}. \tag{5.8}$$

which agrees with the result we obtained earlier using a symmetrized form of the force. As remarked earlier this symmetrized form was discovered from Lagrange's equations.
The Lagrangian and the equation of motion can be easily generalised to the case where there is a body force which can be derived from a potential Φ which is a function of the coordinates. In this case

$$L = \sum_b m_b \left(\frac{1}{2} v_b^2 - u(\rho_b, s_b) - \Phi\right). \tag{5.9}$$

5.2 Conservation Laws

The conservation laws of fluid mechanics are intimately related to the invariance properties. The fundamental theorem is due to Emmy Noether who showed that, if the Lagrangian is invariant to infinitesimal transformations, there will be a conserved quantity. Conserved quantities can be found by playing around with the equations of motion but it is much easier to find these quantities by studying the invariance properties of the Lagrangian.

Momentum Conservation

If the Lagrangian is invariant to a shift in the coordinate system, the momentum is conserved. To see this consider a shift in the coordinate system by the small vector $\mathbf{q}$. In this case the change in $\mathbf{r}$ is $\mathbf{q}$ and the velocity remains

unchanged. If L is invariant to this change then the change in L, to first order, is

$$\delta L = \sum_b \frac{\partial L}{\partial \mathbf{r}_b} \cdot \mathbf{q} \tag{5.10}$$

$$= \mathbf{q} \cdot \sum_b \frac{\partial L}{\partial \mathbf{r}_b} \tag{5.11}$$

$$= \mathbf{q} \cdot \frac{d}{dt} \sum_b \frac{\partial L}{\partial \mathbf{v}_b}, \tag{5.12}$$

where the last equation follows from Lagrange's equations. Since δL must vanish for arbitrary small $\mathbf{q}$ we conclude that the linear momentum

$$\sum m_b \mathbf{v}_b, \tag{5.13}$$

is conserved.

Suppose now that L is invariant to a rotation of the coordinate system through a small angle ϕ about an axis in the direction $\mathbf{k}$. In this case the change in the coordinate is $\delta \mathbf{r} = \phi \mathbf{k} \times \mathbf{r}$ and the change in the velocity is $\delta \mathbf{v} = \phi \mathbf{k} \times \mathbf{v}$. The resulting change in L is then

$$\delta L = \sum_b \left(\frac{\partial L}{\partial \mathbf{r}_b} \cdot \delta \mathbf{r} + \frac{\partial L}{\partial \mathbf{v_b}} \cdot \delta \mathbf{v} \right). \tag{5.14}$$

If we use Lagrange's equation and note that

$$\delta \mathbf{v} = \frac{d}{dt} \delta \mathbf{r}, \tag{5.15}$$

we can write

$$\delta L = \frac{d}{dt} \sum_b \frac{\partial L}{\partial \mathbf{v}_b} \cdot \delta \mathbf{r}_b \tag{5.16}$$

$$= \phi \mathbf{k} \cdot \frac{d}{dt} \sum_b \mathbf{r}_b \times \frac{\partial L}{\partial \mathbf{v}_b}. \tag{5.17}$$

But δL must vanish for arbitrary small $\phi \mathbf{k}$. We therefore conclude that

$$\sum_b \mathbf{r}_b \times \frac{\partial L}{\partial \mathbf{v}_b} = \sum_b m_b \mathbf{r}_b \times \mathbf{v}_b, \tag{5.18}$$

is constant. This quantity is the angular momentum.

Exercise 5.1. Suppose a charged particle moves in the field of a wire bent into a uniform helix. The potential has the following symmetry: if a shift is made along the axis of the helix by α and simultaneously, a rotation by ϕ is made about the axis where $\phi = p\alpha$ (the constant p is determined by the winding of the helix) show that the invariant quantity is a linear combination of the angular momentum about the axis of symmetry and the linear momentum along the axis of symmetry. ∎

Circulation

Another interesting conservation law is circulation. Suppose we consider a fluid where all the particles have the same mass. Imagine a necklace of particles like those illustrated in Figure 2.1. If the particles have the same entropy (so the necklace lies in a constant entropy surface) then nothing will change if each particle is shifted to its neighbour's positions always moving in the same sense around the necklace. The dynamics should therefore be unchanged. We can interpret this as requiring the change in the Lagrangian to be zero.
In this case, if the particles on the necklace are denoted by ℓ then the change in position and velocity of the ℓth particle will be $(\mathbf{r}_{\ell+1} - \mathbf{r}_\ell)$ and $(\mathbf{v}_{\ell+1} - \mathbf{v}_\ell)$ respectively. The change in the Lagrangian to first order is then

$$\delta L = \sum_\ell \left(\frac{\partial L}{\partial \mathbf{r}_\ell} \cdot \delta \mathbf{r}_\ell + \frac{\partial L}{\partial \mathbf{v}_\ell} \cdot \delta \mathbf{v}_\ell \right) \tag{5.19}$$

where now the summation only applies to the particles around the necklace. Using the previous expressions for $\delta\mathbf{r}$ and $\delta\mathbf{v}$ together with Lagrange's equations results in

$$\delta L = \frac{d}{dt} \sum_\ell \mathbf{v}_\ell \cdot (\mathbf{r}_{\ell+1} - \mathbf{r}_\ell) = 0, \tag{5.20}$$

and we deduce that

$$C = \sum_j \mathbf{v}_\ell \cdot (\mathbf{r}_{\ell+1} - \mathbf{r}_\ell), \tag{5.21}$$

is constant and this is true regardless of the necklace in the constant entropy surface. This result is a discrete version of the conservation of circulation of Kelvin's theorem which states that for a fluid which has no dissipation, and the pressure is a function of the density, the circulation

$$C_K = \oint \mathbf{v} \cdot \mathbf{dr}, \tag{5.22}$$

is constant. The integration is around any closed loop. Therefore, by contrast with the conservation of momentum, the circulation invariant is really an infinite number of invariants, one for each loop. Our result is, in general, only approximate because the changes in position and velocity to get from one place in the necklace to its neighbour are discrete whereas exact conservation is only true when infinitesimal transformations are relevant.

Exercise 5.2. Show that we get the same result, but with opposite sign, by going around the necklace in the opposite sense. Combine the two and take account of sign to show that

$$\frac{1}{2} \sum_\ell \mathbf{v}_\ell \cdot (\mathbf{r}_{\ell+1} - \mathbf{r}_{\ell-1}), \tag{5.23}$$

which is a more accurate estimate of the circulation. ∎

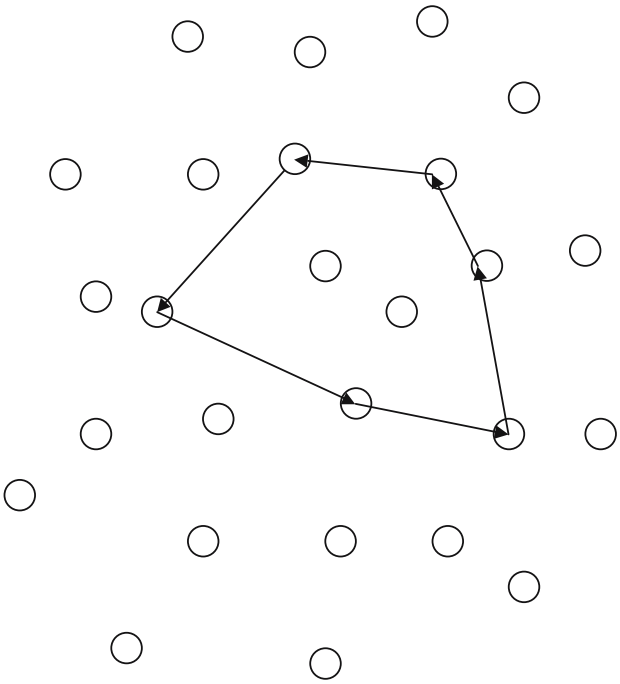


Figure 5.1. The necklace transformation.

Although our expression for the circulation is only approximate it is exactly true in some circumstances as shown in the following exercise.

Exercise 5.3. Suppose the pressure is zero and the fluid moves in a quadratic potential Φ analogous to the Toy star potential considered in the previous section. We define the ith coordinate of any particle by x^i and define the potential by

$$\Phi = \frac{1}{2}\sum_i \sum_j A^{ij} x^i x^j, \tag{5.24}$$

where A^{ij} denote the coefficients of a symmetric matrix. These coefficients may depend on time but, if they do, energy is no longer conserved. The equation of motion for particle ℓ is then

$$m\frac{dv_\ell^j}{dt} = -A^{ij} x_\ell^j. \tag{5.25}$$

Use the definition of C to show that

$$\frac{dC}{dt} = -\sum_\ell A^{ij} x_\ell^j (x_{\ell+1}^i - x_{\ell-1}^i) + \sum_\ell v_\ell^i (v_{\ell+1}^i - v_{\ell-1}^i). \tag{5.26}$$

Show that this vanishes. ∎

The previous analysis assumes that each particle in the necklace has the same entropy. What happens if the necklace has particles with different entropies, and these entropies remain fixed during the motion? If we consider the shift of particles around the necklace the particle ℓ will have to change its entropy when it gets to the neighbouring position. As a consequence δL contains the following entropy term

$$\sum_\ell \frac{\partial L}{\partial s}\delta s = -\sum_\ell \left(\frac{\partial u}{\partial s}\right)_\rho \delta s = -\sum_\ell T_\ell(s_{\ell+1} - s_\ell) \tag{5.27}$$

where T is the temperature.
If, following Eckart (1960), we define a quantity κ by

$$\frac{d\kappa}{dt} = T. \tag{5.28}$$

The extra term can be written, recalling that the entropy of each particle is constant, as

$$\frac{d}{dt}\sum_\ell \kappa_\ell(s_{\ell+1} - s_\ell). \tag{5.29}$$

As before we can take advantage of the fact that we can go around the necklace in either direction, to write this as

$$\frac{1}{2}\frac{d}{dt}\sum_\ell \kappa_\ell(s_{\ell+1} - s_{\ell-1}). \tag{5.30}$$

From the fact that δL should vanish for the necklace transformation we infer the conservation of

$$\frac{1}{2}\sum_\ell \mathbf{v}_\ell \cdot (\mathbf{r}_{\ell+1} - \mathbf{r}_{\ell-1}) - \frac{1}{2}\sum_\ell \kappa_\ell(s_{\ell+1} - s_{\ell-1}), \tag{5.31}$$

where the second term may be called the thermodynamic circulation. The continuum limit of this conserved quantity is

$$C = \oint \mathbf{v}\cdot\mathbf{dr} - \oint \kappa ds. \tag{5.32}$$

The quantity κ is monotonically increasing. Therefore, if we consider C for a loop that involves particles with different entropies, the thermal contribution will force the normal velocity based circulation to change continually. This suggests that such systems can become unstable quite easily.

5.3 The Lagrangian with Constraints

In the simplest form of the SPH equations ρ is defined by a summation over kernels and this means it is a function of the coordinates leading to

the equations of motion given above. However, as we have seen earlier, there may be advantages in working with the continuity equation written in a non-standard way. For example, we can write the SPH continuity equation as

$$\frac{d\rho_a}{dt} = -\rho_a \sum_b \frac{m_b}{\rho_b} \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{5.33}$$

Suppose now we want to use our Lagrangian (5.9). If we go back to the original action principle it requires that the action

$$S = \int L \, dt, \tag{5.34}$$

is stationary for arbitrary and infinitesimal variations $\delta\mathbf{r}$ in the coordinates and corresponding variations $\delta\mathbf{v}$ in the velocities. These variations are related by

$$\frac{d\delta\mathbf{r}}{dt} = \delta\mathbf{v}. \tag{5.35}$$

Suppose then that the only non-zero variation is $\delta\mathbf{r}_a$. The first order change in S is

$$\delta S = \int \left(m_a \mathbf{v}_a \cdot \delta\mathbf{v}_a - \sum_b m_b \frac{\partial u(\rho_b, s)}{\partial \rho_b} \frac{\delta\rho_b}{\delta\mathbf{r}_a} \cdot \delta\mathbf{r}_a \right) dt, \tag{5.36}$$

where

$$\frac{\delta\rho_b}{\delta\mathbf{r}_a} \tag{5.37}$$

denotes the Lagrangian change in ρ_b when the position of particle a changes by $\delta\mathbf{r}_a$at time t. From (5.33) we get

$$\delta\rho_b = -\rho_b \sum_c \frac{m_c}{\rho_c} (\delta\mathbf{r}_b - \delta\mathbf{r}_c) \cdot \nabla_b W_{bc}, \tag{5.38}$$

and therefore

$$\frac{\delta\rho_b}{\delta\mathbf{r}_a} = -\rho_b \sum_c \frac{m_c}{\rho_c} (\delta_{ab} - \delta_{ac}) \nabla_b W_{bc}, \tag{5.39}$$

where δ_{ab} is the Kronecker delta which is 1 if a equals b and zero otherwise. If we substitute this expression into the integral for δS, we find

$$\delta S = \int (m_a \mathbf{v}_a \cdot \delta\mathbf{v}_a - m_a \sum_b m_b \frac{(P_a + P_b)}{\rho_a \rho_b} \nabla_a W_{ab} \cdot \delta\mathbf{r}_a) \, dt. \tag{5.40}$$

If we now integrate the velocity term by parts recalling that $d(\delta\mathbf{r})/dt = \delta\mathbf{v}$, we get

$$\delta S = m_a \int \left(-\frac{d\mathbf{v}_a}{dt} - \sum_b m_b \frac{(P_a + P_b)}{\rho_a \rho_b} \nabla_a W_{ab} \right) \cdot \delta\mathbf{r}_a \, dt. \tag{5.41}$$

Since this must vanish for arbitrary $\delta \mathbf{r}_a$ we conclude that

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \frac{(P_a + P_b)}{\rho_a \rho_b} \nabla_a W_{ab}. \tag{5.42}$$

This is the acceleration equation that is consistent with the continuity equation (5.33). The Lagrangian basis for this choice of the SPH acceleration equation was first pointed out by Bonet.

Exercise 5.4. Generalize this procedure by writing the continuity equation as

$$\frac{d\rho}{dt} = -\left(\frac{\rho}{\Phi}\right) \Phi \nabla \cdot \mathbf{v}, \tag{5.43}$$

where Φ is an arbitrary function. Show that

$$\frac{d\rho}{dt} = -\frac{\rho}{\Phi}(\nabla \cdot (\Phi \mathbf{v}) - \mathbf{v} \cdot \nabla \Phi). \tag{5.44}$$

The SPH form of this equation is

$$\frac{d\rho_a}{dt} = -\frac{\rho_a}{\Phi_a} \sum_b \frac{m_b}{\rho_b} \Phi_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{5.45}$$

∎

If the effect of this constraint on the Lagrangian is calculated following the previous example we find that the acceleration equation consistent with the continuity equation is

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b \frac{m_b}{\rho_a \rho_b} \left(\frac{P_a \Phi_b}{\Phi_a} + \frac{P_b \Phi_a}{\Phi_b} \right) \nabla_a W_{ab}. \tag{5.46}$$

If we choose $\Phi = \rho$ then we recover our first form of the equation of the acceleration equation. If we choose $\Phi = 1$ we recover the second form. If we choose $\Phi = \sqrt{P}$, then the acceleration equation becomes

$$\frac{d\mathbf{v}_a}{dt} = -2 \sum_b m_b \frac{\sqrt{P_a P_b}}{\rho_a \rho_b} \nabla_a W_{ab}, \tag{5.47}$$

which was used by Springel and Hernquist (2002), but they used an inconsistent continuity equation.

5.4 Resolution Varying in Space and Time

In the SPH formulation the density of particle a can be written

$$\rho_a = \sum_b m_b W_{ab}(h_a). \tag{5.48}$$

In many SPH simulations h_a is chosen so that a particle a has a specified number of neighbours. However, to retain the Lagrangian formulations we need to specify h as a function of the coordinates, and this is most easily done by assuming h_a is a function of ρ_a which we denote by $H(\rho_a)$ or H_a. In many astrophysical calculations $H_a \propto (1/\rho_a^{1/d})$ where the number of dimensions is d, but a more general function could be used. For example, to prevent arbitrarily large h when ρ becomes very small we could choose

$$H_a = \frac{A}{1 + B\rho_a^{1/d}},$$

where A and B are constants. Furthermore, while the usual practice is to estimate ρ_a at a given time using the value of h_a from a previous time, it would be possible to calculate ρ_a from (5.48) with h_a a function of ρ_a. This idea was suggested recently by Bonet (2001), and when implemented it gives ρ_a and h_a as functions of the coordinates. Equation (5.48) is a nonlinear function for ρ_a and it can be solved by any standard root solving algorithm using, for example, the value at the previous time step as a first estimate. As an example Figure 5.2 shows the SPH density calculated in this self consistent way for the case of a Gaussian density

$$\rho(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}, \tag{5.49}$$

with total mass 1. We set up 51 SPH particles with equal mass m in the following way. The first particle was placed at $x_0 = 0$ and given $h_0 = 1.3m/\rho(0)$. The next particle was placed at a distance $\Delta x_1 = m/\rho(0)$, then particle 2 at a distance Δx_2 to the right of particle 1 where

$$\rho(1)(\Delta x_1 + \Delta x_2) = m, \tag{5.50}$$

and this was continued up to particle 26. Finally the coordinates of all particles except particle 1 were reflected about the origin. The values of $h_j = 1.3\Delta x_j$. The factor 1.3 could easily be replaced by any number between 1 and 1.5. With these values of h we can expect an SPH sum to give a good estimate of the density by means of the SPH summation. For the self consistent estimate we choose

$$h = 1.3m/\rho. \tag{5.51}$$

All the initial estimates of h at the particles use this formula with the exact density which we can expect to differ slightly from the SPH estimate. The estimates of ρ and therefore h for each particle were found by finding the solution of the equation (5.48) with a Newton-Raphson method. To get convergence to 5 figures no more than 2 iterations were needed. The results are shown in Figure 2.2 where the solid line denotes the exact density.

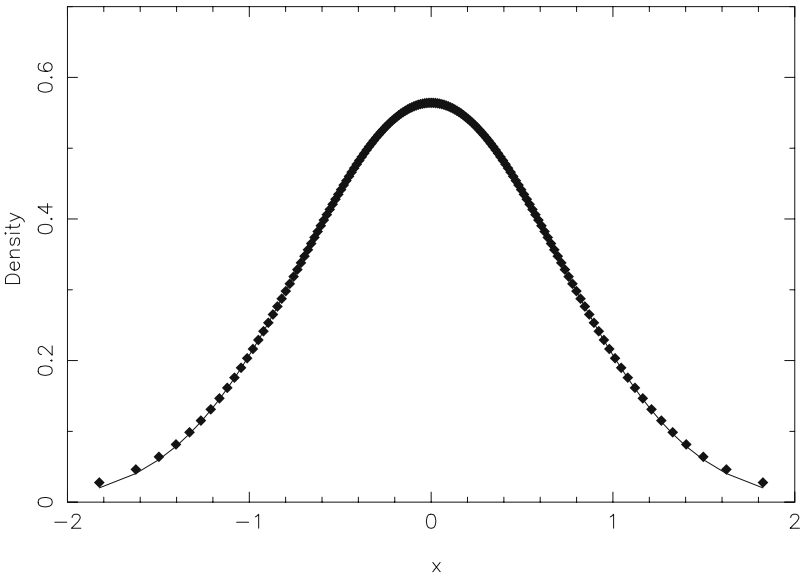


Figure 5.2. The self consistent density is chosen for a set of SPH particles. The line denotes the exact Gaussian density.

The equations of motion follow from varying the action keeping the entropy constant. From Lagrange's equations for particle a we find

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{\partial u}{\partial \rho}\right)_s \frac{\partial \rho_b}{\partial \mathbf{r}_a}. \tag{5.52}$$

From (5.1)

$$\frac{\partial \rho_b}{\partial \mathbf{r}_a} = \sum_c m_c \nabla_a W_{ac}(h_a)\delta_{ab} - m_a \nabla_b W_{ab}(h_b) + \sum_c m_c \frac{\partial W_{bc}}{\partial h_b}\frac{\partial h_b}{\partial \mathbf{r}_a}, \tag{5.53}$$

which we can write as

$$\Omega_b \frac{\partial \rho_b}{\partial \mathbf{r}_a} = \sum_c m_c \nabla_a W_{ac}(h_a)\delta_{ab} - m_a \nabla_b W_{ab}(h_b), \tag{5.54}$$

where the gradient of W_{ab} is taken keeping h constant and

$$\Omega_b = 1 - H_b' \sum_c m_c \frac{\partial W_{bc}(h_b)}{\partial h_b}. \tag{5.55}$$

Here H_b' denotes $\partial H_b/\partial \rho_b$. If the density variation is smooth then $\Omega = 1 + O(h^2)$.

Using the first law of thermodynamics the acceleration equation (5.52) with (5.54) becomes

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\Omega_a \rho_a^2} \nabla_a W_{ab}(h_a) + \frac{P_b}{\Omega_b \rho_b^2} \nabla_a W_{ab}(h_b) \right). \tag{5.56}$$

As in the case where h was held constant, this equation conserves linear and angular momentum as we would expect from the symmetry of the Lagrangian. The equation of motion (5.9) is exactly the same as the equation of motion due to Springel and Hernquist (2001) who introduce constraints on h with a Lagrange multiplier. The important point to note is that despite the fact that the resolution can vary, the equation of motion is not much more complicated than before. The only extra work is to calculate h and ρ self consistently but this can usually be done efficiently because the Newton-Raphson method can be used and a good starting value is always available from the previous step.

6 SPH Heat Conduction

The previous sections have been concerned with non-dissipative dynamics. While there are many interesting problems which involve no dissipation, many of the most important problems involving fluids depend on dissipation which may take the form of heat conduction, diffusion of matter, viscous processes, ohmic heating or friction. In this section we consider the first. In working out SPH forms of the equations we will be guided, as before, by satisfying general physical principles rather than focus on the order of the errors. The first of these principles is that all dissipative processes result in an increase in the entropy of the system. Second, provided the boundaries are sealed the system will conserve energy and matter. Our aim is to design SPH equations which incorporate these principles. The principal applications we have in mind are the shock dynamics of gases.

The equations describing these dissipative processes are parabolic equations which involve second derivatives of spatially varying quantities such as temperature. In an SPH simulation the particles become disordered and calculating second derivatives by differentiating the interpolation formula often results in unacceptable error. To get around this difficulty we return once more to integral interpolants, this time with the aim of constructing second derivatives which are not sensitive to particle disorder. We will find it possible to do this and retain the physical principles mentioned earlier.

The heat conduction equation is

$$\frac{du}{dt} = \frac{1}{\rho} \nabla(\kappa \nabla T), \tag{6.1}$$

where u is the thermal energy per unit mass, T is the absolute temperature, ρ the density, κ the coefficient of thermal conductivity (which in general varies

in space), and d/dt is the derivative following the motion. For simplicity we assume that du can be replaced by $\mathcal{C}_p dT$ where $\mathcal{C}_p$ is the specific heat at constant pressure. We will assume that $\mathcal{C}_p$ is constant. The heat conduction equation then takes the form

$$\mathcal{C}_p \frac{dT}{dt} = \frac{1}{\rho}\nabla(\kappa \nabla T). \tag{6.2}$$

We could approximate this equation by working out spatial derivatives using the interpolation formula given earlier. However, as mentioned above, this can lead to significant errors.

6.1 Derivatives from Integrals

An alternative SPH form of this equation for the change of temperature T_a at particle a can be found by constructing the second derivative term $\nabla(\kappa\nabla T)$ using an integral. To see this consider the integral

$$\int (\kappa(x) + \kappa(x'))\,(T(\mathbf{r}') - T(\mathbf{r}))\,\frac{(\mathbf{r}-\mathbf{r}')\cdot\nabla W(\mathbf{r}-\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|^2}\,\mathbf{dr}', \tag{6.3}$$

where $\mathbf{dr}'$ denotes a volume element.
For convenience we set $\mathbf{q}F(|\mathbf{q}|) = \nabla W(\mathbf{q}, h)$, and we note that $F \leq 0$. The integral then becomes

$$\int (\kappa(x) + \kappa(x'))\,(T(\mathbf{r}') - T(\mathbf{r}))\,F(|\mathbf{r}-\mathbf{r}'|)\,\mathbf{dr}'. \tag{6.4}$$

If we expand the functions of $\mathbf{r}'$ about $\mathbf{r}$, and keep the dominant terms, the integral reduces to $-\nabla\cdot(\kappa\nabla T)$. In making this approximation the reader will note that, for example in two dimensions, integrals like

$$\int \left(\frac{\partial^2 T}{\partial x^2}(x-x')^2 + \frac{\partial^2 T}{\partial y^2}(y-y')^2\right) F(|\mathbf{r}-\mathbf{r}'|)\,\mathbf{dr}' \tag{6.5}$$

occur. From symmetry in a two dimensional space we can equate the integrals as follows

$$\int (x-x')^2 F(|\mathbf{r}-\mathbf{r}'|)\mathbf{dr}' = \int (y-y')^2 F(|\mathbf{r}-\mathbf{r}'|)\mathbf{dr}' = \frac{1}{2}\int (\mathbf{r}-\mathbf{r}')^2 F(|\mathbf{r}-\mathbf{r}'|)\mathbf{dr}'. \tag{6.6}$$

Substituting these results into the expression (6.5) it becomes

$$\frac{1}{2}\nabla^2 T \int (\mathbf{r}-\mathbf{r}')^2 F(|\mathbf{r}-\mathbf{r}'|)\,\mathbf{dr}'. \tag{6.7}$$

If we define $\mathbf{q} = \mathbf{r}' - \mathbf{r}$, and make use of the definition of $F(|\mathbf{q}|)$ we can write the integral in the previous expression as

$$\int \mathbf{q}\cdot\nabla_q W(\mathbf{q}, h)\,\mathbf{dq} = -\int W\nabla\cdot\mathbf{q}\,\mathbf{dq} = -d, \tag{6.8}$$

where d is the number of dimensions which in this case is 2 which cancels the $1/2$ from the series expansion. In 3 dimensions a similar cancellation occurs. If we now write the integral using the usual rule for integral interpolants we find that the heat conduction equation becomes

$$\mathcal{C}_{p,a}\frac{dT_a}{dt} = \sum_b m_b \frac{(\kappa_a + \kappa_b)(T_a - T_b)}{\rho_a \rho_b} F_{ab}. \tag{6.9}$$

The errors in the integral formulation is $O(h^2)$, but there are further errors due to approximating the integral by a summation. The following exercise is an elementary example of the equations satisfying a physical principle.

Exercise 6.1. Does heat travel from a hot particle to a cold particle? From the fact that the contribution of particle b to the rate of change of temperature of particle a is

$$m_b \frac{(\kappa_a + \kappa_b)(T_a - T_b)}{\rho_a \rho_b} F_{ab}. \tag{6.10}$$

Show that, if $T_a > T_b$ then heat flows from a to b as expected physically (that is the temperature of a decreases and that of b increases). Note that if we had constructed the heat conduction equation by second derivatives of an interpolated temperature we could not guarantee this. ∎

6.2 Does the Entropy Increase?

We can also check if the change in the entropy per unit mass s increases. Assuming there are no other processes operating we can write the change of s_a as

$$T_a \frac{ds_a}{dt} = \frac{dQ}{dt} = \sum_b m_b \frac{(\kappa_a + \kappa_b)(T_a - T_b)}{\rho_a \rho_b} F_{ab}, \tag{6.11}$$

where Q denotes the heat per unit mass. Multiplying by m_a and summing gives the change in the total entropy

$$\frac{dS}{dt} = \sum_a m_a \frac{ds_a}{dt} = \sum_a \sum_b m_a m_b \frac{(\kappa_a + \kappa_b)(T_a - T_b)}{\rho_a \rho_b T_a} F_{ab}. \tag{6.12}$$

In the summation we can interchange the dummy suffices a and b. If this is added to the original summation, compensating by a factor $1/2$, we get

$$\frac{dS}{dt} = \sum_a m_a \frac{ds_a}{dt} = \frac{1}{2}\sum_a \sum_b m_a m_b \frac{(\kappa_a + \kappa_b)}{\rho_a \rho_b} \left[(T_a - T_b)\left(\frac{1}{T_a} - \frac{1}{T_b}\right)\right] F_{ab}. \tag{6.13}$$

The factor is square brackets is ≤ 0 and since F_{ab} is also ≤ 0, the terms in the summation are all positive. The entropy therefore increases as a result of heat conduction regardless of the position and temperature of the particles.

Exercise 6.2. The total thermal energy E_{th} is

$$\sum_a m_a C_{p,a} T_a, \tag{6.14}$$

Show that

$$\frac{dE_{th}}{dt} = \sum_a \sum_b m_b m_b \frac{(\kappa_a + \kappa_b)(T_a - T_b)}{\rho_a \rho_b} F_{ab}. \tag{6.15}$$

which vanishes because the summed quantity is antisymmetric in a and b. The equation therefore conserves energy. ∎

The result of the previous exercise means is that if we have a set of SPH particles exchanging heat amongst themselves the total thermal energy remains constant because what we have described is an adiabatic enclosure. To lose heat the SPH particles must communicate with other particles which convey heat away or bring it in. A typical way to do this is to suppose the SPH particles are in a container with walls kept at a fixed temperature. The walls can be simulated by SPH particles. Imagine that we solve our heat conduction equation by some suitable time stepping scheme. At each step the temperature of the wall particles will change corresponding to a loss or gain of heat from the system. We can then set the temperature of the wall particles back to the prescribed value at the end of each step. The heat loss or gain of the entire system can be easily determined by summing the changes which occurred for each wall particle. If the wall, or part of the wall, is adiabatic the wall particles are treated like other particles of the fluid system.

6.3 Discontinuous Thermal Conductivity

When there is more than one material the thermal conductivity may jump discontinuously. The thermal boundary condition at the interface between the two materials is that the flux of heat is continuous and, in finite difference methods, this requires solving the difficult problem of estimating the gradient at a surface which may pass through the cells and vary with time. In the SPH calculation Cleary and Monaghan (1999) showed that the same result could be achieved by replacing the term

$$\kappa_a + \kappa_b, \tag{6.16}$$

by

$$\frac{4\kappa_a \kappa_b}{\kappa_a + \kappa_b}. \tag{6.17}$$

That is, an arithmetic mean is replaced by an harmonic mean. The reason for this is as follows. For convenience suppose that we are using a finite difference scheme in one dimension with the interface at $x = 0$. Let κ be κ_L for $x < 0$ and κ_R for $x > 0$ and assign the temperature T^* to the interface between

point j (the last point of the material on the left) and $(j+1)$ which is the first point of the material on the right of the interface. These two points are assumed to be separated by Δx with the interface half way between them. Then, for the heat flux to be continuous, we require

$$\kappa_L \frac{T^* - T_j}{\Delta x/2} = \kappa_L \frac{T_{j+1} - T^*}{\Delta x/2}. \tag{6.18}$$

Solving this for T^* we get

$$T* = \frac{\kappa_L T_{j+1} + \kappa_R T_j}{\kappa_L + \kappa_R}. \tag{6.19}$$

To solve the heat conduction equation for the material with $x < 0$ we approximate the finite difference heat conduction equation by

$$C_p \frac{dT}{dt} = \kappa_L \left(\frac{T^* - T_j}{\Delta x/2} - \frac{T_j - T_{j-1}}{\Delta x/2} \right). \tag{6.20}$$

If we now substitute for T^* this equation becomes

$$C_p \frac{dT}{dt} = \left(\frac{2\kappa_L \kappa_R}{\kappa_L + \kappa_R} \frac{T_{j+1} - T_j}{\Delta x/2} - \kappa_L \frac{T_j - T_{j-1}}{\Delta x/2} \right). \tag{6.21}$$

This equation shows that to preserve heat flux all we need to do is include the first point of the adjoining region and replace the coefficient of thermal conductivity. The SPH conduction equation then becomes

$$\mathcal{C}_{p,a} \frac{dT_a}{dt} = \sum_b m_b \frac{4\kappa_a \kappa_b}{(\kappa_a + \kappa_b)} \frac{(T_a - T_b)}{\rho_a \rho_b} F_{ab}. \tag{6.22}$$

We can use this form of the equation to deal with heat conduction of two fluids with an interface that may have complicated geometry and may change with time. A simple application of the SPH algorithm is to calculate the temperature conduction in a composite one dimensional medium. In Figure 6.1 we show the temperature distance variation when the density and specific heats are the same, but the thermal conductivity for $x < 0$ is 1.0 and for $x \geq 0$ is 10.0. The temperature at the end points is fixed. At $x = -1$ the temperature is 0.1 and at $x = 1.0$ the temperature is 1. The results are shown after 100 times steps (using a predictor corrector time stepping scheme Cleary and Monaghan (1999)). The continuous line shows the exact results (taken from Carslaw and Jaeger (1990)) and the symbols show the SPH results. The agreement is clearly very good.

6.4 Diffusion of Matter

The diffusion of matter is similar to the diffusion of heat. We consider a liquid which contains dissolved salt. The molecules of salt will diffusion from

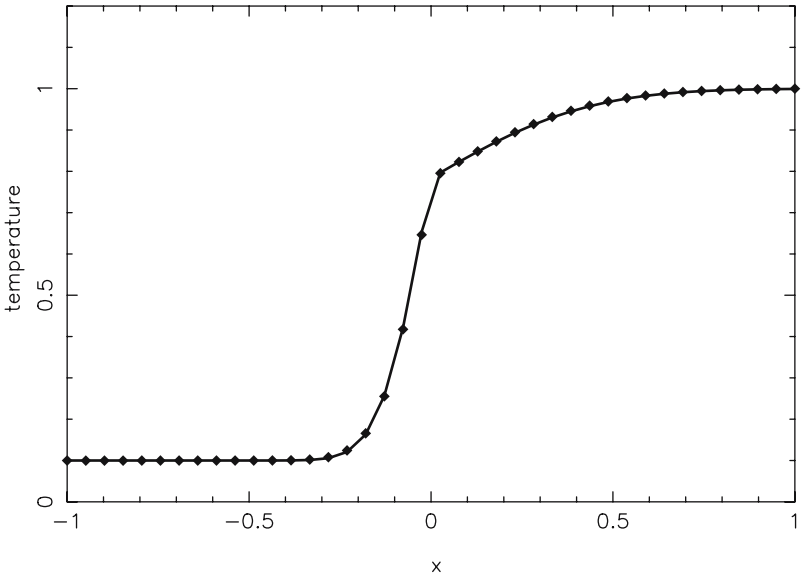


Figure 6.1. The temperature distance variation in a composite one dimensional system, where the thermal conductivity on the right is 10 times that on the left, and the left and right boundaries are kept at constant temperature.

places of high concentration to places of low concentration. We denote the concentration of salt by C so that the mass of salt in a mass M of liquid is CM. The diffusion of the salt is given by an equation similar in form to the heat conduction equation namely

$$\frac{dC}{dt} = \frac{1}{\rho}\nabla(D\nabla C). \tag{6.23}$$

where D is the diffusion coefficient with dimensions of kg/(m s). The SPH form of this equation is

$$\frac{dC_a}{dt} = \sum_b m_b \frac{4D_aD_b}{(D_a + D_b)} \frac{(C_a - C_b)}{\rho_a\rho_b} F_{ab}. \tag{6.24}$$

where m_a is the mass of the liquid particle that contains the salt and D is the coefficient of diffusion. Note that we have written the diffusion coefficients to ensure that the flux of material across an interface is constant in the same way as done previously for the flux of heat.
The total amount of matter in an isolated region is

$$\sum_a m_a C_a \tag{6.25}$$

and the SPH equation shows that this remains constant.

Exercise 6.3. When the composition changes there is a further contribution to the entropy. To deduce this by divide (6.26) by C_a then sum over a followed by an interchange of labels. Show that

$$\frac{d}{dt}\sum_a m_a \ln C_a = -\sum_a \sum_b m_a m_b \frac{4D_a D_b}{(D_a + D_b)} \left(\frac{1}{C_a} - \frac{1}{C_b}\right) \frac{(C_a - C_b)}{\rho_a \rho_b} F_{ab} \geq 0, \tag{6.26}$$

which is the increase of entropy resulting from composition changes. ∎

7 Viscosity

We now consider how to include viscosity in the SPH equations. We begin by deriving an artificial viscosity which is suitable for shock calculations. As before we will be guided by physical principles of which the first are the conservation of linear and angular momentum. In addition we will require that the contributions of viscous dissipation to the thermal energy and the entropy are always positive. Our first viscosity was constructed by comparison with actual gas viscosities. However, for shock problems it is useful to take the Riemann type dissipative terms as a guide (Monaghan (1997)) and this has the advantage of reducing some of the arbitrary features of the first viscosity. Both forms of the viscosity are similar.

