Fast Optimised Wavelet Methods for Control Problems Constrained by Elliptic PDEs

Dissertation

zur Erlangung des Doktorgrades
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Carsten Burstedde
aus Köln

Bonn, 13. September 2005

Angefertigt mit Genehmigung
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachterin: Prof. Dr. Angela Kunoth
2. Gutachter: Prof. Dr. Michael Griebel
Tag der Prüfung: 12.12.2005

Diese Arbeit ist mit Unterstützung des
von der Deutschen Forschungsgemeinschaft getragenen
SFB 611 an der Universität Bonn entstanden

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn
`http://hss.ulb.uni-bonn.de/diss_online/` elektronisch publiziert

Erscheinungsjahr 2005

Für meine Eltern

Zusammenfassung

Schnelle optimierte Waveletmethoden für Kontrollprobleme unter elliptischen PDG-Nebenbedingungen

Carsten Burstedde, Institut für Angewandte Mathematik

Die Betrachtung von Kontrollproblemen mit partiellen Differentialgleichungen (PDG) als Nebenbedingungen hat in jüngster Zeit immens an Bedeutung gewonnen. Praktische Anwendungen reichen von Heiz- und Kühlprozessen und Vorgängen der Strömungsmechanik in industrieller Fertigung und Medizin bis hin zu Problemstellungen der Finanzmathematik. Während bei der Simulation partieller Differentialgleichungen der Zustand eines Systems aus vorgegebenen Randbedingungen und äußeren Einflüssen zu berechnen ist, tritt bei Kontrollproblemen die Fragestellung in den Vordergrund, durch welche Daten und Einstellungen denn ein möglichst optimaler Zustand zu erzielen ist. Hier wird die PDG zur Nebenbedingung für das neu hinzugekommene und übergeordnete Ziel der Optimierung. Vielfach wird die Berechnung derartiger Optimierungsprobleme durch moderne und effiziente numerische Techniken überhaupt erst ermöglicht.

In der vorliegenden Arbeit wird erstmalig die systematische Realisierung eines effizienten numerischen Waveletverfahrens für ein elliptisches Kontrollproblem vorgestellt. Der Waveletansatz wird hier gezielt modifiziert und erweitert, um auf vereinheitlichte Weise die auf vielerlei Ebenen auftretenden prinzipiellen numerischen Schwierigkeiten zu bewältigen. So werden zur Vorkonditionierung elliptischer Operatoren, zur schnellen und getreuen numerischen Auswertung von Sobolevnormen, bei der Entwicklung eines optimalen geschachtelten iterativen Lösungsverfahrens und der Einbindung eines adaptiven Diskretisierungsansatzes neue Beiträge geleistet und aufeinander abgestimmt. Aus diesen Elementen wird schließlich ein Algorithmus synthetisiert und implementiert, der optimale lineare Komplexität aufweist. Dieser wird anhand einer Vielzahl numerischer Beispiele eingehend studiert.

Das untersuchte Kontrollproblem hat die folgende Form: Minimiere das Zielfunktional

$$J(y,u) = \frac{1}{2}\|Ty - y_*\|^2_{H^s} + \frac{\omega}{2}\|u\|^2_{(H^t)'} \tag{J}$$

unter Beachtung der Nebenbedingung

$$\begin{aligned} (-\Delta + 1)y &= f + Eu \qquad && \text{in } \Omega\,, \\ \frac{\partial y}{\partial n} &= 0 && \text{auf } \partial\Omega\,. \end{aligned}$$

Hier seien f und y_* vorgegebene Daten, y bezeichne die Zustandsvariable und u die Kontrollvariable auf einem Gebiet $\Omega \subset \mathbb{R}^n$. Zur Messung von Zustand und Kontrolle sind Sobolevräume reeller Glattheit zugelassen, parametrisiert durch $s, t \in [0, 1]$. Dieses sogenannte linear-quadratische elliptische Kontrollproblem bildet die Basis für viele allgemeinere Fragestellungen, beispielsweise in Bezug auf nichtlineare und

zeitabhängige PDG oder zusätzliche Beschränkungen von Zustand oder Kontrolle in Form von punktweisen Ungleichungen. Jedoch mangelt es selbst für dieses fundamentale Problem an systematischen Studien zum Einsatz effizienter numerischer Verfahren.

Der Grund liegt vor allem in der zusätzlichen Komplexität des Optimierungsansatzes gegenüber der Lösung einer einzelnen PDG. Die schnelle numerische Simulation einer elliptischen PDG führt bereits auf entscheidende Fragen nach der Wahl eines geeigneten iterativen Lösungsverfahrens und eines optimalen Vorkonditionierungsschemas. Diese übertragen sich nun in besonderem Maße auf das Kontrollproblem, da hier die Lösung der PDG-Nebenbedingung vielfach wiederholt als Unterproblem auftritt.

Darüberhinaus eröffnen sich weitere Problemfelder, für die ebenfalls geeignete Lösungen gefordert sind und in dieser Arbeit auch aufgezeigt werden. Zunächst ist die Modellierung zu untersuchen, um die Optimalität eines Zustandes flexibel und gleichzeitig möglichst genau definieren zu können. Die Optimalitätsbedingungen führen stets zu einem System von gekoppelten partiellen Differentialgleichungen, das eine spezielle Sattelpunktstruktur aufweist. Dies läßt sich nicht direkt mit Standarditerationsverfahren behandeln, so daß ein angepaßter Löser zu entwickeln ist. Weiterhin treten im Optimalitätssystem zusätzlich zum Zustand y die Kontrolle u und die adjungierte Variable p auf. Die entsprechend erhöhte Anzahl von Freiheitsgraden gegenüber der einzelnen PDG und das Auftreten unterschiedlicher Singularitäten in diesen drei Variablen motivieren ganz besonders den Einsatz adaptiver Verfahren.

Wavelets als fortschrittliches Werkzeug der numerischen Analysis zeichnen sich dadurch aus, daß sie Anlagen zur Lösung aller oben angesprochenen Schwierigkeiten in sich bergen. Sie beruhen auf einem rigorosen mathematischen Fundament in Gestalt der nichtlinearen Approximationstheorie und bieten trotz einer nichttrivialen Konstruktion große Flexibilität in der Praxis. Jedoch ist dieser Ansatz noch vergleichsweise jung, die Entwicklung ist lebhaft, und es gibt kaum vorgezeichnete Wege für die tatsächliche Realisierung waveletbasierter numerischer Verfahren.

Wavelets sind nach der hier verwendeten Definition lokale Riesz-Basen für Sobolevräume. Gegenüber Finite-Elemente-Diskretisierungen findet ein Paradigmenwechsel statt in dem Sinne, daß das zu lösende Problem zunächst durch eine unendlich-dimensionale Darstellung von Waveletkoeffizienten äquivalent ausgedrückt wird. Bei dieser Transformation geht keinerlei Information verloren, und sämtliche Untersuchungen finden im Unendlichdimensionalen statt, von der Konstruktion von Wavelets über Vorkonditionierung und die Auswertung von Normen bis hin zum expliziten Entwurf eines idealen Lösers inklusive aller Fehlerschranken und Abbruchkriterien.

Im Rahmen dieser Arbeit werden an allen wichtigen Stationen des Entwicklungszyklus' Optimierungen oder Neukonzeptionen vorgestellt mit dem Ziel, für das Gesamtproblem einen effizienten Algorithmus von optimaler linearer Komplexität und mit optimierten Konstanten zu entwickeln. Dies beginnt mit der Konstruktion geeigneter biorthogonaler Waveletbasen mit kompaktem Träger sowohl auf der primalen als auch auf der dualen Seite. Am Beispiel von Finite-Elemente- und B-Spline-Wavelets werden Transformationen vorgeschlagen, die die Kondition der Basis verbessern, die Anzahl notwendiger arithmetischer Operationen reduzieren und zu besseren Symmetrieeigenschaften führen. Von zentraler Bedeutung ist die Eigenschaft von Wavelets, per Konstruktion eine asymptotisch optimale Vorkonditionierung elliptischer PDG zu gestatten. Die absoluten Konditionszahlen lassen sich durch eine hier vorgestellte Technik noch um ein Vielfaches weiter verbessern, was zu einer deutlichen Verkürzung der Programmlaufzeiten führt.

Weiter ermöglicht der Waveletansatz die Behandlung von Sobolevnormen im Kontrollfunktional (J), die ganzzahlige oder fraktionale Glattheit im positiven und/oder negativen Bereich aufweisen. Finite-Elemente-Techniken erlauben hier nur ganzzahlige Parameter s, t. Zusätzlich zum üblichen Regularisierungsparameter ω, der eine globale Gewichtung zwischen dem Datenfit für den Zustand und der Stärke der Kontrolle regelt, eröffnet dies die Möglichkeit, auf die Form der Funktionen einzuwirken. Insbesondere läßt sich damit die Ausprägung von Singularitäten der Zielzustände und Kontrollen kontinuierlich beeinflussen. Um dies auch algorithmisch korrekt realisieren zu können, wird in dieser Arbeit eine neuartige waveletbasierte Konstruktion von Rieszoperatoren vorgestellt. Diese erlaubt auf vereinheitlichte Weise die schnelle numerische Auswertung von Sobolevnormen, die für ganzzahlige (auch negative) Normen und für beliebige Normen konstanter Funktionen exakt ist und die allgemeineren Fälle äquivalent interpoliert.

Ausgehend von einer Reformulierung des Optimalitätssystems über die Kontrollvariable u wird systematisch ein optimierter Algorithmus für das Kontrollproblem entwickelt. Die Kernidee besteht in der numerischen Lösung des Systems von partiellen Differentialgleichungen durch verzahnte innere und äußere iterative Verfahren, die jeweils auf der Methode der konjugierten Gradienten beruhen. Das äußere Verfahren ist dabei als inexakt anzusehen. Das Zusammenspiel dieser beiden Lösungsebenen verlangt eine genaue Abschätzung der Toleranzen und Abbruchkriterien aller beteiligten Unterroutinen. Die Kombination der Wavelet-Vorkonditionierung mit dem Prinzip der verschachtelten Iteration („nested iteration“) stellt schließlich sicher, daß der Algorithmus die asymptotisch optimale lineare Komplexität aufweist.

Die Implementierung wird durch das Programmpaket `BWP` realisiert, das im Rahmen dieser Arbeit konzipiert und in der Sprache `C` erstellt wurde. Zunächst werden mit einer uniformen Diskretisierung großskalige Berechnungen in ein bis zu drei Raumdimensionen durchgeführt, wobei praktisch bis zu zwei Millionen Unbekannte pro Funktionsvariable verwendet werden. Numerisch wird verifiziert, daß die im Rahmen dieser Arbeit entwickelten Optimierungen zu einer erheblichen Reduktion der absoluten Iterationszahlen führen. Diese sind levelunabhängig konstant, was die theoretisch vorhergesagte optimale Komplexität bestätigt. Es werden umfangreiche Studien zu verschiedenen Parametern s, t, ω und Daten f, y_* durchgeführt. Das Konvergenzverhalten erweist sich auch für nichtzulässige Kombinationen als robust.