7.1 A Simple Artificial Shock Viscosity

In one dimension the Navier-Stokes acceleration equation is

$$\frac{dv}{dt} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{1}{\rho}\frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right), \tag{7.1}$$

where μ is the coefficient of viscosity. For a gas

$$\mu \sim \frac{1}{3}\rho \lambda c_s \tag{7.2}$$

where λ is the mean free path of the gas molecules and c_s is the speed of sound. We can write the equation of motion in the form

$$\frac{dv}{dt} = -\frac{1}{\rho}\frac{\partial}{\partial x}\left(P - \mu \frac{\partial v}{\partial x}\right), \tag{7.3}$$

which shows that when $\partial v/\partial x < 0$ that is, when the density is increasing, the viscous term acts like a positive pressure. When the density is decreasing, the viscous term acts as a negative pressure. With this result as a guide, we now write the SPH acceleration equation in the form

$$\frac{dv_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \Pi_{ab}\right) \frac{\partial}{\partial x_a} W_{ab}, \tag{7.4}$$

where Π_{ab} is the SPH viscous dissipation term. From our previous discussion we clearly need

$$\Pi \sim \frac{\mu}{\rho^2}\frac{\partial v}{\partial x}. \tag{7.5}$$

We guess that

$$\frac{\partial v}{\partial x} \sim \frac{v_a - v_b}{x_a - x_b}, \tag{7.6}$$

and that instead of the actual μ for the gas we can take

$$\mu = \alpha \rho_a c_a h, \tag{7.7}$$

since the natural length scale for communication is not the mean free path but the interaction distance of the SPH particles. The constant α is expected to be ~ 1 and, if our intuition is correct, we should be able to choose α in a way which is independent of the particular shock problem. Our first guess for the artificial viscosity is therefore

$$\Pi_{ab} = -\left(\frac{\alpha h c_a}{\rho_a}\right)\left(\frac{v_a - v_b}{x_a - x_b}\right). \tag{7.8}$$

However, to get conservation of momentum we need Π_{ab} to be symmetric in a and b. This is easy to do. We replace c_a, h and ρ_a as follows

$$c_a \to \frac{1}{2}(c_a + c_b) = \bar{c}_{ab}, \tag{7.9}$$

$$\rho \to \frac{1}{2}(\rho_a + \rho_b) = \bar{\rho}_{ab}, \tag{7.10}$$

$$h \to \frac{1}{2}(h_a + h_b) = \bar{h}_{ab}. \tag{7.11}$$

To prevent numerical problems when $v_a \neq v_b$ but $x_a = x_b$, we can write (with $v_{ab} = v_a - v_b$ and with the same notation for x_{ab})

$$\frac{v_a - v_b}{x_a - x_b} \to \frac{v_{ab} x_{ab}}{x_{ab}^2 + \eta^2}, \tag{7.12}$$

and this form is commonly used. The constant $\eta \sim 0.001h^2$ serves to smooth out any singularities resulting from $x_a = x_b$. However, it is simpler to just set the viscous term to zero if $x_a = x_b$ which, in practice, occurs rarely.
We can generalise to an arbitrary number of dimensions and write

$$\Pi_{ab} = -\left(\frac{\alpha \bar{h}_{ab}\bar{c}_{ab}}{\bar{\rho}_{ab}}\right)\left(\frac{\mathbf{v}_{ab}\cdot\mathbf{r}_{ab}}{r_{ab}^2 + \eta^2}\right). \tag{7.13}$$

A further generalization, which gives a higher order viscosity, is to multiply the previous viscosity by any power of

$$\frac{\mathbf{v}_{ab}\cdot\mathbf{r}_{ab}}{\bar{c}_{ab}}. \tag{7.14}$$

Finally we note that, for shock tube problems, it is usual to turn the viscosity on for approaching particles and turn it off for receding particles. In this way the viscosity is used for shocks and not rarefactions. Unfortunately in astrophysical calculations this rule means that the viscosity is turned on when the density is increasing in shock free regions, for example when gravity pulls gas together.

When the viscosity term Π_{ab} was first used (Monaghan and Gingold (1983)) it was found to work well for shocks of moderate strength. However, in astrophysical calculations involving colliding gas clouds, where the Mach number can be very high, it was found that particles from one cloud could stream between the particles of the other cloud. Generally this streaming is limited to a few particle spacings, and is therefore not a severe problem, but it should not occur at all. To prevent it an extra term was added to Π_{ab} which then took the form

$$\Pi_{ab} = -\left(\frac{\bar{h}_{ab}\mathbf{v}_{ab}\cdot\mathbf{r}_{ab}}{\bar{\rho}_{ab}(r_{ab}^2+\eta^2)}\right)(\alpha\bar{c}_{ab}-\beta\mu_{ab}) \tag{7.15}$$

where

$$\mu_{ab} = \frac{\bar{h}_{ab}\mathbf{v}_{ab}\cdot\mathbf{r}_{ab}}{r_{ab}^2+\eta^2}. \tag{7.16}$$

Good results have been obtained with the choice $\alpha = 1$ and $\beta = 2$.

Another form of the viscosity for shock problems can be found using ideas from Riemann solvers as a guide. This viscosity uses

$$\Pi_{ab} = -\frac{Kv_{\rm sig}(a,b)\mathbf{v}_{ab}\cdot\mathbf{r}_{ab}}{\bar{\rho}_{ab}|\mathbf{r}_{ab}|} \tag{7.17}$$

where K is a constant (typically 0.5) and $v_{\rm sig}(a,b)$ is a signal velocity (Monaghan (1997)). This signal velocity automatically includes a term equivalent to the β term in the previous viscosity.

7.2 Invariance Properties

A fundamental property of the fluid dynamical equations is that they are Galilean invariant. That is, if we shift to a coordinate frame moving with constant velocity $\mathbf{V}$ the equations should be unchanged. This is the case for Π_{ab} because it involves differences of velocity and the shift to the new frame simply replaces $\mathbf{v}_a$ by $\mathbf{v}_a - \mathbf{V}$ and $\mathbf{v}_b$ is replaced in the same way. The difference $\mathbf{v}_{ab}$ is unchanged. Similarly if we shift the origin of the coordinate system to $\mathbf{R}$ the equations are unchanged.

If the fluid is rigidly rotating $\mathbf{v}_a = \Omega\times\mathbf{r}_a$, where Ω is the angular velocity. Substitution into Π_{ab} shows that the viscous term disappears in this case as expected.

7.3 Effective Pressure and Viscosity

If particles a and b are approaching each other

$$\mathbf{v}_{ab} \cdot \mathbf{r}_{ab} \leq 0, \tag{7.18}$$

and $\Pi_{ab} \geq 0$ and the contribution to the pressure terms is positive. The viscosity therefore acts to slow down approaching particles. The reverse happens for receding particles.

Exercise 7.1. Take the dot product of the acceleration equation with $\mathbf{v}_a$, followed by multiplying by m_a and summing to get

$$\sum_a m_a \mathbf{v}_a \cdot \frac{d\mathbf{v}_a}{dt} = -\sum_a \sum_b m_a m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \Pi_{ab} \right) \mathbf{v}_a \cdot \nabla_a W_{ab}. \tag{7.19}$$

Note that the left hand side is the rate of change of total kinetic energy so the right hand side must be minus the rate of change of total thermal energy. By interchanging a and b on the right hand side and combining the result with the original right hand side (compensating by a factor 1/2) show that the thermal energy equation is

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab} + \frac{1}{2} \sum_a m_a \sum_b m_b \Pi_{ab} \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{7.20}$$

■

7.4 The Sign of the Dissipation Term

To determine the sign of the SPH dissipation term obtained in the previous exercise we begin by noting that we can write $\nabla_a W_{ab} = \mathbf{r}_{ab} F_{ab}$ where $F_{ab} \leq 0$. Then

$$\Pi_{ab} \mathbf{v}_{ab} \cdot \nabla_a W_{ab} = \Pi_{ab} \mathbf{v}_{ab} \cdot \mathbf{r}_{ab} F_{ab}. \tag{7.21}$$

Referring now to the definition of Π_{ab}, for example to (5.39), we find the viscous dissipation is

$$\Pi_{ab} = -\left(\frac{\alpha \bar{h}_{ab} \bar{c}_{ab}}{\bar{\rho}_{ab}} \right) \frac{F_{ab} (\mathbf{v}_{ab} \cdot \mathbf{r}_{ab})^2}{r_{ab}^2 + \eta^2} \tag{7.22}$$

which is ≥ 0. This confirms that our SPH dissipation increases the thermal energy as it should.

Exercise 7.2. Show that the rate of change of entropy, s, due to viscous dissipation is positive. Begin with the thermodynamic equation

$$T \frac{ds}{dt} = du - \frac{P}{\rho^2} d\rho \tag{7.23}$$

whose SPH form this becomes

$$T_a \frac{ds_a}{dt} = \frac{1}{2} \sum_b m_b \Pi_{ab} \mathbf{v}_{ab} \cdot \nabla_a W_{ab}. \tag{7.24}$$

We showed above that the right hand side is ≥ 0. Since the temperature T is positive, the change to the entropy of any particle due to viscous dissipation is positive. ∎

8 Applications to Shock and Rarefaction Problems

We now have a set of equations which can be used for shock and rarefaction problems. In this section we show the result of applying these equations using a simple predictor corrector time stepping scheme (see Section 7 for details of this and other time stepping schemes).

8.1 Rarefaction Waves

The first case we consider is the rarefaction wave. This can be set up by placing SPH particles in the region $-0.5 \leq x \leq 0.5$. The separation Δx is uniform and the density $\rho = 1$. For this example we use 200 particles and set $\gamma = 1.4$, and the initial $h = 1.5\Delta x$, and the thermal energy/mass to be 2. We integrate the SPH acceleration, continuity and thermal energy equation. The viscosity is turned off for the rarefaction wave. In Figure 8.1 we show the velocity field for $x \geq 0$. The exact velocity field in the wave is a linear function of x shown by the solid line. The SPH velocity is very close to a linear function, and the slope is within ∼2 percent of the exact slope.
In Figure 8.2 we show the SPH density. This follows the exact curve (shown by the solid line) except in the low density region where an oscillation appears. In addition the last particle shows a jump in density. In Figure 8.3 we show the results for a higher resolution (500 particles in the domain $-0.5 \leq x \leq 0.5$). The oscillation and the jump in density remains. In Figure 8.4 we show the density calculated by integrating the continuity equation on odd steps and from the summation on even steps. The jump in density and small oscillations have now disappeared.

8.2 The Sod Shock Tube

We now consider the shock tube used by Sod (1978) as a test for numerical techniques. The system is one dimensional with uniform conditions one each side of a diaphragm which breaks at $t = 0$. To the left of the diaphragm ($x < 0$) the conditions are $\rho, P, v, \gamma = 1.0,\ 1.0,\ 0.0,\ 1.4$ and to the right (0.125, 0.1, 0.0, 1.4). The evolved system consists of (from the left), the undisturbed original conditions, a rarefaction, a contact discontinuity and

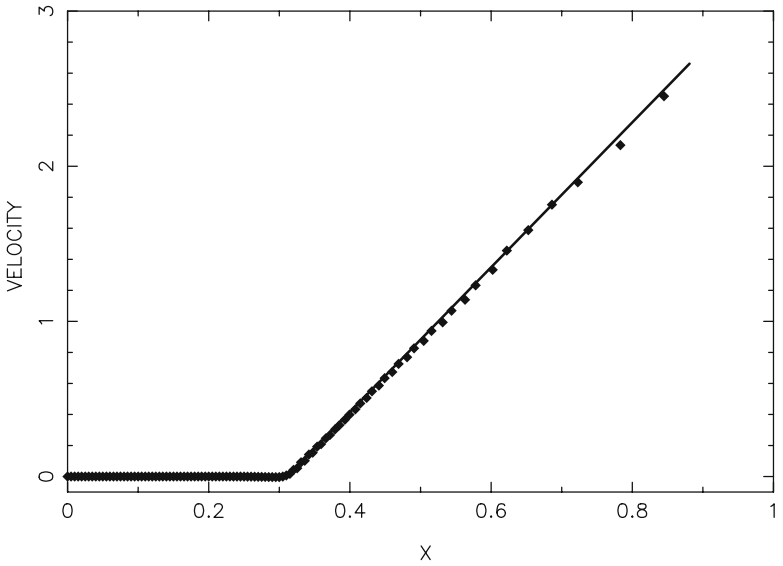


Figure 8.1. The velocity field for the one dimensional rarefaction waves from the expansion of uniform gas initially in the region $-0.5 \leq x \leq 0.5$. We show the results for the right half $x \geq 0$.The exact velocity field is shown by the solid line and the SPH results by the solid diamonds

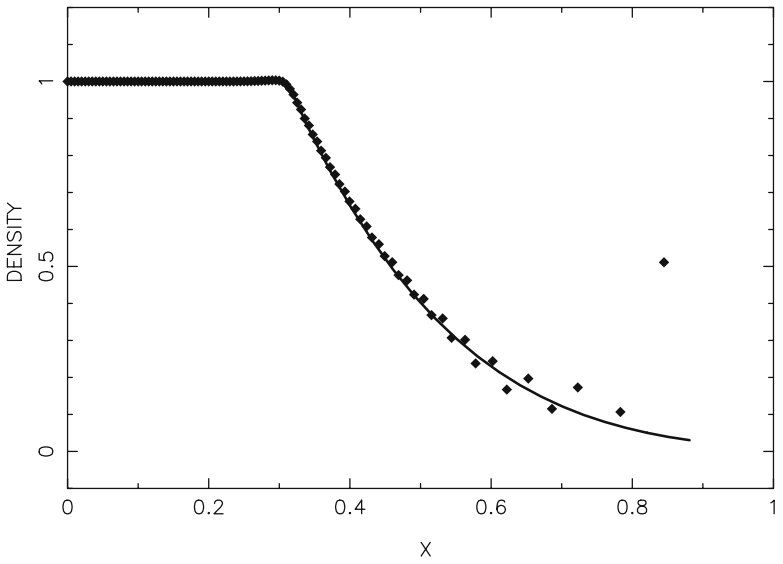


Figure 8.2. The density in the rarefaction waves calculated using the continuity equation. The exact run of density with distance is shown by the solid line and the SPH results by the solid diamonds. Note the jump in density for the last particle and the oscillation in the low density region.

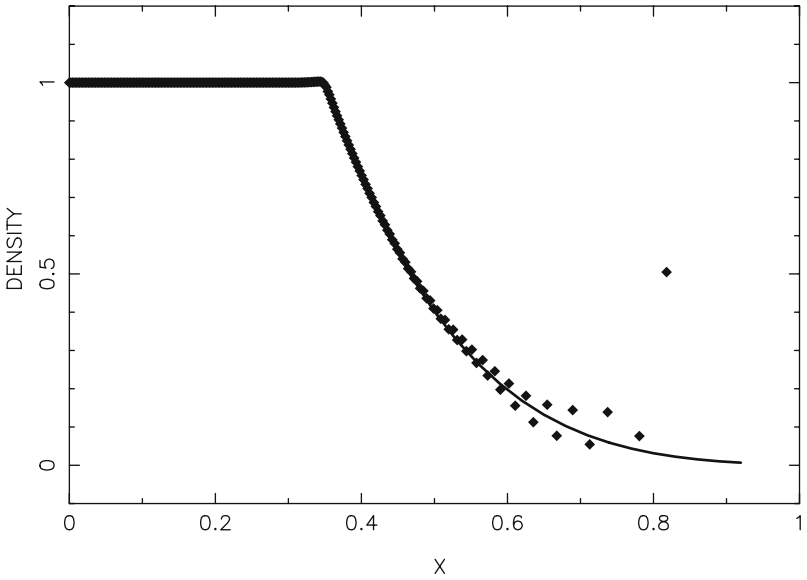


Figure 8.3. The density in the rarefaction waves calculated using the continuity equation with the same conditions as for the previous density but calculated with 500 particles over the domain.

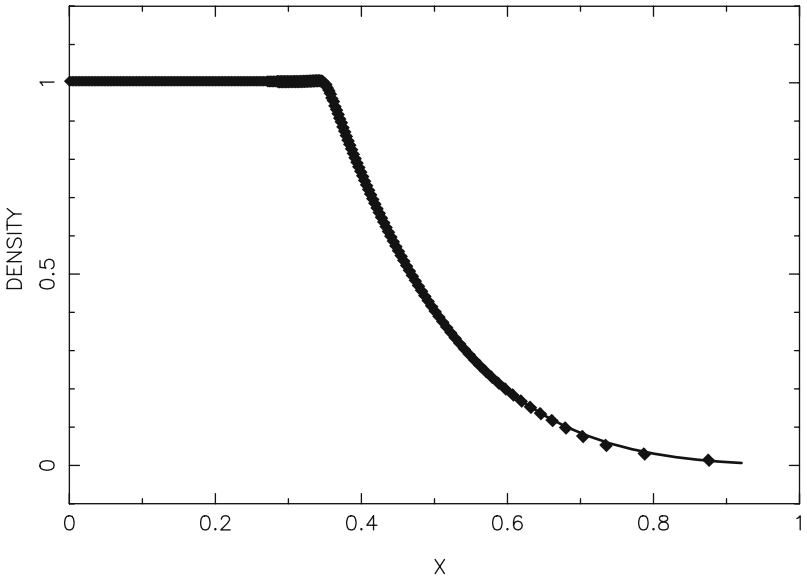


Figure 8.4. The density in the rarefaction waves calculated by using the continuity equation on odd steps and the summation on even steps. Note that the jump in density for the last particle and the oscillation in the low density region have disappeared.

a shock. Between the shock and the rarefaction the pressure and velocity are constant. The density and thermal energy change discontinuously at the contact discontinuity.
We use the viscosity (7.15) with $\alpha = 1$ and $\beta = 2$. Because the density changes we can choose to have the particles equi-spaced or equi-mass, or some other combination. For the present simulations we use equi-mass particles with spacing $\Delta x = 0.005$ on the far right hand side and spacing 0.125 this on the far left hand side. The mass of each particle is then $\rho_+ \Delta x = 0.62510^{-3}$. Because there is an initial discontinuity in all the properties other than the velocity, and because SPH is based on smoothing, we smooth the density and thermal energy at the interface. This means that to be consistent with the particles having constant mass and the density being smoothed we must smooth the spacing.
A simple way to smooth the variables is to define for any quantity A a value on the left A_ℓ, and on the right A_r and then define the smoothed A_S by

$$A_S = \frac{A_\ell + A_r e^{-xk}}{1 + e^{-xk}}, \tag{8.1}$$

where $k = 1/\Delta x$, Δx is the particle spacing on the far right (low density region), and we assume the discontinuity is at $x = 0$. The particle spacing is chosen in the following way. The initial space available to particle j is

$$\frac{1}{2}(x_{j+1} - x_j) + \frac{1}{2}(x_j - x_{j-1}) = \frac{1}{2}(x_{j+1} - x_{j-1}). \tag{8.2}$$

and, with density ρ_j for this particle we require that

$$\frac{1}{2}\rho_j(x_{j-1} - x_{j-2}) = m \tag{8.3}$$

The smoothing (8.1) is satisfactory, but it doesn't guarantee that the system will start with the correct conditions for the shock and we should not be surprised to find there is a perturbation to the solution due to the initial state.
In Figure 8.5 we show the exact and SPH velocity variation with distance x. The exact post-shock velocity is 0.926. The SPH value is 0.921, an error of 0.5 percent. The small bump in the velocity is to an unwanted change in the pressure across the contact discontinuity. The shock front is spread over several particle spacings, but because h and the spacing change across the shock the shock front is 3 of the particle spacings on the low density side. In Figure 8.6 we show the exact and SPH density ρ. The density between the contact discontinuity and the shock front is 0.263 to be compared with the exact value of 0.265.
In Figure 8.7 we show the thermal energy. The post shock thermal energy is 0.284 compared with the exact value of 0.286 and the exact and SPH values to between the rarefaction and the contact discontinuity are 1.78 and

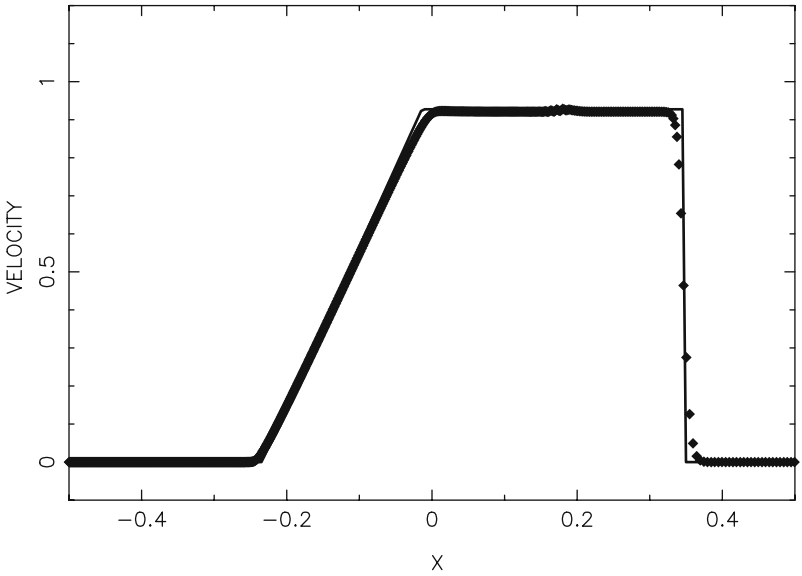


Figure 8.5. The velocity in the Sod shock tube problem. Note the slight deviation in the velocity associated with the small perturbation to the pressure at the contact discontinuity.

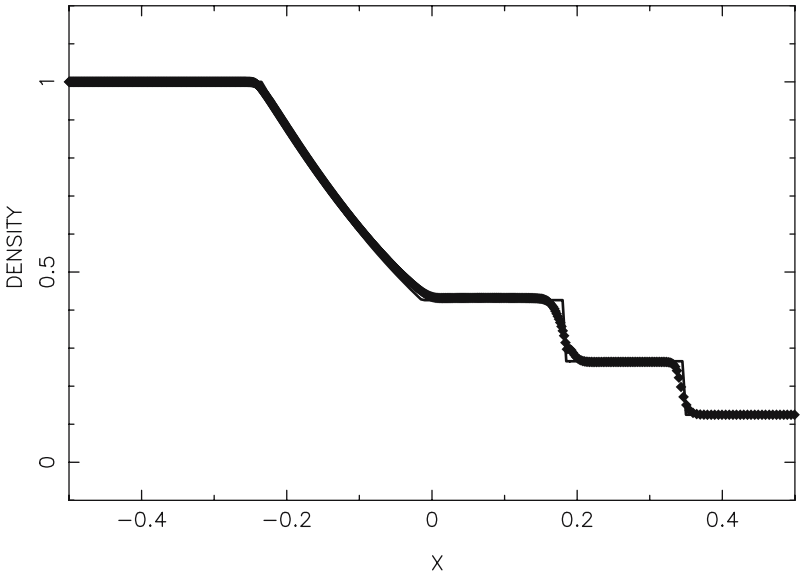


Figure 8.6. The density in the Sod shock tube problem.

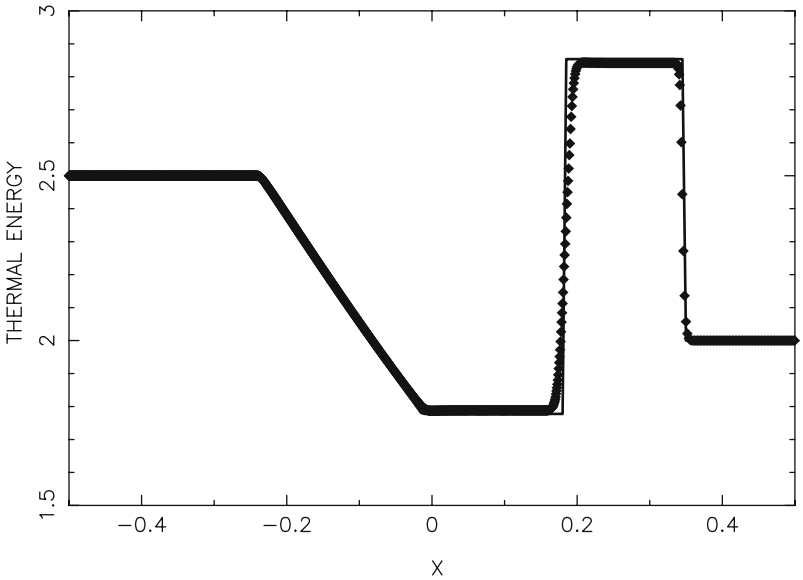


Figure 8.7. The thermal energy in the Sod shock tube with particle spacing on the right of the diaphragm 0.002. The exact run of thermal energy with distance is shown by the solid line. The solid diamonds are the SPH results. Note the sharpening in the SPH profiles compared with the previous figure.

1.79 respectively. In Figure 8.7 we show the thermal energy with the initial resolution ($\Delta x = 0.002$ to the right of the diaphragm). The results are in good agreement with the exact results although there is still significant diffusion near the contact discontinuity.

All of these results are very satisfactory though, for the resolution used the results are not as accurate as those from finely tuned Riemann solvers.

Exercise 8.1. Work out an SPH equation for the rate of change of energy per unit mass $\hat{e}$ defined by

$$\hat{e} = \frac{1}{2}v^2 + u. \tag{8.4}$$

Begin with the acceleration equation

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right) \nabla_a W_{ab} \tag{8.5}$$

and dot it with $\mathbf{v}_a$ to get

$$\frac{1}{2}\frac{d}{dt}\mathbf{v}_a^2 = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right) \mathbf{v}_a \cdot \nabla_a W_{ab}. \tag{8.6}$$

Combine this with the thermal energy equation

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b \mathbf{v}_{ab} \cdot \nabla_a W_{ab}, \tag{8.7}$$

to show

$$\frac{d\hat{e}_a}{dt} = -\sum_b m_b \left(\frac{P_a \mathbf{v}_b}{\rho_a^2} + \frac{P_b \mathbf{v}_a}{\rho_b^2} \right) \cdot \nabla_a W_{ab}. \tag{8.8}$$

If we decode this SPH equation we find it is the SPH form of

$$\frac{d\hat{e}}{dt} = -\frac{P}{\rho^2} \nabla \cdot (\rho \mathbf{v}) - \mathbf{v} \cdot \left(\frac{P}{\rho} \right) = -\frac{1}{\rho} \nabla \cdot P\mathbf{v}, \tag{8.9}$$

and we could have started with this equation and derived the equivalent SPH equation. ∎

In our discussion we have only considered an SPH viscosity suitable for shocks. In many problems we want to mimic physical viscosities. To do this we can make use of the previous viscosity but note that the effective kinematic viscosity is $\alpha h c_s/6$ in two dimensions (but the numerical coefficient depends on the kernel). We can then write

$$\Pi_{ab} = -\frac{12\mu_a\mu_b}{\rho_a\rho_b(\mu_a + \mu_b)} \frac{\mathbf{v}_{ab} \cdot \mathbf{r}_{ab}}{|\mathbf{r}_{ab}|} \tag{8.10}$$

where $\mu = \nu\rho$. This form of the viscosity has been used by Cleary (1998) to model more than one fluid with large differences in viscosity.
Other SPH viscosity calculations are described by Morris et al. (1997), Chaniotis et al. (2002) and Sigalotti et al. (2003). SPH has been used for many other applications. These include elastic fracture (Benz and Asphaug (1995)), relativistic calculations (Chow and Monaghan (1997)) and MHD simulations (Price and Monaghan (2004)).

References

1. Benz, W., and Asphaug, E. *Icarus*, Computer Phys. Communications, **87**, 253-265, (1995).
2. Bonet J. and Lok, T-S.L. *Comp. methods in app. mech. and Eng.*, **180**, 97-115, (1999).
3. Carslaw, H. S., and Jaeger, J. C. (1990) Conduction of Heat in Solids. Oxford Press, Oxford.
4. Chandrasekhar, S. (1995) *Newton's Principia for the Common Reader*, Clarendon Press. Oxford.
5. Chaniotis, A. K., Poulikakos, D., Koumotsakos, P. *J. Comp. Phys.*, **182**, 67, (2002).
6. Chow, E., and Monaghan, J. J. *J. Computat. Phys*, **134**, 296 -305, (1997).

7. Cleary, P. W. *Applied mathematical Modelling*, **22**, 981-983, (1998).
8. Cleary, P. W., and Monaghan, J. J. *J. Computat. Phys.* **148**, 227-264, (1999).
9. Davis, P.J., and Rabinowitz, P. (1967) *Numerical Integration*. Publ. Blaisdell, Waltham.
10. Eckart, C. *Physics of Fluids*, **3**, 421-427, (1960).
11. Feynman, R. P. (1965) *Feynman lectures on Physics Vol II*, Publ. Addison-Wesley.
12. Gingold, R. A., and Monaghan, J. J. *Mon. Not. Roy. Astro. Soc* **181**, 375, (1977).
13. Gingold, R. A., and Monaghan, J. J. *Mon. Not. Roy. Astro. Soc* **184**, 481-499, (1978).
14. Gingold, R. A., and Monaghan, J. J. *Mon. Not. Roy. Astro. Soc* **188**, 45-58, (1979).
15. Gingold, R. A., and Monaghan, J. J. *J. Computat. Phys.* **46**, 429-453, (1982).
16. Gingold, R. A., and Monaghan, J. J.*J. Computat. Phys* **52**, 374, (1983).
17. Goldsbrough, G. R., *Proc. Roy. Soc. A,*, **130**, 157, (1930).
18. Holm, D. D., *J. Fluid Mech*, **227**, 393, (1991).
19. Landau, L. D., and Liftzhitz, E. M. (1960) *Course of Theoretical Physics Vol 1*, Publ. Pergamon.
20. Lucy, L. B. *Astron. J* **82**, 1013, (1977).
21. Monaghan, J. J. *Ann. Rev. Astron. Astro.* **30**, 543 - 574, (1992).
22. Monaghan, J. J. *J. Computat. Phys* **136**, 298, (1997).
23. Monaghan, J. J., and Price, D. J., *Mon. Not. Roy. Astr. Soc*, **350**, 1449-1456, (2004).
24. Morris, J. P., Fox, P. J., and Zhu, Yi *J. computat. Phys*, **136**, 214, (1997).
25. Niedereiter, N., *Bull. American. Math. Soc.* **84**, 957, (1978).
26. Parzen, E. *Ann. Math. Statist.* **33**, 1065-1076, (1962).
27. Price, D., and Monaghan, J. J. *Mon. Not. Roy. Astr. Soc* **328**, 381-392, (2001).
28. Price, D., and Monaghan, J. J. *Mon. Not. Roy. Astr. Soc* **348**, 139-152, (2004).
29. Sod, G., A., *J. Computat. Phys.*, **27**, 1-31, (1978).
30. Shoenberg, I. J. *Quart. J. App. Math*, **IV**, 45, (1946).
31. Sigalotti, L. Di. G., Klapp, J. Sira, Eloy and Melean Ysamin *J. Computat. Phys* **191**, 622, (2003).
32. Springel, V., and Hernquist, L. *Mon. Not. Roy. Astr. Soc.* **333**, 649 -664, (2002).

Efficient Implementation and Parallelization of Meshfree and Particle Methods—The Parallel Multilevel Partition of Unity Method

Marc Alexander Schweitzer

Institut für Numerische Simulation, Rheinische Friedrich–Wilhelms Universität Bonn, Wegelerstraße 6, D–53115 Bonn, Germany.
email: schweitzer@ins.uni-bonn.de

Abstract In these introductory notes, we focus on the efficient implementation and parallelization of meshfree methods. Even though there exist a large number of different meshfree methods, e.g. smoothed particle hydrodynamics (SPH), reproducing kernel particle methods (RKPM), element free Galerkin methods (EFGM), radial basis functions (RBF), generalized finite element methods (GFEM), and partition of unity methods (PUM), the computational challenges are very similar for many of these approaches.
Some of the key issues involved with meshfree Galerkin discretization techniques are the fast construction of the shape functions, the assembly of the stiffness matrix and the load vector, i.e. numerical integration, the treatment of essential boundary conditions, and the efficient solution of the arising linear systems. We shall consider these issues in the context of the PUM, however, the concepts presented are applicable to most meshfree Galerkin approaches.

1 Introduction

Mesh based methods like the finite element method (FEM) [15, 21, 77], the finite difference method (FDM) [48] or the finite volume method (FVM) [13] are classical techniques for the numerical treatment of partial differential equations (PDEs). They exhibit good convergence properties in strong norms and their use is well established in various fields of application such as computational structural mechanics or computational fluid dynamics. However, all mesh based methods are rather involved when it comes to time dependent problems with complicated geometries since they rely on the availability of an appropriate discretization of the domain by a mesh. The construction of good quality meshes is not an easy task and a large portion of the overall computational time is often spent for mesh generation.
Particle schemes [64–67] on the other hand stem directly from physics applications such as Boltzmann equations [1, 36]. They are Lagrangian techniques where the domain of interest is discretized by a set of particles without any fixed connection between them; i.e. they are completely independent of a mesh. Here, the PDE on the computational domain is transformed into a system of ordinary differential equations (ODEs), the equations of motion

for the particles. Then, after time discretization, we obtain a certain particle distribution for each time step and define an approximate solution of the PDE via a density function for the respective particle distribution. Hence, the implementation of particle methods seems to be straightforward and less involved than the implementation of mesh based numerical schemes. However, particle methods generally only exhibit poor convergence properties in weak norms.

Meshfree methods are hybrid schemes which try to merge the best of both worlds. A meshfree method should have the strong convergence properties of a FEM or FDM. For instance, it should allow for a higher order approximation where we utilize the smoothness of the solution to reduce the degrees of freedom. Yet at the same time, the method should be independent of a mesh; i.e. it should not require any fixed connections between the degrees of freedom of the discretization and should only involve minimal assumptions on the distribution of the degrees of freedom.

The approaches to the design and development of meshfree methods are manifold, see e.g. [4, 30, 44, 45, 55] and the references therein. However, the computational challenges in most meshfree methods are very similar. In these introductory notes we focus on the particle-partition of unity method (PUM) [38–43, 76] which is a meshfree Galerkin technique and can be viewed as a generalized finite element method (GFEM). The three key issues we address are

1. The fast construction of the shape functions, which corresponds in some sense to the mesh generation problem in the FEM or the problem of finding all interacting particles in a particle scheme.
2. The development of a suitable weak formulation of the PDE, especially for problems involving essential boundary conditions, and an appropriate numerical integration scheme for the PUM.
3. The fast multilevel solution of the arising linear system.

We present efficient numerical techniques to tackle each of these computational challenges in the context of the PUM. However, the fundamental ideas and concepts (as well as some of the presented algorithms) are applicable to a wide range of mesh based and meshfree methods. The techniques and data structures used for the parallelization of our PUM for instance are also used in parallel particle methods with long range potentials [24, 37, 56, 81, 82] as well as in parallel mesh based multilevel method [86].

The remainder of this paper is organized as follows. In section 2 we review the abstract setting of the PUM. We focus on the construction of the shape functions, their properties and the realization of essential boundary conditions.[1] The efficient implementation of the PUM is discussed in detail in Section 3. There, we present computational techniques and algorithms for the fast construction of the shape functions, for the numerical integration of the weak

[1] For a study of the approximation properties of the PUM we refer to e.g. [8, 9, 61].

form as well as an efficient multilevel solver for our PUM. The parallelization of the overall method is subject of Section 4.

2 Partition of Unity Method

In the following, we consider a general partition of unity method (PUM) for a meshfree discretization of an elliptic partial differential equation. The approach is roughly as follows: The discretization is stated only in terms of points x_i. To obtain a trial and test space V^{PU}, a patch or volume $\omega_i \subset \mathbb{R}^d$ is attached to each point x_i such that the union of these patches form an open cover $C_\Omega = \{\omega_i\}$ of the domain Ω, i.e. $\overline{\Omega} \subset \bigcup \omega_i$. Now, with the help of weight functions $W_i : \mathbb{R}^d \to \mathbb{R}$ with $\mathrm{supp}(W_i) = \overline{\omega_i}$ local shape functions φ_i are constructed by Shepard's method. The functions φ_i form a partition of unity (PU). Then, each partition of unity function φ_i is multiplied by a sequence of local approximation functions ψ_i^n to assemble higher order shape functions. These product functions $\varphi_i \psi_i^n$ are finally plugged into the weak form to set up a linear system of equations via a Galerkin discretization.

2.1 Construction of a Partition of Unity Space

Necessary conditions for a trial and test space to perform well in a Galerkin method are local approximability and inter-element continuity. Here, local approximability means that the shape functions can approximate the exact solution well locally, and interelement continuity means that any linear combination of shape functions satisfies some global continuity condition. In the finite element method we have piecewise polynomial shape functions ϕ where the restriction $\phi|_E$ on an element E is a polynomial. Furthermore, there are certain constraints imposed on these local polynomials on the element boundary ∂E so that the shape function ϕ fulfills the interelement continuity condition. In the partition of unity approach [7–9, 76] we focus on the fulfillment of the condition of interelement continuity via the choice of an appropriate partition of unity $\{\varphi_i\}$ subordinate to a cover $C_\Omega := \{\omega_i\}$. Local expansion of the functions φ_i by the multiplication with local (unconstrained) approximation spaces $V_i^{p_i} = \mathrm{span}\langle\{\psi_i^n\}\rangle$ of order p_i defined on $\omega_i = \mathrm{supp}(\varphi_i)$ causes the generated space

$$V^{\mathrm{PU}} := \sum_i \varphi_i V_i^{p_i} = \sum_i \varphi_i \,\mathrm{span}\langle\{\psi_i^n\}\rangle = \mathrm{span}\langle\{\varphi_i \psi_i^n\}\rangle$$

to fulfill the condition of local approximability. Note that the superscript n only denotes a counting index. The global approximation space V^{PU} inherits the approximation quality of the local spaces $V_i^{p_i}$. Furthermore the space V^{PU} inherits the smoothness of the partition of unity. Here, the approximation property of the space V^{PU} may either be achieved by the smallness of the patches (h-version) or by the approximation quality of $V_i^{p_i}$ (p-version).

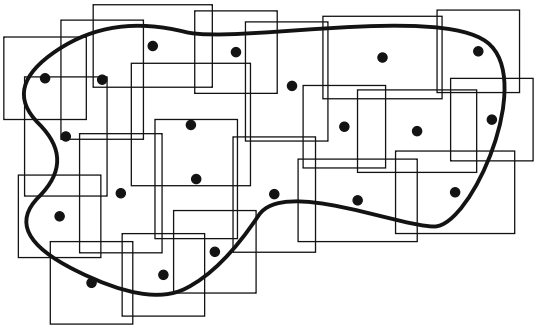


Figure 2.1. Example of an open cover C_Ω of a domain Ω.

The starting point for any meshfree discretization approach is a collection of N independent points

$$P := \{x_i \in \mathbb{R}^d \,|\, x_i \in \overline{\Omega}, i = 1, \ldots, N\}\,.$$

In our PUM, we then attach to each point x_i a d-rectangular patch[2]

$$\omega_i = \{x \in \mathbb{R}^d \,|\, |x_i^l - x^l| < h_i^l, l = 1, \ldots, d\} = \bigotimes_{l=1}^{d} (x_i^l - h_i^l, x_i^l + h_i^l).$$

The construction of appropriate patches ω_i from a given set of points $P = \{x_i\}$ is a first crucial step in the discretization process. Keeping in mind that these patches will be the supports of the trial and test functions in a Galerkin method, the most basic property these patches have to fulfill is that they cover the complete domain $\overline{\Omega} \subset \bigcup_{i=1}^{N} \omega_i$.[3] In other words, for any point $x \in \overline{\Omega}$ there exists at least one patch ω_i which contains x. Figure 2.1 gives an example of an open cover $C_\Omega = \{\omega_i\}$ of a domain Ω with d-rectangular patches ω_i. Note that the cover C_Ω also determines the sparsity pattern of the stiffness matrix via the geometric neighbour relations $\omega_i \cap \omega_j \neq \emptyset$ and thus the number of integrals that have to be evaluated in the Galerkin discretization. The influence on the overall computational cost of our PUM is therefore substantial and special attention should be paid to the appropriate design of a cover C_Ω for general point sets P, see Section 3.1 and [39, 40, 76].

[2] Note that the PUM is not restricted to d-rectangular patches. The geometry of the patches however has a major impact on the computational work involved with numerical integration. See [25, 26] for a partition of unity method based on radial supports.

[3] Other meshfree methods like smoothed particle hydrodynamics (SPH) which was first proposed in [34, 59] and further elaborated in [35, 62, 63] allow for holes in the covering of the domain Ω. Methods based on the moving least squares approach [27, 28, 31] on the other hand have to impose more severe geometric conditions on the cover $C_\Omega = \{\omega_i\}$.

Shepard Partition of Unity
Let us assume that we have constructed such a d-rectangular cover C_Ω. Then we can define a partition of unity $\{\varphi_i\}$ via data fitting techniques [51]. In general, a data fitting method is used to construct special shape functions φ_i for the approximation of a function u from discrete data (e.g. from sampling). Then, a so-called scattered data approximation $\tilde{u}$ usually is defined as

$$\tilde{u}(x) := \sum_{i=1}^{N} u_i \varphi_i(x)$$

where u_i are given data or are derived from that. Shepard's method uses inverse distance weighting for the construction of these shape functions. Here, the shape functions φ_i are defined as

$$\varphi_i(x) := \frac{W_i(x)}{\sum_{j=1}^{N} W_j(x)},$$

with weight functions $W_i(x) = \|x - x_i\|^{-\beta}$ with $\beta > 0$ where $\|\cdot\|$ is the classical Euclidean norm. But since these weight functions W_i have global support also the shape functions φ_i have global support. Hence, the evaluation of a single shape function φ_i involves all weight functions W_j; i.e. each function evaluation requires $O(N)$ operations. Furthermore, the use of globally supported shape functions in a Galerkin method would lead to a dense stiffness matrix and a quadratic storage complexity. We therefore use a localized version of Shepard's approach; i.e. $\text{supp}(W_i) = \overline{\omega_i} \Subset \mathbb{R}^d$. Then, we can restrict the summation of the weight functions to direct neighbours, i.e.

$$\varphi_i(x) = \frac{W_i(x)}{\sum_{\omega_k \in C_i} W_k(x)}, \tag{2.1}$$

where $C_i := \{\omega_j \in C_\Omega \,|\, \omega_i \cap \omega_j \neq \emptyset\}$ denotes the local neighbourhood of a particular Shepard function φ_i, i.e. of its associated patch ω_i. This reduces the complexity of a function evaluation to $O(1)$ and gives a sparse matrix for the stiffness. Here, the neighbourhoods C_i determine the sparsity pattern.
There are basically two variants for this localization: In [54] a locally supported singular weight function such as

$$W_i(x) = L_i(x)\|x - x_i\|^{-\beta}, \text{ where } L_i \in \mathcal{C}^\infty \text{ and } \text{supp}(L_i) = \overline{\omega_i}$$

is used. This approach generates an interpolatory partition of unity, i.e. $\varphi_i(x_j) = \delta_{ij}$. Another approach is to employ a locally supported smooth weight function W_i on a patch ω_i, e.g. W_i is chosen to be a B-spline [51]. The first approach is especially suitable for a collocation discretization, whereas

the latter approach is more suitable for the construction of shape functions for a Galerkin method since the evaluation of singular functions near their singularity is avoided. We are interested in a Galerkin discretization and therefore employ the latter approach in our method.
Since we restrict ourselves to the use of d-rectangular cover patches ω_i, i.e. the ω_i are products of intervals, the most natural choice for a weight function W_i is a product of univariate functions, i.e.

$$W_i(x) = \prod_{l=1}^{d} W_i^l(x^l) = \prod_{l=1}^{d} \mathcal{W}\left(\frac{x - x_i^l + h_i^l}{2h_i^l}\right)$$

with $\operatorname{supp}(\mathcal{W}) = [0,1]$ such that $\operatorname{supp}(W_i) = \overline{\omega_i}$. It is sufficient for this construction to choose a compactly supported univariate weight function $\mathcal{W}$ which is non-negative.[4]
Since we postulate that the union $\bigcup \omega_i$ of the patches ω_i covers the domain $\Omega \subset \bigcup \omega_i$ we are at least able to reproduce constant functions by (2.1); i.e. the functions φ_i form a partition of unity. Therefore, we obtain a consistency order of one in the L^2-norm.

Local Enrichment
First order consistency, however, is not sufficient for the discretization of a second order partial differential equation. Hence, some effort is necessary to improve the order. Here, the moving least squares method [11, 27, 28, 30, 31] allows for the construction of shape functions with higher reproduction and consistency orders but it increases the computational effort dramatically. Furthermore one has to impose severe geometric restrictions on the cover [31] to make the method work at all. Therefore we use a different approach. We use the partition of unity to collect local approximation spaces $V_i^{p_i}$ of order p_i defined on the cover patches ω_i, which generates a global approximation space

$$V^{\mathrm{PU}} := \sum_i \varphi_i V_i^{p_i} = \sum_i \varphi_i \operatorname{span}\langle\{\psi_i^n\}\rangle = \operatorname{span}\langle\{\varphi_i \psi_i^n\}\rangle \tag{2.2}$$

on the domain Ω. Hence, the overall shape functions $\varphi_i \psi_i^n$ in a PUM are product functions. Note that the local spaces $V_i^{p_i}$ can be chosen completely independently of each other. In general, the global space V^{PU} may only reproduce the constant, but the error estimates for the PUM [8,9,61] show that the consistency order of the global space V^{PU} is nevertheless the same as the consistency order of the local spaces $V_i^{p_i}$, see below. If the local spaces $V_i^{p_i}$ are polynomials of degree $\leq p_i$ the resulting global space V^{PU} reproduces polynomials of degree $\min_i p_i$.
Since we only use d-rectangular patches ω_i, a local tensor product space is the most natural choice. We usually employ products of univariate Legendre

[4] We usually employ a normalized B-spline as the generating weight function $\mathcal{W}$.

polynomials as local approximation spaces $V_i^{p_i}$; i.e. we choose

$$V_i^{p_i} = \operatorname{span}\langle \{\psi_i^n \mid \psi_i^n = \prod_{l=1}^{d} \mathcal{L}_i^{\hat{n}_l}, \|\hat{n}\|_1 = \sum_{l=1}^{d} \hat{n}_l \leq p_i\}\rangle, \tag{2.3}$$

where $\hat{n}$ is the multi-index of the polynomial degrees $\hat{n}_l$ of the univariate Legendre polynomials $\mathcal{L}_i^{\hat{n}_l} : [x_i^l - h_i^l, x_i^l + h_i^l] \to \mathbb{R}$, and n is the counting index associated with the product function $\psi_i^n = \prod_{l=1}^{d} \mathcal{L}_i^{\hat{n}_l}$. The use of more general product spaces is nonetheless possible and can improve the overall approximation properties or the computational complexity of the method; e.g. we can use anisotropic spaces or complete tensor product spaces where we use other (generalized) norms instead of $\|\cdot\|_1$ in (2.3). If some *a priori* knowledge about a particular behaviour of the solution (local or global) is available, we can utilize that knowledge in the selection of appropriate local approximation spaces [3, 6, 61, 76]. Note that the selection of the local approximation spaces in our PUM is completely unconstrained, we can use any local space with any basis anywhere in the computational domain without introducing any restriction on the choice of the local spaces elsewhere in the domain.

In summary we can view the construction given above as

$$\begin{pmatrix} \{x_i\} \\ \mathcal{W} \\ \{p_i\} \end{pmatrix} \to \begin{pmatrix} \{\omega_i\} \\ \{W_i\} \\ \{V_i^{p_i} = \operatorname{span}\langle\psi_i^n\rangle\} \end{pmatrix} \to \begin{pmatrix} \{\varphi_i\} \\ \{V_i^{p_i}\} \end{pmatrix} \to V^{\mathrm{PU}} = \sum \varphi_i V_i^{p_i},$$

where the set of points $P = \{x_i\}$, the generating weight function $\mathcal{W}$ and the local approximation orders p_i are assumed to be given. Following this construction we can construct approximate solutions $u^{\mathrm{PU}} \in V^{\mathrm{PU}} = \operatorname{span}\langle\varphi_i\psi_i^n\rangle$ of any order and regularity without additional constraints on the cover C_Ω. The extension to vector-valued PUM spaces is straightforward. We simply change the definition of our local approximation spaces to $V_i^{p_i} = \operatorname{span}\langle\psi_i^{n,l}\rangle = \operatorname{span}\langle\psi_i^n \boldsymbol{e}_l\rangle$ where $\boldsymbol{e}_l$ denotes an appropriate unit vector in $\mathbb{R}^d$, but keep the scalar partition of unity functions φ_i.[5]

Properties

The resulting approximation space V^{PU} and the shape functions $\varphi_i\psi_i^n$ are quite different from their finite element counterparts and have some notable properties.

1. The global PUM space V^{PU} inherits the approximation properties of the local enrichment spaces V^{p_i}. Let $u \in H^1(\Omega)$ be the function to be approximated. Assume that the local approximation spaces $V_i^{p_i}$ have the following approximation properties: On each patch $\Omega \cap \omega_i$, the function u

[5] Note that we may use scalar basis functions ψ_i^n of different type (e.g. polynomial or trigonometric) for the different coordinate directions $\boldsymbol{e}_l$, $l = 1, \ldots, d$.

can be approximated by a function $v_i \in V_i^{p_i}$ such that $\|u - v_i\|_{L^2(\Omega\cap\omega_i)} \leq \hat{\epsilon}_i$, and $\|\nabla(u - v_i)\|_{L^2(\Omega\cap\omega_i)} \leq \tilde{\epsilon}_i$ hold. Then the function

$$u^{\mathrm{PU}} := \sum_i \varphi_i v_i \in V^{\mathrm{PU}} \subset H^1(\Omega)$$

satisfies the estimates

$$\|u - u^{\mathrm{PU}}\|_{L^2(\Omega)} \leq C_1 \Big(\sum_{i=1}^{N} \hat{\epsilon}_i^2 \Big)^{\frac{1}{2}},$$

$$\|\nabla(u - u^{\mathrm{PU}})\|_{L^2(\Omega)} \leq C_2 \Big(\sum_{i=1}^{N} \frac{C_3}{\big(\operatorname{diam}(\omega_i)\big)^2} \hat{\epsilon}_i^2 + C_4 \tilde{\epsilon}_i^2 \Big)^{\frac{1}{2}},$$

where C_1, C_2, C_3, and C_4 only depend on the partition of unity used, see [8, 9] for details.

2. The partition of unity functions φ_i are (in general) non-interpolatory. Furthermore, there are more degrees of freedom in a PUM space than there are points $x_i \in P$ due to the use of (multidimensional) local approximation spaces $V_i^{p_i}$. Therefore, the implementation of essential boundary conditions is not straightforward in the PUM.
3. The PUM shape functions are piecewise rational functions due to the use of piecewise polynomial weights in (2.1) which makes numerical integration more challenging than in the FEM.
4. The regularity of the shape functions $\varphi_i \psi_i^n$ is independent of the number of local degrees of freedom. All shape functions $\varphi_i \psi_i^n$ inherit the regularity of the respective partition of unity function φ_i (if we assume that the local approximation spaces $V_i^{p_i}$ are at least of the same regularity). The partition of unity functions again inherit the smoothness of the weight functions W_i used in (2.1) since $\big(W_i \in \mathcal{C}^k(\mathbb{R}^d) \quad \wedge \quad \forall\, x \ \sum_i W_i(x) \neq 0\big) \Rightarrow \varphi_i \in \mathcal{C}^k(\mathbb{R}^d)$ holds. Therefore, we can improve the regularity of an approximation u^{PU} by changing the generating weight function $\mathcal{W}$ independent of the local approximation spaces $V_i^{p_i}$. Note that this is different from finite element methods. In a FEM the global regularity of an approximation is given by the element regularity which on the other hand is implemented by constraints imposed on the local degrees of freedom. Hence, a higher order regularity may only be achieved by increasing the number of degrees of freedom within an element.
5. The distribution of the point set $P = \{x_i\}$ and the cover $C_\Omega = \{\omega_i\}$ significantly influence the computational effort necessary to evaluate the functions φ_i, since the definition of φ_i in (2.1) involves the weights W_k of all geometric neighbours $\omega_j \in C_i = \{\omega_k \,|\, \omega_i \cap \omega_k \neq \emptyset\}$ of ω_i. The neighbour relations $\omega_j \in C_i$ of the cover C_Ω also define the sparsity pattern of the stiffness matrix. Furthermore, the diameters of the overlaps $\omega_i \cap \omega_j$ for $\omega_j \in C_i$ have a significant effect on the smoothness of φ_i [39, 40]. If the cover is minimal; i.e. there is exactly one patch ω_j for every

$x \in \overline{\Omega}$ with $x \in \overline{\omega_j}$, the partition of unity degenerates to the characteristic functions $\varphi_i = \chi_{\overline{\omega_i}}$ independently of the chosen weight functions W_i. Thus we see that small overlaps will cause large gradients of φ_i close to the boundary of the respective support ω_i. Hence, the diameter of the overlap $\omega_i \cap \omega_j$ of two neighbouring patches ω_i, ω_j should be bounded, i.e.

$$\operatorname{diam}(\omega_i \cap \omega_j) \geq C \min(\operatorname{diam}(\omega_i), \operatorname{diam}(\omega_j)), \tag{2.4}$$

so that the gradients $\nabla\varphi_i$ and $\nabla\varphi_j$ of the PU functions φ_i and φ_j are bounded.

Note that in our PUM we use a rather small overlap. In the FEM for instance the overlap of two supports $\operatorname{supp}(\phi_i) \cap \operatorname{supp}(\phi_j)$ is the size of an element, whereas in many meshfree methods the overlap is usually three to five times larger. Hence, in the FEM we obtain a stiffness matrix with approximately 3^d entries per row but in many meshfree methods the stiffness matrix is much more dense e.g. 7^d entries per row and more. In our PUM, however, we allow for an overlap which is even smaller than in the FEM (but fulfills (2.4)) and obtain a stiffness matrix with a similar sparsity pattern as the FEM.

6. The PUM shape functions $\varphi_i\psi_i^n$ are not shape equivalent due to the meshfree construction. The shape of each single PUM function $\varphi_i\psi_i^n$ is dependent on the geometric neighbour relations $\omega_i \cap \omega_j \neq \emptyset$, the associated weight functions W_j and the local basis $\{\psi_i^n\}$. Hence, the numerical integration (in general) cannot be carried out in a single reference configuration as we have in the FEM but it must rather be carried out in the physical space for each entry of the stiffness matrix.
7. In very special situations the shape functions $\varphi_i\psi_i^n$ can be linearly dependent which makes the solution of the resulting linear system more involved than in the FEM. For instance, the GFEM [2–5, 78, 79], where the PU comes from an h-version FEM, leads to linearly dependent shape functions $\varphi_i\psi_i^n$ (the so-called nullity of the method). This is essentially due to the fact that in the GFEM the partition of unity functions φ_i already reconstruct the linear polynomials. Consider the one-dimensional situation, where we have one element, i.e. a single interval, and two nodes (the interval boundaries) with their associated linear shape function as φ_i. Assume that we use linear polynomials as local approximations spaces $V_i^{p_i}$. The shape functions $\varphi_i\psi_i^n$ are (global) polynomials due to this construction. The number of shape functions is four and the maximal polynomial degree is two. Since the quadratic polynomials in one dimension can be generated by three basis functions, we see that the GFEM shape functions are linearly dependent. Hence, the solution of the arising linear system is a very challenging task in the GFEM.

   Note also that the construction of approximations with a higher degree of regularity in the GFEM requires the use of a more complex elements for the construction of the PU. With our general Shepard approach for the

PU construction we can construct a PU with a higher degree of regularity simply by a change of the generating weight function $\mathcal{W}$. Furthermore, we can easily avoid the linear dependence of the shape functions by enforcing $\varphi_i \equiv 1$ on $\tilde{\omega}_i \subset \omega_i$ with $\mathrm{vol}(\omega_i) \leq C\,\mathrm{vol}(\tilde{\omega}_i)$.

8. In general a smooth coefficient vector does not correspond to a smooth function since the coefficients are not directly related to the function value due to the non-interpolatory character of the shape functions. Furthermore, the choice of the local basis functions ψ_i^n and their ordering determine e.g. the discrete representation of the constant function. This can have a significant impact on the iterative solution of the resulting linear system since e.g. an algebraic multigrid method [80] (usually) assumes that the constant vector represents the constant function.

2.2 Variational Formulation and Boundary Conditions

We are interested in the approximate solution of an elliptic boundary value problem of the type

$$\begin{aligned} Lu &= f \text{ in } \Omega \subset \mathbb{R}^d, \\ Bu &= g \text{ on } \partial\Omega, \end{aligned} \tag{2.5}$$

where L is a symmetric partial differential operator of second order and B expresses suitable boundary conditions. Here, we are faced with two major computational tasks: We need to discretize the partial differential operator efficiently and we need to deal with boundary conditions properly.
Our PUM shape functions $\varphi_i\psi_i^n$ are non-interpolatory since the partition of unity functions φ_i are (in general) non-interpolatory, i.e. $\varphi_i(x_j) \neq \delta_{ij}$. Furthermore, the use of local approximation spaces $V_i^{p_i}$ with $\dim(V_i^{p_i}) > 1$ generates an approximation space $V^{\mathrm{PU}} = \sum_i \varphi_i V_i^{p_i}$ with more degrees of freedom than interpolation nodes x_i. Therefore, we have to cope with the problem: How do we fulfill the boundary conditions?
First consider (2.5) with $L = -\Delta$ and Neumann boundary conditions $Bu = u_\nu := \partial u/\partial\nu := \nabla u \cdot \nu = g$ on $\partial\Omega$, where ν denotes the outer normal. Then the continuous and elliptic bilinear form induced by L on $H^1(\Omega)$ is given by $a(u,v) = \langle \nabla u, \nabla v \rangle_{L^2}$ and we learn from the variational formulation

$$F(v) := \frac{1}{2}a(v,v) - \langle f, v \rangle_{L^2} - \int_{\partial\Omega} gv \to \min\{v \in H^1(\Omega)\}, \tag{2.6}$$

that the trial functions v have to fulfill no additional constraint besides being from the definition space $H^1(\Omega)$ of the differential operator L in its weak form. The boundary conditions are not imposed explicitly on the function space. Therefore, the basis of a finite-dimensional subspace $V \subset H^1(\Omega)$ used to approximate the solution of (2.6) may be compiled of arbitrary functions $v \in H^1(\Omega)$. The basis functions do not need to be interpolatory. Hence, we may use our functions $\varphi_i\psi_i^n$ as trial and test functions in a Galerkin procedure without any modification.

However, Dirichlet boundary conditions $Bu = u = g$ on $\partial\Omega$ explicitly impose the values of the solution u on the boundary $\partial\Omega$. Therefore, the trial space of the usual weak formulation

$$\text{Find } u \in H^1_g(\Omega): \quad a(u,v) = \langle f, v\rangle_{L^2} \text{ for all } v \in H^1_0(\Omega)$$

is not the complete space $H^1(\Omega)$ but $H^1_g(\Omega) := \{v \in H^1(\Omega) \,|\, u = g \text{ on } \partial\Omega\}$, whereas the test space is $H^1_0(\Omega)$. Note that we can enforce vanishing Dirichlet boundary conditions within the PUM by the selection of appropriate local approximation spaces $V_i^{p_i}$; i.e. we need $\psi_i^n|_{\partial\Omega} \equiv 0$ for all local basis functions ψ_i^n [2,31]. But the implementation of a trial space $V \subset H^1_g(\Omega)$ with $g \not\equiv 0$ by the selection of appropriate local approximation spaces is not feasible.
There are many different approaches to the treatment of Dirichlet boundary conditions with meshfree methods [2,31,39,46,50,53,58,61,76]. From our point of view, however, the most natural approach seems to be Nitsche's method [68] which is a variational technique that allows for the use of subspaces $V_N \subset H^1(\Omega)$ which do not have to fulfill the boundary conditions explicitly, yet it gives the optimal rate of convergence. The main advantages of Nitsche's method over other techniques are:

1. It does *not* introduce constraints on the distribution of the points $x_i \in P$.
2. The problem formulation only involves a single function space V_N defined on Ω. There is no need for an additional appropriate function space on the boundary $\partial\Omega$.
3. The method leads to symmetric positive definite linear systems. We do not need to be concerned with linear solvers for saddle-point problems.

Let us consider the Poisson problem

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega \subset \mathbb{R}^d, \\ u &= g \text{ on } \partial\Omega, \end{aligned} \tag{2.7}$$

for reasons of simplicity. We are interested in finding an approximate solution $u_N \in V_N \subset H^1(\Omega)$ to (2.7) — within optimal error bounds. Nitsche proposed in [68] to minimize the functional

$$J_N(w) := \int_\Omega |\nabla w|^2 - 2\int_{\partial\Omega} w w_\nu + \beta_N \int_{\partial\Omega} w^2,$$

for the error $w = v - u$ among all $v \in V_N$ where u is the solution of (2.7), and $\beta_N > 0$ only depends on the subspace V_N; i.e. the approximation u_N is given by $J(u_N - u) \stackrel{!}{=} \inf_{v\in V_N} J_N(v-u)$. Note that the subscript ν denotes the normal derivative, i.e. $w_\nu = \nabla w \cdot \nu$, whereas the subscript N indicates a dependence on the discretization space $V_N \subset H^1(\Omega)$. The minimizer $u_N \in V_N$ can be computed from the input data f and g of (2.7) since

$$\begin{aligned} J_N(v-u) &= J_N(v) + J_N(u) - 2\left(\int_\Omega \nabla v \nabla u + \int_{\partial\Omega} \beta_N uv - vu_\nu - uv_\nu\right) \\ &= J_N(v) + J_N(u) - 2\left(\int_\Omega fv + \int_{\partial\Omega} \beta_N gv - gv_\nu\right). \end{aligned}$$

The corresponding weak formulation is given by $a_N(u_N, v) = l_N(v)$ for all $v \in V_N$ where

$$a_N(w,v) := \int_\Omega \nabla v \nabla w - \int_{\partial\Omega} v w_\nu - \int_{\partial\Omega} w v_\nu + \beta_N \int_{\partial\Omega} vw,$$
$$l_N(v) := \int_\Omega fv - \int_{\partial\Omega} g v_\nu + \beta_N \int_{\partial\Omega} gv.$$

Although the bilinear form $a_N(\cdot,\cdot)$ is indefinite on the space $H^1(\Omega)$ it is symmetric positive definite on the subspace V_N under the assumptions that

$$\|v_\nu\|_{L^2(\partial\Omega)} \le C_N \|\nabla v\|_{L^2(\Omega)} \tag{2.8}$$

holds for all $v \in V_N$ with $C_N > 0$ and that $\beta_N > 2\ C_N^2$ since

$$\begin{aligned} a_N(v,v) &= \|\nabla v\|^2_{L^2(\Omega)} - 2\int_{\partial\Omega} v v_\nu + \beta_N \|v\|^2_{L^2(\partial\Omega)} \\ &\ge \|\nabla v\|^2_{L^2(\Omega)} - 2\ C_N \|v\|_{L^2(\partial\Omega)} \|\nabla v\|_{L^2(\Omega)} + \beta_N \|v\|^2_{L^2(\partial\Omega)} \\ &\ge \frac{1}{2}\|\nabla v\|^2_{L^2(\Omega)} + (\beta_N - 2\ C_N^2)\|v\|^2_{L^2(\partial\Omega)}. \end{aligned}$$

Nitsche furthermore proved optimal error estimates if the relation

$$C_N^2 = O(\mathrm{diam}(\mathrm{supp}(\phi))^{-1}) \tag{2.9}$$

for C_N in (2.8) holds for all basis functions $\phi \in V_N$ and the respective approximation property in V_N is given. The proportionality (2.9) is valid e.g. if we have estimates of the form

$$\int_{\partial\Omega} |\phi_\nu|^2 \le C_{N,1} (\mathrm{diam}(\mathrm{supp}(\phi)))^{d-1} \tag{2.10}$$

and

$$\int_\Omega |\nabla\phi|^2 \ge C_{N,2} (\mathrm{diam}(\mathrm{supp}(\phi)))^{d} \tag{2.11}$$

for all basis functions $\phi \in V_N$ with $\mathrm{supp}(\phi) \cap \partial\Omega \neq \emptyset$. Note that (2.10) and (2.11) essentially introduce some geometric constraints on the intersections $\mathrm{supp}(\phi) \cap \Omega$ and $\mathrm{supp}(\phi) \cap \partial\Omega$; i.e. in our meshfree context on the cover C_Ω or in the finite element context on the regularity of the mesh.
In general a proof of (2.9) is simplified when we only need to consider a regular reference configuration; i.e. where the map to the reference configuration is affine. Here, we find

$$\frac{\int_{\partial\,\mathrm{supp}(\phi)} |\phi_\nu|^2}{\int_{\mathrm{supp}(\phi)} |\nabla\phi|^2} = \frac{\det(J_{\partial T}) \int_{\partial\omega_{\mathrm{ref}}} |\phi_\nu \circ \partial T|^2}{\det(J_T) \int_{\omega_{\mathrm{ref}}} |\nabla\phi \circ T|^2} = \frac{\det(J_{\partial T})}{\det(J_T)} C_{N,\mathrm{ref}}$$
$$\approx (\mathrm{diam}(\mathrm{supp}(\phi)))^{-1} C_{N,\mathrm{ref}}$$

where $C_{N,\mathrm{ref}}$ only depends on the polynomial degree of the shape function ϕ. So if we limit ourselves to the use of uniform covers and a fixed local approximation space we only need to consider very few reference cases (depending on the number of edges of $\mathrm{supp}(\varphi\psi_i^n) \cap \partial\Omega$ and the local polynomial degree p_i of $V_i^{p_i}$). But for the general situation where we have an irregular point distribution and locally varying approximation spaces this approach cannot be pursued. Furthermore, from a computational point of view we must be interested in the *value* of C_N in (2.8), not only the type of the proportionality (2.9). Only a large enough value of C_N will lead to a definite problem formulation. Yet, a wrong choice of the regularization parameter may have an impact on the condition number of the linear system or other adverse effects on the applicability of certain linear solvers. Since the parameter C_N is not only dependent on the support sizes but also on the selected local basis functions ψ_i^n for the spaces $V_i^{p_i}$ and the local approximation orders p_i, we need to be concerned with the *automatic computation* of a reliable estimate of C_N. Here, we decided to approach condition (2.8) as a generalized eigenvalue problem

$$\mathcal{A}x = \lambda \mathcal{B}x \tag{2.12}$$

where

$$\mathcal{A}_{(i,n),(j,m)} := \int_{\partial\Omega} (\varphi_j \psi_j^m)_\nu (\varphi_i \psi_i^n)_\nu$$

and

$$\mathcal{B}_{(i,n),(j,m)} := \int_{\Omega} \nabla(\varphi_j \psi_j^m) \nabla(\varphi_i \psi_i^n)$$

for all index pairs (i,n), and (j,m) which correspond to shape functions which overlap the boundary $\partial\Omega$; i.e. $\omega_i \cap \partial\Omega \neq \emptyset$ and $\omega_j \cap \partial\Omega \neq \emptyset$. Solving (2.12) for the maximal eigenvalue $\lambda_{\max}$ we get a good estimate for C_N^2. Hence, a choice of $\beta_N = w\lambda_{\max}$ with $w > 2$ will lead to a symmetric positive definite system.

Note that the assembly of the matrices $\mathcal{A}$ and $\mathcal{B}$ does *not* introduce a significant amount of additional computational cost. The entries of $\mathcal{B}$ are needed for the stiffness matrix and can be reused. The remaining additional cost associated with the computation of the regularization parameter β_N come from the solution of the eigenvalue problem (2.12). The eigenvalue $\lambda_{\max}$ can be computed very efficiently by a simultaneous Rayleigh-quotient minimization method [14, 57] due to the similar structure of the matrices $\mathcal{A}$ and $\mathcal{B}$. On average we need about five to ten conjugate gradient iterations to compute $\lambda_{\max}$ with five digits accuracy. Here, the minimization of $\frac{x^T \mathcal{B}x}{x^T \mathcal{A}x}$ involves only matrix-vector-products. We do not need to solve a linear system. Furthermore, the eigenvalue problem (2.12) involves only boundary degrees of freedom and is therefore of smaller dimension. Hence, the computational cost associated with the assembly of a Dirichlet problem are comparable to the cost associated with the respective Neumann problem.

For the sake of completeness, we give the weak formulation of a Poisson problem

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \subset \mathbb{R}^d, \\ u &= g_D \text{ on } \Gamma_D \subset \partial\Omega, \\ u_\nu &= g_N \text{ on } \Gamma_N = \partial\Omega \setminus \Gamma_D, \end{aligned} \tag{2.13}$$

with mixed boundary conditions where the Dirichlet boundary conditions are realized with Nitsche's method and the Neumann boundary conditions are implemented in the standard fashion as an additional surface term on the right-hand side. The respective weak formulation $a(u,v) = l(v)$ is given by

$$\begin{aligned} a(u,v) &:= \int_\Omega \nabla u \nabla v + \int_{\Gamma_D} u(\beta v - v_\nu) - u_\nu v, \\ l(v) &:= \int_\Omega f v + \int_{\Gamma_D} g_D(\beta v - v_\nu) + \int_{\Gamma_N} g_N v, \end{aligned} \tag{2.14}$$

where β now denotes the respective regularization parameter.
The extension of Nitsche's method to vector-valued problem is straightforward. Let us consider the Navier–Lamé equations

$$-\mu \boldsymbol{\Delta} u - (\lambda + \mu) \nabla \operatorname{div}(u) = f \quad \text{in} \quad \Omega \subset \mathbb{R}^d, \quad d = 2, 3$$

together with suitable boundary conditions $u_D = g_D$ on $\Gamma_D \subset \partial\Omega$ and $\boldsymbol{\sigma}(u) \cdot \nu = g_N$ on $\Gamma_N = \partial\Omega \setminus \Gamma_D$ where $\boldsymbol{\sigma}(u) := \lambda \operatorname{div}(u) \mathbb{I} + 2\mu\, \boldsymbol{\epsilon}(u)$ denotes the symmetric stress tensor, $\mathbb{I}$ is the identity operator and $\boldsymbol{\epsilon}(u) := \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ the strain tensor associated with the displacement field $u = (u_i)$, $i = 1, \ldots, d$.[6] The associated bilinear form arising from Nitsche's approach is given by

$$a(u,v) = \int_\Omega \boldsymbol{\sigma}(u) : \boldsymbol{\epsilon}(v) + \int_{\Gamma_D} 2\mu\beta_{\boldsymbol{\epsilon}} u \cdot v + \lambda \beta_{\operatorname{div}} (u \cdot \nu)(v \cdot \nu) - \big((\boldsymbol{\sigma}(u) \cdot \nu) \cdot v + u \cdot (\boldsymbol{\sigma}(v) \cdot \nu) \big)$$

where $\boldsymbol{\sigma}(u) : \boldsymbol{\epsilon}(v) := \sum_{i,j} \boldsymbol{\sigma}(u)_{i,j}\, \boldsymbol{\epsilon}(v)_{i,j}$. The linear form on the right-hand side is given by

$$l(v) = \int_\Omega f \cdot v + \int_{\Gamma_N} g_N \cdot v + \int_{\Gamma_D} 2\mu\beta_{\boldsymbol{\epsilon}} g_D \cdot v + \lambda \beta_{\operatorname{div}} (g_D \cdot \nu)(v \cdot \nu) - g_D \cdot (\boldsymbol{\sigma}(v) \cdot \nu).$$

Here, we compute the regularization parameter $\beta_{\boldsymbol{\epsilon}} = w C_{\boldsymbol{\epsilon}}$ associated with the strain term from the generalized eigenvalue problem

$$\mathcal{A}x = \int_{\partial\Omega} (\boldsymbol{\epsilon}(u) \cdot \nu) \cdot (\boldsymbol{\epsilon}(v) \cdot \nu) \le C_{\boldsymbol{\epsilon}} \int_\Omega \boldsymbol{\epsilon}(u) : \boldsymbol{\epsilon}(v) = C_{\boldsymbol{\epsilon}} \mathcal{B} x.$$

[6] The parameters λ and μ are the so-called Lamé parameters. They are related to the Poisson ratio ν and the Young modulus E of the material via $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$.

The regularization parameter $\beta_{\text{div}} = wC_{\text{div}}$ for the divergence term is computed with the help of

$$\mathcal{A}x = \int_{\partial\Omega} \operatorname{div}(u)\operatorname{div}(v) \leq C_{\text{div}} \int_{\Omega} \operatorname{div}(u)\operatorname{div}(v) = C_{\text{div}}\mathcal{B}x.$$

Note that the use of Nitsche's method for the implementation of Dirichlet boundary conditions leads to a so-called level dependent weak formulation; i.e. the bilinear form as well as the linear form on the right-hand side involve regularization parameters which depend on the discretization space used. Hence, if we change the discretization space, for instance by increasing the resolution via refinement, we obtain a different weak formulation $a(u, v) = l(v)$.