In einer Simulation sind Zustand, Adjungierte und Kontrolle oft unterschiedlich glatt und besitzen auch unterschiedliche singuläre Stellen. Eine adaptive Waveletdiskretisierung erlaubt es hier, den verschiedenen Variablen unterschiedliche Auflösungsmuster zuzuweisen. Dies ist eine hervorstechende Eigenschaft gegenüber Finite-Elemente-Methoden, die in der Regel mit einem Gitter für alle Variablen arbeiten. Zudem existieren für adaptive Finite-Elemente-Ansätze für Kontrollprobleme bisher keine Aussagen zu Konvergenz oder Konvergenzraten. In dieser Arbeit wird der Ansatz vorgeschlagen, bestimmte Routinen, die als Bestandteile bereits existierender adaptiver Waveletverfahren konzipiert wurden, in das oben beschriebene zweistufige Iterationsschema zu integrieren. Dessen Toleranzen werden dazu auf die durch Vergröberungsoperationen und approximative Matrixanwendungen auftretenden zusätzlichen Approximationsfehler abgestimmt. So entsteht ein neuer Typ von Waveletverfahren, der Elemente uniformer und adaptiver Diskretisierungsstrategien enthält. Dieses wird ebenfalls im Rahmen des Programmpakets `BWP` umgesetzt. Anhand umfangreicher numerischer Experimente läßt sich experimentell beobachten, wie sich die lokale Verteilung der Koeffizienten automatisch gemäß den Regularitätseigenschaften der Daten für jede Variable individuell einstellt. Die asymptotisch levelunabhängige Anzahl an Iterationen wird wie schon beim uniformen auch beim adaptiven Verfahren beobachtet. Zudem finden sich Hinweise auf eine superlineare Konvergenzrate.

Insgesamt wird in der vorliegenden Arbeit ein weiter Bogen gespannt, um auf Basis einer optimierten Waveletkonstruktion ein effizientes zweistufiges Iterationsverfahren zur numerischen Lösung des Kontrollproblems zu konzipieren und umzusetzen. Dieses kombiniert einen nested-iteration Ansatz mit inexakten konjugierten Gradientenverfahren. Der rigorose theoretische Hintergrund von Waveletmethoden wird dabei zur Gewinnung von Aussagen zu Modellierung, Vorkonditionierung, Adaptivität und Konvergenz herangezogen. Anhand extensiver numerischer Experimente wird verifiziert, daß die benötigte Rechenleistung linear mit der Anzahl der Unbekannten skaliert und der hier vorgestellte Algorithmus somit die theoretisch vorhergesagte optimale lineare Komplexität bietet. Die Freiheit in der Modellierung durch die Einführung der waveletspezifischen Parameter s und t und deren neuentwickelte algorithmische Realisierung schlägt sich in einer großen Vielfalt der numerisch berechenbaren und berechneten Ergebnisse nieder. Zusätzlich wird eine deutliche Reduktion an verwendeten Koeffizienten durch die adaptive Weiterentwicklung des Verfahrens beobachtet. Abschließend läßt sich feststellen, daß die hier vorgestellte zielgerichtete Anpassung, Verbindung und Erweiterung der auf verschiedenen Ebenen gegebenen inhärenten Vorzüge einer Waveletdiskretisierung auf ein numerisches Verfahren von ganz eigener Art und überzeugender Flexibilität, Effizienz und Leistungsfähigkeit führt.

Inhaltsverzeichnis

1 Introduction 1

I Wavelet Methods for Linear Elliptic Partial Differential Equations 9

2 About Wavelets 11
2.1 Introduction 11
2.2 Theoretical Concepts 12
2.2.1 Multiresolution 12
2.2.2 Biorthogonality 15
2.2.3 Riesz Bases for Sobolev Spaces 17
2.3 Construction Principles 19
2.3.1 Stable Completions 20
2.3.2 Change of Bases 22
2.3.3 Multivariate Wavelet Bases 23

3 Two Constructions on the Interval 27
3.1 Introduction 27
3.2 Finite Element Wavelets 28
3.2.1 The Basic Setting 29
3.2.2 Construction of the Single-Scale Basis 30
3.2.3 A Standard Stable Completion 33
3.2.4 Improvements to the Stable Completion 35
3.2.5 Transformation to the Nodal Basis 41
3.3 Spline Wavelets 42
3.3.1 A Biorthogonal B-Spline Multiresolution on the Interval 42
3.3.2 Identifying a Stable Completion 48
3.3.3 Construction Details and Modifications 51
3.3.4 Plotting of Representations in the Dual Basis 52
3.3.5 Transformation to the Nodal Basis 58
3.3.6 Numerical Properties 61

4 Wavelet Methods for Linear Elliptic Partial Differential Equations 65
4.1 Introduction 65
4.2 Numerical Solution of Elliptic Boundary Value Problems 66
4.2.1 Weak Formulation 66
4.2.2 A-priori Estimates 68
4.3 A Wavelet Method for Elliptic Problems 69
4.3.1 A Wavelet Galerkin Method 69
4.3.2 Fast Solution over Finite-Dimensional Subspaces 71

4.3.3 Additional Preconditioning 75
4.3.4 Pre- and Postprocessing 77

II Fast Solution of a PDE-Constrained Optimal Control Problem in Wavelet Discretisation 79

5 A Linear-Quadratic Elliptic Optimal Control Problem in Wavelet Coordinates 81
5.1 Introduction 81
5.2 Mathematical Formulation 83
5.2.1 Definitions and Notation 83
5.2.2 Example Problems 85
5.3 Numerical Evaluation of Sobolev Norms 87
5.3.1 Evaluation of Norms on L_2 and H^1 88
5.3.2 Evaluation of Dual Norms 89
5.3.3 Evaluation of Fractional Sobolev Norms 90
5.3.4 Normalisation by Means of Constant Functions 92
5.4 Wavelet Representation 95
5.5 Optimality Conditions 97
5.5.1 Existence and Uniqueness 97
5.5.2 Formulation of the Optimality System 98

6 A Fast Wavelet Algorithm for the Control Problem 103
6.1 Introduction 103
6.2 An Inexact Conjugate Gradient Method 105
6.2.1 Basic Algorithm and Error Bounds 106
6.2.2 Inexact Outer Iterations 110
6.2.3 Nested Iteration 111
6.3 Numerical Results 113
6.3.1 Conforming examples 113
6.3.2 Further Characterisation of the Method 120
6.3.3 Higher Dimensions 130

7 An Adaptive Wavelet Method for the Control Problem 139
7.1 Introduction 139
7.2 Nonlinear Wavelet Approximation 140
7.2.1 Weak ℓ_τ Spaces 141
7.2.2 Connection to Besov Spaces 142
7.2.3 Coarsening Strategy 143
7.2.4 Application of Quasi-Sparse Matrices 143
7.3 An Adaptive Algorithm for the Control Problem 146
7.3.1 Properties of the Original Richardson Algorithm 146
7.3.2 An Adaptive Conjugate Gradient Method for the Control Problem 147
7.4 Numerical Results 150
7.4.1 The Solution of a Single Elliptic System 151
7.4.2 Results for the Optimal Control Problem 163

8 Conclusion and Outlook 183

List of Symbols 187

Bibliography 191

Chapter 1

Introduction

In this thesis, we develop a wavelet method for the numerical solution of an optimal control problem constrained by a linear elliptic partial differential equation. The particular challenge here lies in considering and combining two areas of research. On the one hand, we have to deal with the efficient solution of an elliptic partial differential equation. On the other, we face an optimisation problem specified by a target functional and PDE constraints.

Wavelets constitute a sophisticated tool for analysis and numerics, which provides key features for both of the above subjects. Consequently employing the unique properties of an optimised wavelet discretisation, a synergy emerges which allows us to develop a fully specified fast iterative solution scheme of optimal computational complexity. Let us now comment step by step on the relevant mathematical ideas.

Elliptic Partial Differential Equations

A large variety of phenomena in physics, engineering and mathematical finance are described by partial differential equations or shortly PDEs. Famous examples are the Navier Stokes equation for general fluid flow, which is reasonably accurate for water, or the heat equation which accounts for the distribution of temperature over time in a solid medium, e.g. metal. The deformations of diverse elastic substances such as glass, steel or plastic are modelled by a partial differential equation, as well as some relations in the pricing of option derivatives in the stock market. There exists an abundance of further examples where processes of nature can be modelled with this class of equations.

A partial differential equation describes the dependence of the state of a system on exterior forces. The classical case is Laplace's equation complemented by homogeneous Dirichlet boundary conditions,

$$-\Delta y = f \qquad \text{in } \Omega\,, \tag{1.1a}$$

$$y = 0 \qquad \text{on } \partial\Omega\,. \tag{1.1b}$$

Here y and f are functions of $x \in \Omega \subset \mathbb{R}^n$, where y denotes the unknown solution to be computed, while f represents the data which is given a priori. Physically speaking, the state y is a function of the forces or sources denoted by f. The partial differential operator Δ is defined as

$$\Delta y = \sum_{i=1}^{n} \frac{\partial^2 y}{\partial x_i^2}\,. \tag{1.2}$$

Equation (1.1) is the prototype of an elliptic boundary value problem. It is stationary, that is, independent of time.

The mathematical discipline of numerical analysis is largely concerned with the development and study of algorithms for problems of continuous mathematics. As such, it covers the numerical solution of partial differential equations. Its history begins hundreds of years ago, far before the development of mathematical automata and computing machines. Now, the use of modern computers allows to handle quantities of mathematical information larger than ever imagined by humans at the early times, whereas many central mathematical and algorithmic concepts have basically remained unchanged for centuries. On the other hand, the availability of computers has tremendously inspired the mathematical community and led to the development of a variety of recent numerical methods for a multitude of problems.

To handle equations such as (1.1) numerically, the continuous functions y and f have to be replaced by a finite amount of mathematical information. This process is called discretisation, where major techniques are finite differences or finite elements, see e.g. [22,36]. In these approaches, the domain Ω is divided into a finite number of cells, where a constant amount of coefficients per cell locally characterises the function. Naturally, this can only be an approximation. The discretisation error can be reduced by increasing the number of cells while shrinking their size. A high resolution is thus necessary to achieve a sufficiently realistic and physically meaningful simulation.