2.3 Galerkin Discretization

The PUM is a meshfree generalized finite element method, hence we discretize a partial differential equation using the respective weak formulation $a(u, v) = l(v)$ and a Galerkin approach. That is, we need to compute the entries of the stiffness matrix

$$A = (A_{(i,n),(j,m)}), \text{ with } A_{(i,n),(j,m)} = a\left(\varphi_j\psi_j^m, \varphi_i\psi_i^n\right)$$

and the entries of the right-hand side vector

$$\hat{f} = (\hat{f}_{(i,n)}), \text{ with } \hat{f}_{(i,n)} = l(\varphi_i\psi_i^n)$$

to set up the respective linear system $A\tilde{u} = \hat{f}$, where $\tilde{u} = (\tilde{u}_{(i,n)})$ denotes a coefficient vector and $\hat{f}$ denotes a moment vector. Let us now consider this assembly step for (2.14). Here we have to compute the integrals

$$\int_{\Omega} f\varphi_i\psi_i^n + \int_{\Gamma_D} g_D(\beta\varphi_i\psi_i^n - ((\varphi_i\psi_i^n)_\nu) + \int_{\Gamma_N} g_N\varphi_i\psi_i^n$$

for the right-hand side $\hat{f}$, and the integrals

$$\int_{\Omega} \nabla\varphi_i\psi_i^n\nabla\varphi_j\psi_j^m + \int_{\Gamma_D} \varphi_i\psi_i^n(\beta\varphi_j\psi_j^m - (\varphi_j\psi_j^m)_\nu) - (\varphi_i\psi_i^n)_\nu\varphi_j\psi_j^m \quad (2.15)$$

for the stiffness matrix A. Recall that φ_i is defined by (2.1), i.e.

$$\varphi_i(x) = \frac{W_i(x)}{\sum\limits_{\omega_k \in C_i} W_k(x)} = \frac{W_i(x)}{\sum\limits_{k=1}^{N} W_k(x)}.$$

Now we carry out the differentiation in (2.15). With the notation $\mathcal{S} := \sum_{k=1}^{N} W_k$, $\mathcal{T} := \sum_{k=1}^{N} \nabla W_k$ and $\mathcal{G}_i := \nabla W_i \mathcal{S} - W_i \mathcal{T}$ we end up with the integrals

$$\begin{aligned} a(\varphi_j\psi_j^m, \varphi_i\psi_i^n) = &\int_\Omega \mathcal{S}^{-4}\mathcal{G}_i\psi_i^n\mathcal{G}_j\psi_j^m + \int_\Omega \mathcal{S}^{-2}W_i\nabla\psi_i^n W_j\nabla\psi_j^m \\ &+ \int_\Omega \mathcal{S}^{-3}\big(\mathcal{G}_i\psi_i^n W_j\nabla\psi_j^m + W_i\nabla\psi_i^n\mathcal{G}_j\psi_j^m\big) \\ &- \int_{\Gamma_D} \mathcal{S}^{-3}\big(\mathcal{G}_i\psi_i^n W_j\psi_j^m + W_i\psi_i^n\mathcal{G}_j\psi_j^m\big)\cdot\nu \\ &- \int_{\Gamma_D} \mathcal{S}^{-2}\big(W_i\nabla\psi_i^n W_j\psi_j^n + W_i\psi_i^n W_j\nabla\psi_j^m\big)\cdot\nu \\ &+ \int_{\Gamma_D} \beta\mathcal{S}^{-2}W_i\psi_i^n W_j\psi_j^m \end{aligned} \tag{2.16}$$

for the stiffness matrix and the integrals

$$\begin{aligned} l(\varphi_i\psi_i^n) = &\int_\Omega \mathcal{S}^{-1}W_i\psi_i^n f + \int_{\Gamma_N} \mathcal{S}^{-1}W_i\psi_i^n + \int_{\Gamma_D} \beta\mathcal{S}^{-1}W_i g_D \\ &- \int_{\Gamma_D} \big(\mathcal{S}^{-2}\mathcal{G}_i\psi_i^n + \mathcal{S}^{-1}W_i\nabla\psi_i^n\big)g_D\cdot\nu \end{aligned} \tag{2.17}$$

for the right-hand side. The functions $\mathcal{T}$ and $\mathcal{G}_i$ may have a large number of jumps (or kinks) due to the overlap of the support patches ω_i and the use of piecewise polynomial weights W_i in (2.1). Therefore, the integrals (2.16) and (2.17) should not be computed by a simple quadrature scheme which does not respect these discontinuities and the algebraic structure of the shape functions.[7] The numerical integration of the weak form is a major computational task in any meshfree Galerkin discretization [10, 19, 20, 29, 39, 40]. Furthermore, we must be aware that an inappropriate solution to the integration problem can lead to stability problems.

2.4 Solution of Resulting Linear System

Finally, it remains to solve the discrete system of linear equations $A\tilde{u} = \hat{f}$. For our PUM space we have $\mathsf{dof} = O(Np^d)$ where $N = \mathrm{card}(C_\Omega)$ denotes the number of patches ω_i and p the order of approximation. The number of nonzeros nnz of the stiffness matrix A is of the order $O(Np^{2d})$. The stiffness matrix is sparse with respect to N due to the compactness of the cover patches ω_i and the sparsity pattern is given by the local neighbourhoods C_i. The higher order p-dependence is due to the use of multi-dimensional local

[7] The integrals for the right-hand side are oftentimes not directly evaluated but rather approximated via the coefficient vectors $\tilde{f}$, $\tilde{g}_N$, $\tilde{g}_D$ and (generalized) mass matrices (on the boundary).

approximation spaces $V_i^{p_i}$. Overall, the stiffness matrix is a sparse block-matrix with dense matrix blocks $A_{i,j} = (A_{(i,n),(j,m)})$. A single block $A_{i,j}$ corresponds to a local discretization of the PDE on the domain $\omega_i \cap \omega_j \cap \Omega$. The blocks $A_{i,j}$ are dense matrices and may have different dimensions corresponding to the dimensions of the local approximation spaces $V_j^{p_j}$ and $V_i^{p_i}$.

Note that the use of an inappropriate solver can drive up the computational time as well as the storage demand dramatically. For an optimal scalability of the overall methods, the linear solver used should have a complexity of $O(\mathsf{nnz})$. Classical direct solvers for dense matrices like Gaussian elimination or LU-decomposition have a storage requirement of $O(\mathsf{dof}^2)$ and the number of operations even scales with $O(\mathsf{dof}^3)$, where dof denotes the number of degrees of freedom. More advanced direct solvers for sparse linear systems can reduce these complexities to some extend only. In the special case of regular meshes in two dimensions for instance, a nested dissection solver requires $O(\mathsf{dof}^{3/2})$ operations and $O(\mathsf{dof}\ln(\mathsf{dof}))$ storage [32] whereas the optimal complexity is $O(\mathsf{dof}) = O(\mathsf{nnz})$ with nnz being the number of nonzeros of the matrix A. Hence, the optimal storage and operation complexity will be lost when a direct solver is employed.

With the classical iterative schemes like the Jacobi- or Gauss–Seidel method we do not have a significant increase in the storage requirements, but the number of operations necessary to obtain the solution of the linear system does not scale with the optimal complexity. A class of sophisticated iterative methods which not only show an optimal scaling in the storage demand but also in the operation count is the class of the so-called multilevel iterative solvers [83] or multigrid methods [47]. These solvers, however, are not general algebraic methods but involve a substantial amount of information about the discretization and possibly the PDE.[8] Hence, we cannot expect an existing multilevel solver which was designed for a completely different type of discretization to solve our linear system from a PUM discretization. We rather need to translate the essential multigrid ideas to the meshfree setting. However, the design of an efficient multilevel solver for meshfree methods is complicated by the fact that it is in general not feasible to construct a sequence of nested function spaces.

[8] There are algebraic multigrid (AMG) methods [80] but their construction is (in general) based on the assumption of an interpolatory linear basis. These methods are very involved and a generalization of AMG to meshfree discretizations is not an easy task. Furthermore, we usually try to mimic the behaviour of geometric multigrid methods with AMG. Hence, a first step in the design of an AMG method for meshfree discretizations must be the development of a geometric multilevel solver which can provide guidelines for a meshfree AMG.

3 Efficient Implementation

In the following we focus on the efficient implementation of the PUM. According to the presentation given above, the three major issues in the realization of the PUM, and many other mesh-based and meshfree methods, are:

1. The fast construction of the shape functions; i.e. the construction of an appropriate cover $C_\Omega = \{\omega_i\}$ of the domain Ω and the computation of the local neighbourhoods $C_i = \{\omega_j \,|\, \omega_i \cap \omega_j \neq \emptyset\}$.
2. The efficient and reliable integration of the weak form to set up a valid approximation of the stiffness matrix A and right-hand side $\hat{f}$.
3. The fast solution of the resulting linear system $A\tilde{u} = \hat{f}$.

We present specialized algorithms for each of these challenges. Even though these algorithms are specifically designed for the PUM, the underlying concepts are applicable to a number of meshfree (and mesh-based) numerical methods. The tree-based cover construction algorithm we present in the following for instance can be modified easily to be suitable for mesh generation [12], for the implementation of an adaptive multilevel FEM [86] or Lagrangian particle schemes [24, 37, 56, 81, 82].

3.1 Cover Construction

Following the construction given in Section 2.1, the first critical step in the implementation of the PUM is the efficient construction of an appropriate cover C_Ω for a general given point set. Here, we need to consider not only the construction of patches ω_i which cover the domain Ω but the respective neighbourhoods C_i which significantly influence the overall computational cost. Hence, our cover construction should have the following properties:

1. The only input data are the domain Ω and a set of points $P = \{x_i \in \mathbb{R}^d\}$. There are no assumptions on the distribution of the points x_i. The construction is independent of the dimension d.
2. The union of the constructed cover patches ω_i covers the complete domain Ω including the boundary $\partial\Omega$, i.e. $\bigcup_i \omega_i \supset \overline{\Omega}$.
3. The number of constructed cover patches is minimal, i.e. $\mathrm{card}(C_\Omega) = O(\mathrm{card}(P))$. The geometric shape of a cover patch ω_i is simple and all patches are shape-regular.
4. The neighbourhood $C_i = \{\omega_j \in C_\Omega \,|\, \omega_i \cap \omega_j \neq \emptyset\}$ of a particular patch is easily computable.
5. The size of the overlaps $\omega_i \cap \omega_j$ can be controlled and the number of neighbours $\mathrm{card}(C_i)$ is of the order $O(1)$ and can be controlled.

Many different geometric algorithms have been used for the construction of a suitable cover. The nearest neighbour or direct covering for instance has been used in [31, 39, 40, 76]. With this cover construction, however, the

covering property is not ensured automatically for a general point set and an explicit validation procedure is necessary. Furthermore, the problem of finding all neighbouring patches $\omega_j \in C_i$ remains. The computation of the neighbourhoods C_i is essentially a geometric search problem [75]. Hence, tree-based techniques which have been used successfully for searching and sorting problems in many areas [52,74] can be used to tackle this problem. But if such a tree-based algorithm must be employed to compute the neighbourhoods C_i even when we have a cover C_Ω, the question arises, if we can use a tree-based method also for the construction of C_Ω itself.

We have developed a hierarchical cover construction algorithm [40,76] based on d-binary trees (quadtrees, octrees) which allows us to construct a valid cover C_Ω and enables us to compute the local neighbourhoods C_i very efficiently using a single data structure. In the following we denote the set of points x_i which are used for the partition of unity construction by P whereas we denote a given initial point set by $\tilde{P}$ since these two sets may differ. Our tree-based cover construction algorithm employs a decomposition approach for the domain Ω to assign patches $\omega_i \subset \mathbb{R}^d$ to the points x_i of a newly constructed point set $P \supset \tilde{P}$ in such a way that these patches cover the complete domain $\Omega \subset \bigcup \omega_i$. Note that this hierarchical algorithm does not need any additional input besides the point set $\tilde{P}$ and the domain Ω. Furthermore, there is no need for an explicit validation of the covering property $\overline{\Omega} \subset \bigcup \omega_i$.

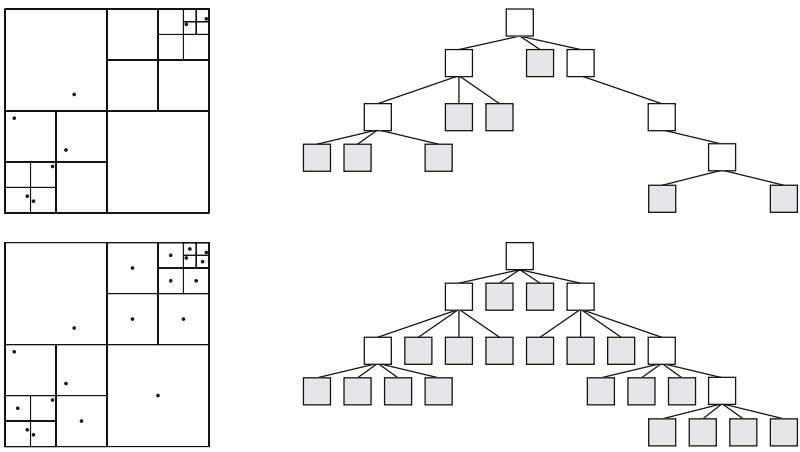


Figure 3.1. Hierarchical cover construction with Algorithm 3.1 in two dimensions. The cell decomposition induced by $\tilde{P}$ (upper left) and its corresponding tree representation (upper right, white: INNER tree nodes, gray shaded: LEAF tree nodes) after step 3 of Algorithm 3.1. Here, the leaves of the tree correspond to the points $x_i \in \tilde{P}$. The final cell decomposition with all points $x_L \in P$ (lower left) and its tree representation (lower right) after the completion of Algorithm 3.1. Now, the leaves of the tree correspond to the points $x_L \in P$.

Algorithm 3.1. Hierarchical Cover Construction

1. Given the domain $\Omega \subset \mathbb{R}^d$, a bounding box $R_\Omega = \bigotimes_{i=1}^d [l_\Omega^i, u_\Omega^i] \supset \overline{\Omega}$ and a scalar $\alpha \geq 1$.
2. Given the initial point set $\tilde{P} = \{x_j \,|\, x_j \in \overline{\Omega}, j = 1, \ldots, \tilde{N}\}$.
3. Build a d-binary tree[9] over R_Ω such that per leaf L at most one $x_i \in \tilde{P}$ lies within the associated cell $\mathcal{C}_L := \bigotimes_{i=1}^d [l_L^i, u_L^i]$; see Figure 3.1.
4. Set $P = \emptyset$, $C_\Omega = \emptyset$.
5. For the root cell $\mathcal{C}_L = \bigotimes_{i=1}^d [l_L^i, u_L^i] = R_\Omega$:
   (a) If current tree cell $\mathcal{C}_L$ is an INNER tree node and $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Descend tree for all successors $\mathcal{C}_S$ of $\mathcal{C}_L$. ($\rightarrow$ 5(a))
      ii. Set patch ω_L such that $\bigcup \omega_S \subset \omega_L$.
   (b) Else if $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. If $x_k \in \mathcal{C}_L$ for a $x_k \in \tilde{P}$:
         Set $x_L = x_k$.
      ii. Else:
         Choose $x_L \in \mathcal{C}_L$, e.g. $x_L^i = l_L^i + \frac{1}{2}(u_L^i - l_L^i)$.
      iii. Set $h_L^i = \alpha \max\{u_L^i - x_L^i, x_L^i - l_L^i\}$.
      iv. Set patch $\omega_L = \bigotimes_{i=1}^d [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$.
      v. Set $P = P \cup \{x_L\}$, $C_\Omega = C_\Omega \cup \{\omega_L\}$.

With this hierarchical cover construction algorithm we have $P = \tilde{P} \cup Q$ where Q is an *automatically* constructed set of additional points.[10]
We ensure not only the covering property $\overline{\Omega} \subset \bigcup \omega_i$ without additional input data with Algorithm 3.1, but also some control over the neighbourhoods C_i; i.e. the nonzero blocks of the stiffness matrix, and to some extent we can regulate the smoothness of the functions φ_i. Furthermore, we can easily compute the neighbourhoods C_i with the help of the tree of patches by Algorithm 3.2.

Algorithm 3.2. Computation of Neighbourhood C_i
For the root cell $\mathcal{C}_L = \bigotimes_{j=1}^d [l_L^j, u_L^j] = R_\Omega$:

1. If current tree cell $\mathcal{C}_L$ is an INNER tree node and $\mathcal{C}_L \cap \Omega \neq \emptyset$:
   (a) If for current patch $\omega_L \cap \omega_i \neq \emptyset$:
      Descend tree for all successors $\mathcal{C}_S$ of $\mathcal{C}_L$. ($\rightarrow$ 1)
2. Else if $\mathcal{C}_L \cap \Omega \neq \emptyset$:
   (a) If for current patch $\omega_L \cap \omega_i \neq \emptyset$:
      Set $C_i = C_i \cup \{\omega_L\}$.

[9] Samet [75] refers to a point region (PR) quadtree for our two-dimensional construction in Figure 3.1.

[10] The additional points are necessary to ensure the shape regularity of the tree cells (and patches). A similar tree-based algorithm for the construction of shape-regular triangulations with an almost-minimal number of vertices was proposed in [12]. Analogously, Algorithm 3.1 may have a similar almost-optimal property: If m is the minimal number of shape-regular d-rectangles required to cover the given point set in such way that all d-rectangles contain at most one point, then the cover C_Ω constructed by the presented algorithm is of size $\mathrm{card}(C_\Omega) = O(m)$.

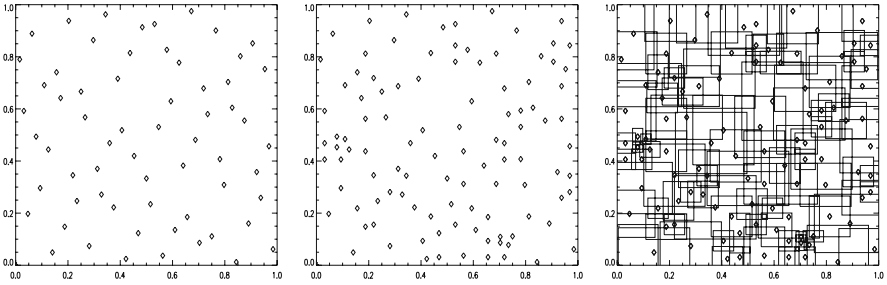


Figure 3.2. Points of an initial Halton$(2,3)$ point set $\tilde{P}$ with $\tilde{N} = \mathrm{card}(\tilde{P}) = 64$ points distributed in $R_\Omega = \Omega = [0,1]^2$ (left), the points of the generated point set P with $N = \mathrm{card}(P) = 106$ (centre) after Algorithm 3.1, and the constructed cover C_Ω (right) with $\alpha = 1.25$.

Note that the number of neighbours $\mathrm{card}(C_i)$ of a particular cover patch ω_i constructed by Algorithm 3.1 is small, yet the amount of overlap of any two neighbouring patches is of significant size. Certainly, these features do come at a price we have to pay: The constructed point set P is larger than the given point set $\tilde{P}$; see Figures 3.1 and 3.2. This increases the number of cover patches $N = \mathrm{card}(C_\Omega)$; i.e. the number of block-rows of the stiffness matrix, and seemingly the overall computational cost. However, the total number of nonzero blocks of a stiffness matrix based on our hierarchical algorithm is comparable to the number of nonzero blocks of a stiffness matrix based on the direct covering for uniformly distributed point sets $\tilde{P}$ and it is substantially less for highly irregular point sets $\tilde{P}$, see [40, 76]. Furthermore, the proposed algorithm enables the user to control the amount of overlap $\omega_i \cap \omega_j$ of two neighbouring patches completely by the choice of $x_L \in C_L$ in step 5(b)ii, and the choice of the parameter α in step 5(b)iii.[11] Hence, this construction leads to smoother PU functions φ_i and allows for the use of cheaper quadrature schemes (compared with the nearest neighbour covering) during the assembly of the stiffness matrix. Note however, that the functions φ_i are still more complex than FE shape functions (see Figure 3.3).

Note that we obtain d-rectangular cover patches ω_i independent of the shape of the bounding box R_Ω. However, the aspect ratios are bounded; see Figure 3.2. In Section 3.2 we present a regularized version of Algorithm 3.1 which

[11] The original algorithm presented in [40] employs a further parameter $k \in \mathbb{N}$ in step 3 which controls the local imbalance of the tree, e.g. with a choice of $k = 0$ the constructed cell decomposition always corresponds to the cells of a uniform grid. Here, we allow for $k = \infty$, i.e. we impose no restrictions on the local imbalance of the tree. If we limit k, e.g. enforce $k = 1$, then there is no need for a complete neighbour search. In this situation we can directly compute the set of (possible) neighbours. Such a restricted tree construction is also employed in [60, 86].

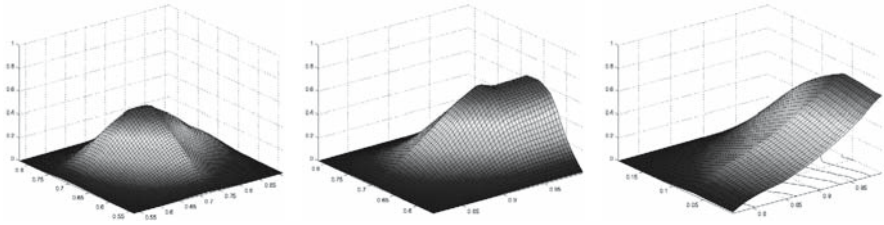


Figure 3.3. The PU functions φ_i on $\Omega \cap \omega_i$ generated by Algorithm 3.1 with the input data from Figure 3.2 for an interior point (left), a boundary point (centre), and a corner point (right) using linear B-splines in the Shepard construction (2.1).

guarantees the shape-regularity of the cover patches where all patches inherit the geometry and aspect ration of the bounding box R_Ω.

Remark 3.1.
Note that we usually use a very *small* overlap parameter $\alpha \in (1, 2]$ in our PUM. Here, a choice of $\alpha = 2$ leads to an overlap with $\text{diam}(\omega_i \cap \omega_j) \approx \text{diam}(\mathcal{C}_i)$ which corresponds to the amount of overlap of two finite element shape functions in a mesh-based method. Hence, we have a similar neighbour structure and sparsity pattern as in a FEM with a choice of $\alpha \in (1, 2]$. This is in contrast to many other meshfree methods where a rather large overlap is chosen. ∎

Remark 3.2.
The extension of Algorithm 3.1 for a least squares approximation or a Lagrangian particle method is straightforward. We only need to change step 5(b) where we introduce additional points x_L in empty tree cells $\mathcal{C}_L$ and have to choose the overlap parameter α with respect to the interaction potential of the particles. ∎

Computational Complexity
The computational complexity of our tree-based algorithm is of order $O(NJ)$ where $N = \text{card}(P) = \text{card}(C_\Omega)$ and J denotes the resulting depth of the tree, i.e. the number of levels of the tree, after all $\text{card}(\tilde{P})$ insert operations. For uniformly distributed point sets $\tilde{P}$ we have $N = \text{card}(P) = O(\text{card}(\tilde{P}))$ and $J = O(\ln N)$ such that the overall complexity of our tree-based cover construction algorithm is $O(N \ln N)$.
Note that the patches ω_L associated with INNER tree nodes in Algorithm 3.1 are used for the efficient neighbour search in the computation of the neighbourhoods C_i only, see Algorithm 3.2. They are not used for the construction of the partition of unity. If we assign patches $\omega_L \supset \bigcup \omega_S$ of minimal size to INNER tree nodes in step 5(a)ii, the computation of the local neighbourhood C_i of a particular patch ω_i with Algorithm 3.2 in general only requires $O(J)$ operations. Hence, for uniform point sets $\tilde{P}$ the computation of *all* neighbourhoods can be completed in $O(N \ln N)$ operations.

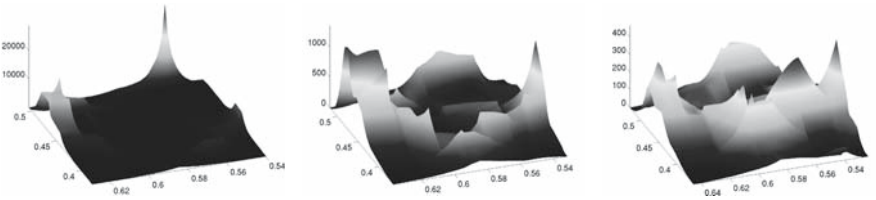


Figure 3.4. Surface plots of $\nabla\varphi_i \nabla\varphi_i$ for a partition of unity function based on a cover from Algorithm 3.1 using a linear spline as generating weight function $\mathcal{W}$ and an overlap parameter of $\alpha = 1.1$ (left), $\alpha = 1.3$ (centre), and $\alpha = 1.5$ (right).

3.2 Numerical Integration

In the following we focus on the efficient assembly of the stiffness matrix; i.e. on the numerical integration problem. Let us assume that the PU is given by an h-mesh construction like we have in the GFEM. Then, we know how to resolve the piecewise character of the integrands: We subdivide the integration domains $\omega_{ij} := \omega_i \cap \omega_j \cap \Omega$ with the help of the geometric elements of the h-mesh to obtain smooth (polynomial) integrands on the integration cells/elements. With our general PUM, however, we do not have a mesh or geometric elements. But we have support patches ω_i and weight functions W_i which define the partition of unity functions φ_i by (2.1). From this information only, we have to find an appropriate subdivision of the support patches ω_i and subsequently the integration domains. Furthermore, we have to cope with rational integrands on the cells of such a subdivision in our general PUM due to (2.1), cf. Figure 3.4.

We can obtain a suitable decomposition $D_{\omega_{ij}}$ of the integration domains ω_{ij} into disjoint integration cells $D^s_{\omega_{ij}}$ utilizing the product structure of the cover patches ω_i and the product structure of the weight functions W_i used during the construction (2.1) of the partition of unity $\{\varphi_i\}$ [39, 40]. The resulting decomposition $D_{\omega_{ij}} := \{D^s_{\omega_{ij}}\}$ is optimal in the sense that all discontinuities of the derivatives of the partition of unity functions, see (2.16), are resolved[12] with a minimal number of cells. Since the integrands are smooth on the disjoint integration cells a higher order quadrature rule can be successfully used. A reduction in the computational cost associated with the numerical integration requires the construction of partition of unity functions with simpler algebraic structure; i.e. which allow for a decomposition with less integration cells. To this end, we later present a regularized version of our cover construction which reduces the computational cost substantially, but requires only some minor changes to Algorithm 3.1.

[12] Note that this is not feasible when we use radial weight functions in Shepard's method. The integration scheme presented in [26] does not resolve all discontinuities of all derivatives of the respective Shepard functions.

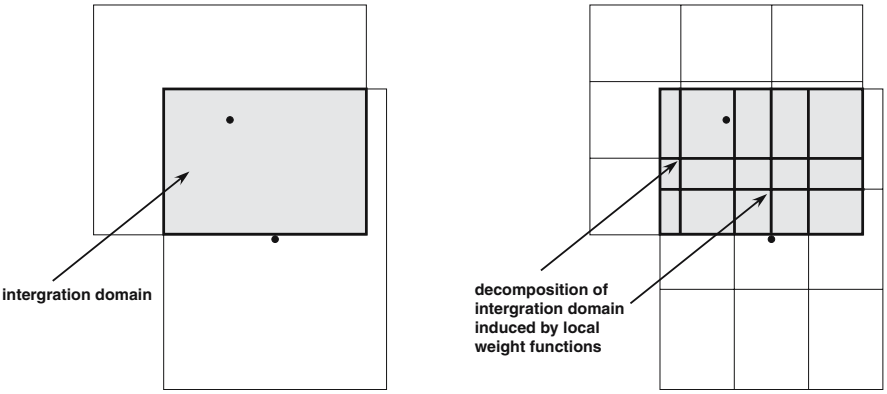


Figure 3.5. Integration domain $\Omega_{ij} = \omega_i \cap \omega_j$ (left). The decomposition $E_{\omega_{ij}}$ of the integration domain ω_{ij} via the subdivision induced by the weight functions W_i and W_j (right). Here, the weights are products of quadratic B-splines.

Let us consider the integration domain $\omega_{ij} = \omega_i \cap \omega_j \subset \Omega$ which is a product of intervals since our cover patches ω_i, ω_j are products of intervals, see Figure 3.5 (left). Moreover, the weight functions used W_k are products of normalized B-splines of order l, i.e. they are piecewise polynomials of degree l. Therefore, the weight function W_k induces a subdivision of the respective cover patch ω_k into $(l+1)^d$ sub-patches ω_k^q on which $W_k\big|_{\omega_k^q}$ is polynomial. Furthermore, these sub-patches ω_k^q are also products of intervals. With the help of the sub-patches ω_i^q, ω_j^q we can define a first decomposition $E_{\omega_{ij}} = \{E^s_{\omega_{ij}}\}$ of ω_{ij}, see Figure 3.5 (right). On the cells $E^s_{\omega_{ij}}$ of this decomposition we have that $W_i\big|_{E^s_{\omega_{ij}}}$ and $W_j\big|_{E^s_{\omega_{ij}}}$ are polynomials of degree l, but the weights $W_k\big|_{E^s_{\omega_{ij}}}$ for all other neighbours $\omega_k \in C_{ij} := C_i \cap C_j$ may still be piecewise polynomial only. Therefore, we further refine the decomposition $E_{\omega_{ij}}$ by subdividing the cells $E^s_{\omega_{ij}}$ with the help of the ω_k^q sub-patches for all $\omega_k \in C_{ij}$, see Figure 3.6. The resulting decomposition $D_{\omega_{ij}} = \{D^s_{\omega_{ij}}\}$ consists of d-rectangular cells $D^s_{\omega_{ij}}$ on which all weight functions $W_k|_{D^s_{\omega_{ij}}}$ are polynomials of degree l. The number of cells $\mathrm{card}(D_{\omega_{ij}})$ of the decomposition $D_{\omega_{ij}} = \{D^s_{\omega_{ij}}\}$ depends on the polynomial degree l of the weight functions W_k used during the Shepard construction (2.1) for the partition of unity, the number of neighbours $\mathrm{card}(C_{ij})$ and their geometric location.

Since all weights W_k are polynomial on the cells $D^s_{\omega_{ij}}$, the functions $\mathcal{T}$ and $\mathcal{G}_i$ (see Section 2.3) are non-singular rational functions on $D^s_{\omega_{ij}}$. Hence, any standard quadrature rule for smooth functions is applicable for the numerical integration of the weak form e.g. (2.16) and (2.17) on the cells $D^s_{\omega_{ij}}$ (if we assume that the local basis functions ψ_i^n and ψ_j^m are smooth on ω_{ij}). Independently of the local quadrature rule used on $D^s_{\omega_{ij}}$ we can utilize the product structure of the shape functions $\varphi_i\psi_i^n$ to reduce the computational cost of an

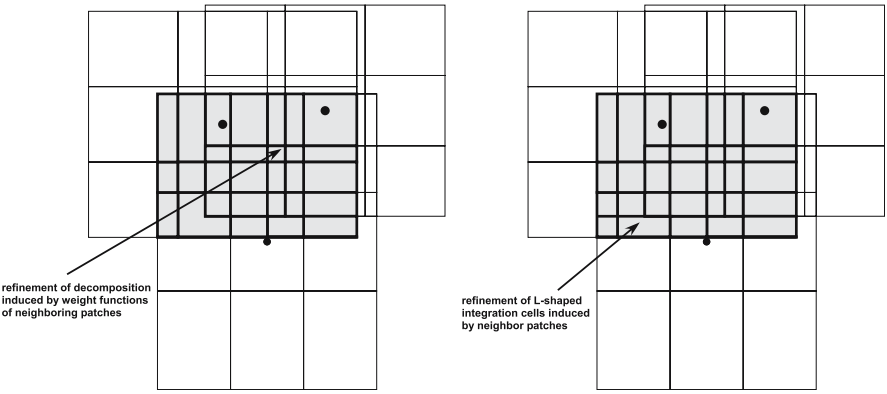


Figure 3.6. Refinement of the decomposition $E_{\omega_{ij}}$ of the integration domain ω_{ij} via the subdivision induced by the weight function W_k (product of quadratic B-splines) of one neighbouring patch ω_k (left). The resulting decomposition $D_{\omega_{ij}}$ after the refinement step for the neighbouring weight function W_k (right).

evaluation of the weak form at a quadrature point. To this end we evaluate the complete block $A_{i,j} = a(\varphi_j\psi_j^m, \varphi_i\psi_i^n) \in \mathbb{R}^{\dim(V_i^{p_i})\times\dim(V_j^{p_j})}$ of the stiffness matrix simultaneously rather than evaluating every single scalar entry $A_{(i,n),(j,m)} = a(\varphi_j\psi_j^m, \varphi_i\psi_i^n) \in \mathbb{R}$ for fixed n and m. Thereby, we reduce the number of (relatively expensive) evaluations of the PU functions φ_i and φ_j. Furthermore, this block-approach also allows for a hierarchical evaluation of the local basis functions ψ_i^n and ψ_j^m (which is available for the chosen Legendre polynomials) which reduces the computational cost of an evaluation of the weak form significantly (especially for higher order approximations).
A further reduction of the computational cost associated with the assembly of the stiffness matrix can be achieved by the simultaneous integration of a complete block-row of the discrete operator. Here, we evaluate all block-entries of the form $A_{i,\cdot} = a(\cdot, \varphi_i\psi_i^n)$ simultaneously.[13] Thereby, we reduce the number of evaluations of the weight functions W_l for $\omega_l \in C_i$. This approach is possible due to our decomposition scheme. Since the decomposition $D_{\omega_{ij}}$ of an integration domain ω_{ij} involves all neighbouring patches $\omega_l \in C_i \cap C_j$ it is clear that each cell $D^{\tilde{s}}_{\omega_{ij}}$ of such a decomposition $D_{\omega_{ij}}$ is also a cell $D^s_{\omega_{ii}}$ of the decomposition $D_{\omega_{ii}}$ for the diagonal block-entry, i.e. $D_{\omega_{ij}} \subset D_{\omega_{ii}}$ for all $\omega_j \in C_i$. Furthermore, the evaluation of the partition of unity function φ_i on the test side of the weak form already involves all non-vanishing weight functions W_l for $\omega_l \in C_i$. Therefore, we can compute the values for all non-vanishing PU functions φ_l for $\omega_l \in C_i$ simultaneously; i.e. we can evaluate all PU functions φ_j on the trial side from the data necessary for the computation of the value of the PU function φ_i on the test side of the weak form. Hence,

[13] This simultaneous integration procedure can be viewed as a generalization of the assembly of the stiffness matrix by element matrices in the FEM.

every non-vanishing weight function W_l is evaluated only once per quadrature point independent of the quadrature rule used.
For the selection of an appropriate quadrature rule on the cells $D^s_{\omega_{ij}}$ we now can assume the smoothness of the integrands due to our decomposition approach. But still the quadrature rule has to be applicable to general situations (general covers, weights and local basis functions ψ^n_i, etc.). Hence, we have to find a fast converging, cheap quadrature rule on $D^s_{\omega_{ij}}$ which allows for a reliable dynamic stopping criterion for a wide range of integrands.
So-called *sparse grid quadrature* [33] rules are multidimensional interpolatory rules with a substantially smaller number of integration nodes compared with a tensor product rule. They are defined as special products of one-dimensional interpolatory quadrature rules. Although the number of evaluations of the integrand is significantly less for a sparse grid quadrature rule, the order of the approximation is comparable to that of a full tensor product rule. Here, we only state the fundamental construction principles and error bounds, see [33] and the references cited therein for further details.
Consider a sequence of nested one-dimensional quadrature rules $\{Q^1_l \mid Q^1_l f := \sum_{i=1}^{n^1_l} w_{li} f(x_{li}),\ n^1_l = O(2^l)\}$ for univariate functions $f : \mathbb{R} \to \mathbb{R}$ with weights w_{li}, nodes x_{li} and an error bound $|Q^1_l f - \int f| = O(2^{-lr})$ where f is assumed to be r-times continuously differentiable. These assumptions hold for example for the Clenshaw–Curtis and Gauss–Patterson [71] rules. With the help of the difference quadrature rules Δ^1_k

$$\Delta^1_k f := (Q^1_k - Q^1_{k-1}) f \quad \text{with} \quad Q^1_0 f := 0$$

we can define the sparse grid quadrature rule Q^d_l on level l in d dimensions as

$$Q^d_l f := \sum_{\sum_{i=1}^{d} k_i \le l + d - 1} (\Delta^1_{k_1} \otimes \cdots \otimes \Delta^1_{k_d}) f$$

with $f : \mathbb{R}^d \to \mathbb{R}$ now denoting a multivariate function, $l \in \mathbb{N}$ and $k = (k_i)_{i=1}^d \in \mathbb{N}^d$. Due to the restriction $\sum_{i=1}^d k_i \le l + d - 1$ in the summation, the number n^d_l of quadrature points x^d_i of the resulting sparse grid quadrature rule Q^d_l is only

$$n^d_l = O(2^l l^{d-1}).$$

Hence, the number of function evaluations for a sparse grid quadrature rule is dramatically less (see Figure 3.7) than for a full tensor product rule where the integrand has to be evaluated at $O(2^{ld})$ quadrature points. This reduction of the computational cost, however, does not compromise the approximation quality significantly for smooth functions. When f is assumed to be r-times continuously differentiable the following estimate holds:

$$|Q^d_l f - \int f| = O(2^{-lr} l^{(d-1)(r+1)}).$$

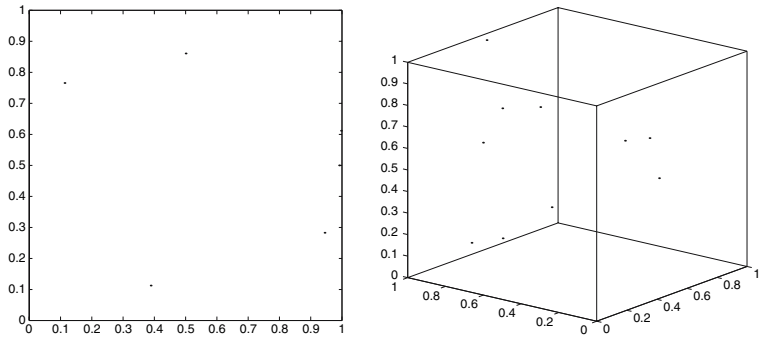


Figure 3.7. Quadrature nodes of two sparse grid Gauss–Patterson rules, level $l = 6$ with 769 nodes in two dimension (left) and level $l = 5$ with 1023 nodes in three dimension (right).