One straightforward way to discretise a partial differential equation is to employ a linear combination of the type $y = \sum_{i=1}^{N} c_i \phi_i$ with unknown coefficients $c_i \in \mathbb{R}$ and a set of functions $\phi_i : \Omega \to \mathbb{R}$. Thereby, a linear PDE is directly translated into a linear system of equations over the space of coefficients. For nonlinear partial differential equations however, several problems arise. First of all, the superposition principle no longer holds and consequently the conversion into a discrete system becomes more difficult. Furthermore, existence and uniqueness of the solution need not be assured at all. For example, global existence is unknown for the Navier Stokes equations in three dimensions.

Independent of the respective type of equation, we define the primary goal of numerical efficiency as follows. First choose a discretisation with a total number of N coefficients which delivers a discretisation error ϵ_{d} that is appropriate in the context of the problem. This selects the order of approximation d, given by the relation $\epsilon_{\mathrm{d}} = \mathcal{O}(N^{-d})$. Then calculate the discrete solution with minimal computational complexity, that is, use a possibly large N with a possibly small demand of computer memory and computing time. Since all coefficients have to be evaluated at least once, the optimal complexity is $\mathcal{O}(N)$.

Discretisations of linear elliptic partial differential equations generally lead to large linear systems of equations. The system or stiffness matrix of elliptic problems has a special sparse structure which depends on the discretisation, see e.g. [37]. The study of appropriate solvers is a large field, where direct and iterative methods are distinguished. Direct solvers yield an exact solution but usually have a complexity of at least $\mathcal{O}(N^2)$. Only iterative solvers can achieve optimal linear complexity. They deliver the solution up to a prescribed error ϵ_{s} in a certain number of k steps, $\epsilon_{\mathrm{s}} = \mathcal{O}(\rho^k)$, with $\rho < 1$.

The convergence rate ρ of iterative solvers depends directly on the so-called condition number $\kappa(\mathbf{A})$ of the system matrix $\mathbf{A}$, that is, the ratio of the largest and smallest eigenvalue. Optimal complexity can be guaranteed for a uniform condition number $\kappa(\mathbf{A}) = \mathcal{O}(1)$, independent of the amount of unknowns N. However, naive approaches for the discretisation of Laplace's operator yield a condition number which grows exponentially with N. Therefore, a large effort of research has been dedicated to techniques for preconditioning. The practically most successful concept today are multi-level methods.

By introducing a multi-level hierarchy of space decompositions, which separates the errors for different frequencies, solvers for linear elliptic PDEs can be designed which guarantee optimal complexity. The class of multigrid methods appeared first [85], and a large amount of literature exists on this topic, see e.g. [22, 23, 26, 85, 138]. The idea of multi-level preconditioners has been examined and refined in many ways, see e.g. [74, 144] and [24, 47, 102].

To further increase the numerical performance, additional important concepts have been developed, namely adaptivity and parallelisation. Firstly, when a function consists of both smooth and rough parts, it may make sense to represent it adaptively. This means to spend most coefficients for the rough part, while the smooth part is approximated accurately with very few coefficients. Consequently, a smaller number of total coefficients is required for the same error of approximation than with a standard uniform discretisation, and the computational cost can be drastically reduced depending on the smoothness of the function.

Secondly, parallelisation denotes the distribution of the computational work among several identical computers. In general, this requires non-trivial algorithmic concepts for the division of the data and the communication between the computers. Ideally, the amount of coefficients to be handled, but also the financial and environmental demands increase proportionally to the number of processors. Several industrial applications naturally call for an amount of memory which necessitates the use of many thousands of machines, and as such motivate the development of parallel algorithms. A parallel multigrid PDE solver is for example described in [78]. Here, we concentrate on the reduction of the runtime and/or the improvement of the accuracy of a numerical method for a given amount of computer memory and a fixed core frequency. To this end, we focus on an adaptive strategy to eliminate unnecessary degrees of freedom. An additional discussion of parallelisation would be beyond the scope of this work and may be considered at a later stage. —

This thesis deals with wavelet methods for the solution of control problems constrained by linear elliptic partial differential equations [28]. We choose a biorthogonal wavelet discretisation [49] precisely because of its strong virtues with respect to the numerical treatment of such PDEs. Wavelets are local multi-scale bases of functions which satisfy the Riesz basis property for a range of Sobolev spaces. This implies that the stiffness matrix for Laplace's equation is uniformly well-conditioned, and an iterative solver like the method of conjugate gradients converges with optimal complexity [47]. Furthermore, wavelets provide a solid theoretical fundament in nonlinear approximation theory [59]. They are thus particularly suited to implement adaptivity, and permit a rigorous convergence theory not only for elliptic partial differential equations [39], but also for the optimal control problem with elliptic PDE constraints [48].

Although the available literature on wavelet methods has grown considerably in the last decade, wavelets can still be considered as a novel, advanced and progressive tool for the numerical solution of PDEs. There is no such thing as a single best or standard wavelet basis, and the development of wavelet approaches to various problems in mathematics and computer science is rapid. Therefore, we need to adapt the practical construction of suitable wavelet bases and the choice of appropriate solvers specifically for the type of application discussed in this thesis. Consequently, we discuss the subject of wavelets in some detail. After the introduction of the notation and properties of wavelet bases in Chapter 2, we treat two concrete constructions of wavelets explicitly in Chapter 3. For both of them, we propose additional transformations which optimise important quantities such as the size of their support, the symmetry and the condition number of the wavelet basis. The application to elliptic PDEs is covered in Chapter 4. Here we develop a technique to further improve the condition number of the stiffness matrix in wavelet representation by an adaption to the elliptic operator, and conclude with a fast and optimal iterative solution scheme.

Optimal Control Problems

The subject of optimisation is an active field at the front of modern research. It deals with techniques to find one element out of an admissible set such that a so-called objective or cost functional is minimised. Thereby the admissible set is generally determined by constraints in the form of systems of equalities or inequalities.

A primary motivation to examine this type of problem is to reduce the cost in manufacturing and maintenance of industrial systems. Consider for example the following applications.

- Optimise the shape of tubes and valves in industrial production plants to increase the throughput of flow and the efficiency of chemical reactions [87, 113].
- Employ the injection or suction of air at the surface of aeroplane wings to inhibit drag and turbulence and consequently reduce fuel consumption.
- Control heating and cooling processes to enhance material properties in steel or glass production [64, 137].

In all of the above cases, the underlying physical system is described by a partial differential equation. The control variables enter in the form of unknown boundary conditions or exterior forces. It is generally necessary to repeatedly solve the PDE for different sets of control variables to find the optimal combination. Optimisation is thus a complex process which involves the solution of the PDE constraint as a subproblem.

We concentrate here on the class of linear-quadratic elliptic optimal control problems. This means that the state obeys a linear partial differential equation of elliptic type similar to (1.1). Specifically, we consider the prototype

$$(-\Delta + 1)y = f + Eu \qquad \text{in } \Omega\,, \tag{1.3a}$$

$$\frac{\partial y}{\partial n} = 0 \qquad \text{on } \partial\Omega\,. \tag{1.3b}$$

In this situation, the control u enters the state equation on the right hand side. The extension operator E is used to map u from the control space into the space of permissible sources f. It is thus possible to implement both Neumann boundary and distributed control, i.e., u may be defined on $\partial\Omega$ or Ω, respectively.

We examine the following type of cost functional,

$$J(y, u) = \frac{1}{2}\|Ty - y_*\|_Z^2 + \frac{\omega}{2}\|u\|_U^2\,, \tag{1.4}$$

where norms on the spaces Z and U will be specified later. The first term is of *tracking type*, which means that we wish to obtain a state y which is close to a pre-defined target state y_*. The operator T is introduced to allow an observation of y on general submanifolds or traces. The second term is often called *regularisation* with the parameter $\omega > 0$. The functional only contains quadratic contributions. In particular, this implies that it is differentiable, which is useful to derive appropriate criteria for minimisation.

The solution of optimal control problems with linear PDE constraints poses several major challenges in addition to the above stated difficulties arising from the solution of one partial differential equation alone. First of all, the necessary conditions for a minimisation of (1.4) under the constraints (1.3) lead to a coupled system of linear systems of equations, the so-called optimality system. Additional unknowns such as the control and the Lagrangian multiplier for the constraint enter the picture, leading to a threefold increase in the amount of degrees of freedom. Moreover, the optimality system has a peculiar saddle point structure, which means that the development of an optimal solver is a non-trivial task. Also the fast and accurate evaluation of suitable norms on the spaces Z and U comes into play. For maximal freedom of modelling, it is desirable to employ Sobolev norms of arbitrary real smoothness indices.

Nonlinearities generally give rise to additional issues with respect to discretisation and optimisation, such as convexity, global uniqueness of the minimiser and well-posedness of the problem. In this situation, additional techniques such as SQP methods are required [89,92,142]. An introduction is provided in [136], covering also Karush-Kuhn-Tucker-theory in function spaces and regularity issues. An extension of this thesis to nonlinear stationary constraints may be possible on the basis of [41]. Since nonlinear PDE constraints are often handled by linearisation at some point, the linear-quadratic case constitutes the core problem for many more general applications. Still, even for this fundamental case systematic numerical experiments lack behind theoretical investigations.

An overview on modern PDE-constrained optimisation is given in [11]. The optimal control of fluid flow problems is maybe the most important area today. See [80] for a general survey, which also covers different technical approaches via the optimality system, the analysis of sensitivities or adjoint-based methods. The discussion of algorithmic concepts is very active, see e.g. [124, 126, 135]. Some applications to the Navier Stokes equations are discussed in [83, 95], and reaction diffusion systems have been treated in [17].

Other attempts are concerned with novel ways to increase the numerical efficiency, for example model reduction by proper orthogonal decomposition [65, 101, 142], checkpointing techniques [79] which reduce the requirements on computer memory, and algorithmic differentiation [44]. Adaptive finite element approaches have been suggested using residual-based error estimators [108] and a framework based on duality [8]. However, these do currently offer neither a convergence theory nor complexity estimates.

The multi-level ansatz becomes increasingly popular for the numerical solution of optimal control problems as well. Multigrid methods for elliptic constraints on uniformly refined grids have been proposed early in [84]. Recently, multigrid optimisation for selected applications has been discussed [63], and specific multigrid algorithms for elliptic or parabolic constraints have been developed [15, 16]. Lately, wavelet methods have entered the scene [48, 103, 104]. —

We demonstrate in this thesis that a wavelet ansatz for the linear-quadratic optimal control problem allows for a systematic approach and a unified solution to all of the numerical difficulties mentioned above. By a reformulation of the optimality system in terms of the control u as the principal variable, we obtain a uniformly well-conditioned system of equations. Its iterative solution contains in each step the numerical inversion of the stiffness matrix as a subproblem which is also uniformly well-conditioned. By utilising the method of conjugate gradients in conjunction with a nested iteration strategy for this two-layer approach, we obtain discretisation error accuracy with optimal computational complexity.

We also propose a concept to handle the numerical evaluation of integral and fractional Sobolev norms in the wavelet framework. This allows for the modelling of the whole range of smoothness between -1 and 1 in a continuous manner. To consider negative norms is for example motivated by goal-oriented approaches, which employ distributional formulations. Technically, Sobolev norms of functions are computed by the evaluation of Riesz operators in wavelet discretisation. For fractional smoothness, this is possible up to equivalence. We propose an enhanced version here which provides equality of the original Sobolev norm and its discrete form for a larger class of functions than the standard wavelet approaches in [46, 48].

Furthermore we exploit the inherent potential of wavelet bases to implement adaptivity. The Riesz basis property can be employed to choose the most significant coefficients, discarding small contributions in a controlled fashion. As opposed to finite element techniques, it is not necessary to track the refinement of triangular or quadrilateral grids, since all information lies exclusively in the index and magnitude of the individual coefficients. In the adaptive context, the notion of asymptotically optimal complexity is interpreted in the sense that the number of unknowns N to achieve a particular error ϵ_{a} depends on the smoothness of the exact solution $\mathbf{u}$ measured in weak ℓ_τ sequence spaces,

$$N \leq C\epsilon_{\mathrm{a}}^{-1/\sigma} \|\mathbf{u}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma}, \tag{1.5}$$

with the convergence rate $\sigma > 0$ [39]. Here we incorporate elements from the adaptive algorithm proposed in [48] into our solver for the optimal control problem, and conduct numerical experiments to estimate the rate.

The definition of the optimal control problem and its transformation into wavelet coordinates is covered in Chapter 5. We discuss the reformulation of the optimality system and introduce the concept for the fast and accurate evaluation of Sobolev norms. In Chapters 6 and 7, we cover in detail the algorithm `n𝕀cG/2` using uniform refinement, and its adaptive enhancement δ-`An𝕀cG/2`, respectively. We compute the necessary error bounds and stopping criteria and provide a variety of numerical results.

Scope of this Thesis

Wavelets are a sophisticated tool in numerical analysis with enormous potential for both partial differential equations and optimisation problems. We unify the specific capabilities of wavelets for preconditioning elliptic operators, the evaluation of norms, fast iterative solvers on uniform grids and the realisation of adaptivity in one large scale programming framework for the efficient numerical solution of linear-quadratic elliptic optimal control problems with distributed or Neumann boundary control.

Technically, the discretisation is based on biorthogonal B-spline wavelets which we have optimised with respect to symmetry and conditioning. We introduce an enhanced concept to evaluate integral and fractional Sobolev norms of functions in wavelet discretisation, and propose and implement a novel inner-outer nested iteration conjugate gradient solver which computes the solution up to discretisation error with optimal computational complexity.

The implementation has been realised through the programme framework `BWP` written from scratch in pure `C`. `BWP` has been employed for the computation of all sets of transformation matrices, condition numbers and other constants and thresholds, and of course to create the main programmes implementing the algorithms `n𝕀cG/2` and δ-`An𝕀cG/2`. Since all operations require a computational effort linear in the number of unknowns, the size of the problems to be covered is only limited by the availability of main memory.

This work constitutes the first systematic realisation of a fast and accurate wavelet method for an optimal control problem constrained by linear elliptic PDEs. New concepts have been introduced with respect to modelling, discretisation and implementation. The numerical experiments suggest that the proposed optimised method is competitive with finite element approaches in terms of efficiency on simple domains, and additionally offers a superior way to model norms of arbitrary smoothness and a natural adaptive concept.

Outline

We now give a short summary of the chapters of this document. We structured them as self-consistent as possible. Naturally, later chapters will make use of important definitions and results stated earlier, but they usually do not require an especially detailed retrospection.

Chapter 2

First of all, we establish the basic properties of biorthogonal wavelet bases. This chapter is intended for anyone not familiar with wavelets and to introduce notation which will be used throughout this document.

Chapter 3

This chapter is dedicated to two different approaches to construct biorthogonal wavelet bases on the interval. These are covered in some detail, and we also add a couple of improvements by the author. This chapter may be skipped by anyone not interested in details concerning the construction, as the subjects discussed in the following chapters can generally be understood in terms of the abstract framework from Chapter 2.

Chapter 4

Discussing the solution of linear elliptic operator equations in wavelet discretisation, we derive the fundamental result that such systems are uniformly well-conditioned. We also develop a technique to improve the absolute values of the condition number of the stiffness matrix, and propose a generic nested iteration conjugate gradient solver.

Chapter 5

This chapter contains the definition of the full optimal control problem and its reformulation in terms of wavelet coordinates. To assure that the minimisers of the continuous and the discrete functional coincide for a possibly large variety of settings, we propose a framework for the accurate evaluation of norms on Sobolev spaces of integral and fractional smoothness.

Chapter 6

In this chapter we specify a numerical method for the solution of the optimal control problem and derive the necessary error bounds for several calculations and subroutines. We conclude with various numerical examples for a uniform discretisation in one to three dimensions. We display the solutions for the state and the control and corresponding convergence histories which confirm that the computational work is proportional to the number of unknowns.

Chapter 7

The last chapter is devoted to the development of an adaptive algorithm for the optimal control problem. We first summarise some results from nonlinear approximation theory and then describe how to incorporate the corresponding additional subroutines into the existing algorithm. We close the chapter with numerical results.

Chapter 8

Finally, we provide a summary and interpretation of the observations and results accumulated in this thesis, and outline perspectives for future research.

Implementation

All computations have been performed within the newly contrived programme framework `BWP`, which is an acronym for *Burstedde's Biorthogonal Wavelet Project*. It provides a framework for general biorthogonal wavelets with a strict focus on linear computational complexity. So far, it has been employed for the numerical solution of linear-elliptic optimal control problems and a full weak space-time discretisation of parabolic partial differential equations. The package is written in pure `C` and has been designed to be self-consistent, non-redundant and modular. `BWP` comprises the following three structural layers.

General computations

This lowest layer realises generic vector and matrix operations and the assembling and indexing of tensor products in arbitrary dimensions. Furthermore, it contains routines for the n-dimensional integration and interpolation of functions, iterative solvers and forward and inverse power methods to compute largest and smallest eigenvalues. It also provides capabilities for numerical file input and output to interface with `gnuplot` and `octave`.

Wavelet toolbox

All operations which are specific to biorthogonal wavelets are concentrated here within a generic multi-level framework. While I have included an implementation of the B-spline wavelets described in Section 3.3, other wavelet families can be easily integrated by providing corresponding boundary information and the primal and dual wavelet transforms. Moreover, this layer contains the exact and approximate application of operators in wavelet representation, different variants of diagonal scalings and hooks for the implementation of additional preconditioning techniques. It also realises the concept of nested iteration and encapsulates several repeatedly used subalgorithms. Besides, it offers an export filter for Postscript graphics.

Execution layer

The topmost tier of `BWP` comprises auxiliary programmes for the computation of operator bounds and condition numbers, and, of course, the main programmes in uniform and adaptive wavelet discretisation, respectively. All of them make use of the extensive abstraction framework offered by the two lower layers to avoid the overlap of code.

The `C` package uses the `autoconf/automake` framework. It is complemented by a collection of `Perl` programmes for the analysis and postprocessing of the results.

Part I

Wavelet Methods for Linear Elliptic Partial Differential Equations

Chapter 2

About Wavelets

2.1 Introduction

This thesis is concerned with a wavelet method for optimal control problems constrained by an elliptic partial differential equation. Since there exists a variety of different notions of wavelets, and the scope of applications covers many areas including signal processing and image compression, we begin with a specification of what we understand by wavelets here.

A wavelet basis Ψ for a Hilbert space H with inner product $(\cdot,\cdot)_H$ and induced norm $\|\cdot\|_H := (\cdot,\cdot)_H^{1/2}$ is a collection of functions,

$$\Psi := \{\psi_\lambda : \lambda \in I\!I\} \subset H\,, \tag{2.1.1}$$

with some characteristic properties. First of all, the indices $\lambda \in I\!I$ contain information about both the *scale* or *frequency* of each function, abbreviated as $j := |\lambda|$, and the *location*, which is denoted by k in the simplest case. For multivariate bases, additional information such as a type of wavelet may also be included in the index. The following three requirements are essential for the applications in Chapters 4 through 7.

Riesz basis property. The wavelets form a Riesz basis for H, which means that there exists a norm equivalence,

$$\|v\|_H \sim \|\mathbf{v}\|_{\ell_2} \qquad \text{for all} \qquad H \ni v = \mathbf{v}^T\Psi := \sum_\lambda \mathbf{v}_\lambda \psi_\lambda\,. \tag{R}$$

Here we have interpreted the expansion coefficients $\mathbf{v} = \{\mathbf{v}_\lambda\}_\lambda$ and the collection of wavelet functions Ψ as column vectors, since we always assume that the indices from $I\!I = \{\lambda\}$ are ordered in a fixed manner. The coefficient vectors are measured in the space ℓ_2 of all square summable sequences with norm $\|\mathbf{v}\|_{\ell_2} := (\sum_\lambda \mathbf{v}_\lambda^2)^{1/2}$. The relation $a \sim b$ means that $ca \leq b \leq Ca$ with constants c, C which are independent of any parameters on which a or b may depend. Likewise, we introduce the notation $a \lesssim b$ for $a \leq Cb$, and $a \gtrsim b$ is naturally defined as $b \lesssim a$.

Specifically, we will employ wavelets which satisfy the Riesz basis property for Sobolev spaces $H := H^s := H^s(\Omega)$, $\Omega \subset \mathbb{R}^n$, for a whole range of smoothness $s \in (-\tilde{\gamma}, \gamma) \subset \mathbb{R}$, with γ, $\tilde{\gamma} > 0$. This includes negative order spaces, which we define by the dual $H^{-s} := (H^s)'$.

Locality. The wavelets have *compact support*,

$$\operatorname{diam}\operatorname{supp}(\psi_\lambda) \lesssim 2^{-|\lambda|}\,. \tag{L}$$

It is often said that wavelets are localised in space *and* frequency at the same time. This distinguishes them from Fourier decompositions, which satisfy (R) but violate (L), or classical finite element bases on uniform grids, which satisfy (L) but do not fulfil (R).

Cancellation Property. The integration of a function v against a wavelet annihilates the smooth part of the function,

$$|\langle v, \psi_\lambda \rangle| \lesssim 2^{-|\lambda|(\tilde{d}+\frac{n}{2})} \|v\|_{W^{\tilde{d}}_\infty(\operatorname{supp}\psi_\lambda)}\,, \tag{CP}$$

where $\tilde{d}$ is the order of vanishing moments.

2.2 Theoretical Concepts

There exist by now very different constructions of wavelets [42, 54, 55, 68], and various extensions and modifications for particular cases [53, 61, 107]. Introductions and surveys are available [56, 59, 112], and also material which focuses on the application to partial differential equations [2, 38, 46]. However, constructions which are optimised with respect to practical use in numerical analysis lack behind. In particular, the constants implied in (R) directly influence the spectral condition numbers of stiffness matrices. Therefore, we will propose improvements to existing constructions, which necessitates to present the basic mathematical ideas in detail.