In summary, sparse grid quadrature rules are not only cheaper to evaluate (especially in higher dimensions) compared with tensor product rules, but rather their overall efficiency with respect to accuracy is significantly better. In [33] the fast convergence of sparse grid quadrature rules based on Gauss–Patterson rules (see Figure 3.7) is shown for a wide variety of function classes. In fact, they have a polynomial exactness of $3 \cdot 2^{l-1} - 1$ and converge exponentially for smooth integrands. Since, our integrands are smooth on the cells $D^s_{\omega_{ij}}$ of the constructed decomposition $D_{\omega_{ij}}$ we use Gauss–Patterson sparse grid rules for the numerical integration of the entries of the stiffness matrix. To ensure a reliable accuracy of our quadrature scheme, we use a simple three level dynamic stopping criterion [69]. The quadrature on a cell $D^s_{\omega_{ij}}$ is stopped if

$$|Q^d_{l-1} f - Q^d_{l-2} f| \leq c_1 \epsilon_a + c_2 \epsilon_r |Q^d_{l-1} f| \quad \text{and} \quad |Q^d_l f - Q^d_{l-1} f| \leq \epsilon_a + c_3 \epsilon_r |Q^d_l f|$$

hold for all the integrals $A_{(i,n),(j,m)}$ of each block $A_{i,j}$. Here, c_1, c_2 and c_3 are non-negative constants and ϵ_a and ϵ_r are user supplied absolute and relative tolerances. These tolerances which determine the accuracy of the integration have to be chosen with respect to the approximation space. Here, the diameters $\mathrm{diam}(\omega_i)$, $\mathrm{diam}(\omega_j)$ of the cover patches ω_i, ω_j, the number of integration cells $\mathrm{card}(D_{\omega_{ij}})$, their respective diameters $\mathrm{diam}(D^s_{\omega_{ij}})$ and the local approximation orders p_i and p_j have to be considered. An automatic selection of the tolerances ϵ_a and ϵ_r which minimizes the computational work but at the same time does not compromise the accuracy of the discretization [77] is an open problem. Note that we limit the stopping criterion to the integrals $A_{(i,n),(i,m)}$ associated with the block-diagonal entry $A_{i,i}$ of the stiffness matrix if we compute all integrals of a complete block-row simultaneously. The overall numerical scheme for the assembly of the stiffness matrix is given in Algorithm 3.3. Note that in step 4(b)iii we need to evaluate all weight functions W_j only once per integration node p_l.

Algorithm 3.3. Assembly of Stiffness Matrix $A = (A_{i,j}) = (A_{(i,n),(j,m)})$
For all $i = 1, \ldots, N$:

1. For all $\omega_j \in C_i$:
   Set $A_{i,j} = 0$.
2. Compute decomposition $D_{\omega_{ii}} = \{D^s_{\omega_{ii}}\}$ of patch ω_i via neighbouring patches $\omega_j \in C_i$ and respective weight functions W_j.
3. Compute decomposition $D^{\Omega}_{\omega_{ii}} = \{D^{s,\Omega}_{\omega_{ii}}\}$ of integration domain $\omega_i \cap \Omega$.
4. For all integration cells $D^{s,\Omega}_{\omega_{ii}} \in D^{\Omega}_{\omega_{ii}}$:
   (a) Set transformation $T_{\omega_{ii}} : [-1,1]^d \to D^{s,\Omega}_{\omega_{ii}}$.
   (b) For all $\omega_j \in C_i$ with $\omega_j \cap D^{s,\Omega}_{\omega_{ii}} \neq \emptyset$:
      i. Set integration level $l = 1$.
      ii. Set $A^{s,\Omega}_{i,j} = 0$.
      iii. For all integration points p_l on level l:
         Set $p_l^{\omega_{ii}} = T_{\omega_{ii}}(p_l)$.
         Set $A^{s,\Omega}_{i,j} = A^{s,\Omega}_{i,j} + a(\varphi_j(p_l^{\omega_{ii}})\psi_j^m(p_l^{\omega_{ii}}), \varphi_i(p_l^{\omega_{ii}})\psi_i^n(p_l^{\omega_{ii}}))$.
      iv. If stopping criteria for all $A^{s,\Omega}_{i,j}$ are fulfilled:
         Set $A_{i,j} = A_{i,j} + A^{s,\Omega}_{i,j}$. ($\to$ 4)
      v. Else:
         If $l < 6$: Increase integration level $l = l + 1$. ($\to$ 4(b)ii)
         Else: Refine $D^{s,\Omega}_{\omega_{ii}}$. ($\to$ 4)

Remark 3.3.
In step 4(b)ii of Algorithm 3.3 we need to employ a volume integration rule as well as a surface integration rule since the bilinear form $a(\cdot,\cdot)$ in general consists of volume and surface integrals due to Nitsche's approach. To this end, we use the sparse grid construction in d dimensions for the volume terms and in $d-1$ dimensions for the surface terms. ∎

Remark 3.4.
From the product construction of the sparse grid quadrature rules it is obvious that the proposed numerical integration scheme will be most effective for PUM shape functions $\varphi_i\psi_i^n$ where the higher order approximation function ψ_i^n is a product function itself; e.g., a tensor product of local univariate polynomials. If we use other approximation functions with radial characteristic or augment a local polynomial space by singular or other custom functions [3, 6, 7, 61] we may need to employ other quadrature rules (at least for the singular shape functions) in step 4(b)iii of Algorithm 3.3. But since the local basis $\{\psi_i^n\}$ used on patch ω_i is known in advance we can select an appropriate quadrature rule prior to the assembly of the stiffness matrix. ∎

Remark 3.5.
Since the discontinuities of the derivatives of the partition of unity functions φ_i are located on the boundaries $\partial D^{s,\Omega}_{\omega_{ii}}$ of the integration cells $D^{s,\Omega}_{\omega_{ii}}$, the local quadrature scheme (for the volume terms) should not involve integration nodes p_l on the boundary $\partial D^{s,\Omega}_{\omega_{ii}}$ of the integration cells; i.e. so-called open rules like the Gauss–Patterson rules should be used. ∎

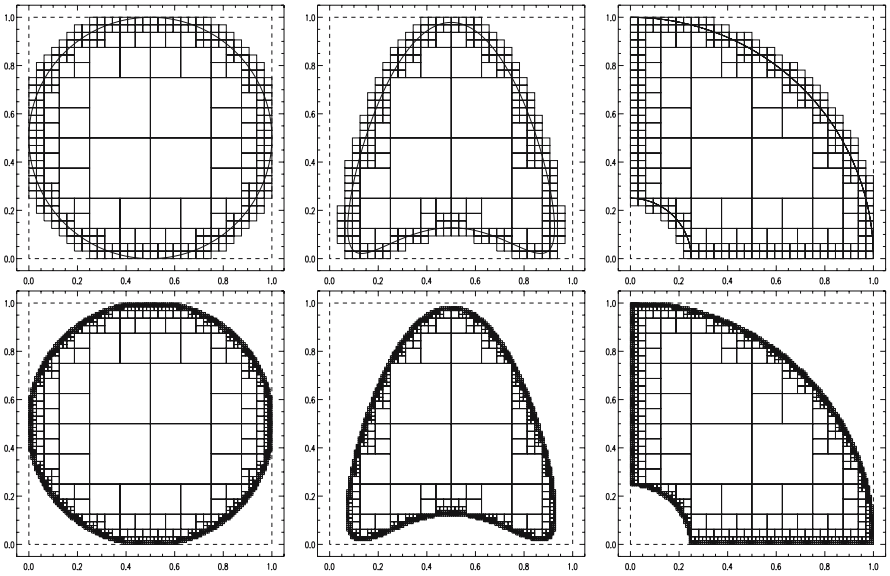


Figure 3.8. Tree-based approximation of a spherical domain (left), a smooth non-convex domain (centre), and a quarter of a spherical domain with a spherical hole (right) on level $k = 5$ (upper row) and level $k = 7$ (lower row).

Note that in step 3 we assume that we can easily compute the intersection of the d-rectangular cell $D^s_{\omega_{ii}}$ with the domain Ω. Furthermore, in step 4 we assume that a (smooth) transformation $T_{\omega_{ii}} : [-1, 1]^d \to D^{s,\Omega}_{\omega_{ii}}$ exists for each integration cell $D^{s,\Omega}_{\omega_{ii}} = D^s_{\omega_{ii}} \cap \Omega$. These assumptions are essentially due to the use of integration schemes based on products of univariate quadrature rules. In general, they can only be fulfilled if we employ an appropriate approximation to the domain Ω. With the help of our tree construction we can obtain such an approximation without much additional work.

Domain Approximation

Recall that in steps 5(a)ii and 5(b)iv of Algorithm 3.1 we only consider tree cells $\mathcal{C}_L$ which intersect the computational domain Ω; i.e. we need to define shape functions on a patch ω_L only if for the respective tree cell $\mathcal{C}_L \cap \Omega \neq \emptyset$ holds. Then we set the respective patches ω_L in such a way that they cover their associated tree cell $\mathcal{C}_L$, i.e. $\omega_L \supset \mathcal{C}_L$. Hence, the cover patches do not only cover the computational domain Ω but also the union of all tree cells $\mathcal{C}_L$ with $\mathcal{C}_L \cap \Omega \neq \emptyset$. Therefore, the cover $C_\Omega = \{\omega_L\}$ is a valid cover for Ω as well as for $\Omega_A := \bigcup_L \mathcal{C}_L$ with $\mathcal{C}_L \cap \Omega \neq \emptyset$. The approximation $\Omega_A \supset \Omega$ can be simplified with the help of the tree to only consist of a minimal number of cells. To this end, we assign to each tree node the information if the respective

tree cell $\mathcal{C}_L$ is completely contained in the domain Ω or if it intersects the boundary $\partial\Omega$. With these data we can easily obtain an approximation to the domain Ω and its boundary $\partial\Omega$ with a minimal number of cells by descending the tree only for those nodes which intersect the boundary $\partial\Omega$, see Figure 3.8. By construction the cells of this approximation are d-rectangular so that the assumptions in Algorithm 3.3 are automatically fulfilled.

Remark 3.6.
Other approximation techniques for the computational domain can also be used within the PUM. Note, however, that appropriate quadrature rules must be available for the respective domain cells. For instance, if the domain is approximated via a triangulation then quadrature rules on triangles must be used in Algorithm 3.3. ∎

Remark 3.7.
The tree-based approximation Ω_A given above can also serve as a base for a triangulation $\mathcal{T}_h$ of the domain Ω, see e.g. [12]. ∎

The overall computational cost of Algorithm 3.3 depends on the number of cells $\mathrm{card}(D_{\omega_{ii}})$ of the decomposition, i.e. on the order l of the weight functions, the geometric location of the neighbours $\omega_j \in C_i$, their number $\mathrm{card}(C_i)$, and the local quadrature rule used on the integration cells. The order l of the spline weight function is determined by the global continuity requirements. We always use the smallest allowable order l to minimize the computational work, i.e. for a PDE of second order we use a linear spline weight function so that $l = 1$ and $\varphi_i \in \mathcal{C}^0$. Due to our tree construction and the small overlap parameter $\alpha \in (1, 2]$ we obtain small neighbourhoods C_i which limits the number of integration cells necessary to resolve the discontinuities of the derivatives of φ_i. Furthermore, the use of sparse grid rules on each integration cell reduces the computational cost with respect to a single integration cell significantly compared with tensor product rules.

Regularized Cover Construction
A further reduction of the computational cost associated with the assembly of the stiffness matrix can be achieved only by reducing the number of cells of the decomposition $D_{\omega_{ij}}$.[14] This can be attained by the alignment of the cover patches ω_k and their subdivisions $\{\omega_k^q\}$. Taking into account that we limit ourselves to the use of tensor product B-splines as weight functions W_i in the

[14] The decomposition itself, however, is minimal in the sense that it has a minimal number $\mathrm{card}(D_{\omega_{ij}})$ of integration cells necessary to resolve the piecewise character of the PU functions. In our construction (2.1) of the PU we have to allow for higher orders t of the B-spline weights to be able to construct global solutions u^{PU} with higher order regularity; i.e. $u^{\mathrm{PU}} \in \mathcal{C}^{t-1}$. Therefore, the remaining influences on the computational effort involved with the numerical integration of the stiffness matrix entries are the geometric neighbouring relations of our cover patches ω_i.

construction of the PU (2.1) we can align the cover patches to simplify the algebraic structure of the resulting partition of unity functions φ_i. Here, we eliminate some of the flexibility in step 5(b) of Algorithm 3.1 for the choices of x_L and α. This, however, does not lead to a significantly larger number of neighbours. Hence, the number of nonzero blocks of the stiffness matrix stays (almost) constant. Yet, the number of integration cells $\mathrm{card}(D_{\omega_{ij}})$ is substantially reduced by this modification, see [40, 76] for details.
Recall that we split the integration domain ω_{ij} into several cells by its intersections $\omega_{ij} \cap \omega_k^q$ with the cells ω_k^q of the subdivision induced by the weight W_k on $\omega_k \in C_{ij}$ during the construction of the decomposition $D_{\omega_{ij}}$. Hence, we align these intersections $\omega_{ij} \cap \omega_k^q$, which subsequently induce at least one integration cell $D^s_{\omega_{ij}}$, if we align the neighbouring cover patches ω_k with respect to their subdivisions $\{\omega_k^q\}$. Therefore, many of the $\omega_{ij} \cap \omega_k^q$ will lead to the *same* integration cell $D^s_{\omega_{ij}}$, and the overall number of integration cells $\mathrm{card}(D_{\omega_{ij}})$ will be reduced significantly. This alignment of the cover patches ω_k and their subdivisions $\{\omega_k^q\}$ is achieved by the following algorithm where we make changes to the original Algorithm 3.1 only in step 5(b).[15] However, we give the overall algorithm for the sake of completeness.

Algorithm 3.4. Regular Hierarchical Cover Construction

1. Given the domain $\Omega \subset \mathbb{R}^d$ and a bounding box $R_\Omega = \bigotimes_{i=1}^d [l_\Omega^i, u_\Omega^i] \supset \overline{\Omega}$.
2. Given the initial point set $\tilde{P} = \{x_j \,|\, x_j \in \overline{\Omega}, j = 1, \ldots, \tilde{N}\}$.
3. Build a d-binary tree over R_Ω such that per leaf L at most one $x_i \in \tilde{P}$ lies within the associated cell $\mathcal{C}_L := \bigotimes_{i=1}^d [l_L^i, u_L^i]$; see Figure 3.1.
4. Set $P = \emptyset$, $C_\Omega = \emptyset$.
5. For the root cell $\mathcal{C}_L = \bigotimes_{i=1}^d [l_L^i, u_L^i] = R_\Omega$:
   (a) If current tree cell $\mathcal{C}_L$ is an INNER tree node and $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Descend tree for all successors $\mathcal{C}_S$ of $\mathcal{C}_L$. ($\rightarrow$ 5(a))
      ii. Set patch ω_L such that $\bigcup \omega_S \subset \omega_L$.
   (b) Else if $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Set patch $\omega_L = \bigotimes_{i=1}^d [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$ where $h_L^i = \frac{\alpha_t}{2}(u_L^i - l_L^i)$, $x_L^i = l_L^i + \frac{1}{2}(u_L^i - l_L^i)$, and $\alpha_t > 1$.
      ii. Set $P = P \cup \{x_L\}$, $C_\Omega = C_\Omega \cup \{\omega_L\}$.

The parameter $\alpha_t \in (1, 2]$ in the computation of the support size in step 5(b)i is dependent only on the weight function used in (2.1); i.e. the order t of the B-spline. By construction the one-dimensional distances from a point $x_L \in P$ to its direct neighbouring point $x_j \in P$, i.e. the point x_j corresponding to the sibling tree cell $C_j = \bigotimes_{i=1}^d [l_j^i, u_j^i]$, are $|x_L^i - x_j^i| = u_L^i - l_L^i = u_j^i - l_j^i$, where $C_L = \bigotimes_{i=1}^d [l_L^i, u_L^i]$ is the cell associated with x_L. Hence, if we choose α_t in such a way that condition (3.1) is fulfilled, we align not only the patch ω_L with its direct neighbouring patch ω_j but rather also their corresponding

[15] Hence, the approximation of the computational domain is not affected by this regularization technique.

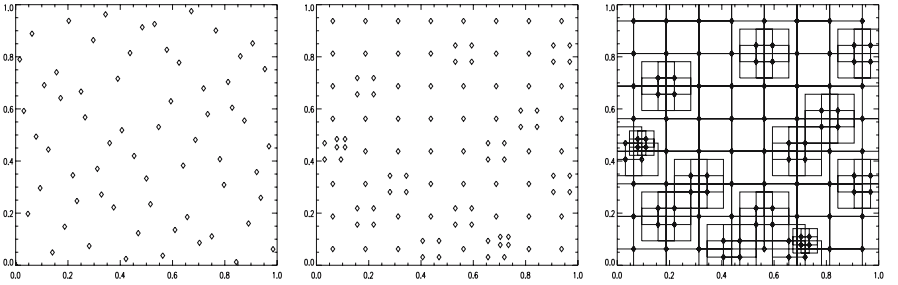


Figure 3.9. Points of an initial Halton$(2,3)$ point set $\tilde{P}$ with $\tilde{N} = \mathrm{card}(\tilde{P}) = 64$ points distributed in $R_\Omega = \Omega = [0,1]^2$ (left), the points of the generated point set P with $\mathrm{card}(P) = 106$ (centre) after Algorithm 3.4, and the constructed cover C_Ω with $\alpha_t = 2$ (right).

subdivisions $\{\omega_L^q\}$ and $\{\omega_j^q\}$ induced by the weight functions W_L and W_j; see Figure 3.9. Moreover, this alignment of the patches does not increase the number of neighbours $\mathrm{card}(C_L)$. With the notation $h_t^i := \frac{\alpha_t}{t+1}(u_L^i - l_L^i)$ for the B-spline interval size, the condition reads

$$x_L^i + \tfrac{t+1}{2} h_t^i = x_j^i - \left(\tfrac{t+1}{2} - m\right) h_t^i = x_L^i + (u_L^i - l_L^i) - \left(\tfrac{t+1}{2} - m\right) h_t^i \quad (3.1)$$

for the ith coordinate with $i = 1, \ldots, d$. Here, the parameter $m \in \mathbb{N}$ indicates the amount of overlap $\omega_L \cap \omega_j \sim \bigotimes_{i=1}^d m h_t^i$ for the neighbour $\omega_j \in C_L$. Any integer m with $1 \le m \le \frac{t+1}{2}$ leads to minimal neighbourhoods C_L and minimal decompositions $D_{\omega_{Lj}}$; i.e. the number of nonzero entries of the stiffness matrix $\sum_L \mathrm{card}(C_L)$ and the number of integration cells $\sum_{L,j} \mathrm{card}(D_{\omega_{Lj}})$ are (almost) constant. Therefore, it is advisable to choose the largest such integer to control the gradients of the PU function φ_i. Solving (3.1) for α_t we have

$$\alpha_t = \frac{t+1}{t+1-m}.$$

With the choice of $t = 2n-1$ and maximal $m = n$, this yields $\alpha_t = 2$; in general, we have $1 < \alpha_t \le 2$. Due to this construction many of the points $x_i \in P$ are covered only by the corresponding ω_i. Therefore, we have $\varphi_i(x_j) = \delta_{ij}$ for many PU functions φ_i and points $x_j \in P$; see Figure 3.10. In fact, $\varphi_i(x) = 1$ holds not only for the point $x = x_i$ if we have $\alpha_t < 2$ but rather on a subpatch $\tilde{\omega}_i \subset \omega_i$ with $x_i \in \tilde{\omega}_i \sim \bigotimes_{i=1}^d h_t^i$; i.e. $\varphi_i|_{\tilde{\omega}_i} \equiv 1$; see Figure 3.10.
When we compare the covers C_Ω (Figures 3.2 and 3.9), the partition of unity functions φ_i (Figures 3.3 and 3.10), and the respective integrands (Figures 3.4 and 3.11) generated by Algorithms 3.1 and 3.4, we clearly see the effect of the alignment of the cover patches.
Note also that the cover patches ω_L constructed with Algorithm 3.4 and the bounding box R_Ω always have the same aspect ratio; see Figure 3.9. If we apply the algorithm given above to $\Omega = R_\Omega = [0,1]^d$ with $\alpha_t = 2$ to

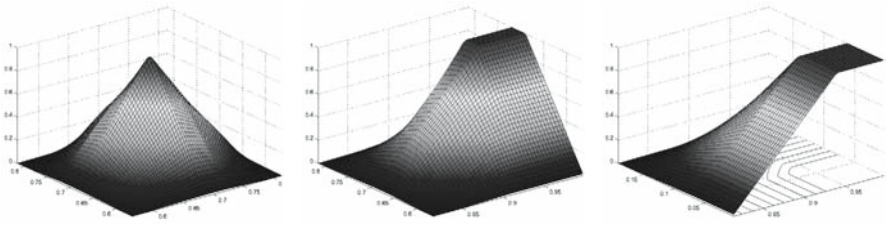


Figure 3.10. The PU functions φ_i on $\Omega \cap \omega_i$ generated by Algorithm 3.4 with the input data from Figure 3.9 for an interior point (left), a boundary point (centre), and a corner point (right) using linear B-splines ($t = 1$, $\alpha_t = 2$) in the Shepard construction (2.1).

a uniformly distributed set of points $\tilde{P}$, we construct a uniform grid (or at least an r-irregular grid with very small r depending only on the quality of the initial point set $\tilde{P}$; see Figure 3.9). Here, also the cells $D^s_{\omega_{ij}}$ of the decomposition $D_{\omega_{ij}}$ are (geometrically) identical to a bilinear finite element. Furthermore, the PU $\{\varphi_i\}$ generated by (2.1) will again be piecewise linear for $t = 1$ just like their finite element counterpart in the GFEM (see Figure 3.10). Hence, in this situation our method does reconstruct functions φ_i that are identical to bilinear finite element functions, and also our general decomposition algorithm will recover the corresponding geometric elements. Hence, the number of integrals to be evaluated in this situation with our method or a FEM/GFEM are the same.

So far we were only concerned with the computational cost during the integration and the influence the shape functions $\varphi_i\psi_i^n$ have on the computational efficiency of our PUM. Another important issue, however, is the stability of the basis of our PUM space. Here, we also have to address the question of whether the functions $\varphi_i\psi_i^n$ are indeed a basis. In the case of $t = 1$ and $\alpha_t = 2$ the alignment of the cover patches ω_i and their respective weight subdivisions $\{\omega_i^q\}$ leads to the reconstruction of the finite element hat functions for the PU. Hence, our PUM reduces to the GFEM in this situation. It is well-known [8,9,78] that the GFEM (in general) generates linearly dependent shape functions $\varphi_i\psi_i^n$, the so-called nullity of the method. This is essentially due to the fact that in the GFEM the PU functions φ_i already reconstruct the linear polynomial. With our approach, the φ_i only reconstruct the linear polynomial away from the boundary; close to the boundary we have $\varphi_i \equiv 1$. Therefore, the shape functions are not linearly dependent. However, since the small boundary layer where $\varphi_i \equiv 1$ decreases with larger N, the condition number κ of the mass matrix is dependent on N; i.e. the basis is not stable. A simple cure for this stability problem is to use $m < 1$ in (3.1) when we have $t = 1$; i.e. we limit ourselves to $1 < \alpha_t < 2$ when $t = 1$. With $\alpha_t < 2$ we can find a subpatch $\tilde{\omega}_i \subset \omega_i$ with $\mathrm{vol}(\omega_i) \leq C\,\mathrm{vol}(\tilde{\omega}_i)$, where $\varphi_i\,|_{\tilde{\omega}_i} \equiv 1$ for many i. Therefore, the PU functions φ_i no longer reconstruct the linear polynomial independent of N, and the resulting shape functions form a stable basis. We therefore allow for any value $1 < \alpha_t < 2$ in Algorithm 3.4

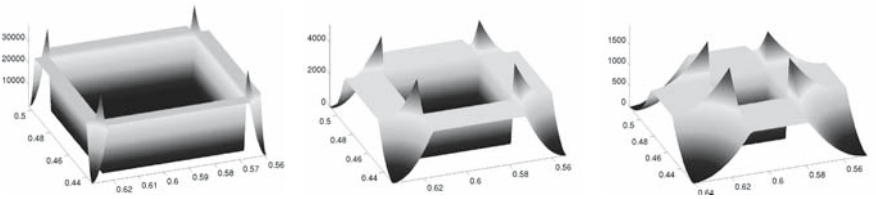


Figure 3.11. Surface plots of $\nabla\varphi_i\nabla\varphi_i$ for a partition of unity function based on a cover from Algorithm 3.4 using a linear spline as generating weight function $\mathcal{W}$ and an overlap parameter of $\alpha = 1.1$ (left), $\alpha = 1.3$ (centre), and $\alpha = 1.5$ (right).

if $t = 1$.[16] The number of integration cells increases somewhat due to this generalization. The patches ω_i are still aligned, but their respective weight subdivisions are not. However, a comparison of the average number of integration cells required to resolve the piecewise character of the resulting PU functions [40, 76] showed that we still need substantially fewer integration cells compared with the covers from Algorithm 3.1.

Note that with Algorithm 3.4 we now may have $P \cap \tilde{P} = \emptyset$. However, this is admissible due to the non-interpolatory character of the PUM shape functions $\varphi_i\psi_i^n$. We can interpret this change in the point set P as a change of the weight functions W_k used during the Shepard construction (2.1). So far the weight functions W_k and the cover patches ω_k were assumed to be centred on the given point x_k (cf. Section 2.1), this, however, is not a necessary condition for the PUM to work. Note that the constructed point set P is only part of the implementation of the function space. The given point set $\tilde{P}$ is still the set of all relevant points for the resolution of the function space and the approximation of the domain. We can either store a separate copy of the initial point set $\tilde{P}$ which is used in time dependent settings to generate covers for future time steps [39] or allow for the associated patch ω_k and weight function W_k to be centered at a point other than x_k, e.g. with Algorithm 3.4 the weight functions W_k and cover patches ω_k are now centered at $l_L + \frac{1}{2}(u_L - l_L)$ rather than x_k.[17] This leaves the given points at their original location and as before we have $P = \tilde{P} \cup Q$ with Algorithm 3.1.

Computational Complexity

The optimal computational complexity associated with the assembly of the stiffness matrix A corresponds to the number of nonzeros of A, in our case $O(Np^{2d})$. If numerical integration is used this optimal complexity can hardly be realized. First of all, we need an *a priori* estimate on the allowable error due to numerical integration [77] to establish a stable approximation.

[16] A similar problem arises for higher order splines $t > 1$ only if $\alpha_t > 2$; e.g., we need $\alpha_t = 4$ with $t = 2$. Therefore, we can stay with the minimal values of $\alpha_t = 1.5$ if $t = 2$ and $\alpha_t = 2$ if $t = 3$.

[17] Here, a cover patch ω_k now needs to store the coordinates of the given point x_k, the centre x_L of the associated tree cell $\mathcal{C}_L$ and the radii h_L.

For the PUM and most meshfree Galerkin methods this is an open problem, see Remark 3.8 below. Hence, we currently employ a dynamic stopping criterion with user supplied relative and absolute tolerances for our numerical integration scheme which makes the analysis of the computational cost associated with the assembly of the stiffness matrix very challenging.[18] Under some reasonable assumptions, however, we can attain an estimate of the computational work.

In general the cost $\mathcal{C}_{\mathrm{NI}}$ associated with the numerical integration of a single entry of the stiffness matrix A is given by

$$\mathcal{C}_{\mathrm{NI}} = O(n_{\mathrm{IC}}\ n_{\mathrm{IN}}\ \mathcal{C}_{\mathrm{EI}})$$

where n_{IC} denotes the number of integration cells, n_{IN} the number of integration nodes per cell, and $\mathcal{C}_{\mathrm{EI}}$ the cost associated with the evaluation of the integrand. The number of integration cells n_{IC} in our implementation, see Algorithm 3.3, is determined by the number of jumps of the derivatives of φ_i. Due to our regularized cover construction, see Algorithm 3.4, we have $n_{\mathrm{IC}} = O(3^d(t+1)^d)$ where t denotes the order of the spline weight functions $\mathcal{W}$ used.

On each of the integration cells the integrands are smooth so that a higher order scheme can be used. We use a sparse grid Gauss–Patterson scheme with $n_{\mathrm{IN}} = O(2^q q^{d-1})$ integration nodes where q denotes the refinement level of the univariate Gauss–Patterson rule which has a polynomial exactness of $3 \cdot 2^{q-1} - 1$. Note that the sparse grid construction preserves this exactness for higher dimensions d.

Due to our regularized cover construction we may make the assumption that a partition of unity function φ_i can be well approximated by a piecewise polynomial of degree l, cf. Figure 3.10. If we further assume that the coefficients of $a(\cdot,\cdot)$ are piecewise constant functions, we can approximate all integrands by a polynomial of degree $(p+t)^2$ on each integration cell and we can choose the refinement level $q \approx \ln(p+t)$ for the sparse grid Gauss–Patterson scheme. Under these assumptions we can estimate the computational cost $\mathcal{C}_{\mathrm{A,NI}}$ associated with the assembly of the stiffness matrix by

$$\mathcal{C}_{\mathrm{A,NI}} = O(N(p+t)(\ln(p+t))^{d-1}(dp + p^d + p^{2d})).$$

Hence, $\mathcal{C}_{\mathrm{A,NI}}$ is optimal up to factor of $O(p(\ln p)^{d-1})$ for a fixed weight function $\mathcal{W}$.

Remark 3.8.

An open problem in the PUM and most meshfree methods is the question of selecting an appropriate accuracy for the stopping criterion in the numerical integration of the entries of the stiffness matrix. Here, the goal is to allow for the largest admissible integration error to reduce the computational cost, yet

[18] Note also that the approximation may become instable if the user prescribed tolerances are too crude.

to maintain the order of approximation of the overall discretization [77]. Since the size of an integration cell in Algorithm 3.3 may not be comparable to the support size it is not an easy task to obtain an estimate which balances the error due to integration and the approximation error without very restrictive assumptions on the distribution of the points $x_i \in P$. ∎

Remark 3.9.
The alignment of the cover patches ω_i in Algorithm 3.4 leads to partition of unity functions with simpler algebraic structure, see Figure 3.10. Moreover, this alignment generates regions in the domain where the neighbourhoods C_i of the respective patches ω_i are geometrically equivalent. Hence, the associated partition of unity functions φ_i are only shifted versions of the same function $\tilde{\varphi}$. Therefore, the respective entries in the stiffness matrix are identical if the PDE has constant coefficients and if the same local basis functions are employed on these shifted patches. In this case, we can reduce the number of integrals that need to be evaluated dramatically; i.e. we only compute a specific stencil once and reuse the computed values for equivalent patches. ∎

Remark 3.10.
In a FEM the integrals associated with the right hand-side vector $\hat{f}$ are often not evaluated directly. If a coefficient vector $\tilde{f}$ of the right hand-side f of the PDE is available it is sufficient to approximate $\hat{f}$ by the product $M\tilde{f}$ of the mass matrix M and the coefficient vector $\tilde{f}$. This can reduce the computational cost substantially especially when the same problem is solved for multiple right hand-sides. Often an interpolation of f is used to obtain a coefficient vector $\tilde{f}$. Since the shape functions of the PUM are non-interpolatory we need to find a different approach to attain an acceptable approximation to $\tilde{f}$. The construction of this approximation, however, must be very efficient so that the overall computational cost are in fact reduced. In Section 3.3 we present a very cheap localized projection technique which can be used to obtain such a valid approximation $\tilde{f}$.
Note that in our PUM not only need the mass matrix M but also more general moment matrices $M_{\partial\Omega}$ on the boundary to approximate the surface terms due to Nitsche's method. The caching technique discussed in Remark 3.9 can be used for the mass and the moment $M_{\partial\Omega}$ matrices on the boundary. ∎

Remark 3.11.
With Algorithm 3.3 we implicitly define a particular integration scheme for a certain bilinear form. For a different bilinear form we may obtain a different decomposition and might employ local integration rules on a different level l. Hence, if we use Algorithm 3.3 for instance for the assembly of the stiffness matrix associated with a Poisson problem like (2.7) and for the assembly of the respective mass matrix independently, the two resulting approximations may be incompatible; i.e. they are evaluated at different integration nodes p_l. If multiple operators, i.e. bilinear forms, need to be assembled for a single simulation it is advisable to evaluate all operators simultaneously using

the same integration nodes. To this end, we can either modify Algorithm 3.3 to deal with multiple bilinear forms simultaneously or we can implement a modification where a pre-computed decomposition is reused. Here, the stopping criteria should be evaluated for the highest order term. Similarly, the right hand-side should also be evaluated at the same integration nodes as the operator on the respective left hand-side. ∎

Remark 3.12.
Note that the assembly of the stiffness matrix A does not make explicit use of the tree data structure. Here, we employ a sparse matrix data structure (for sparse block-matrices with dense blocks) to store the matrix A. Once the neighbourhoods C_i are known the evaluation of a partition of unity function and the matrix assembly are independent of the tree construction. ∎

3.3 Multilevel Solution of Linear System

For a PUM discretization in d dimensions, the number of degrees of freedom is of the order $\mathsf{dof} = O(Np^d)$ where N denotes the number of cover patches and p is the approximation order. The number of nonzeros nnz of the stiffness matrix is of the order $O(Np^{2d})$. To allow for an efficient and scalable meshfree simulation the employed linear solver should have a similar complexity. Since there is no such optimal solver based on general algebraic methods, non-optimal (sparse) direct solvers are often employed in meshfree methods or generalized finite element methods [79]. Our goal is the development of an iterative multilevel solver with optimal complexity for the PUM; i.e. the number of iterations required to solve the linear system should be independent of the number of points N and the approximation order p, and the computational cost associated with a single iteration should be close to $O(Np^{2d})$.
Multigrid [47] and multilevel methods [83] have been developed in the late 1970s and early 1980s for the efficient solution of linear systems derived from grid-based discretizations. The fundamental observation which led to the development of multigrid methods was that classical iterative schemes like the Jacobi- or the Gauss–Seidel method reduce oscillatory error components very efficiently but their convergence behaviour breaks down for smooth errors. Such smooth errors, however, can be approximated very well on a coarser mesh. Furthermore, these formerly smooth functions (with respect to the original mesh-width) are now again more oscillatory (with respect to the coarser mesh-width). Hence, a classical iterative scheme on the coarser mesh will again start to converge very efficiently. Now, we can either apply this idea recursively or we can use a direct solver on the coarser mesh since the number of degrees of freedom is smaller than on the original mesh. Finally, we only need to correct the current iterate on the original mesh by the computed solution on the coarse mesh to obtain a better approximation to the solution of the linear system on the fine level. Hence, a multigrid method essentially consist of two operations: the application of a classical iterative

method (the so-called *smoother*) on the current mesh and the transfer of information between two successive meshes (the so-called *interlevel transfer*). Obviously, certain properties of these two components and their interplay are the key to the optimal convergence of multigrid methods [16]. The standard prerequisites and basic assumptions for a multilevel algorithm are:

1. Let $V_0, \ldots, V_J$ be a sequence of (nonnested) finite dimensional vector spaces where V_J is the finest discretization space.
2. Assume that we have a linear prolongation operator $I_{k-1}^k : V_{k-1} \to V_k$ for $k = 1, \ldots, J$.
3. Assume that we have a linear restriction operator $I_k^{k-1} : V_k \to V_{k-1}$ for $k = 1, \ldots, J$.
4. Assume that we have a symmetric positive definite bilinear form $a(\cdot, \cdot)$ on the function space V and its respective representation A_k on the discretization spaces V_k for $k = 0, \ldots, J$.
5. Assume that we have linear smoothing operators $S_k^{\text{pre}} : V_k \times V_k \to V_k$ and $S_k^{\text{post}} : V_k \times V_k \to V_k$ on the spaces V_k for $k = 1, \ldots, J$.

With these spaces and operators we can define an abstract multiplicative multilevel iteration, see Algorithm 3.5.

Algorithm 3.5. Multilevel Iteration $M_\gamma^{\nu_1,\nu_2}(k, x_k, b_k)$

1. if $k > 0$:
   (a) For $l = 1, \ldots, \nu_1$:
   Set $x_k = S_k^{\text{pre}}(x_k, b_k)$.
   (b) Set $d_{k-1} := I_k^{k-1}(b_k - A_k x_k)$.
   (c) Set $e_{k-1} := 0$.
   (d) For $i = 1, \ldots, \gamma$: $e_{k-1} = M_\gamma^{\nu_1,\nu_2}(k-1, e_{k-1}, d_{k-1})$.
   (e) Set $x_k = C_k(x_k, e_{k-1}) := x_k + I_{k-1}^k e_{k-1}$.
   (f) For $l = 1, \ldots, \nu_2$:
   Set $x_k = S_k^{\text{post}}(x_k, b_k)$.
2. else:
   (a) Set $x_k = A_k^{-1} b_k$.