2.2.1 Multiresolution

The construction of wavelets can be based on the concept of a *multiresolution analysis.* This has been introduced in [111], see also [55, 112]. We will collect some basic facts here, largely based on [46, 54, 128].

Definition 2.1. *Let H be a Hilbert space over the domain $\Omega \subset \mathbb{R}^n$ with inner product $(\cdot,\cdot)_H$ and norm $\|\cdot\|_H := (\cdot,\cdot)_H^{1/2}$. A* multiresolution sequence $\mathcal{S} = \{S_j\}_{j \ge j_0}$ *is a set of nested closed subspaces $\mathcal{S}_j \subset H$ with the following properties,*

$$S_{j_0} \subset S_{j_0+1} \subset \ldots \subset H\,, \qquad \operatorname{clos}_H \left(\bigcup_{j \ge j_0} S_j \right) = H\,, \tag{2.2.1}$$

where $j_0 \in \mathbb{Z}$ denotes the coarsest level of resolution.

Introducing the notation

$$S(\Phi) := \operatorname{clos}_H(\operatorname{span}\Phi) \tag{2.2.2}$$

for any countable collection of functions $\Phi \subset H$, the subspaces S_j typically have the form

$$S_j = S(\Phi_j) \qquad \text{with} \qquad \Phi_j := \{\phi_{j,k} : k \in \Delta_j\}\,, \tag{2.2.3}$$

where $\phi_{j,k}$ are suitable basis functions for S_j over corresponding (possibly infinite) index sets Δ_j with cardinality

$$N_j := \#\Delta_j\,. \tag{2.2.4}$$

Multivariate wavelets for dimensions $n \ge 2$ require a slightly more sophisticated structure of the indices in Δ_j. To keep notation as transparent as possible, we omit this generalisation here and refer to Section 2.3.3 for details.

The *generator bases* or *single-scale bases* $\{\Phi_j\}_{j \ge j_0}$ defined in (2.2.3) are chosen to be *uniformly stable* in the sense that

$$\|\mathbf{c}\|_{\ell_2(\Delta_j)} \sim \|\mathbf{c}^T \Phi_j\|_H \tag{2.2.5}$$

for any coefficient vector $\mathbf{c} \in \ell_2(\Delta_j)$. In the following, we will drop the subscript on norms over the sequence spaces ℓ_2, simply writing $\|\cdot\|$, unless we wish to emphasise the choice of the index set. This applies equally to vectors and matrices. Furthermore, we only consider dyadic refinement, that is, the number of basis functions grows geometrically between two levels,

$$N_j \sim 2^{nj} \,. \tag{2.2.6}$$

For numerical purposes, it is important that the basis functions satisfy the *locality condition*

$$\text{diam supp}(\phi_{j,k}) \sim 2^{-j} \,. \tag{2.2.7}$$

As the spaces S_j are nested, the generator functions of adjacent scales obey the *two-scale relations*

$$\Phi_j^T = \Phi_{j+1}^T \mathbf{M}_{j,0} \,, \tag{2.2.8}$$

which define certain $N_{j+1} \times N_j$ *refinement matrices* $\mathbf{M}_{j,0} : \ell_2(\Delta_j) \to \ell_2(\Delta_{j+1})$. As a consequence of locality (2.2.7), the number of entries per row and column is uniformly bounded by a constant. It follows that the matrices $\mathbf{M}_{j,0}$ are *uniformly sparse*, that is, the number of nonzero entries is proportional to the number of unknowns.

The multiresolution framework presented up to this point applies equally well to standard finite element bases. In the following, we move on to introduce wavelets as *multi-scale* bases which provide explicit representations of *complement spaces*. To this end, we infer from (2.2.1) that there exist sequences $\Psi_j \coloneqq \{\psi_{j,k} : k \in \nabla_j\}$ such that

$$S(\Phi_{j+1}) = S(\Phi_j) \oplus S(\Psi_j) \,, \tag{2.2.9}$$

where the direct sum $\oplus$ is not necessarily meant to be orthogonal. This relation trivially implies $\#\Delta_{j+1} = (\#\Delta_j) + (\#\nabla_j)$. For multivariate wavelets ($n \geq 2$), the indices from ∇_j have a slightly more general form, just as those from Δ_j (see Section 2.3.3). With

$$W_j \coloneqq S(\Psi_j) \tag{2.2.10}$$

and (2.2.3), the relation (2.2.9) can be abbreviated as

$$S_{j+1} = S_j \oplus W_j \,. \tag{2.2.11}$$

The *complement basis* Ψ_j is usually selected such that the collections $\Phi_j \cup \Psi_j$ are uniformly stable in the sense of (2.2.5) for all $j \geq j_0$. It follows further that there exist matrices $\mathbf{M}_{j,1} : \ell_2(\nabla_j) \to \ell_2(\Delta_{j+1})$ satisfying

$$\Psi_j^T = \Phi_{j+1}^T \mathbf{M}_{j,1} \,. \tag{2.2.12}$$

It is important for us that the basis Ψ_j also has compact support analogous to (2.2.7), which causes $\mathbf{M}_{j,1}$ to be uniformly sparse as well. The complete two-scale relations now read

$$(\Phi_j^T, \Psi_j^T) = \Phi_{j+1}^T \mathbf{M}_j \qquad \text{with} \qquad \mathbf{M}_j \coloneqq (\mathbf{M}_{j,0}, \mathbf{M}_{j,1}) : \ell_2(\Delta_{j+1}) \to \ell_2(\Delta_{j+1}) \,. \tag{2.2.13}$$

From (2.2.9) it follows that $\mathbf{M}_j$ is invertible. Moreover, $\Phi_j \cup \Psi_j$ is uniformly stable if and only if

$$\|\mathbf{M}_j\| \sim 1 \,, \qquad \|\mathbf{M}_j^{-1}\| \sim 1 \,. \tag{2.2.14}$$

Definition 2.2. *Any matrix $\mathbf{M}_{j,1}$ for which the completed matrix $\mathbf{M}_j$ satisfies the relations* (2.2.14) *will be called a* stable completion *of $\mathbf{M}_{j,0}$.*

Iterating the two-scale transformations up to a specific highest level J, we decompose the space S_J according to

$$S(\Phi_J) = S(\Phi_{j_0}) \bigoplus_{j=j_0}^{J-1} S(\Psi_j) \,. \tag{2.2.15}$$

Thus, the *multi-scale basis* $\Psi_{(J)}$ for S_J, as opposed to the single-scale basis Φ_J, can be assembled from the generator basis at the coarsest level and all intermediate complement bases,

$$\Psi_{(J)} = \Phi_{j_0} \bigcup_{j=j_0}^{J-1} \Psi_j \,. \tag{2.2.16}$$

As the union of subspaces S_j is dense in H, a basis for the full space H is given by

$$\Psi \coloneqq \Phi_{j_0} \bigcup_{j=j_0}^{\infty} \Psi_j \,. \tag{2.2.17}$$

Defining $\Psi_{j_0-1} \coloneqq \Phi_{j_0}$ and $\nabla_{j_0-1} \coloneqq \Delta_{j_0}$, (2.2.17) can be conveniently abbreviated

$$\Psi = \bigcup_{j=j_0-1}^{\infty} \Psi_j \,. \tag{2.2.18}$$

Naturally, we wish to find a transformation matrix $\mathbf{W}_j : \ell_2(\Delta_J) \to \ell_2(\Delta_J)$ between the single-scale basis Φ_J and the multi-scale basis $\Psi_{(J)}$,

$$\Psi_{(J)}^T = \Phi_J^T \mathbf{W}_J \,. \tag{2.2.19}$$

Such a matrix can indeed be obtained by iterating the two-scale transformations (2.2.13). Consequently, the resulting *multi-scale transformation* $\mathbf{W}_J$ is composed of two-scale operators,

$$\mathbf{W}_J \coloneqq \mathbf{W}_{J,J-1} \cdots \mathbf{W}_{J,j_0} \qquad \text{with} \qquad \mathbf{W}_{J,j} \coloneqq \begin{pmatrix} \mathbf{M}_j & 0 \\ 0 & \mathbf{I} \end{pmatrix} . \tag{2.2.20}$$

It can be used to switch between single-scale and multi-scale representations of functions. To this end, let the expansion coefficients of a function $v \in S_J$ in the generator basis be denoted by $\mathbf{c}$, and the coefficients in the multi-scale basis by $\mathbf{d}$, i.e.,

$$v = \mathbf{c}^T \Phi_J = \mathbf{d}^T \Psi_{(J)} \,. \tag{2.2.21}$$

The coefficient vector $\mathbf{d} \coloneqq (\mathbf{c}_{j_0}^T, \mathbf{d}_{j_0}^T, \ldots, \mathbf{d}_{J-1}^T)^T$ contains contributions from all scales. The first part holds the coefficients of the generator basis on the coarsest level which provides a rough approximation of v. The following coefficients are ordered by scale and add successive layers of *detail information* to the approximation. Inserting (2.2.19) into (2.2.21), the transformation from the detail coefficients to the single-scale coefficients emerges as

$$\mathbf{c} = \mathbf{W}_J \mathbf{d} \,. \tag{2.2.22}$$

The multi-scale transformation $\mathbf{W}_J$ can be applied to a vector $\mathbf{d}$ by successive applications of the two-level transforms (2.2.20). This is visualised in the following *pyramid scheme*,

$$\begin{tikzcd}
\mathbf{c}_{j_0} \arrow[r, "\mathbf{M}_{j_0,0}", "\mathbf{M}_{j_0,1}"'] & \mathbf{c}_{j_0+1} \arrow[r, "\mathbf{M}_{j_0+1,0}", "\mathbf{M}_{j_0+1,1}"'] & \mathbf{c}_{j_0+2} \arrow[r] & \cdots \arrow[r, "\mathbf{M}_{J-1,0}", "\mathbf{M}_{J-1,1}"'] & \mathbf{c}_J \\
\mathbf{d}_{j_0} \arrow[ur] & \mathbf{d}_{j_0+1} \arrow[ur] & \mathbf{d}_{j_0+2} \arrow[ur, "\cdots"'] & \mathbf{d}_{J-1} \arrow[ur] &
\end{tikzcd} , \tag{2.2.23}$$

where $\mathbf{c} = \mathbf{c}_J$ is the result. If each matrix $\mathbf{M}_{j,e}$ can be applied in $\mathcal{O}(N_j)$ operations, it follows by a geometric series argument using (2.2.6) that the complete multi-scale transformation can be performed in $\mathcal{O}(N_J)$ operations, i.e., with *linear complexity.* The application of $\mathbf{W}_J$ is therefore called *fast wavelet transform*, FWT.

It is important for us to find a uniformly stable multi-scale basis where the inverse multi-scale transformation can also be applied in linear time. Let the inverse two-scale transformation matrices be defined by

$$\begin{pmatrix} \mathbf{G}_{j,0} \\ \mathbf{G}_{j,1} \end{pmatrix} := \mathbf{G}_j := \mathbf{M}_j^{-1} \,. \tag{2.2.24}$$

The blocking is performed according to the splitting (2.2.9). The inversion of $\mathbf{W}_J$ can be understood through (2.2.20) and arranged in a pyramid scheme analogously to (2.2.23),

$$\begin{tikzcd}
\mathbf{c}_J \arrow[r, "\mathbf{G}_{J-1,0}"] \arrow[dr, "\mathbf{G}_{J-1,1}"'] & \mathbf{c}_{J-1} \arrow[r, "\mathbf{G}_{J-2,0}"] \arrow[dr, "\mathbf{G}_{J-2,1}"'] & \mathbf{c}_{J-2} \arrow[r] \arrow[dr] & \cdots \arrow[r, "\mathbf{G}_{j_0,0}"] \arrow[dr, "\mathbf{G}_{j_0,1}"'] & \mathbf{c}_{j_0} \\
 & \mathbf{d}_{J-1} & \mathbf{d}_{J-2} & \cdots & \mathbf{d}_{j_0}
\end{tikzcd} \,. \tag{2.2.25}$$

It follows that $\mathbf{W}_J^{-1}$ can be applied in linear time if the matrices $\mathbf{G}_j$ are uniformly sparse. In this case, $\mathbf{W}_J^{-1}$ is called *inverse fast wavelet transform*, IFWT. To achieve the sparsity of $\mathbf{G}_j$ is a crucial point in both constructions of wavelets which we present in Chapter 3.

2.2.2 Biorthogonality

The concept of multiresolution analysis introduced in the previous section will now be expanded by duality. This leads to *biorthogonal space decompositions* and biorthogonal wavelet bases, which offer more flexibility for the envisaged applications than strictly orthogonal wavelets.

To emphasise the role of duality, we will write $\langle v, \tilde{v} \rangle$ for the dual pairing of two functions $v \in H$, $\tilde{v} \in H'$, where H denotes a general Hilbert space with dual H'. (For the special case $H = L_2$, this reduces to the standard inner product.) For collections of functions $\Phi = \{\phi\} \subset H$, which we consistently interpret as (possibly infinite) column vectors, and some single function $f \in H'$, the term $\langle \Phi, f \rangle$ is to be understood as a column vector, and $\langle f, \Phi \rangle$ as a row vector, according to

$$\langle \Phi, f \rangle := \big(\langle \phi, f \rangle \big)_{\phi \in \Phi} \,, \qquad \langle f, \Phi \rangle := \langle \Phi, f \rangle^T \,. \tag{2.2.26}$$

Consequently, by-element dual pairings of two function sequences $\Phi \subset H$ and $\tilde{\Phi} = \{\tilde{\phi}\} \subset H'$ are written as (not necessarily square and possibly infinite) matrices $\langle \Phi, \tilde{\Phi} \rangle$ with entries

$$\langle \Phi, \tilde{\Phi} \rangle := \big(\langle \phi, \tilde{\phi} \rangle \big)_{\phi \in \Phi, \tilde{\phi} \in \tilde{\Phi}} \,. \tag{2.2.27}$$

It follows that linear transformations of two function sequences Φ, $\tilde{\Phi}$ by (not necessarily square) matrices $\mathbf{A}$ and $\mathbf{B}$ satisfy

$$\langle \mathbf{A}\Phi, \mathbf{B}\tilde{\Phi} \rangle = \mathbf{A} \langle \Phi, \tilde{\Phi} \rangle \mathbf{B}^T \,. \tag{2.2.28}$$

Recall that we denote by H^s the Sobolev space of (possibly fractional) index of smoothness $s \in \mathbb{R}$ over a domain $\Omega \subset \mathbb{R}^n$, and abbreviate its dual as $H^{-s} := (H^s)'$ [1]. Using this notation, we formulate the following theorem and corollary which contain the fundamental statements about biorthogonal space decompositions, see e.g. [45, 49, 54].

Theorem 2.3. *Let $S_{j_0} \subset S_{j_0+1} \subset \ldots$ and $\tilde{S}_{j_0} \subset \tilde{S}_{j_0+1} \subset \ldots$ be two sequences of nested subspaces of L_2.*

(a) Suppose that

$$\inf_{0 \neq v \in S_j} \sup_{0 \neq \tilde{v} \in \tilde{S}_j} \frac{|\langle v, \tilde{v} \rangle|}{\|v\|_{L_2} \|\tilde{v}\|_{L_2}} \gtrsim 1 \,, \tag{2.2.29}$$

and the analogous condition (2.2.29)* *with interchanged roles of S_j and $\tilde{S}_j$ hold.*

Then there exists a sequence $\{Q_j\}_j$ *of uniformly bounded projectors* $Q_j : L_2 \to L_2$ *with* $\mathrm{Im}(Q_j) = S_j$ *and* $\mathrm{Im}(I - Q_j) = (\tilde{S}_j)^{\perp_{L_2}}$*. For the adjoint projectors* $(\tilde{Q}_j)_j$ *holds analogously* $\mathrm{Im}(\tilde{Q}_j) = \tilde{S}_j$ *and* $\mathrm{Im}(I - \tilde{Q}_j) = (S_j)^{\perp_{L_2}}$*. Additionally, it follows that* $Q_j Q_{j+1} = Q_j$ *and* $\tilde{Q}_j \tilde{Q}_{j+1} = \tilde{Q}_j$*, respectively.*

(b) Under the extra assumptions that there exists $0 < \gamma < d$ *such that*

$$\inf_{w \in S_j} \|v - w\|_{L_2} \lesssim 2^{-js} \|v\|_{H^s} \tag{2.2.30}$$

for $v \in H^s$, $0 \le s \le d$ *(which is usually called a* direct *or* Jackson estimate*) and*

$$\|v\|_{H^s} \lesssim 2^{js} \|v\|_{L_2} \tag{2.2.31}$$

for $v \in S_j$, $0 \le s \le \gamma$ *(which is usually referred to as* inverse *or* Bernstein *estimate), and that analogous assumptions* (2.2.30)*, (2.2.31)* *with constants* $0 < \tilde{\gamma} < \tilde{d}$ *hold for* $\{\tilde{S}_j\}_j$*, it follows (with the definitions* $Q_{j_0-1} := \tilde{Q}_{j_0-1} := 0$*) that*

$$\Big\| \sum_{j=j_0}^{\infty} v_j \Big\|_{H^s}^2 \lesssim \sum_{j=j_0}^{\infty} 2^{2js} \|v_j\|_{L_2}^2 \qquad \textit{for} \qquad v_j \in \mathrm{Im}(Q_j - Q_{j-1}), s \in (-\tilde{d}, \gamma)\,, \tag{2.2.32}$$

and

$$\sum_{j=j_0}^{\infty} 2^{2js} \|(Q_j - Q_{j-1})v\|_{L_2}^2 \lesssim \|v\|_{H^s}^2 \qquad \textit{for} \qquad v \in H^s, s \in (-\tilde{\gamma}, d)\,. \tag{2.2.33}$$

Corollary 2.4. *In particular, for* $s \in (-\tilde{\gamma}, \gamma)$, $v \mapsto ((Q_j - Q_{j-1})v)_j$ *is a bounded mapping from* H^s *into* $\ell_{2,s}(Q)$*, where*

$$\ell_{2,s}(Q) := \Big\{ (v_j)_j : v_j \in \mathrm{Im}(Q_j - Q_{j-1}), \|(v_j)_j\|_{\ell_{2,s}(Q)} := \Big(\sum_{j=0}^{\infty} 2^{2js} \|v_j\|_{L_2}^2 \Big)^{\frac{1}{2}} < \infty \Big\}\,, \tag{2.2.34}$$

with bounded inverse $(v_j)_j \mapsto \sum_{j=0}^{\infty} v_j$*. Therefore the symbols* $\lesssim$ *in* (2.2.32) *and* (2.2.33) *can be replaced by* $\sim$ *symbols in this case. This leads to the* norm equivalence

$$\|v\|_{H^s}^2 \sim \sum_{j=j_0}^{\infty} 2^{2js} \|(Q_j - Q_{j-1})v\|_{L_2}^2\,, \qquad v \in H^s, s \in (-\tilde{\gamma}, \gamma)\,. \tag{2.2.35}$$

Analogous results hold for interchanged roles of (γ, d) *and* $(\tilde{\gamma}, \tilde{d})$*, with* Q_j *replaced by* $\tilde{Q}_j$*.*

These statements about projectors are connected to the multiresolution sequences introduced in the previous section, with $H = L_2$, by the following remark.

Remark 2.5. *We can identify* $W_j = \mathrm{Im}(Q_{j+1} - Q_j)$*, cf.* (2.2.11)*, and its dual counterpart* $\tilde{W}_j = \mathrm{Im}(\tilde{Q}_{j+1} - \tilde{Q}_j)$*. Consequently, we get*

$$W_j = (\tilde{S}_j)^{\perp_{L_2}} \cap S_{j+1} \qquad \textit{and} \qquad \tilde{W}_j = (S_j)^{\perp_{L_2}} \cap \tilde{S}_{j+1}\,. \tag{2.2.36}$$

It follows trivially that the W_j *and* $\tilde{W}_j$ *are biorthogonal between levels,*

$$W_{j_1} \perp \tilde{W}_{j_2} \qquad \textit{for} \qquad j_1 \neq j_2\,. \tag{2.2.37}$$

Therefore, the collection of spaces W_j, $\tilde{W}_j$ is called a biorthogonal decomposition of L_2. This includes the spaces $W_{j_0-1} := S_{j_0}$ and $\tilde{W}_{j_0-1} := \tilde{S}_{j_0}$. The functions in W_j and $\tilde{W}_j$ will be called *primal* and *dual wavelets*.