The parameter γ in Algorithm 3.5 determines the recursive cycling scheme of the algorithm and thereby its overall computational complexity. The multilevel algorithm $M_\gamma^{\nu_1,\nu_2}(k, x_k, b_k)$ with $\gamma = 1$ is referred to as the V-cycle, and for a choice of $\gamma = 2$ we get the so-called W-cycle [18].
Let us now turn to the question of how we can design a multilevel solver for our partition of unity method. According to the multigrid motivation given above there are essentially three major issues we need to address: First, the question of how to construct an appropriate sequence of partition of unity space V_k^{PU}. Then, we must consider the transfer of information between two partition of unity spaces V_{k-1}^{PU} and V_k^{PU} on different scales. Finally, the selection of an appropriate smoother for our multilevel partition of unity method is the last crucial decision.

Construction of a Sequence of PUM Spaces

The hierarchical construction of a (fine level) cover $C_\Omega = C_\Omega^J$ enables us to define a sequence of covers C_Ω^k for $k = 0, \ldots, J$ with similar properties at no significant extra cost. This sequence of covers C_Ω^k can then be used to define the sequence of PUM spaces V_k^{PU} needed for our multilevel solver. To this end, we need to set appropriate patches $\omega_{i,k}$ on coarser levels $k < J$, i.e. for INNER tree nodes, and to specify the respective polynomial degrees $p_{i,k}$.

Recall that the amount of overlap $\omega_{i,k} \cap \omega_{j,k}$ of two neighbouring patches $\omega_{i,k}$ and $\omega_{j,k}$ have significant impact on the smoothness of the PU functions $\varphi_{i,k}$ and $\varphi_{j,k}$. Hence, it is not sufficient to choose coarser patches $\omega_{i,k}$ for $k < J$ which only cover their respective successor patches $\omega_{S,k+1}$; compare step 5(a)ii of Algorithm 3.4. A coarser cover patch must be larger than the union of its successor patches to control the size of the gradients of the partition of unity functions. Hence, we choose the size of a coarser cover patch to be twice as large as the size of its successor patches; i.e. we fix the ratio of $\operatorname{diam}(\omega_{i,k})$ and $\operatorname{diam}(\omega_{i,k} \cap \omega_{j,k})$ independent of the level k.

Algorithm 3.6. Multilevel Cover Construction

1. Given the domain $\Omega \subset \mathbb{R}^d$ and a bounding box $R_\Omega = \bigotimes_{i=1}^d [l_\Omega^i, u_\Omega^i] \supset \overline{\Omega}$.
2. Given the initial point set $\tilde{P} = \{x_j \,|\, x_j \in \overline{\Omega}, j = 1, \ldots, \tilde{N}\}$.
3. Build a d-binary tree over R_Ω such that per leaf L at most one $x_i \in \tilde{P}$ lies within the associated cell $\mathcal{C}_L := \bigotimes_{i=1}^d [l_L^i, u_L^i]$.
4. Set J to the finest refinement level of the tree.
5. Set $P_k = \emptyset$, $C_\Omega^k = \emptyset$ for $k = 0, \ldots, J$.
6. For the root cell $\mathcal{C}_L = \bigotimes_{i=1}^d [l_L^i, u_L^i] = R_\Omega$:
   (a) If current tree cell $\mathcal{C}_L$ is an INNER tree node and $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Descend tree for all successors $\mathcal{C}_S$ of $\mathcal{C}_L$. ($\rightarrow$ 6(a))
      ii. Set patch $\omega_L = \bigotimes_{i=1}^d [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$ where $x_L = \frac{1}{2^d} \sum x_S$ is the centre of its successors points x_S and $h_L^i = 2 \max h_S^i$ is twice the maximum radius of its successors h_S^i.
      iii. Set active levels $l_L^{\min} = l_L^{\max} = \min l_S^{\min} - 1$ and update for all successors $l_S^{\min} = \min l_S^{\min}$.
      iv. Set polynomial degree $p_L := \min p_S$ to minimal degree of its successors.
   (b) Else if $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Set patch $\omega_L = \bigotimes_{i=1}^d [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$ where $h_L^i = \frac{\alpha_t}{2}(u_L^i - l_L^i)$, $x_L^i = l_L^i + \frac{1}{2}(u_L^i - l_L^i)$, and $\alpha_t > 1$.
      ii. Set active levels $l_L^{\min} = l_L^{\max} = J$.
      iii. Set polynomial degree to some given value p_L.
      iv. Set $P_J = P_J \cup \{x_L\}$, $C_\Omega^J = C_\Omega^J \cup \{\omega_L\}$.
7. For $k = 0, \ldots, J-1$:
   (a) Set $P_k = \{x_L \,|\, l_L^{\min} \leq k \leq l_L^{\max}\}$.
   (b) Set $C_\Omega^k = \{\omega_L \,|\, l_L^{\min} \leq k \leq l_L^{\max}\}$.

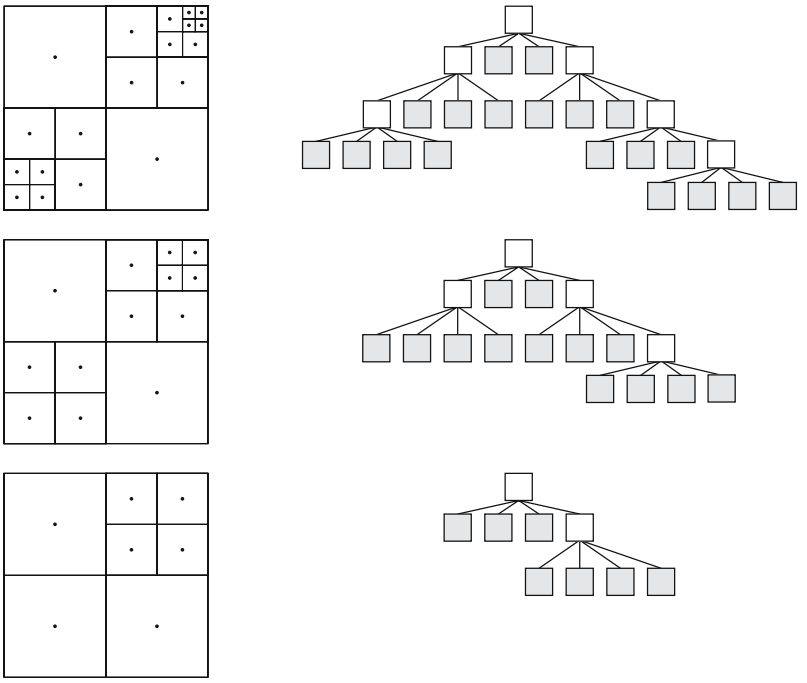


Figure 3.12. Multilevel cover construction with Algorithm 3.6 in two dimensions. The cell decompositions and its respective tree representation (upper right, white: INNER tree nodes, gray shaded: LEAF tree nodes) for the fine level point set $P_J = P_4$ (upper row), and two coarser level point sets P_3 (centre row) and P_2 (lower row). The leaves of the tree correspond to the points $x_L \in P_k$.

Note that the active (discretization) levels k with $l_L^{\min} \leq k \leq l_L^{\max}$ of a particular patch ω_L can be completely unrelated to the respective tree refinement level, see steps 6(a)iii and 6(b)ii. With this algorithm[19] we define a coarser cover C_Ω^{k-1} to a cover C_Ω^k by collapsing those leaves of the tree into its parent tree node whose siblings are also leaves (with respect to the current level k), see Figure 3.12. Note however that the corresponding coarser patch $\omega_{j,k-1}$ is not the agglomerate of its successor patches $\omega_{i,k}$. A coarser patch needs to be slightly larger than that to control the amount of overlap on coarser levels, i.e. to control the gradients of coarser partition of unity functions $\varphi_{j,k-1}$. Furthermore, the described cell agglomeration principle does not translate (in general) to a nested sequence of function spaces V_k^{PU} due to the Shepard construction (2.1) for the partition of unity. Each PUM space V_k^{PU} with $k = 0, \ldots, J$ is defined according to the single level construction presented in

[19] We can also construct a sequence of more general covers (Algorithm 3.1) by changing step 6(b)i accordingly. The definition of coarser patches in step 6(a)ii is not affected by such a change.

Section 2.1; i.e. starting from the respective cover C_Ω^k we set up the Shepard partition of unity $\{\varphi_{i,k}\}$ via (2.1) and define the global PUM space

$$V_k^{\mathrm{PU}} := \sum \varphi_{i,k} \operatorname{span}\langle\{\psi_{i,k}^{p_{i,k}}\}\rangle = \operatorname{span}\langle\{\varphi_{i,k}\psi_{i,k}^{p_{i,k}}\}\rangle.$$

Note that a geometric patch ω_L may be resident on several (discretization) levels k, e.g. $\omega_L = \omega_{i,k} = \omega_{j,k-1}$ so that $\omega_L \in C_\Omega^k$ and $\omega_L \in C_\Omega^{k-1}$, see Figure 3.12. Nevertheless, the corresponding shape functions on level k may differ from those on level $k-1$. Since the geometric neighbourhoods $C_{i,k}$ and $C_{j,k-1}$ and the weight functions of the respective neighbours on different levels can change, the corresponding partition of unity function may change, i.e. $\varphi_{i,k} \neq \varphi_{j,k-1}$. Hence, the shape functions $\varphi_{i,k}\psi_{i,k}^n$ associated with $\omega_{i,k} = \omega_L$ on level k are different from those $\varphi_{j,k-1}\psi_{j,k-1}^n$ on level $k-1$, even if the local approximation space $V_{i,k}^{p_{i,k}} = V_{j,k-1}^{p_{j,k-1}} = V_L^{p_L}$ on the cover patch $\omega_{i,k} = \omega_{j,k-1} = \omega_L$ is not changed between levels k and $k-1$. Therefore, the PUM spaces V_k^{PU} and V_{k-1}^{PU} on two successive levels k and $k-1$ are in general nonnested, i.e. $V_k^{\mathrm{PU}} \not\supset V_{k-1}^{\mathrm{PU}}$.

Remark 3.13.
Note that the tree construction of the cover patches $\omega_{i,k}$ leads to a sequence of covers C_Ω^k where for each cover patch $\omega_{i,k} \in C_\Omega^k$ on level k we have exactly one cover patch $\omega_{\tilde{i},k-1} \in C_\Omega^{k-1}$ such that $\omega_{i,k} \subseteq \omega_{\tilde{i},k-1}$. Every cover patch ω_L corresponds to a tree-cell C_L and vice versa. Either a fine cover patch $\omega_{i,k}$ is also element of the coarse cover C_Ω^{k-1}, then we have $\omega_{i,k} = \omega_{\tilde{i},k-1}$, or the cover patch $\omega_{\tilde{i},k-1}$ which corresponds to the parent tree-cell of $\omega_{i,k}$ is element of C_Ω^{k-1} and is the only coarse patch $\omega_{l,k-1}$ that fulfills $\omega_{l,k-1} \supseteq \omega_{i,k}$; i.e. in this case $\omega_{j,k-1} \supset \omega_{i,k}$ holds, see Figure 3.12. ∎

Remark 3.14.
The neighbourhoods $C_{i,k} := \{\omega_{j,k} \in C_\Omega^k \,|\, \omega_{i,k} \cap \omega_{j,k} \neq \emptyset\}$ on all levels k can be computed with Algorithm 3.2 if we extend step 2(a) to include a test of the active level $l_L^{\min} \leq k \leq l_L^{\max}$. ∎

Now that we have a sequence of PUM function spaces V_k^{PU} which are in general nonnested, i.e.

$$V_0^{\mathrm{PU}} \not\subset V_1^{\mathrm{PU}} \not\subset V_2^{\mathrm{PU}} \not\subset \cdots \not\subset V_J^{\mathrm{PU}},$$

the next step in the development of a multilevel solver is the design of appropriate interlevel transfer operators

$$I_{k-1}^k : V_{k-1}^{\mathrm{PU}} \to V_k^{\mathrm{PU}} \quad \text{and} \quad I_k^{k-1} : V_k^{\mathrm{PU}} \to V_{k-1}^{\mathrm{PU}}.$$

Interlevel Transfer
In addition to the fact that coarser shape functions $\varphi_{i,k-1}\psi_{i,k-1}^n$ cannot be represented exactly on finer levels, i.e.

$$\varphi_{i,k-1}\psi_{i,k-1}^n \neq \sum \beta_{j,k}^m \varphi_{j,k}\psi_{j,k}^m,$$

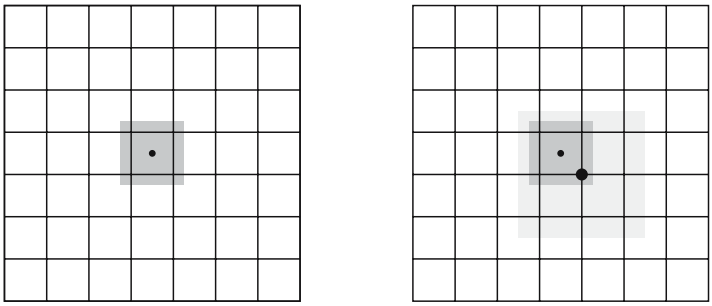


Figure 3.13. Uniform cells on level l and support of a single shape function (dark gray shaded) on level l which overlaps 3^d cells (left). The support of a coarser shape function (gray shaded) on level $l-1$ overlaps 4^d cells on level l (right).

due to the nonnestedness of the spaces V_k^{PU}, we also have to deal with non-interpolatory shape functions $\varphi_{i,k-1}\psi^n_{i,k-1}$ and $\varphi_{j,k}\psi^m_{j,k}$. Therefore, the two classical approaches to the interlevel transfer problem, natural injection and interpolation, are not available for our multilevel partition of unity method. One approach toward the construction of the prolongation operators I^k_{k-1} for our nonnested spaces is the use of L^2-projections $\Pi^k_{k-1} : V^{\mathrm{PU}}_{k-1} \to V^{\mathrm{PU}}_k$ as prolongations I^k_{k-1} from V^{PU}_{k-1} onto V^{PU}_k which are given by

$$\Pi^k_{k-1} = (M^k_k)^{-1}(M^k_{k-1})$$

where the storage requirement of Π^k_{k-1} is given by the sparsity patterns of the mass matrix M^k_k on level k and the interlevel mass matrix M^k_{k-1}. Therefore, we need to be concerned with the number of interlevel neighbours $\mathrm{card}(C_{j,k-1,k})$, where

$$C_{j,k-1,k} := \{\omega_{i,k} \in C^k_\Omega \,|\, \omega_{i,k} \cap \omega_{j,k-1} \neq \emptyset\}.$$

Let us assume for now that the tree in our cover construction is fully saturated; i.e. all tree cells correspond to grid cells of a uniformly refined grid with mesh-width $h_k = 2^{-k}$, i.e. $\omega_{i,k} = \bigotimes_{l=1}^d (x^l_{i,k} - \alpha\frac{h_k}{2}, x^l_{i,k} + \alpha\frac{h_k}{2})$. Then it is easy to see that the number of interlevel neighbours $\mathrm{card}(C_{j,k-1,k}) = 4^d$ is even larger than the number of (intralevel) neighbours $\mathrm{card}(C_{i,k}) = 3^d$, see Figure 3.13. Hence, it becomes clear that the global projection Π^k_{k-1} suffers from three major drawbacks:

1. The mass matrix M^k_k has to be inverted. Although the global basis $\{\varphi_{j,k}\psi^m_{j,k}\}$ is stable with respect to the number of cover patches $\mathrm{card}(C^k_\Omega)$, see Section 2.1, the condition number κ_k of M^k_k is dependent on the local approximation orders $p_{j,k}$.
2. The sparsity pattern of the mass matrix M^k_k is identical to that of the operator matrix A_k and therefore the storage requirement per level k is doubled.

3. The sparsity pattern of the interlevel mass matrix M_{k-1}^k is given by the geometric neighbour relations $\omega_{i,k} \cap \omega_{j,k-1} \neq \emptyset$. Due to the overlap of the cover patches the number of interlevel neighbours is rather large which further increases the storage requirement per level.

These issues make the use of the global L^2-projection Π_{k-1}^k too expensive in practice. We need to find a way to avoid the inversion of the mass matrix M_k^k and we also have to reduce the overall storage demand associated with the interlevel transfer.
Within the PUM context we can construct a very cheap prolongation operator based on a localized L^2-projection approach, see [41, 76] for details. The localization of the L^2-projection Π_{k-1}^k consists of two steps. At first consider the basic PUM error estimate [8, 9]

$$\|v - v^{\mathrm{PU}}\|^2_{L^2(\Omega)} \leq C \sum_i \|v - v_i\|^2_{L^2(\omega_i \cap \Omega)}, \tag{3.2}$$

where $v^{\mathrm{PU}} := \sum_i \varphi_i \sum_n u_i^n \psi_i^n$ and $v_i := \sum_n u_i^n \psi_i^n$. From (3.2) we know that it is sufficient to control the local errors $\|v - v_i\|_{L^2(\omega_i \cap \Omega)}$ on each cover patch ω_i. Now choose $v = u_{k-1}^{\mathrm{PU}} = \sum_j \varphi_{j,k-1} u_{j,k-1} = \sum_j \varphi_{j,k-1} \sum_m u_{j,k-1}^m \psi_{j,k-1}^m$ and $v^{\mathrm{PU}} = I_{k-1}^k u_{k-1}^{\mathrm{PU}} = \sum_i \varphi_{i,k} u_{i,k} = \sum_i \varphi_{i,k} \sum_n u_{i,k}^n \psi_{i,k}^n$ so that (3.2) reads

$$\|u_{k-1}^{\mathrm{PU}} - I_{k-1}^k u_{k-1}^{\mathrm{PU}}\|^2_{L^2(\Omega)} \leq C \sum_i \|u_{k-1}^{\mathrm{PU}} - u_{i,k}\|^2_{L^2(\omega_{i,k} \cap \Omega)}. \tag{3.3}$$

Hence, we observe that we can approximate the global coarse function u_{k-1}^{PU} locally on the fine cover patches $\omega_{i,k}$ using the local basis functions $\psi_{i,k}^n$, rather than approximating u_{k-1}^{PU} by the global shape functions $\varphi_{i,k}\psi_{i,k}^n$ on the finer level k. Now in a second step we establish an upper bound for each of the terms on the right-hand side of (3.3) utilizing the geometric hierarchy of our tree. Due to our tree-based cover construction we can find exactly one coarse patch $\omega_{\tilde{i},k-1}$ for every fine patch $\omega_{i,k}$ such that $\omega_{i,k} \subset \omega_{\tilde{i},k-1}$ holds, see Remark 3.13. Hence, we can introduce the respective coarse local function $u_{\tilde{i},k-1}$ associated with the unique coarse patch $\omega_{\tilde{i},k-1}$ into each term $\|u_{k-1}^{\mathrm{PU}} - u_{i,k}\|_{L^2(\omega_{i,k} \cap \Omega)}$ of (3.3) so that we obtain the estimate

$$\begin{aligned} \|u_{k-1}^{\mathrm{PU}} - u_{i,k}\|_{L^2(\omega_{i,k} \cap \Omega)} \leq{}& \|u_{k-1}^{\mathrm{PU}} - u_{\tilde{i},k-1}\|_{L^2(\omega_{i,k} \cap \Omega)} \\ &+ \|u_{\tilde{i},k-1} - u_{i,k}\|_{L^2(\omega_{i,k} \cap \Omega)} \end{aligned} \tag{3.4}$$

by the triangle inequality. This estimate allows us to approximate each coarse local function $u_{\tilde{i},k-1}$, independent of all other local components $u_{j,k-1}$ of u_{k-1}^{PU}, on the respective fine cover patch $\omega_{i,k}$ with $\omega_{i,k} \subset \omega_{\tilde{i},k-1}$ since the first term of (3.4) is small by definition of u_{k-1}^{PU}. Hence, we can set up our prolongation operators I_{k-1}^k via the so-called local-to-local L^2-projection. To this end, we project each local approximation $u_{i,k-1}$ on level $k-1$ independently

to the finer level k using the hierarchical condition $\omega_{i,k} \subseteq \omega_{\tilde{i},k-1}$ instead of the geometric neighbour relation $\omega_{i,k} \cap \omega_{j,k-1} \neq \emptyset$ only. The respective matrix representation of this prolongation is given by

$$\begin{aligned} I_{k-1}^k := \widetilde{\Pi}_{k-1}^k &:= (\widetilde{M}_k^k)^{-1}(\widetilde{M}_{k-1}^k) && \text{with} \\ (\widetilde{M}_k^k)_{(i,n),(i,m)} &:= \langle \psi_{i,k}^m, \psi_{i,k}^n \rangle_{L^2(\omega_{i,k}\cap\Omega)} && \text{and} \\ (\widetilde{M}_{k-1}^k)_{(i,n),(\tilde{i},m)} &:= \langle \psi_{\tilde{i},k-1}^m, \psi_{i,k}^n \rangle_{L^2(\omega_{i,k}\cap\Omega)}. \end{aligned}$$

The storage requirement of $I_{k-1}^k = \widetilde{\Pi}_{k-1}^k$ is minimal. We only need to store a single block-entry $(\widetilde{M}_k^k)_{i,i}^{-1}(\widetilde{M}_{k-1}^k)_{i,\tilde{i}}$ for each patch $\omega_{i,k}$ on level k since $I_{k-1}^k = \widetilde{\Pi}_{k-1}^k$ and $I_k^{k-1} = (I_{k-1}^k)^T$ involve the hierarchical neighbours

$$\begin{aligned} C_{j,k-1,k}^H &:= \{\omega_{i,k} \in C_\Omega^k \,|\, \omega_{i,k} \subseteq \omega_{j,k-1}\}, \\ C_{i,k,k-1}^H &:= \{\omega_{j,k-1} \in C_\Omega^{k-1} \,|\, \omega_{i,k} \subseteq \omega_{j,k-1}\} \end{aligned}$$

rather than all neighbours $C_{i,k-1,k}$. Moreover, the respective integrals only involve the local basis functions $\psi_{\tilde{i},k-1}^m$ and $\psi_{i,k}^n$ and can be computed very efficiently. Overall, the projection operator $\widetilde{\Pi}_{k-1}^k$ can be computed with $O(Np_{\max}^{3d})$ operations in general (and with $O(Np_{\max}^d)$ if we use orthogonal polynomials locally) where $p_{\max} = \max_i p_{i,k}$. Furthermore, it is exact for polynomials of degree $p_{\min} = \min_i p_{i,k}$ and therefore suitable also for higher order approximations. Note that the construction is symmetric so that the prolongation $I_{k-1}^k = \widetilde{\Pi}_{k-1}^k$ as well as the restriction operators $I_k^{k-1} = (I_{k-1}^k)^T$ are well-suited for multilevel schemes for general PUM spaces with varying local basis functions [76].

Smoothing Operators

The remaining ingredients for our multilevel solver, Algorithm 3.5, are the smoothers S_k^{pre} and S_k^{post}. Recall that a smoother should damp highly oscillatory error components, so that the smoothed error can be well-approximated on a coarser level. Note that we only coarsen the h-components of our PUM approximation space, i.e. the partition of unity functions, independent of the polynomial degree p. Hence, the quality of our multilevel solver with respect to the approximation order p is essentially determined by the quality of the smoother; i.e. we need to employ a p-robust smoother to obtain a p-robust multilevel solver.

Smoothers usually are classical iterative schemes like the Jacobi- or Gauss–Seidel iteration. These classical smoothing schemes as well as overlapping domain decomposition methods and even multigrid methods can be interpreted in the framework of subspace correction methods (SCM) [17,23,49,70,83,84]. Hence, let us shortly review the abstract setting of an SCM.

The general idea is as follows: First, we write the discretization space $\mathcal{V} = \sum_{j=1}^{\mathcal{N}} \mathcal{V}_j$ as the sum[20] of subspaces $\mathcal{V}_j$ with maps $P_j : \mathcal{V}_j \to \mathcal{V}$.[21] Then, we

[20] Note that we do not assume that the splitting is a direct sum.

[21] It is sufficient to require $\mathcal{V} = \sum_j P_j \mathcal{V}_j$, i.e. the condition $\mathcal{V}_j \subset \mathcal{V}$ is not necessary.

choose symmetric positive definite bilinear forms $b_j(\cdot,\cdot)$ on each $\mathcal{V}_j$ represented by operators B_j such that solutions to the systems of linear equations $B_j u_j = f_j$ on $\mathcal{V}_j$ are easily computable, and B_j^{-1} can be considered as an approximate inverse to the restriction of A to $\mathcal{V}_j$. Finally, we combine these local approximate inverses B_j^{-1} appropriately to define a global approximate inverse to A on the discretization space $\mathcal{V}$. There are essentially two approaches to the definition of an approximate inverse of A by the B_j^{-1}, the additive approach and the multiplicative approach.
In the so-called parallel subspace correction (PSC) or additive Schwarz method we set up an iterative solution process via the operator

$$M_{PSC} := \mathbb{I} - \omega \sum_{j=1}^{\mathcal{N}} P_j T_j = \mathbb{I} - \omega \Big(\sum_{j=1}^{\mathcal{N}} P_j B_j^{-1} R_j \Big) A, \tag{3.5}$$

where ω is a relaxation parameter and the operators involved are defined by

$$a(u,v) = \langle Au, v \rangle_{\mathcal{V}}, \qquad b_j(u_j, v_j) = \langle B_j u_j, v_j \rangle_{\mathcal{V}_j},$$
$$\langle R_j u, v_j \rangle_{\mathcal{V}} = \langle u, P_j v_j \rangle_{\mathcal{V}}, \; b_j(T_j u, v_j) = a(u, P_j v_j).$$

The iteration operator of the successive subspace correction (SSC) or multiplicative Schwarz method is given by

$$M_{SSC} := \prod_{j=1}^{\mathcal{N}} \big(\mathbb{I} - P_j T_j \big) = \prod_{j=1}^{\mathcal{N}} \big(\mathbb{I} - P_j B_j^{-1} R_j A \big). \tag{3.6}$$

Note that the PSC operator (3.5) can also be interpreted as a preconditioned Richardson iteration where the preconditioner is given by

$$\mathcal{C}_{PSC} := \sum_{j=1}^{\mathcal{N}} P_j B_j^{-1} R_j. \tag{3.7}$$

Let us now restrict ourselves to the case of $B_j := A|_{\mathcal{V}_j}$ which means that we only consider exact subspace solvers. Then, we have two degrees of freedom in the design of our smoothing scheme: The splitting of the discretization space and the type of the iteration, namely the additive scheme (3.5) or the multiplicative scheme (3.6). For instance the classical Jacobi- or the Gauss–Seidel iteration is based on a splitting into one-dimensional subspaces; i.e. each single shape function $\varphi_{i,k}\psi_{i,k}^n$ defines a particular subspace. Such a splitting in to one-dimensional subspaces, however, is not very natural for our PUM due to the specific product structure of the shape functions. The subspace splitting should respect the product structure of the PUM shape functions $\varphi_{i,k}\psi_{i,k}^n \in V_k^{\mathrm{PU}}$. For the ease of notation we assume $p_{i,k} = p$ and omit the level index k in the following.

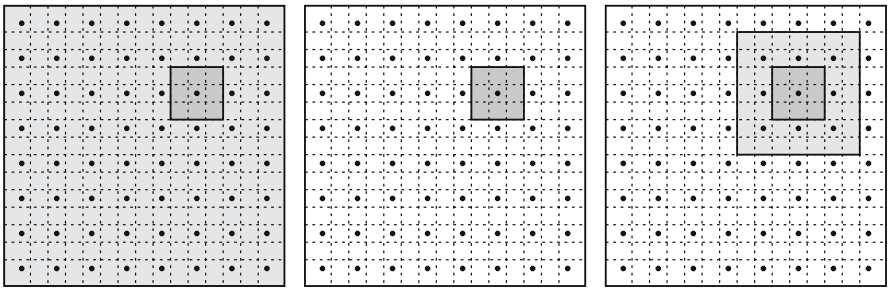


Figure 3.14. Sub-domains (light gray shaded) associated with the subspaces $\widehat{\mathcal{V}}_n$ (left), $\mathcal{V}_i$ from (3.8) (centre), and $\widetilde{\mathcal{V}}_l$ from (3.9) (right) and the support of the shape functions $\varphi_i\psi_i^n$ (dark gray shaded) based on a cover with $\alpha = 1.5$.

The product structure of the shape functions $\varphi_i\psi_i^n$ implies two natural subspace definitions. For instance, we can define the subspaces

$$\widehat{\mathcal{V}}_n := \operatorname{span}_i\langle\varphi_i\psi_i^n\rangle := \{v \in V^{\mathrm{PU}} \,|\, v = \sum_i \varphi_i v_i^n \psi_i^n\}.$$

These subspaces, however, contain functions with *global* support on the domain Ω, see Figure 3.14 (left), and the dimension of each subspace is of the order $O(N)$. Therefore, a direct solution of $A|_{\widehat{\mathcal{V}}_n}$ is not feasible. We would need to resort to fast iterative solution techniques for these subspace problems. Furthermore, we are interested in smoothing schemes S_k for Algorithm 3.5 based on our multilevel cover sequence C_Ω^k. Hence, there is no additional benefit from the fact that the solutions to $A|_{\widehat{\mathcal{V}}_n}$ contain global information and the computational cost associated with the solution of the subspace problems make this splitting unsuitable for our construction. A more appropriate subspace definition is given by

$$\mathcal{V}_i := \varphi_i V_i^p = \operatorname{span}_n\langle\varphi_i\psi_i^n\rangle := \{v \in V^{\mathrm{PU}} \,|\, v = \sum_n \varphi_i v_i^n \psi_i^n\}. \tag{3.8}$$

These spaces contain functions only with local supports, see Figure 3.14 (centre). Furthermore, the dimension of the subspace $\mathcal{V}_i$ is given by the dimension $O(p^d)$ of the local approximation spaces V_i^p. Hence, we can compute the inverse $(A|_{\mathcal{V}_i})^{-1}$ of each of the subspace problems with acceptable complexity of $O(p^{3d})$; i.e. one iteration of a PSC or SSC iteration based on this splitting is of the order $O(Np^{3d})$.

Note that both subspace definitions lead to a *direct* splitting of our PUM function space $V^{\mathrm{PU}} = \sum_i \mathcal{V}_i = \sum_n \widehat{\mathcal{V}}_n$; i.e. every basis function $\varphi_i\psi_i^n$ is contained in exactly one subspace. In terms of the index pairs (i, n) we have a disjoint decomposition of the index set $\{(i, n)\}$ which induces a specific partitioning of the PUM stiffness matrix $A = (A_{(i,n),(j,m)})$. A PSC iteration (3.5) based on the direct splitting $V^{\mathrm{PU}} = \sum_i \mathcal{V}_i$ corresponds to the classical block-Jacobi iteration and the SSC iteration (3.6) corresponds to the block-Gauss–Seidel iteration (BGS) where we only have a small overlap between the

supports of functions from different subspaces, see Figure 3.14 (centre). Even though we consider a direct splitting and employ an exact solver $(A|_{\mathcal{V}_i})^{-1}$ within a specific subspace $\mathcal{V}_i$ there are still couplings between the subspaces due to the overlap of the supports via the global problem A. The quality of the PSC and SSC iterations is obviously determined by the strength of these couplings. The two parameters within our PUM which can influence the strength of the couplings between two different subspaces $\mathcal{V}_i$ and $\mathcal{V}_j$, and hence the quality of the iterations, are the overlap parameter α used in our cover construction and the polynomial degree p. Therefore, the BGS smoothing scheme will not be robust with respect to p; i.e. the quality of the smoother will depend on p, see Figure 3.15.
One approach to overcome this p-dependence is to consider subspace splittings $V^{\mathrm{PU}} = \sum_l \widetilde{\mathcal{V}}_l$ which are no longer direct splittings, i.e. a basis function $\varphi_i \psi_i^n$ may belong to several subspaces $\widetilde{\mathcal{V}}_l$. Consider the subspace definition

$$\widetilde{\mathcal{V}}_l := \sum_{\omega_i \cap \omega_l \neq \emptyset} \mathcal{V}_i = \mathrm{span}_{(i,n), i \in C_l} \langle \varphi_i \psi_i^n \rangle \tag{3.9}$$

where $C_l := \{i \,|\, \omega_i \cap \omega_l \neq \emptyset\}$ denotes the neighbourhood of the cover patch ω_l, see Figure 3.14 (right). The subspace $\widetilde{\mathcal{V}}_l$ contains *all* functions $\varphi_i \psi_i^n$ whose support ω_i has a non-vanishing intersection with the patch ω_l. Hence, when we solve the subspace problem $A|_{\widetilde{\mathcal{V}}_l}$ we resolve all couplings involving the basis functions $\varphi_l \psi_l^q$. Therefore, for each patch ω_l there is one subspace problem $A|_{\widetilde{\mathcal{V}}_l}$ which resolves *all* couplings involving the associated basis functions $\varphi_l \psi_l^q$ independent of the overlap parameter α and the polynomial degree p.
Since the subspace splitting into $\widetilde{\mathcal{V}}_l$ is not a direct splitting it does not correspond to a simple partitioning scheme of the stiffness matrix A. Here, we have to assemble the (discrete) local subproblems $A_{l,l}$ from the global linear system via the Galerkin products $A_{l,l} := P_l^T A P_l$ where P_l denotes the discrete extension operator which embeds the subspace $\widetilde{\mathcal{V}}_l$ in the global PUM space V^{PU}. In our case P_l is just a mask matrix, i.e. a reduced identity matrix. With the matrices A and P_l the application of the SSC iteration operator (3.6) to a linear system $A\tilde{u} = \hat{f}$ can be realized by Algorithm 3.7. In the following this iteration referred to as a multiplicative overlapping Schwarz (MOS) smoother.

Algorithm 3.7. Successive subspace correction method
For all $l = 1, \ldots, N$:

1. Compute local residual $\hat{f}_l := P_l^T(\hat{f} - A\tilde{u})$.
2. Solve subspace problem $(P_l^T A\ P_l)\tilde{u}_l = A_{l,l}\tilde{u}_l = \hat{f}_l$.
3. Update global iterate $\tilde{u} = \tilde{u} + P_l \tilde{u}_l$.

In Figure 3.15 we give the smoothing results obtained after one iteration of the BGS and the MOS smoother for $p = 1$ and $p = 5$. From these surface plots we can clearly observe that the MOS smoother gives much smoother

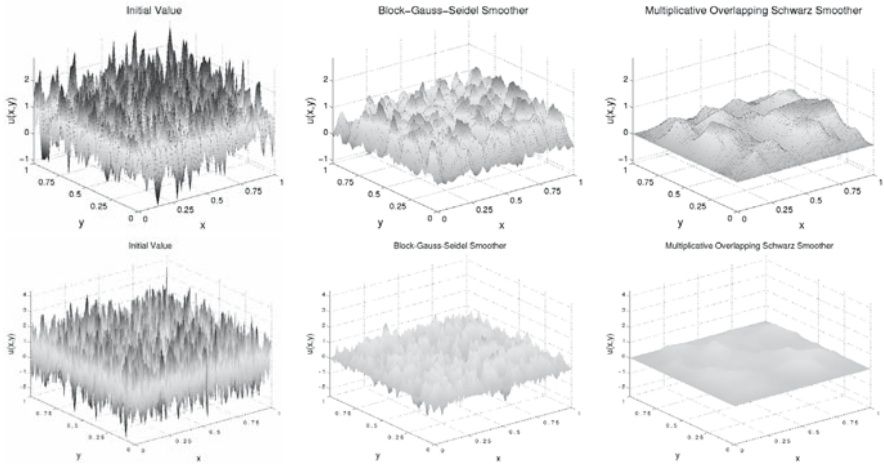


Figure 3.15. Random valued initial guess (left) and smoothing results using a block-Gauss–Seidel (centre) and a multiplicative overlapping Schwarz smoother (right). Depicted are the current iterates after a single application of the smoother. The discretization was based on a uniform node arrangement on level 5 and employed polynomials of degree $p = 1$ (upper row) and $p = 5$ (lower row).

iterates than the *BGS* smoother. More notably, the quality of the *BGS* smoother deteriorates for higher order approximations. The results for $p = 5$ are not as smooth as for $p = 1$. For the *MOS* smoother we find a completely different behaviour. There is no deterioration in the quality for larger p. In fact it even seems that the results for $p = 5$ are better than for $p = 1$.

Computational Complexity

To estimate the computational cost associated with the multilevel iteration operator $M_\gamma^{(\nu_1,\nu_2)}$, see Algorithm 3.5, we need to consider the assembly of the prolongation I_{k-1}^k and restriction $I_k^{k-1} = (I_{k-1}^k)^T$ operators, and the setup of the smoothing schemes S_k^{pre} and S_k^{post} on all levels.