The prerequisites for Theorem 2.3 can be verified by the following considerations. When the spaces S_j and $\tilde{S}_j$ are chosen as finite element spaces, direct and inverse estimates (2.2.30), (2.2.31) are standard facts, see e.g. [22, 54, 117]. For functions which are dyadically refined (2.2.6) and local (2.2.7), the inverse estimates (2.2.31) are satisfied with $\gamma = t + \frac{3}{2}$ and $\tilde{\gamma} = \tilde{t} + \frac{3}{2}$, provided that the functions from the spaces S_j and $\tilde{S}_j$ are piecewise smooth and globally t, respectively $\tilde{t}$, times continuously differentiable. For the case of discontinuous finite elements, this holds with $t = \tilde{t} = -1$. The direct estimates (2.2.30) are usually enforced by the requirement that, relative to their meshes, S_j and $\tilde{S}_j$ contain all t or $\tilde{t}$ times continuously differentiable piecewise polynomials of degree $d-1$ and $\tilde{d}-1$, respectively. Considering (2.2.37), this also implies that the wavelets have $\tilde{d}$, respectively d, vanishing moments,

$$\langle v, (\cdot)^{\tilde{r}} \rangle = 0\,, \qquad v \in W_j\,, \qquad \tilde{r} = 0, \ldots, \tilde{d}-1\,, \tag{2.2.38a}$$

$$\langle \tilde{v}, (\cdot)^{r} \rangle = 0\,, \qquad \tilde{v} \in \tilde{W}_j\,, \qquad r = 0, \ldots, d-1\,. \tag{2.2.38b}$$

Up to this point, we have dealt with statements about spaces S_j, W_j and general projectors Q_j, and their dual counterparts. Now we will provide a means to verify condition (2.2.29) by the choice of appropriate bases. We first cite the following general lemma [54].

Lemma 2.6. *Let Φ_j and $\tilde{\Phi}_j$ be L_2-stable bases of S_j and $\tilde{S}_j$, respectively, with the common index set Δ_j. Define the square matrix $\mathbf{B}_j : \ell_2(\Delta_j) \to \ell_2(\Delta_j)$ according to*

$$\mathbf{B}_j := \left(\frac{\langle \phi_{j,k}, \tilde{\phi}_{j,l} \rangle}{\|\phi_{j,k}\|_{L_2} \|\tilde{\phi}_{j,l}\|_{L_2}} \right)_{k,l \in \Delta_j} . \tag{2.2.39}$$

Then (2.2.29) is equivalent to $\|\mathbf{B}_j \mathbf{c}_j\| \gtrsim \|\mathbf{c}_j\|$ for all $\mathbf{c}_j = (c_{j,k})_{k \in \Delta_j} \in \ell_2(\Delta_j)$, and analogously (2.2.29)* is equivalent to $\|\mathbf{B}_j^T \mathbf{c}_j\| \gtrsim \|\mathbf{c}_j\|$ for all $\mathbf{c}_j$.

When we demand in addition that Φ_j and $\tilde{\Phi}_j$ are biorthogonal,

$$\langle \Phi_j, \tilde{\Phi}_j \rangle = \mathbf{I}\,, \qquad \mathbf{I}_{k,l} := \delta_{k,l} := \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{else,} \end{cases} \tag{2.2.40}$$

we can establish the following theorem which also provides concrete representations of the biorthogonal projectors [29].

Theorem 2.7. *Let Φ_j and $\tilde{\Phi}_j$ be biorthogonal, uniformly stable bases of the nested spaces S_j and $\tilde{S}_j$, respectively. Then, (2.2.29) is satisfied, and the projectors defined by*

$$Q_j v := \langle v, \tilde{\Phi}_j \rangle \Phi_j\,, \qquad \tilde{Q}_j \tilde{v} := \langle \tilde{v}, \Phi_j \rangle \tilde{\Phi}_j \tag{2.2.41}$$

fulfil the conditions from Theorem 2.3.

Summarising these results, we can ensure the norm equivalence (2.2.35) by the choice of biorthogonal bases for the spaces S_j and $\tilde{S}_j$ which are uniformly stable and local and yield polynomial exactness of orders d and $\tilde{d}$, respectively. We will discuss two independent constructions of bases which fulfil these requirements in Chapter 3.

2.2.3 Riesz Bases for Sobolev Spaces

We will show here how wavelets as defined in Section 2.2.1 can be used to substitute the differences $(Q_j - Q_{j-1})v$ from (2.2.35). This will in turn yield the Riesz basis property (R).

Definition 2.8. *A collection of functions* $\Sigma \subset H$ *is called a Riesz system for* H *if*

$$\|v\|_H = \|\mathbf{c}^T\Sigma\|_H \sim \|\mathbf{c}\|\,, \qquad v = \mathbf{c}^T\Sigma \in H\,. \tag{2.2.42}$$

A Riesz system is called a Riesz basis if it is also a basis of H*. In other words, the norm of a function* $v = \mathbf{c}^T\Sigma \in H$ *can be estimated from below and above by the sequence norm of its expansion coefficients* $\mathbf{c}$*.*

We have already encountered this form of equivalence in (2.2.5), which means that Φ_j forms a Riesz basis for S_j. Here, we will extend the Riesz basis property to the full infinite multi-scale basis Ψ. To this end, we make the additional assumption that the bases for the complement spaces $\Psi_j \subset W_j$ and $\tilde{\Psi}_j \subset \tilde{W}_j$ as defined in (2.2.10) are biorthogonal,

$$\langle \Psi_j, \tilde{\Psi}_j \rangle = \mathbf{I}\,. \tag{2.2.43}$$

We will show a recipe for such a construction in Section 2.3.1. Together with (2.2.37), we deduce biorthogonality of the full multi-scale basis Ψ, $\tilde{\Psi}$ as defined in (2.2.17),

$$\langle \Psi, \tilde{\Psi} \rangle = \mathbf{I}\,. \tag{2.2.44}$$

We can then obtain a discrete norm equivalence from (2.2.35) by replacing the projectors Q_j with their definition (2.2.41) and explicitly computing the differences $(Q_j - Q_{j-1})v$ [49].

Theorem 2.9. *Let* Ψ*,* $\tilde{\Psi}$ *be a pair of uniformly stable, biorthogonal multi-scale bases in the sense of* (2.2.14) *and* (2.2.44)*, and let the direct and inverse inequalities* (2.2.30) *and* (2.2.31) *hold. Then it follows that*

$$\|v\|_{H^s}^2 \sim \sum_{j=j_0-1}^{\infty} 2^{2js}\|\langle v, \tilde{\Psi}_j\rangle\|^2\,, \qquad v \in H^s, s \in (-\tilde{\gamma}, \gamma)\,, \tag{2.2.45}$$

and the dual relation, obtained by switching the roles of Ψ_j *and* $\tilde{\Psi}_j$*, holds for* $s \in (-\gamma, \tilde{\gamma})$*.*

By setting $s = 0$, we conclude that Ψ and $\tilde{\Psi}$ are Riesz bases for L_2, and the expansion of a function $v \in L_2$ can be written as

$$v = \langle v, \Psi\rangle\tilde{\Psi} = \langle v, \tilde{\Psi}\rangle\Psi\,, \tag{2.2.46}$$

with

$$\|v\|_{L_2} \sim \|\langle v, \Psi\rangle\| \sim \|\langle v, \tilde{\Psi}\rangle\|\,. \tag{2.2.47}$$

For general s, the formulation (2.2.45) suggests to absorb the scaling factors 2^{js} into the wavelet basis. To this end, we define

$$\Psi^s := \mathbf{D}^{-s}\Psi = \{2^{-js}\psi_{j,k}\}_{j \geq j_0-1, k \in \nabla_j}\,, \tag{2.2.48}$$

where $\mathbf{D}$ is the diagonal matrix consisting of entries 2^j, j being the level of each function. For the multivariate case, the index sets take a slightly more complicated form, see Section 2.3.3. The dual basis is defined by

$$\tilde{\Psi}^s := \mathbf{D}^s\tilde{\Psi}\,. \tag{2.2.49}$$

In view of (2.2.44), these definitions imply

$$\langle \Psi^s, \tilde{\Psi}^s \rangle = \mathbf{I}\,. \tag{2.2.50}$$

In summary, the main theorem characterising Sobolev spaces by wavelet expansions emerges as follows.

Corollary 2.10. *Under the assumptions of Theorem 2.9 and for any* $s \in (-\tilde{\gamma}, \gamma)$*,* Ψ^s *is a Riesz basis for* H^s*, and* $\tilde{\Psi}^s$ *is a Riesz basis for* H^{-s}*, namely,*

$$\|\cdot\|_{H^s} \sim \|\langle \tilde{\Psi}^s, \cdot\rangle\| \qquad \textit{and} \qquad \|\cdot\|_{H^{-s}} \sim \|\langle \Psi^s, \cdot\rangle\|\,. \tag{2.2.51}$$

Thus, wavelets can be used to characterise function spaces in terms of expansion coefficients. This will play an important role with respect to preconditioning of elliptic problems and the accurate evaluation of Sobolev norms and Riesz operators later in this document. The transformation between Sobolev spaces of different smoothness can be achieved through the definition of a shift operator as follows.

Corollary 2.11. *Let the* shift operator *be defined by*

$$D_t : H^s \to H^{s-t}\,, \qquad v = \mathbf{c}_s^T \Psi^s \mapsto D_t v := (\mathbf{D}^t \mathbf{c}_s)^T \Psi^s = \mathbf{c}_s^T \Psi^{s-t}\,. \tag{2.2.52}$$

This is a well-defined operation for s, $s-t \in (-\tilde{\gamma}, \gamma)$, $v \in H^s$, and it holds

$$\|D_t v\|_{H^{s-t}} \sim \|v\|_{H^s}\,. \tag{2.2.53}$$

Analogous statements hold for the dual basis. In any case, a positive shift t decreases the smoothness of a function, while a negative exponent smoothens a function.

To transform between the single-scale basis of L_2 and a certain wavelet basis of smoothness s, the scaling in terms of the diagonal matrix $\mathbf{D}$ can also be incorporated into the multi-scale transformation (2.2.19) according to

$$(\Psi^s_{(J)})^T = \Phi_J^T \mathbf{W}_J^s \qquad \text{with} \qquad \mathbf{W}_J^s := \mathbf{W}_J \mathbf{D}^{-s}\,. \tag{2.2.54}$$

The following result relates the conditioning of $\mathbf{W}_J^s$, which can be understood as a generalisation of (2.2.14), to the Riesz basis property for Ψ^s [46].

Theorem 2.12. *The transformations $\mathbf{W}_J^s$ are well-conditioned in the sense that*

$$\|\mathbf{W}_J^s\| \sim 1\,, \qquad \|(\mathbf{W}_J^s)^{-1}\| \sim 1\,, \tag{2.2.55}$$

if and only if Ψ^s is a Riesz basis of H^s.

As an orthogonal basis is by definition a self-dual Riesz basis, it fits into the framework described here. However, orthogonality can be considered as a requirement which is too strong. Although compactly supported orthogonal wavelet bases for the Sobolev spaces considered here have been constructed [55], they have some disadvantages which make them harder to handle in numerical simulations. In particular, they are not piecewise polynomials, and they are only implicitly defined.

The framework of biorthogonal bases offers much more flexibility to construct multi-scale bases which are explicitly defined, well-conditioned in the sense of (2.2.55), and lead to multi-scale transformations which can be applied with linear computational complexity.