The local-to-local projection $I_{k-1}^k = \widetilde{\Pi}_{k-1}^k$ on a particular level k can in general be computed with $O(N_k p^3 d_k)$ operations where $N_k = \text{card}(C_\Omega^k)$ and $p_k = \max_i p_{i,k}$. Similarly, the *BGS* and the *MOS* smoothers require $O(N_k p_k^{3d})$ operations due to the computation of $A_{i,i}^{-1}$ for all $i = 1, \ldots, N_k$. The application of the prolongation operator $I_{k-1}^k = \widetilde{\Pi}_{k-1}^k$, the restriction operator $I_k^{k-1} = (I_{k-1}^k)^T$, and the smoothers $S_k^{\text{pre}} = S_k^{\text{post}}$ requires only order $O(N_k p_k^{2d})$ operations.

Hence, one iteration of our multilevel solver is of optimal complexity with respect to the number of patches $N_J = \text{card}(C_\Omega^J)$ if the series

$$\sum_{k=0}^{J} \gamma^k \frac{N_{J-k}}{N_J} < \infty \quad \text{for} \quad J \to \infty \tag{3.10}$$

converges. If the number of patches $N_k = \mathrm{card}(C_\Omega^k)$ is reduced at a constant rate from level to level, (3.10) holds and our multilevel iteration $M_\gamma^{(\nu_1,\nu_2)}$ can be applied with $O(N_J)$ operations.

With respect to the polynomial degree $p = \max_{i,k} p_{i,k}$, the optimal complexity is $O(p^{2d})$ since the local matrix blocks $A_{i,j}$ are generally dense. Yet, due to the direct solves for the diagonal blocks $A_{i,i}$ one iteration of our multilevel solver $M_\gamma^{(\nu_1,\nu_2)}$ is optimal up to a factor of $O(p^d)$ only. Therefore, we obtain the solution of a linear system $A\tilde{u} = \hat{f}$ up to a prescribed relative accuracy ϵ with $O(\ln(1/\epsilon)N_J p^{3d})$ operations if the rate of convergence of the solver is independent of the number of patches N_J and independent of the polynomial degree p. The results presented in [41, 76] show that the $V(1,1)$-cycle, i.e. the $M_1^{(1,1)}$ iteration, converges with a rate which is independent of N_J for the BGS smoother. However, the rate is not independent of the polynomial degree p. On the other hand the $V(1,1)$-cycle with the MOS smoother converges with a rate which is independent of the number of patches N_J and the polynomial degree p [38]. However, the MOS smoother is much more expensive since it involves a rather large constant so that for a practical range of N_J and p the multigrid solver Algorithm 3.5 with the BGS smoother may give a faster solver with respect to actual computational time.

4 Parallelization

With the algorithms given in the previous section we can carry out simulations with several hundred thousand degrees of freedom efficiently on a single processor. However, the storage limitations of a single processor machine in general render a simulation with millions of degrees of freedom not feasible. For very large simulations we must resort to distributed memory parallel computers. Hence, we need to parallelize the algorithms given above to be able to deal with large scale problems.

Our parallelization follows the data decomposition approach. Here, the main ingredients are a parallel key-based tree implementation and a space filling curve load balancing scheme. The overall method can be split into three major steps: The initial tree construction and load balancing step, the assembly step where we set up the stiffness matrices A_k on all levels $k = 0, \ldots, J$ and the interlevel transfers I_k^{k-1} and I_{k-1}^k, and finally the solution step where we use a multiplicative multilevel iteration to solve the linear system $A_J\tilde{u}_J = \hat{f}_J$. The load balancing step as well as the assembly step require some information about the neighbouring patches. The neighbour search in parallel computations is the most challenging task since we need to determine the communication pattern and have to exchange the appropriate data between the processors. This is further complicated by our multilevel construction and the necessary increase in the support sizes on coarser levels, see Section 4.4.

4.1 Parallel Data Decomposition

In general there are two main tasks associated with the efficient parallelization of any numerical computation on distributed memory computers. The first is to split up the data evenly among the participating processors; i.e. the associated computational work should be well-balanced. The second is to allow for an efficient access to data stored by another processor; i.e. on distributed memory parallel computers also the amount of remote data needed by a processor should be small.
In a data decomposition approach we partition the data, e.g. the computational domain or mesh, among the participating processors [72]. Then, we simply restrict the operations of the global numerical method to the assigned part of the data/domain. A processor has read and write access to its local data but only read access to remote data it may need to complete its local computation. On distributed memory machines these required data have to be exchanged explicitly in distinct communication steps.
The quality of the partition of the domain/data essentially determines the efficiency of the resulting parallel computation. The local parts of the data assigned to each processor should induce a similar amount of computational work so that each processor needs roughly the same time to complete its local computation. Here, a processor may need to access the data of the neighbouring sub-domains to solve its local problem. Hence, the geometry of the sub-domains should be simple to limit the number of communication steps and the communication volume. The number of neighbouring processors (which determines the number of communication steps) should be small and the geometry of the local boundary (which strongly influences the communication volume) should be simple, i.e. its size should be small.

Key Based Tree Implementation
In a classical tree implementation the topology of the tree is explicitly encoded via pointers from a tree node to its successors. Such a pointer based implementation, however, is not easily parallelized especially on distributed memory machines. Hence, we use a different implementation of a d-binary tree [76, 81, 82]. Here, the tree is realized with the help of a hashed associative container. To this end, a unique label is assigned to each possible tree cell and instead of linking a cell directly to its successor cells, the labelling scheme implicitly defines the topology of the tree and allows for the easy access to successors and ancestors of a particular tree cell. Furthermore, we can randomly access any cell of the tree via its unique label. This allows us to catch accesses to non-local data in parallel computations and we can easily compute the communication pattern and send and receive all necessary data to complete the local computation.
The labelling scheme must encode the topology of the tree. To this end, the labelling scheme maps tree cells $\mathcal{C}_L = \bigotimes_{i=1}^{d} [c_L^i, c_L^i + h_L^i] \subset \mathbb{R}^d$ to a single integer value $\mathsf{k}_L \in \mathbb{N}_0$, the *key*. For instance, we can use the *d-binary path* as the key value k_L associated with a tree cell $\mathcal{C}_L$. The d-binary path k_L is

successor cell	binary key value	integer key value
$[c_L^1, c_L^1 + \frac{1}{2}h_L^1] \times [c_L^2, c_L^2 + \frac{1}{2}h_L^2]$	$\mathsf{k}_L 00$	$4\mathsf{k}_L$
$[c_L^1, c_L^1 + \frac{1}{2}h_L^1] \times [c_L^2 + \frac{1}{2}h_L^2, c_L^2 + h_L^2]$	$\mathsf{k}_L 01$	$4\mathsf{k}_L + 1$
$[c_L^1 + \frac{1}{2}h_L^1, c_L^1 + h_L^1] \times [c_L^2, c_L^2 + \frac{1}{2}h_L^2]$	$\mathsf{k}_L 10$	$4\mathsf{k}_L + 2$
$[c_L^1 + \frac{1}{2}h_L^1, c_L^1 + h_L^1] \times [c_L^2 + \frac{1}{2}h_L^2, c_L^2 + h_L^2]$	$\mathsf{k}_L 11$	$4\mathsf{k}_L + 3$

Table 4.1. Path key values for the successor cells of a tree cell $\mathcal{C}_L = \bigotimes_{i=1}^{d} [c_L^i, c_L^i + h_L^i]$ with associated key k_L in two dimensions.

defined by the search path that has to be completed to find the respective cell in the tree. Starting at the root of the tree, we set $\mathsf{k}_L = 1$ and descend the tree in the direction of the cell $\mathcal{C}_L$. Here we concatenate the current key value k_L (in binary representation) and the d Boolean values 0 and 1 associated with the decisions to which successor cell the descent continues to reach the respective tree cell $\mathcal{C}_L$. In Table 4.1 we give the resulting path key values k_L for a two dimensional example. Note that the key value $\mathsf{k}_L = 1$ for the root cell is essentially a stop bit which is necessary to ensure the uniqueness of the key values.

Parallel Key Based Tree Implementation

The data structure which describes the computational domain in our PUM is a d-binary tree (quadtree, octree) used for the cover construction and the fast neighbour search for the evaluation of the Shepard PU functions (2.1). The use of a global unique integer key for each cell of the tree allows for a simple description of a partitioning of the computational domain. The set of all admissible[22] keys $\{0, 1, \ldots, \mathsf{k}_{\max}\}$ is simply split into $\wp$ subsets which are then assigned to the $\wp$ processors. We subdivide the range of keys into $\wp$ intervals

$$0 = r_0 \leq r_1 \leq \cdots \leq r_\wp = \mathsf{k}_{\max}$$

and assign the interval $[r_q, r_{q+1})$ to the qth processor, i.e. the set of tree cells assigned to the qth processor is $\{\mathcal{C}_L \,|\, \mathsf{k}_L \in [r_q, r_{q+1})\}$. With this very simple decomposition each processor can identify which processor stores a particular tree cell $\mathcal{C}_L$. A processor has to compute only the key value k_L for the tree cell $\mathcal{C}_L$ and the respective interval $[r_q, r_{q+1})$ with $\mathsf{k}_L \in [r_q, r_{q+1})$ to determine the processor q which stores this tree cell $\mathcal{C}_L$. The question now arises if such a partition of the domain with the path keys k_L (see Section 3 is a reasonable choice? Obviously the partitioning of the tree should be done in such a way that complete sub-trees are assigned to a processor to allow for efficient tree traversals. But the path key labelling scheme given above orders the tree cells rather horizontally (see Figure 4.1) instead of vertically. Therefore, we need to transform the path keys k_L to so-called domain keys k_L^D.

A simple transformation which leads to a vertical ordering of the tree cells is the following: First, we remove the leading bit (the initial root key value)

[22] The maximal key value $\mathsf{k}_{\max}$ is a constant depending on the architecture of the parallel computer.

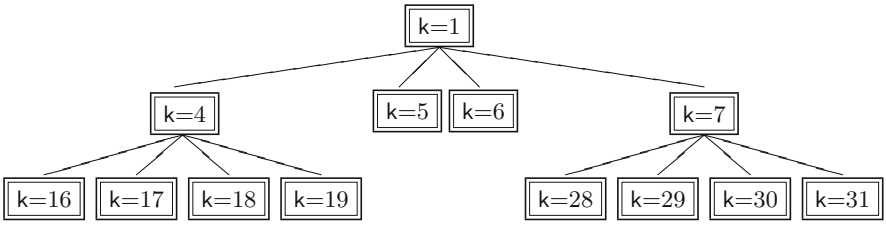


Figure 4.1. Horizontal ordering of a tree induced by the path key values k.

from the key's binary representation. Then we shift the remaining bits all the way to the left so that the leading bit of the path information is now stored in the most significant bit.[23] Assume that the key values are stored as an 32 bit integer and that we are in two dimensions. Then this simple transformation of a path key value k_L to a respective domain key value k_L^D is given by

$$\begin{aligned} \mathsf{k}_L &= 00000000000000000000001\underbrace{01110010}_{\text{path}} \\ \mathsf{k}_L^D &= \underbrace{01110010}_{\text{path}}00000000000000000000000. \end{aligned} \tag{4.1}$$

With these domain keys k_L^D the tree is now ordered vertically and we can assign complete sub-trees to a processor using the simple interval domain description $[r_q, r_{q+1})$.

Remark 4.1.
Note that the transformed keys are no longer unique and cannot be used as the key value for the associative container to store the tree itself. Obviously, a successor cell $\mathcal{C}_S$ of a tree cell $\mathcal{C}_L$ can be assigned the same domain key as the tree cell, i.e. $\mathsf{k}_S^D = \mathsf{k}_L^D$. Hence, we use the unique path keys k_L for the container and the associated domain keys k_L^D for the domain description, i.e. for the associated interval boundaries $[r_q, r_{q+1})$. ∎

Note that the description of the data partition via the intervals $[r_q, r_{q+1})$ defines a minimal refinement stage of the tree which has to be present on all processors to ensure the consistency of the tree. In the following we refer to this top part of the tree as the *common global tree*. The leaves $\mathcal{C}_L$ of the common global tree are characterized by the fact that they are the coarsest tree cells for which all possible successor cells are stored on the same processor, see Figure 4.2. The domain key values k_S^D of all possible successor cells $\mathcal{C}_S$ lie in the same interval $[r_q, r_{q+1})$ as the domain key k_L^D. We therefore refer to the leaves of the common global tree as *local sub-tree roots*.

[23] This transformation needs $O(1)$ operations if we assume that the current refinement level of the tree is known, otherwise it is of the order $O(J)$, where J denotes the number of levels of the tree.

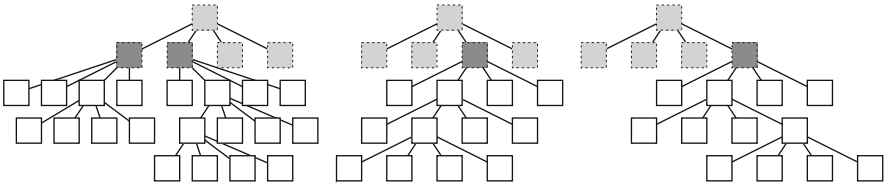


Figure 4.2. Common global tree (dashed, gray shaded) for a partition onto three processors. Local sub-tree roots (dark gray shaded) and the local sub-tree cells (white) for the first (left), second (centre) and third processor (right).

The order of the tree cells induced by the domain keys k_L^D given above is often referred to as bit-interleaving, the Morton-order, the Z-order or the N-order. The curve induced by mapping the domain keys to the associated cell centres corresponds to the Lebesgue curve (Figure 4.3 (upper left)) which is a space filling curve [73]. There are many space filling curves with different properties which might be more suitable for our needs; e.g. the sub-domains generated by the Lebesgue curve may be not connected [86] even for a d-rectangle, see Figure 4.3 (upper right). This increases the size of the local boundary and thereby the communication volume and possibly the number of communication steps.

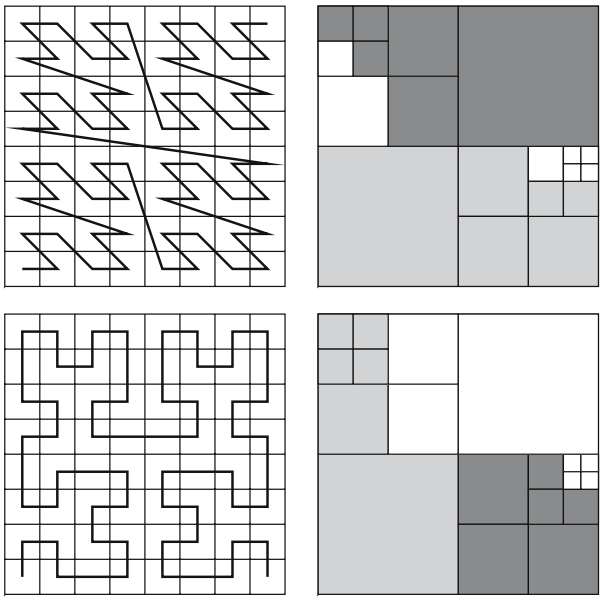


Figure 4.3. The Lebesgue curve (upper left) and the constructed sub-domains (upper right) for a partition onto three processors. The sub-domains are not connected since the curve does not have the locality property. The Hilbert curve (lower left) and the constructed sub-domains (lower right) for a partition onto three processors. The sub-domains are connected due to the locality property of the curve.

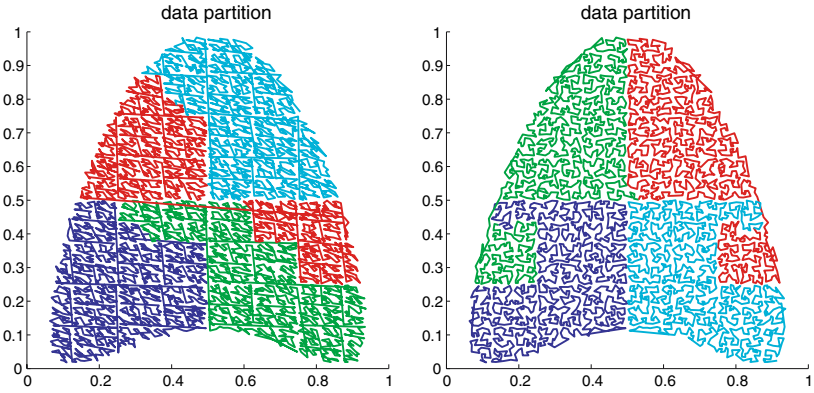


Figure 4.4. A partition of the point set P_J onto four processors (colour coded) using the domain keys k_L^D based on the Lebesgue curve (left), and the domain keys k_L^D based on the Hilbert curve (right).

4.2 Load Balancing with Space Filling Curves

The properties of space filling curves with respect to partitioning data for parallel computations have been studied in [85, 86]. Here, it turns out that the Hilbert curve (Figure 4.3 (lower left)) is more suitable for partitioning irregular data than the Lebesgue curve, see Figure 4.4. It provides a better data locality, e.g. the constructed sub-domains for a d-rectangle are connected (Figure 4.3 (lower right)) and the size of the local boundaries is of optimal order. Hence, we use the Hilbert curve instead of the Lebesgue curve to order the tree in our implementation; i.e. we use a different transformation than (4.1) to map the path keys k_L to domain keys k_L^D. This transformation of the path key values to Hilbert curve keys is more involved than the transformation (4.1) to Lebesgue curve keys, but it can also be realized with fast bit manipulations, see [76, Appendix B] for details.[24]

By changing the interval boundaries $\{r_q \,|\, q = 0, \ldots, \wp\}$, which describe the decomposition of our tree, we can balance the load among the processors. To this end we assign estimated work loads w_L as weights to the leaves $\mathcal{C}_L$ of the tree. Then we compute the current load estimate $w^{\hat{q}} = \sum w_L$ on every processor $\hat{q}$ and gather all remote load estimates w^q with $q \neq \hat{q}$. In the next step, the global load estimate $w = \sum_{q=0}^{\wp-1} w^q$, and the balanced load distribution $w_b^q = \frac{qw}{\wp}$ are computed. Then, every processor $\hat{q}$ iterates over its current set of leaves $\mathcal{C}_L$ of the tree in ascending order of the domain keys k_L^D and sets new (intermediate and inconsistent) local interval boundaries $\{\tilde{r}_q^{\hat{q}} \,|\, q = 0, \ldots, \wp\}$ accordingly. Finally, a reduction operation over all (local

[24] In general the transformation of a given key k_L to its associated Hilbert domain key k_L^D requires $O(J)$ operations, even if the current tree level J is known. But since we are interested in the domain keys k_L^D keys for all cells (or at least for all leaves) of the tree we can merge the transformation with the tree traversal which reduces the complexity of the transformation of a single key to $O(1)$.

intermediate) sets $\{\tilde{r}_q^{\tilde{q}} \,|\, q = 0, \ldots, \wp\}$ of the $\wp$ participating processors $\tilde{q}$ gives the new (global and consistent) interval boundaries $\{r_q \,|\, q = 0, \ldots, \wp\}$ which balance the estimated load w. Note that this load balancing scheme itself is completed in parallel.

Algorithm 4.1. Load Balancing

1. For all local leaves $\mathcal{C}_L$ of the tree:
   Assign estimated work load w_L.
2. Compute local estimate $w^{\hat{q}} = \sum_L w_L$ (on processor $\hat{q}$).
3. Gather remote estimates w^q with $q = 0, \ldots, \wp - 1$ and $q \neq \hat{q}$.
4. Compute global load estimate $w = \sum_{q=0}^{\wp-1} w^q$.
5. Set local estimate $w_g^{\hat{q}} = \sum_{q=0}^{q<\hat{q}} w^q$ (on processor $\hat{q}$).
6. Set balanced load distribution $w_b^q = \frac{qw}{\wp}$ for $q = 0, \ldots, \wp$.
7. For all local leaves $\mathcal{C}_L$ (in ascending order of domain keys k_L^D):
   Set local intermediate interval boundary $\tilde{r}_q^{\hat{q}} = k_L^D$ (on processor $\hat{q}$) where $q \in \{0, \ldots, \wp\}$ is the smallest integer with $w_g^{\hat{q}} \leq w_b^q$ and update estimate $w_g^{\hat{q}} = w_g^{\hat{q}} + w_L$.
8. Set (global) interval boundaries $r_q = \max_{\tilde{q}} \tilde{r}_q^{\tilde{q}}$ for all $q \in \{0, \ldots, \wp\}$ by reducing the set of all (local) intermediate boundaries $\{\tilde{r}_q^{\tilde{q}}\}$ over all processors $\tilde{q}$, force $r_0 = 0$ and $r_\wp = \mathsf{k}_{\max}$.

The quality of the load balancing scheme is essentially determined by the local load estimate w_L. With respect to the finest level J we can estimate the computational work associated with a particular patch $\omega_{i,k}$ for instance by the number of degrees of freedom assigned to the patch. Such an estimate is very cheap to compute, however, its quality is rather poor. We can obtain a very accurate estimate of the local computational work by considering the number of integration points used on a particular patch. Since we use a dynamic stopping criterion in the assembly of the stiffness matrix, such an estimate however cannot be attained with reasonable effort. Yet, if we assume that the integration of all nonzero entries of the stiffness matrix can be computed with a similar amount of work, we can use the number of nonzeros per block-row as our load estimate, i.e.

$$w_L = w_{i,J} = \dim(V_{i,J}^{p_{i,J}}) \sum_{\omega_k \in C_{i,J}} \dim(V_{k,J}^{p_{k,J}}).$$

If all polynomial degrees $p_{i,J} = p$ we can use $w_L = w_{i,J} = \mathrm{card}(C_{i,J})$. Hence, to balance the load with this work load estimate, we need to compute all neighbourhoods $C_{i,J}$ (in parallel) on the finest level J (in the unbalanced tree). Therefore, the computation of an accurate load estimate for parallel simulations is already a challenging task.

Remark 4.2.
The computational cost associated with the estimation of the current load

can often be reduced. In a time dependent setting or in adaptive refinement we usually have a pretty good load estimate from a previous time step or a coarser level without extra computations. This estimate can either be used directly to partition the data or it can be updated with only a few operations. Furthermore, we typically have to redistribute only a small amount of data in these situations. ∎

Remark 4.3.
With the load balancing scheme given in Algorithm 4.1 we balance the load with respect to only the finest level J. For our PUM discretization this is sufficient, since the largest amount of work is due to the finest level and we already assume that we coarsen the number of patches at a constant rate from level to level to obtain an optimal complexity multilevel iteration. However, sometimes it might be necessary to balance the load with respect to all levels simultaneously. Then, we need to modify the load balancing scheme in such a way that we consider all nodes of the tree rather than only the leaves. Note, however, that the design of an appropriate load estimate is somewhat more involved in such cases. ∎

Remark 4.4.
In the case of a PDE with (piecewise) constant coefficients we can employ a caching technique due to our regularized cover construction, see Remark 3.9, in some regions of the domain. Hence, not all integrals associated with the stiffness matrix are computed explicitly, many entries are computed only once and reused in the assembly. Therefore, when this caching technique is employed we need to update our load estimate to account for this change in computational work. Yet, we must be aware that the load has to balanced also with respect to the solution phase, i.e. the scalability of a single matrix-vector product should be retained. ∎

4.3 Parallel Cover Construction

Now that the computational domain is partitioned in an appropriate fashion among the processors we turn to the algorithmic changes for our parallel implementation, e.g. the computation of the communication pattern. The first task in our PUM is the multilevel cover construction (cf. Section 3.3) which is essentially a post-order tree operation. Due to our tree decomposition which assigns complete sub-trees to processors most work can be done completely in parallel. When we reach elements of the common global tree we need to gather the respective tree cells from remote processors. Then, all processors can complete the cover construction on the common global tree. The parallel version of the multilevel cover construction algorithm (cf. Algorithm 3.6) reads as:

Algorithm 4.2. Parallel Multilevel Cover Construction

1. Given the domain $\Omega \subset \mathbb{R}^d$ and a bounding box $R_\Omega = \bigotimes_{i=1}^{d} [l_\Omega^i, u_\Omega^i] \supset \overline{\Omega}$.

2. Given the interval boundaries $\{r_q \,|\, q = 0, \ldots, \wp\}$ and the local part $\tilde{P}_{\hat{q}}$ of the initial point set $\tilde{P} = \{x_j \,|\, x_j \in \overline{\Omega}, j = 1, \ldots, \tilde{N}\}$, i.e. $\mathsf{k}_j^D \in [r_{\hat{q}}, r_{\hat{q}+1})$ for all $x_j \in \tilde{P}_{\hat{q}}$.[25]
3. Initialize the common global d-binary tree (quadtree, octree) according to the $\wp$ intervals $[r_q, r_{q+1})$.
4. Build parallel d-binary sub-trees over local sub-tree roots, such that per leaf L at most one $x_i \in \tilde{P}_{\hat{q}}$ lies within the associated cell $\mathcal{C}_L := \bigotimes_{i=1}^{d} [l_L^i, u_L^i]$.
5. Set J to the finest refinement level of the tree.
6. For all local sub-tree roots $\mathcal{C}_L = \bigotimes_{i=1}^{d} [l_L^i, u_L^i]$:
   (a) If current tree cell $\mathcal{C}_L$ is an INNER tree node and $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Descend tree for all successors $\mathcal{C}_S$ of $\mathcal{C}_L$. ($\rightarrow$ 6(a))
      ii. Set patch $\omega_L = \bigotimes_{i=1}^{d} [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$ where $x_L = \frac{1}{2^d} \sum x_S$ is the centre of its successors points x_S and $h_L^i = 2 \max h_S^i$ is twice the maximum radius of its successors h_S^i.
      iii. Set active levels $l_L^{\min} = l_L^{\max} = \min l_S^{\min} - 1$ and update for all successors $l_S^{\min} = \min l_S^{\min}$.
      iv. Set polynomial degree $p_L := \min p_S$ to minimal degree of its successors.
   (b) Else if $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Set patch $\omega_L = \bigotimes_{i=1}^{d} [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$ where $x_L^i = l_L^i + \frac{1}{2}(u_L^i - l_L^i)$ and $h_L^i = \frac{\alpha_t}{2}(u_L^i - l_L^i)$.
      ii. Set active levels $l_L^{\min} = l_L^{\max} = J$.
      iii. Set polynomial degree to some given value p_L.
      iv. Set $P_{J\hat{q}} = P_{J\hat{q}} \cup \{x_L\}$, $C_{\Omega\hat{q}}^J = C_{\Omega\hat{q}}^J \cup \{\omega_L\}$.
7. Broadcast patches ω_L associated with local sub-tree roots $\mathcal{C}_L$ to all processors.
8. For the common global root cell $\mathcal{C}_L = \bigotimes_{i=1}^{d} [l_L^i, u_L^i] = R_\Omega$:
   (a) If current tree cell $\mathcal{C}_L$ is not the root of any complete processor sub-tree, and an INNER tree node with $\mathcal{C}_L \cap \Omega \neq \emptyset$:
      i. Descend tree for all successors of $\mathcal{C}_L$. ($\rightarrow$ 8(a))
      ii. Set patch $\omega_L = \bigotimes_{i=1}^{d} [x_L^i - h_L^i, x_L^i + h_L^i] \supset \mathcal{C}_L$ where $x_L = \frac{1}{2^d} \sum x_S$ is the centre of its successors points x_S and $h_L^i = 2 \max h_S^i$ is twice the maximum radius of its successors h_S^i.
      iii. Set active levels $l_L^{\min} = l_L^{\max} = \min l_S^{\min} - 1$ and update for all successors $l_S^{\min} = \min l_S^{\min}$.
      iv. Set polynomial degree $p_L := \min p_S$ to minimal degree of its successors.
9. For $k = 0, \ldots, J-1$:
   (a) Set $P_{k\hat{q}} = \{x_L \,|\, l_L^{\min} \leq k \leq l_L^{\max} \text{ and } \mathsf{k}_L^D \in [r_{\hat{q}}, r_{\hat{q}+1})\}$.
   (b) Set $C_{\Omega\hat{q}}^k = \{\omega_L \,|\, l_L^{\min} \leq k \leq l_L^{\max} \text{ and } \mathsf{k}_L^D \in [r_{\hat{q}}, r_{\hat{q}+1})\}$.

[25] An initial partition can easily be constructed by choosing uniform interval boundaries $\{r_q\}$ and partitioning the initial point set $\tilde{P}$ according to the domain keys on the finest possible tree level.

Note that the main difference between this parallel cover construction algorithm and Algorithm 3.6 is the use of different entry points for insert operations into the (global) tree. With Algorithm 3.6 we always insert points starting at the (global) root of the tree whereas in parallel each processor will essentially insert points into one of its local sub-tree roots only. Therefore, Algorithm 4.2 will yield the same sequence of covers C_Ω^k as Algorithm 3.6 only if the initial common global tree of step 3 (which is induced by the interval boundaries of step 2) is reasonable, cf. Section 4.1. Otherwise there can be slight differences in the constructed covers using different processor numbers $\wp$ for small initial point sets $\tilde{P}$.

4.4 Parallel Neighbour Search

Although most neighbours $\omega_{j,k}$ of a patch $\omega_{i,k}$ are stored on the local processor, the patch $\omega_{i,k}$ may well overlap patches which are stored on a remote processor. Hence, a processor may need copies of certain patches from a remote processor for the assembly of its assigned block-rows of the global stiffness matrices A_k. The computation of a single block-entry $(A_k)_{i,j}$ involves $\varphi_{i,k}$ and $\varphi_{j,k}$. Hence, it seems that we not only need remote patches $\omega_{j,k}$ but also all their neighbours $\omega_{l,k} \in C_{j,k}$ for the evaluation of the integrands involved in the block-row corresponding to the local patch $\omega_{i,k}$. This would significantly increase the communication volume and storage overhead due to parallelization. But since all function evaluations of $\varphi_{j,k}$ are restricted to the support of $\varphi_{i,k}$—recall that the integration domain for the block entry is $\Omega \cap \omega_{i,k} \cap \omega_{j,k}$—every neighbouring patch $\omega_{l,k} \in C_{j,k}$ that contributes a nonzero weight $W_{l,k}$ to the PU function $\varphi_{j,k}$ (on the integration domain) must also be a neighbour of $\omega_{i,k}$. Hence, it is sufficient to store copies of remote patches $\omega_{j,k}$ which are direct neighbours of a local patch $\omega_{i,k}$. There is no need to store neighbours of neighbours for the assembly of the stiffness matrix.

But how does a processor detect which neighbours $\omega_{j,k}$ exist on a remote processor? In fact, a processor cannot determine which patches to request from a remote processor. But a processor can certainly determine which of its local patches $\omega_{i,k}$ overlap the remote sub-trees with the help of the leaves of the common global tree. Hence, a processor can compute which local patches a remote processor may need to complete its neighbour search. Therefore, we need to perform only a parallel communication step where a processor sends its local patches which overlap the remote sub-trees prior to the computation of the neighbourhoods $C_{i,k}$.

Our cover construction algorithm constructs patches with increasing overlap on coarser levels $k < J$ to control the gradients $\nabla\varphi_{i,k}$ for $k < J$. Hence, many local patches $\omega_{i,\hat{k}}$ will overlap a remote sub-tree root patch $\omega_{j,\tilde{k}}$. But for the computation of the neighbourhoods $C_{j,\hat{k}}$ on level $\hat{k} > \tilde{k}$ the remote processor may not need the local patch $\omega_{i,\hat{k}}$. The remote patches $\omega_{j,\hat{k}}$ on level $\hat{k}$ might

not overlap $\omega_{i,\hat{k}}$, even though the coarser patch $\omega_{j,\tilde{k}}$ with $\tilde{k} < \hat{k}$ does overlap $\omega_{i,\hat{k}}$. Hence, the patch $\omega_{i,\hat{k}}$ is not needed by the remote processor to complete its computation and $\omega_{i,\hat{k}}$ should not be sent. This problem can be cured if we first compute a minimal coarse cover. Here, the patches associated with the tree cells are computed without increasing the overlap from level to level. Recall that in the single-level cover construction with Algorithm 3.1 and Algorithm 3.4 we already computed such a cover. There, we assigned patches ω_L to INNER tree nodes which only cover the union of their respective successor patches ω_S. But for the multilevel cover construction it was necessary to increase the size of these coarser patches in Algorithm 3.6 to control the size of the gradients of the associated partition of unity functions. Now in parallel computations we need to employ both types of coarse patches. We need the minimal patches for the partitioning of the domain, and we need the larger PUM patches for the construction of the shape functions. Note however that we only need the minimal patches for the common global tree. Therefore, we compute a minimal coarse cover, essentially with a parallel version of Algorithm 3.4, and then we store separate copies of the computed minimal patches associated with the leaves of the common global tree before we compute the correct PUM cover sequence with Algorithm 4.2. A processor can now test its local PUM patches against the minimal patches associated with remote sub-tree roots to compute the correct overlap with respect to the finest level J.

For the computation of the neighbourhoods $C_{i,k}$ on coarser levels $k < J$ we have to keep in mind that the complete tree is coarsened from level to level. Hence, we need to coarsen the common global tree as well. Furthermore, we also have to update the minimal patches associated with the coarser cells of the common global tree to compute the current overlaps.[26] After the exchange of the respective overlaps the neighbour search can be completed on each processor as before with Algorithm 3.2.

Remark 4.5.

The precise estimation of the communication volume is essential to obtain a scalable parallel implementation. It is very important to employ a conservative estimate, i.e. to over-estimate the amount of data, since all required data should be exchanged in a single communication step. However, if too many unnecessary data are sent and too many copies are stored locally we may lose the optimal complexity of the original algorithm due to an unsuitable non-optimal parallelization. ∎

[26] Under certain constraints on the overlap parameter α in the cover construction and the regularity of the tree we can compute the neighbourhoods $C_{i,k}$ on coarser levels $k < J$ directly from the neighbourhoods $C_{i,J}$ on the finest level J and there is no need for an overlap computation of coarser levels. But this does not improve the overall complexity since we still need to search for neighbours on the finest level J.

4.5 Parallel Matrix Assembly

Now that we have constructed the covers C_Ω^k in a distributed fashion, we come to the Galerkin discretization of a PDE in parallel. Here, we simply restrict the assembly of the stiffness matrix (and the transfer operators) on each of the $\wp$ processors to the block-rows associated with its assigned patches $\omega_{i,k}$. A processor $\hat{q}$ computes all block-entries

$$(A_k)_{i,j} = (A_{k(i,n),(j,m)})\,, \text{ with } A_{k(i,n),(j,m)} = a\,(\varphi_{j,k}\psi_{j,k}^m, \varphi_{i,k}\psi_{i,k}^n)\,, \quad (4.2)$$

where $\varphi_{i,k}$ is the PU function associated with one of its assigned patches $\omega_{i,k}$, i.e. the domain key $\mathsf{k}_{i,k}^D = \mathsf{k}_i^D$ associated with the patch $\omega_{i,k}$ is element of $[r_{\hat{q}}, r_{\hat{q}+1})$. The block-sparsity pattern of the respective block-row is determined by the neighbourhood $C_{i,k} = \{\omega_{j,k} \in C_\Omega^k \,|\, \omega_{i,k} \cap \omega_{j,k} \neq \emptyset\}$. Hence, a processor needs to access all geometric neighbours $\omega_{i,k} \cap \omega_{j,k} \neq \emptyset$ of its patches $\omega_{i,k}$ to compute its assigned part of the stiffness matrix A_k on level k. These neighbourhoods $C_{i,k}$ need to be computed prior to the assembly of the stiffness matrix A_k on level k so that during the assembly of the stiffness matrix A_k local copies of all neighbouring patches are available. After the exchange of all required neighbours $\omega_{j,k} \in C_{i,k}$ we can use Algorithm 3.3 for the assembly of the stiffness matrix without any modification. Note that the neighbourhoods $C_{i,k}$ are needed to pre-allocate the correct storage for the sparse block-matrix A_k as well as for each function evaluation of $\varphi_{i,k}$. Hence, we compute all $C_{i,k}$ only once and store them, i.e. the respective keys $\mathsf{k}_{i,k}$, in a separate sparse data structure.

4.6 Parallel Multilevel Solution

The first challenge we encounter in the parallelization of our multilevel solver is the question of smoothing in parallel. Recall that our smoothing schemes, namely the *BGS* and *MOS* iterations, are *SSC* methods. Hence, they are inherently sequential and their efficient parallelization is in general not feasible. Therefore, we need to modify the smoothing schemes in parallel computations. A common approach to circumvent the complete parallelization of *SSC* iterations is the domain decomposition approach, i.e. a sub-domain-blocking approach. Here, the *SSC* iteration is only applied locally within a processor's assigned sub-domain and these local iterates are then merged using an outer *PSC* iteration, i.e. by a sub-domain-block-Jacobi iteration. Note that this approach may lead to a change in the subspace splitting and in the overall iteration for different numbers of sub-domains, i.e. varying processor numbers $\wp$. For the *MOS* iteration for instance we now define the subspaces

$$\tilde{\mathcal{V}}_{l,q} := \text{span}_{(i,n),i\in C_l\cap C_{\Omega_q}} \langle \phi_i \psi_i^n \rangle$$

where C_{Ω_q} denotes the set of all patches assigned to processor q. The respective composite iteration can then be carried out with Algorithm 4.3.

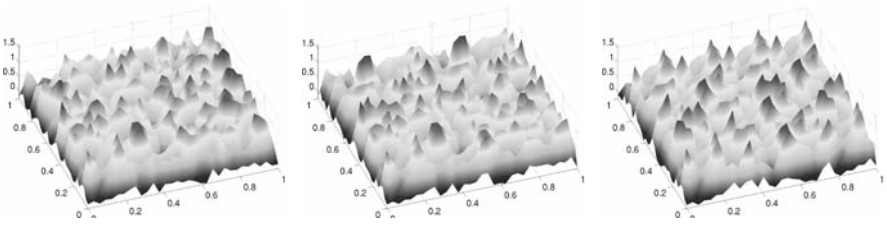


Figure 4.5. Smoothing results obtained with one iteration of the *BGS* smoother (left) and the composite *PSC-BGS* smoother (centre: with 4 sub-domains, right: with 16 sub-domains).

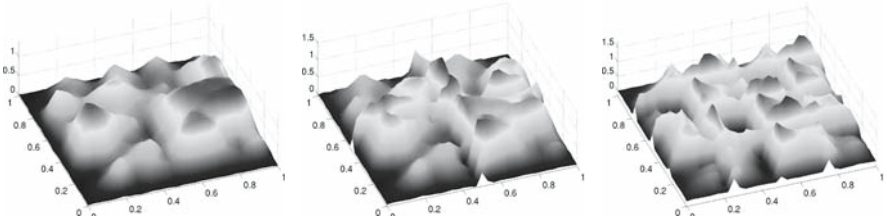


Figure 4.6. Smoothing results obtained with one iteration of the *MOS* smoother (left) and the composite *PSC-MOS* smoother (centre: with 4 sub-domains, right: with 16 sub-domains).

Algorithm 4.3. Composite *PSC-SSC* iteration

1. Exchange coefficients $\tilde{u}$ required for parallel matrix-vector product $A\tilde{u}$.
2. For all local $l \in C_{\Omega_q}$ on processor q:
   (a) Compute local residual $\hat{f}_{l,q} := P_{l,q}^T(\hat{f} - A\tilde{u})$.
   (b) Solve subspace problem $(P_{l,q}^T A\, P_{l,q})\tilde{u}_{l,q} = \hat{f}_{l,q}$.
   (c) Update iterate $\tilde{u} = \tilde{u} + P_{l,q}\tilde{u}_{l,q}$.

Note that the computation of the residual in step 2(a) employs an inconsistent coefficient vector $\tilde{u}$ since we exchange the coefficients only once per iteration; i.e. in step 2(c) we update coefficients $k \in C_{\Omega_q}$ on processor q but we do not update the respective copies stored on other processors.
The error reduction rate of such a composite *PSC-SSC* iteration is somewhat reduced compared with the rate obtained by the original *SSC* scheme but it is often still superior to that of the respective *PSC* scheme (for large sub-domains). Since we are interested in a parallel smoothing scheme for our multilevel solver, however, we must also be interested in the global smoothness of the respective iterates. From the plots depicted in Figures 4.5 and 4.6 we can observe that the composite *PSC-BGS* smoother with $\wp = 4$ and $\wp = 16$ sub-domains gives iterates with similar smoothness as the sequential *BGS* smoother ($\wp = 1$). However, for the composite *PSC-MOS* smoother

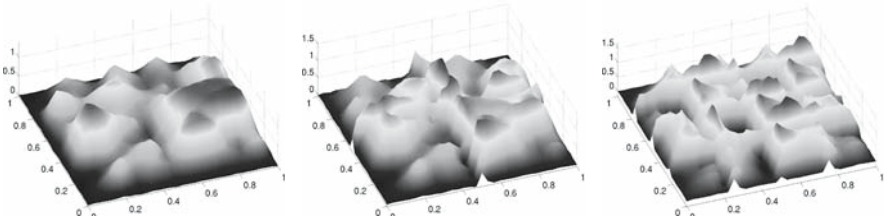


Figure 4.7. Smoothing results obtained with the MOS smoother (left) and a multiplicative smoother, i.e. an SSC iteration, based on the subspaces $\tilde{\mathcal{V}}_{l,q}$ (centre: with 4 sub-domains, right: with 16 sub-domains).

this is not the case. Here, we can clearly see the artifacts introduced by the sub-domain splitting. In the interior of each processor sub-domain we have the excellent error reduction and smoothing property of the MOS scheme. But close to the boundary of the processor sub-domain we lose the error reduction property and hence the global smoothness of the iterates. Note that the subspaces $\tilde{\mathcal{V}}_{l,q}$ near the sub-domain boundary become very similar to the non-overlapping subspaces $\mathcal{V}_l$ employed in a BGS smoother. Furthermore, the quality of the MOS smoother relies very much on the fact that smoothing is done in a successive fashion in the entire domain. In Figure 4.7 we give the iterates obtained by a multiplicative iteration, i.e. an SSC iteration, using the subspace splitting $\tilde{\mathcal{V}}_{l,q}$.[27] From these plots we see that the iterates are much smoother than those obtained with the composite PSC-MOS smoother. Hence, the outer PSC iteration seems to be the main reason for the substantial loss in the quality of the smoother composite PSC-MOS smoother. Note that the loss in smoothing for the composite PSC-MOS iteration is so severe that the respective multilevel iteration may diverge. Hence, for the parallelization of the MOS smoother we cannot pursue a simple sub-domain blocking approach together with an outer PSC iteration. Here, a more involved parallelization approach using multi-colour strategies must be taken [22]. However, the implementation of such techniques for unstructured data is rather cumbersome and sometimes even not feasible.

Since the MOS smoother is rather expensive and not easily parallelizable we usually employ the composite PSC-BGS smoother in our parallel multilevel solver. The computational complexity of this parallel smoother with respect to the number of operations, the storage requirements and the communication demands is comparable to that of a parallel matrix-vector product.

The second basic operation of our multilevel iteration is the application of the prolongation and restriction operators. In our implementation we completely assemble the prolongation as well as the restriction operators in an analogous

[27] Here we employ a consistent coefficient vector in the residual computation and update all copies of the coefficients on all processors, i.e. we employ multiple communication steps per iteration.

fashion as described above for the stiffness matrices A_k. This increases somewhat the storage overhead but on the other hand we do not need an explicit transposition or a transpose matrix-vector-product in parallel. We need only a parallel matrix-vector-product to transfer information between levels. Since we assign complete sub-trees to a processor most block-coefficients per processor are stored locally. Therefore the communication volume in the smoother as well as in the interlevel transfer is small. Here, the local-to-local projection has an especially simple communication demand due to its minimal block-sparsity pattern and our tree partitioning scheme. The ith block-row of the restriction operator I_k^{k-1} only consists of a single block-entry ${I_k^{k-1}}_{i,j}$ which corresponds to the coarser cover patch $\omega_{j,k-1} \in C^H_{i,k,k-1}$ associated with the ancestor tree-cell of the current fine level patch $\omega_{i,k}$ (cf. Section 3.3). Most of these ancestors are located on the same processor as the current patch due to our partition of the tree. Hence, the application of a local-to-local transfer operator involves very little communication.

4.7 Computational Complexity

The complexity of the parallel multilevel cover construction including the setup of the tree is given by $O(\frac{N}{\wp}J + \wp \log \wp)$, where $N = \text{card}(P_J)$ and P_J is the point set for our PUM space V_J^{PU} on the finest level J, i.e. $\text{card}(P_J)$ corresponds to the number of leaves of the tree, and $\wp$ is the number of processors. For the load balancing step we need to compute the neighbourhoods $C_{i,J}$ as the local load estimate. If we assume that the load imbalance is not too severe, this estimate can be computed with $O(\frac{N}{\wp}J)$ operations. The complexity of the necessary overlap computation is given by $O(J(\log \wp)^2)$ and the respective communication volume is of the order $O((\frac{N}{\wp})^{\frac{d-1}{d}})$.[28] Hence, the complexity of the tree construction and load balancing step is given by $O(\frac{N}{\wp}J + (\frac{N}{\wp})^{\frac{d-1}{d}} + J(\log \wp)^2 + \wp \log \wp)$.

Note that in our implementation we pre-compute the neighbourhoods $C_{i,k}$ on all levels $k = 0, \ldots, J$ prior to the assembly of the stiffness matrices A_k. Again, the complexity of the neighbourhood computation is given by $O(\frac{N}{\wp}J)$. These neighbourhoods, i.e. the respective keys, are stored in an additional sparse data structure since they determine not only the sparsity pattern of the stiffness matrix but they are also needed for the function evaluation of the PU functions $\varphi_{i,k}$. Hence, we compute the neighbourhoods $C_{i,k}$ only once and utilize the $O(1)$ random access capabilities of our key-based tree implementation so that the single function evaluation of $\varphi_{i,k}$ is of the order $O(1)$. Hence, the assembly of the stiffness matrices does not involve any searching operations so that its complexity is of the order $O(\frac{\mathcal{C}_{\text{A,NI}}}{\wp})$ in parallel.

[28] The complexity of the overlap computation may be reduced to $O(J \log \wp)$ if we employ a second tree data structure to store a complete copy of the common global tree.

The computation of the hierarchical neighbourhoods $C^H_{i,k,l}$ does not involve a complete search process. Here, we only need to check the active levels of the particular patch $\omega_{i,k}$. Either the patch itself is the only element of $C^H_{i,k,l}$ or we can directly compute the key for the ancestor patch (or the successor patches) with a constant number of operations. Therefore, we can assemble the local-to-local transfers also with $O(\frac{\mathcal{C}_{\tilde{\Pi},\mathrm{NI}}}{\wp})$ operations.
Finally, we need to consider the complexity of our multiplicative multilevel solver in parallel. In essence, the iteration given in Algorithm 3.5 consists of three operations: the application of a smoothing scheme, a matrix-vector product and a scalar product. Since our parallel *PSC-BGS* smoothing scheme has a similar complexity as a parallel matrix-vector product, we only need to consider the complexities of the parallel matrix-vector product and the parallel scalar product. Obviously, a scalar product can be computed in parallel with $O(\frac{N}{\wp} + \log p)$ operations and communication steps. A parallel matrix-vector product (for sparse matrices) has a parallel complexity of $O(\frac{N}{\wp} + (\frac{N}{\wp})^{\frac{d-1}{d}})$. To obtain the overall complexity of our parallel multilevel solver we need to consider that we apply these operations on all levels. Note that the overall number of arithmetic operation is not effected by the multilevel structure.[29] However, we need to communicate on each level $k = 0, \ldots, J$. Hence, the overall complexity of our parallel multilevel solver is given by $O(\frac{N}{\wp} + (\frac{N}{\wp})^{\frac{d-1}{d}} + J + \log \wp)$.

References

1. H. Babovsky, *Die Boltzmann–Gleichung*, B. G. Teubner, 1998.
2. I. Babuška, U. Banerjee, and J. E. Osborn, *Meshless and Generalized Finite Element Methods: A Survey of Some Major Results*, in Meshfree Methods for Partial Differential Equations, M. Griebel and M. A. Schweitzer, eds., vol. 26 of Lecture Notes in Computational Science and Engineering, Springer, 2002, pp. 1–20.
3. ———, *On Principles for the Selection of Shape Functions for the Generalized Finite Element Method*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 5595–5629.
4. ———, *Survey of Meshless and Generalized Finite Element Methods: A Unified Approach*, Acta Numerica, (2003), pp. 1–125.
5. ———, *Generalized Finite Element Methods—Main Ideas, Results, and Perspective*, Inter. J. Comput. Meth., 1 (2004), pp. 67–103.
6. ———, *On the Approximability and the Selection of Particle Shape Functions*, Numer. Math., 96 (2004), pp. 601–640.
7. I. Babuška, G. Caloz, and J. E. Osborn, *Special Finite Element Methods for a Class of Second Order Elliptic Problems with Rough Coefficients*, SIAM J. Numer. Anal., 31 (1994), pp. 945–981.

[29] We assume that the number of degrees of freedom is reduced at a geometric rate from level to level.

8. I. Babuška and J. M. Melenk, *The Partition of Unity Finite Element Method: Basic Theory and Applications*, Comput. Meth. Appl. Mech. Engrg., 139 (1996), pp. 289–314. Special Issue on Meshless Methods.
9. ———, *The Partition of Unity Method*, Int. J. Numer. Meth. Engrg., 40 (1997), pp. 727–758.
10. S. Beissel and T. Belytschko, *Nodal Integration of the Element-Free Galerkin Method*, Comput. Meth. Appl. Mech. Engrg., 139 (1996), pp. 49–74.
11. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, *Meshless Methods: An Overview and Recent Developments*, Comput. Meth. Appl. Mech. Engrg., 139 (1996), pp. 3–47. Special Issue on Meshless Methods.
12. M. Bern, D. Eppstein, and J. Gilbert, *Provably Good Mesh Generation*, J. Comput. Sys. Sci., 48 (1994), pp. 384–409.
13. J. Bey, *Finite-Volumen- und Mehrgitter-Verfahren für elliptische Randwertprobleme*, Advances in Numerical Mathematics, Teubner, 1998.
14. W. W. Bradburry and R. Fletcher, *New Iterative Methods for the Solution of the Eigenproblem*, Numer. Math., 9 (1966), pp. 259–267.
15. D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, Cambridge University Press, 2001.
16. D. Braess and W. Hackbusch, *A New Convergence Proof for the Multigrid Method Including the V-Cycle*, SIAM J. Numer. Anal., 20 (1983), pp. 967–975.
17. J. H. Bramble and X. Zhang, *Handbook of Numerical Analysis*, in The Analysis of Multigrid Methods, P. G. Ciarlet and J. L. Lions, eds., vol. VII, Elsevier, 2000, pp. 173–416.
18. W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM, 2nd ed., 2000.
19. J. S. Chen, C. T. Wu, and S. Yoon, *Non-linear Version of Stabilized Conforming Nodal Integration for Galerkin Mesh-free Methods*, Int. J. Numer. Meth. Engrg., 53 (2002), pp. 2587–2615.
20. J. S. Chen, C. T. Wu, S. Yoon, and Y. You, *A Stabilized Conforming Nodal Integration for Galerkin Mesh-free Methods*, Int. J. Numer. Meth. Engrg., 50 (2001), pp. 435–466.
21. P. G. Ciarlet, *The Finite Element Methods for Elliptic Problems*, North–Holland, 1980.
22. J. Culberson, *Graph Coloring Page*. www.cs.ualberta.ca/ joe/Coloring/.
23. W. Dahmen, *Multiscale Analysis, Approximation, and Interpolation Spaces*, in Approximation Theory VIII, C. K. Chui and L. L. Schumaker, eds., vol. 2, World Scientific, 1995, pp. 47–88.
24. R. Dave, J. Dubinski, and L. Hernquist, *Parallel TreeSPH*, New Astronomy, 2 (1997), pp. 277–297.
25. S. De and K. J. Bathe, *The Method of Finite Spheres*, Comput. Mech., 25 (2000), pp. 329–345.
26. ———, *The Method of Finite Spheres with improved Numerical Integration*, Comput.s & Struct., 79 (2001), pp. 2183–2196.
27. G. A. Dilts, *Moving-Least-Square-Particle Hydrodynamics I: Consistency and Stability*, Int. J. Numer. Meth. Engrg., 44 (1999), pp. 1115–1155.
28. ———, *Moving-Least-Square-Particle Hydrodynamics II: Conservation and Boundaries*, Int. J. Numer. Meth. Engrg., 48 (2000), pp. 1503–1524.
29. J. Dolbow and T. Belytschko, *Numerical Integration of the Galerkin Weak Form in Meshfree Methods*, Comput. Mech., 23 (1999), pp. 219–230.

30. C. A. M. Duarte, *A Review of Some Meshless Methods to Solve Partial Differtial Equations*, Tech. Rep. 95-06, TICAM, University of Texas, 1995.
31. C. A. M. Duarte and J. T. Oden, *hp Clouds – A Meshless Method to Solve Boundary Value Problems*, Numer. Meth. for PDE, 12 (1996), pp. 673–705.
32. J. A. George, *Nested Dissection of a Regular Finite Element Mesh*, SIAM J. Num. Anal., 10 (1973), pp. 345–363.
33. T. Gerstner and M. Griebel, *Numerical Integration using Sparse Grids*, Numer. Alg., 18 (1998), pp. 209–232.
34. R. A. Gingold and J. J. Monaghan, *Smoothed Particle Hydrodynamics: Theory and Application to non-spherical Stars*, Mon. Not. R. Astr. Soc., 181 (1977), pp. 375–389.
35. ———, *Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics*, J. Comput. Phys., 46 (1982), pp. 429–453.
36. R. T. Glassey, *The Cauchy Problem in Kinetic Theory*, SIAM, 1996.
37. M. Griebel, S. Knapek, G. Zumbusch, and A. Caglar, *Numerische Simulation in der Molekulardynamik*, Springer, 2003.
38. M. Griebel, P. Oswald, and M. A. Schweitzer, *A Particle-Partition of Unity Method—Part VI: A p-robust Multilevel Solver*, in Meshfree Methods for Partial Differential Equations II, M. Griebel and M. A. Schweitzer, eds., vol. 43 of Lecture Notes in Computational Science and Engineering, Springer, 2004, pp. 71–92.
39. M. Griebel and M. A. Schweitzer, *A Particle-Partition of Unity Method for the Solution of Elliptic, Parabolic and Hyperbolic PDE*, SIAM J. Sci. Comput., 22 (2000), pp. 853–890.
40. ———, *A Particle-Partition of Unity Method—Part II: Efficient Cover Construction and Reliable Integration*, SIAM J. Sci. Comput., 23 (2002), pp. 1655–1682.
41. ———, *A Particle-Partition of Unity Method—Part III: A Multilevel Solver*, SIAM J. Sci. Comput., 24 (2002), pp. 377–409.
42. ———, *A Particle-Partition of Unity Method—Part IV: Parallelization*, in Meshfree Methods for Partial Differential Equations, M. Griebel and M. A. Schweitzer, eds., vol. 26 of Lecture Notes in Computational Science and Engineering, Springer, 2002, pp. 161–192.
43. ———, *A Particle-Partition of Unity Method—Part V: Boundary Conditions*, in Geometric Analysis and Nonlinear Partial Differential Equations, S. Hildebrandt and H. Karcher, eds., Springer, 2002, pp. 517–540.
44. ———, eds., *Meshfree Methods for Partial Differential Equations*, vol. 26 of Lecture Notes in Computational Science and Engineering, Springer, 2002.
45. ———, eds., *Meshfree Methods for Partial Differential Equations II*, vol. 43 of Lecture Notes in Computational Science and Engineering, Springer, 2005.
46. F. C. Günther and W. K. Liu, *Implementation of Boundary Conditions for Meshless Methods*, Comput. Meth. Appl. Mech. Engrg., 163 (1998), pp. 205–230.
47. W. Hackbusch, *Multi-Grid Methods and Applications*, vol. 4 of Springer Series in Computational Mathematics, Springer, 1985.
48. ———, *Elliptic Differential Equations. Theory and Numerical Treatment*, Springer, 1992.
49. ———, *Iterative Solution of Large Sparse Linear Systems of Equations*, Springer, 1994.

50. W. Han and X. Meng, *Some Studies of the Reproducing Kernel Particle Method*, in Meshfree Methods for Partial Differential Equations, M. Griebel and M. A. Schweitzer, eds., vol. 26 of Lecture Notes in Computational Science and Engineering, Springer, 2002, pp. 193–210.
51. J. Hoschek and D. Lasser, *Grundlagen der geometrischen Datenverarbeitung*, B. G. Teubner, 1992.
52. D. E. Knuth, *The Art of Computer Programming*, vol. 3 Searching and Sorting, Addison Wesley, Second ed., 1998.
53. Y. Krongauz and T. Belytschko, *Enforcement of Essential Boundary Conditions in Meshless Approximations using Finite Elements*, Comput. Meth. Appl. Mech. Engrg., 131 (1996), pp. 133–145.
54. P. Lancaster and K. Salkauskas, *Surfaces Generated by Moving Least Squares Methods*, Math. Comp., 37 (1981), pp. 141–158.
55. S. Li and W. K. Liu, *Meshfree Particle Methods*, Springer, 2004.
56. C. Lia and G. Carraro, *A Parallel Tree SPH Code for Galaxy Formation*, Month. Not. Roy. Astro. Soc., 314 (2000), pp. 145–161.
57. D. E. Longsine and S. F. McCormick, *Simultaneous Rayleigh-Quotient Minimization Methods for $Ax = \lambda Bx$*, Lin. Alg. Appl., 34 (1980), pp. 195–234.
58. Y. Y. Lu, T. Belytschko, and L. Gu, *A New Implementation of the Element Free Galerkin Method*, Comput. Math. Appl. Mech. Engrg., 113 (1994), pp. 397–414.
59. L. B. Lucy, *A Numerical Approach to the Testing of the Fission Hypothesis*, Astro. J., 82 (1977), pp. 1013–1024.
60. M. Macri, S. De, and M. S. Shepard, *Hierarchical Tree-based Discretization in the Method of Finite Spheres*, Comput. & Struct., 81 (2003), pp. 789–803.
61. J. M. Melenk, *On Approximation in Meshless Methods*, in Durham 2004, J. Blowey and A. Craig, eds., Springer, 2004. this volume.
62. J. J. Monaghan, *Why Particle Methods Work*, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 422–433.
63. ———, *An Introduction to SPH*, Comput. Phys. Comm., 48 (1988), pp. 89–96.
64. K. Nanbu, *Direct Simulation Scheme derived from the Boltzmann Equation*, J. Phys. Soc. Japan, 49 (1980), pp. 20–49.
65. ———, *Theoretical Basis on the Direct Simulation Monte Carlo Method*, in Rarefied Gas Dynamics, V. Boffi and C. Cercignani, eds., vol. 1, Teubner, 1986.
66. H. Neunzert, A. Klar, and J. Struckmeier, *Particle Methods: Theory and Applications*, Tech. Rep. 95-153, Arbeitsgruppe Technomathematik, Universität Kaiserslautern, 1995.
67. H. Neunzert and J. Struckmeier, *Particle Methods for the Boltzmann Equation*, Acta Numerica, (1995), pp. 417–457.
68. J. Nitsche, *Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind*, Abh. Math. Sem. Univ. Hamburg, 36 (1970–1971), pp. 9–15.
69. E. Novak, K. Ritter, R. Schmitt, and A. Steinbauer, *On a Recent Interpolatory Method for High Dimensional Integration*, J. Comput. Appl. Math., 15 (1999), pp. 499–522.
70. P. Oswald, *Multilevel Finite Element Approximation*, Teubner Skripten zur Numerik, Teubner, 1994.
71. T. N. L. Patterson, *The Optimum Addition of Points to Quadrature Formulae*, Math. Comp., 22 (1968), pp. 847–856.

72. A. Pothen, *Graph Partitioning Algorithms with Applications to Scientific Computing*, in Parallel Numerical Algorithms, D. E. Keyes, A. Sameh, and V. Venkatakrishnan, eds., Kluwer Academic Publishers, 1997, pp. 323–368.
73. H. Sagan, *Space-Filling Curves*, Springer, 1994.
74. H. Samet, *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*, Addison–Wesley, 1990.
75. ———, *The Design and Analysis of Spatial Data Structures*, Addison–Wesley, 1990.
76. M. A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*, vol. 29 of Lecture Notes in Computational Science and Engineering, Springer, 2003.
77. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice–Hall, 1973.
78. T. Strouboulis, I. Babuška, and K. Copps, *The Design and Analysis of the Generalized Finite Element Method*, Comput. Meth. Appl. Mech. Engrg., 181 (2000), pp. 43–69.
79. T. Strouboulis, K. Copps, and I. Babuška, *The Generalized Finite Element Method*, Comput. Meth. Appl. Mech. Engrg., 190 (2001), pp. 4081–4193.
80. U. Trottenberg, C. W. Osterlee, and A. Schüller, *Multigrid*, Academic Press, 2001, Appendix A: An Introduction to Algebraic Multigrid by K. Stüben, pp. 413–532.
81. M. S. Warren and J. K. Salmon, *A Parallel Hashed Oct-Tree N-Body Algorithm*, in Supercomputing '93, IEEE Comput. Soc., 1993, pp. 12–21.
82. ———, *A Portable Parallel Particle Program*, Comput. Phys. Comm., 87 (1995).
83. J. Xu, *Iterative Methods by Space Decomposition and Subspace Correction*, SIAM Review, 34 (1992), pp. 581–613.
84. H. Yserentant, *Old and New Convergence Proofs for Multigrid Methods*, Acta Numerica 93, (1993), pp. 285–326.
85. G. W. Zumbusch, *On the Quality of Space-Filling Curve Induced Partitions*, Z. Angew. Math. Mech., 81 Suppl. 1 (2001), pp. 25–28.
86. ———, *Parallel Multilevel Methods. Adaptive Mesh Refinement and Loadbalancing*, Advances in Numerical Mathematics, Teubner, 2003.

Universitext

Aguilar, M.; Gitler, S.; Prieto, C.: Algebraic Topology from a Homotopical Viewpoint

Aksoy, A.; Khamsi, M. A.: Methods in Fixed Point Theory

Alevras, D.; Padberg M. W.: Linear Optimization and Extensions

Andersson, M.: Topics in Complex Analysis

Aoki, M.: State Space Modeling of Time Series

Arnold, V. I.: Lectures on Partial Differential Equations

Audin, M.: Geometry

Aupetit, B.: A Primer on Spectral Theory

Bachem, A.; Kern, W.: Linear Programming Duality

Bachmann, G.; Narici, L.; Beckenstein, E.: Fourier and Wavelet Analysis

Badescu, L.: Algebraic Surfaces

Balakrishnan, R.; Ranganathan, K.: A Textbook of Graph Theory

Balser, W.: Formal Power Series and Linear Systems of Meromorphic Ordinary Differential Equations

Bapat, R.B.: Linear Algebra and Linear Models

Benedetti, R.; Petronio, C.: Lectures on Hyperbolic Geometry

Benth, F. E.: Option Theory with Stochastic Analysis

Berberian, S. K.: Fundamentals of Real Analysis

Berger, M.: Geometry I, and II

Bliedtner, J.; Hansen, W.: Potential Theory

Blowey, J. F.; Coleman, J. P.; Craig, A. W. (Eds.): Theory and Numerics of Differential Equations

Blowey, J.; Craig, A.: Frontiers in Numerical Analysis. Durham 2004

Blyth, T. S.: Lattices and Ordered Algebraic Structures

Börger, E.; Grädel, E.; Gurevich, Y.: The Classical Decision Problem

Böttcher, A; Silbermann, B.: Introduction to Large Truncated Toeplitz Matrices

Boltyanski, V.; Martini, H.; Soltan, P. S.: Excursions into Combinatorial Geometry

Boltyanskii, V. G.; Efremovich, V. A.: Intuitive Combinatorial Topology

Bonnans, J. F.; Gilbert, J. C.; Lemarchal, C.; Sagastizbal, C. A.: Numerical Optimization

Booss, B.; Bleecker, D. D.: Topology and Analysis

Borkar, V. S.: Probability Theory

Brunt B. van: The Calculus of Variations

Carleson, L.; Gamelin, T. W.: Complex Dynamics

Cecil, T. E.: Lie Sphere Geometry: With Applications of Submanifolds

Chae, S. B.: Lebesgue Integration

Chandrasekharan, K.: Classical Fourier Transform

Charlap, L. S.: Bieberbach Groups and Flat Manifolds

Chern, S.: Complex Manifolds without Potential Theory

Chorin, A. J.; Marsden, J. E.: Mathematical Introduction to Fluid Mechanics

Cohn, H.: A Classical Invitation to Algebraic Numbers and Class Fields

Curtis, M. L.: Abstract Linear Algebra

Curtis, M. L.: Matrix Groups

Cyganowski, S.; Kloeden, P.; Ombach, J.: From Elementary Probability to Stochastic Differential Equations with MAPLE

Dalen, D. van: Logic and Structure

Das, A.: The Special Theory of Relativity: A Mathematical Exposition

Debarre, O.: Higher-Dimensional Algebraic Geometry

Deitmar, A.: A First Course in Harmonic Analysis

Demazure, M.: Bifurcations and Catastrophes

Devlin, K. J.: Fundamentals of Contemporary Set Theory

DiBenedetto, E.: Degenerate Parabolic Equations

Diener, F.; Diener, M.(Eds.): Nonstandard Analysis in Practice

Dimca, A.: Sheaves in Topology

Dimca, A.: Singularities and Topology of Hypersurfaces

DoCarmo, M. P.: Differential Forms and Applications

Duistermaat, J. J.; Kolk, J. A. C.: Lie Groups

Edwards, R. E.: A Formal Background to Higher Mathematics Ia, and Ib

Edwards, R. E.: A Formal Background to Higher Mathematics IIa, and IIb

Emery, M.: Stochastic Calculus in Manifolds

Emmanouil, I.: Idempotent Matrices over Complex Group Algebras

Endler, O.: Valuation Theory

Erez, B.: Galois Modules in Arithmetic

Everest, G.; Ward, T.: Heights of Polynomials and Entropy in Algebraic Dynamics

Farenick, D. R.: Algebras of Linear Transformations

Foulds, L. R.: Graph Theory Applications

Franke, J.; Hrdle, W.; Hafner, C. M.: Statistics of Financial Markets: An Introduction

Frauenthal, J. C.: Mathematical Modeling in Epidemiology

Freitag, E.; Busam, R.: Complex Analysis

Friedman, R.: Algebraic Surfaces and Holomorphic Vector Bundles

Fuks, D. B.; Rokhlin, V. A.: Beginner's Course in Topology

Fuhrmann, P. A.: A Polynomial Approach to Linear Algebra

Gallot, S.; Hulin, D.; Lafontaine, J.: Riemannian Geometry

Gardiner, C. F.: A First Course in Group Theory

Gårding, L.; Tambour, T.: Algebra for Computer Science

Godbillon, C.: Dynamical Systems on Surfaces

Godement, R.: Analysis I, and II

Goldblatt, R.: Orthogonality and Spacetime Geometry

Gouvêa, F. Q.: p-Adic Numbers

Gross, M. et al.: Calabi-Yau Manifolds and Related Geometries

Gustafson, K. E.; Rao, D. K. M.: Numerical Range. The Field of Values of Linear Operators and Matrices

Gustafson, S. J.; Sigal, I. M.: Mathematical Concepts of Quantum Mechanics

Hahn, A. J.: Quadratic Algebras, Clifford Algebras, and Arithmetic Witt Groups

Hájek, P.; Havránek, T.: Mechanizing Hypothesis Formation

Heinonen, J.: Lectures on Analysis on Metric Spaces

Hlawka, E.; Schoißengeier, J.; Taschner, R.: Geometric and Analytic Number Theory

Holmgren, R. A.: A First Course in Discrete Dynamical Systems

Howe, R., Tan, E. Ch.: Non-Abelian Harmonic Analysis

Howes, N. R.: Modern Analysis and Topology

Hsieh, P.-F.; Sibuya, Y. (Eds.): Basic Theory of Ordinary Differential Equations

Humi, M., Miller, W.: Second Course in Ordinary Differential Equations for Scientists and Engineers

Hurwitz, A.; Kritikos, N.: Lectures on Number Theory

Huybrechts, D.: Complex Geometry: An Introduction

Isaev, A.: Introduction to Mathematical Methods in Bioinformatics

Istas, J.: Mathematical Modeling for the Life Sciences

Iversen, B.: Cohomology of Sheaves

Jacod, J.; Protter, P.: Probability Essentials

Jennings, G. A.: Modern Geometry with Applications

Jones, A.; Morris, S. A.; Pearson, K. R.: Abstract Algebra and Famous Inpossibilities

Jost, J.: Compact Riemann Surfaces

Jost, J.: Dynamical Systems. Examples of Complex Behaviour

Jost, J.: Postmodern Analysis

Jost, J.: Riemannian Geometry and Geometric Analysis

Kac, V.; Cheung, P.: Quantum Calculus

Kannan, R.; Krueger, C. K.: Advanced Analysis on the Real Line

Kelly, P.; Matthews, G.: The Non-Euclidean Hyperbolic Plane

Kempf, G.: Complex Abelian Varieties and Theta Functions

Kitchens, B. P.: Symbolic Dynamics

Kloeden, P.; Ombach, J.; Cyganowski, S.: From Elementary Probability to Stochastic Differential Equations with MAPLE

Kloeden, P. E.; Platen; E.; Schurz, H.: Numerical Solution of SDE Through Computer Experiments

Kostrikin, A. I.: Introduction to Algebra

Krasnoselskii, M. A.; Pokrovskii, A. V.: Systems with Hysteresis

Kurzweil, H.; Stellmacher, B.: The Theory of Finite Groups. An Introduction

Lang, S.: Introduction to Differentiable Manifolds

Luecking, D. H., Rubel, L. A.: Complex Analysis. A Functional Analysis Approach

Ma, Zhi-Ming; Roeckner, M.: Introduction to the Theory of (non-symmetric) Dirichlet Forms

Mac Lane, S.; Moerdijk, I.: Sheaves in Geometry and Logic

Marcus, D. A.: Number Fields

Martinez, A.: An Introduction to Semiclassical and Microlocal Analysis

Matoušek, J.: Using the Borsuk-Ulam Theorem

Matsuki, K.: Introduction to the Mori Program

Mazzola, G.; Milmeister G.; Weissman J.: Comprehensive Mathematics for Computer Scientists 1

Mazzola, G.; Milmeister G.; Weissman J.: Comprehensive Mathematics for Computer Scientists 2

Mc Carthy, P. J.: Introduction to Arithmetical Functions

McCrimmon, K.: A Taste of Jordan Algebras

Meyer, R. M.: Essential Mathematics for Applied Field

Meyer-Nieberg, P.: Banach Lattices

Mikosch, T.: Non-Life Insurance Mathematics

Mines, R.; Richman, F.; Ruitenburg, W.: A Course in Constructive Algebra

Moise, E. E.: Introductory Problem Courses in Analysis and Topology

Montesinos-Amilibia, J. M.: Classical Tessellations and Three Manifolds

Morris, P.: Introduction to Game Theory

Nikulin, V. V.; Shafarevich, I. R.: Geometries and Groups

Oden, J. J.; Reddy, J. N.: Variational Methods in Theoretical Mechanics

Øksendal, B.: Stochastic Differential Equations

Øksendal, B.; Sulem, A.: Applied Stochastic Control of Jump Diffusions

Poizat, B.: A Course in Model Theory

Polster, B.: A Geometrical Picture Book

Porter, J. R.; Woods, R. G.: Extensions and Absolutes of Hausdorff Spaces

Radjavi, H.; Rosenthal, P.: Simultaneous Triangularization

Ramsay, A.; Richtmeyer, R. D.: Introduction to Hyperbolic Geometry

Rees, E. G.: Notes on Geometry

Reisel, R. B.: Elementary Theory of Metric Spaces

Rey, W. J. J.: Introduction to Robust and Quasi-Robust Statistical Methods

Ribenboim, P.: Classical Theory of Algebraic Numbers

Rickart, C. E.: Natural Function Algebras

Rotman, J. J.: Galois Theory

Rubel, L. A.: Entire and Meromorphic Functions

Ruiz-Tolosa, J. R.; Castillo E.: From Vectors to Tensors

Runde, V.: A Taste of Topology

Rybakowski, K. P.: The Homotopy Index and Partial Differential Equations

Sagan, H.: Space-Filling Curves

Samelson, H.: Notes on Lie Algebras

Schiff, J. L.: Normal Families

Sengupta, J. K.: Optimal Decisions under Uncertainty

Séroul, R.: Programming for Mathematicians

Seydel, R.: Tools for Computational Finance

Shafarevich, I. R.: Discourses on Algebra

Shapiro, J. H.: Composition Operators and Classical Function Theory

Simonnet, M.: Measures and Probabilities

Smith, K. E.; Kahanpää, L.; Kekäläinen, P.; Traves, W.: An Invitation to Algebraic Geometry

Smith, K. T.: Power Series from a Computational Point of View

Smoryński, C.: Logical Number Theory I. An Introduction

Stichtenoth, H.: Algebraic Function Fields and Codes

Stillwell, J.: Geometry of Surfaces

Stroock, D. W.: An Introduction to the Theory of Large Deviations

Sunder, V. S.: An Invitation to von Neumann Algebras

Tamme, G.: Introduction to Étale Cohomology

Tondeur, P.: Foliations on Riemannian Manifolds

Toth, G.: Finite Mbius Groups, Minimal Immersions of Spheres, and Moduli

Verhulst, F.: Nonlinear Differential Equations and Dynamical Systems

Wong, M. W.: Weyl Transforms

Xambó-Descamps, S.: Block Error-Correcting Codes

Zaanen, A.C.: Continuity, Integration and Fourier Theory

Zhang, F.: Matrix Theory

Zong, C.: Sphere Packings

Zong, C.: Strange Phenomena in Convex and Discrete Geometry

Zorich, V. A.: Mathematical Analysis I

Zorich, V. A.: Mathematical Analysis II