In summary, biorthogonality is as far as one can deviate from orthogonality. Essential features of orthogonality are preserved: Biorthogonal wavelets satisfy the Riesz basis property for the Sobolev spaces in question, and the expansion coefficients of a function can be computed by inner products with (dual) basis functions.

2.3 Construction Principles

After the introduction of the basic theoretical concepts, we will now address some practical aspects. These will mostly be related to the construction of biorthogonal bases for the complement spaces W_j and $\tilde{W}_j$, that is, complement bases which satisfy (2.2.43). To prepare the optimisation of specific constructions of wavelets for our numerical purposes, we need to recall and further develop the construction principles in detail.

2.3.1 Stable Completions

We will show next how to construct biorthogonal two-scale bases from a biorthogonal generator basis (2.2.40) and an initial stable completion. The key ingredient is the use of biorthogonal projectors as defined in (2.2.41). This recipe contains many degrees of freedom which can be used to tune the properties of the resulting wavelet basis.

The starting point for the construction is the existence of one stable completion (2.2.12). From there, the whole family of stable completions can be constructed according to the following scheme [29].

Theorem 2.13. *Suppose that the bases* Φ_j *are uniformly stable with refinement matrices* $\mathbf{M}_{j,0}$ *and let* $\check{\mathbf{M}}_{j,1}$ *be any uniformly stable completion of* $\mathbf{M}_{j,0}$*. Let* $\check{\mathbf{G}}_j = \begin{pmatrix} \check{\mathbf{G}}_{j,0} \\ \check{\mathbf{G}}_{j,1} \end{pmatrix}$ *denote the inverse of* $\check{\mathbf{M}}_j = (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1})$*. Then* $\mathbf{M}_{j,1}$ *is also a stable completion of* $\mathbf{M}_{j,0}$ *if and only if there exist*

$$\mathbf{L}_j : \ell_2(\nabla_j) \to \ell_2(\Delta_j)\,, \qquad \mathbf{K}_j : \ell_2(\nabla_j) \to \ell_2(\nabla_j) \tag{2.3.1}$$

such that $\mathbf{L}_j$*,* $\mathbf{K}_j$ *and* $\mathbf{K}_j^{-1}$ *are uniformly bounded as operators and*

$$\mathbf{M}_{j,1} = \mathbf{M}_{j,0}\mathbf{L}_j + \check{\mathbf{M}}_{j,1}\mathbf{K}_j\,. \tag{2.3.2}$$

In this case, the inverse $\mathbf{G}_j = \begin{pmatrix} \mathbf{G}_{j,0} \\ \mathbf{G}_{j,1} \end{pmatrix}$ *of* $\mathbf{M}_j = (\mathbf{M}_{j,0}, \mathbf{M}_{j,1})$ *is given by*

$$\mathbf{G}_{j,0} = \check{\mathbf{G}}_{j,0} - \mathbf{L}_j\mathbf{K}_j^{-1}\check{\mathbf{G}}_{j,1}\,, \qquad \mathbf{G}_{j,1} = \mathbf{K}_j^{-1}\check{\mathbf{G}}_{j,1}\,. \tag{2.3.3}$$

In other words, once a stable completion $\check{\mathbf{M}}_{j,1}$ has been identified, any other possible stable completion can be obtained by a specific choice of operators $\mathbf{L}_j$ and $\mathbf{K}_j$, provided they satisfy the conditions formulated in the theorem given above. For example, the particular case $\mathbf{K}_j = \mathbf{I}$ corresponds to the *lifting scheme* introduced in [131]. We will use this freedom to build biorthogonal primal and dual complement bases Ψ_j and $\tilde{\Psi}_j$.

An *initial* stable completion $\check{\mathbf{M}}_{j,1}$ corresponds to a matching basis Ξ_j for W_j, cf. (2.2.12),

$$\Xi_j^T = \Phi_{j+1}^T\check{\mathbf{M}}_{j,1}\,. \tag{2.3.4}$$

Intuitive choices for Ξ_j are for example variants of the nodal basis. To satisfy the biorthogonality conditions (2.2.36), we can use a biorthogonal projection of the initial basis Ξ_j. Let

$$S_j = S(\Phi_j)\,, \qquad \tilde{S}_j = S(\tilde{\Phi}_j)\,, \qquad \langle \Phi_j, \tilde{\Phi}_j \rangle = \mathbf{I}\,. \tag{2.3.5}$$

Since the spaces S_j and $\tilde{S}_j$ are nested, the primal and dual refinement relations hold according to

$$\Phi_j^T = \Phi_{j+1}^T\mathbf{M}_{j,0} \qquad \text{and} \qquad \tilde{\Phi}_j^T = \tilde{\Phi}_{j+1}^T\tilde{\mathbf{M}}_{j,0}\,. \tag{2.3.6}$$

From these relations and (2.3.4), several identities between matrices and scalar products of function sequences can be derived, for example,

$$\mathbf{M}_{j,0} = \langle \tilde{\Phi}_{j+1}, \Phi_j \rangle\,, \qquad \tilde{\mathbf{M}}_{j,0} = \langle \Phi_{j+1}, \tilde{\Phi}_j \rangle\,, \qquad \check{\mathbf{M}}_{j,1} = \langle \tilde{\Phi}_{j+1}, \Xi_j \rangle\,. \tag{2.3.7}$$

It follows that if the bases Φ_j, $\tilde{\Phi}_j$ and Ξ_j are uniformly local, then the matrices $\mathbf{M}_{j,0}$, $\tilde{\mathbf{M}}_{j,0}$ and $\check{\mathbf{M}}_{j,1}$ are uniformly sparse.

To establish biorthogonality, we eliminate the part of Ξ_j which is not perpendicular to $\tilde{W}_j$, defining the projected basis Ψ_j by

$$\Psi_j := (I - Q_j)\Xi_j = \Xi_j - \langle \Xi_j, \tilde{\Phi}_j \rangle \Phi_j\,. \tag{2.3.8}$$

Inserting (2.3.4) and (2.3.6) into this equation, and using (2.2.12) and (2.3.7), the matching stable completion is identified as

$$\mathbf{M}_{j,1} = (\mathbf{I} - \mathbf{M}_{j,0}\tilde{\mathbf{M}}_{j,0}^T)\check{\mathbf{M}}_{j,1} \,. \tag{2.3.9}$$

This matrix is uniformly sparse if all matrices appearing in this expression are uniformly sparse. By comparing it to (2.3.2), we recognise that this definition is precisely an application of the lifting scheme with

$$\mathbf{L}_j = -\tilde{\mathbf{M}}_{j,0}^T\check{\mathbf{M}}_{j,1} \,, \qquad \mathbf{K}_j = \mathbf{I} \,. \tag{2.3.10}$$

Finally, if we specify the basis for the dual complement spaces $\tilde{W}_j = S(\tilde{\Psi}_j)$ as

$$\tilde{\Psi}_j^T := \tilde{\Phi}_{j+1}^T\tilde{\mathbf{M}}_{j,1} \qquad \text{with} \qquad \tilde{\mathbf{M}}_{j,1} := \check{\mathbf{G}}_{j,1}^T \,, \tag{2.3.11}$$

it follows that

$$\tilde{\mathbf{M}}_j := (\tilde{\mathbf{M}}_{j,0}, \tilde{\mathbf{M}}_{j,1}) = \mathbf{G}_j^T = \mathbf{M}_j^{-T} \,. \tag{2.3.12}$$

Under the assumptions of Theorem 2.13, $\tilde{\mathbf{M}}_{j,1}$ is guaranteed to be a stable completion of $\tilde{\mathbf{M}}_{j,0}$. By comparison to (2.3.3) we gain the additional identity

$$\tilde{\mathbf{M}}_{j,0}^T = \check{\mathbf{G}}_{j,0} + \tilde{\mathbf{M}}_{j,0}^T\check{\mathbf{M}}_{j,1}\check{\mathbf{G}}_{j,1} \,. \tag{2.3.13}$$

After these preparations, we can state that the specific choice of lifting in (2.3.10) indeed yields a biorthogonal basis.

Theorem 2.14. *The complement bases defined by* (2.3.8) *and* (2.3.11) *are biorthogonal.*

Proof. The refinement relations for the primal and dual bases can be summarised as

$$(\Phi_j^T, \Psi_j^T) = \Phi_{j+1}^T\mathbf{M}_j \,, \tag{2.3.14a}$$

$$(\tilde{\Phi}_j^T, \tilde{\Psi}_j^T) = \tilde{\Phi}_{j+1}^T\tilde{\mathbf{M}}_j \,. \tag{2.3.14b}$$

Biorthogonality then follows from (2.3.5) and (2.3.12),

$$\left\langle \begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix}, \begin{pmatrix} \tilde{\Phi}_j \\ \tilde{\Psi}_j \end{pmatrix} \right\rangle = \langle \mathbf{M}_j^T\Phi_{j+1}, \tilde{\mathbf{M}}_j^T\tilde{\Phi}_{j+1}\rangle = \mathbf{M}_j^T\tilde{\mathbf{M}}_j = \mathbf{I} \,, \tag{2.3.15}$$

which completes the proof. □

We can also conclude that the dual refinement matrix $\tilde{\mathbf{M}}_j$ is uniformly sparse if both $\tilde{\mathbf{M}}_{j,0}$ and $\check{\mathbf{G}}_{j,1}$ are uniformly sparse. While the first matrix is generally sparse by locality, to establish the sparsity of $\check{\mathbf{G}}_{j,1}$ (which is the lower half of the inverse of $\check{\mathbf{M}}_j$) is non-trivial. This problem is resolved in different ways for the two constructions of wavelets which we describe in Chapter 3.

The dual multi-scale transformation $\tilde{\mathbf{W}}_J$ is defined analogously to the primal $\mathbf{W}_J$, see (2.2.20),

$$\tilde{\mathbf{W}}_J := \tilde{\mathbf{W}}_{J,J-1}\cdots\tilde{\mathbf{W}}_{J,j_0} \qquad \text{with} \qquad \tilde{\mathbf{W}}_{J,j} := \begin{pmatrix} \tilde{\mathbf{M}}_j & 0 \\ 0 & \mathbf{I} \end{pmatrix} . \tag{2.3.16}$$

It can be applied with linear complexity when all matrices $\tilde{\mathbf{M}}_j$ are sparse. As a consequence of (2.3.12) we obtain the relation $\mathbf{W}^{-1} = \tilde{\mathbf{W}}^T$.

2.3.2 Change of Bases

The biorthogonal wavelet basis which resulted from the projection of an initial stable completion is not unique. In view of (2.2.17), which states that the multi-scale basis Ψ consists of generator functions for the coarsest space $S(\Phi_{j_0})$ and wavelets spanning the completion spaces $S(\Psi_j)$, we will propose modifications for each of these two parts, which transform one biorthogonal basis into another.

This change of bases may be used to reduce the absolute value of the condition number of the wavelet transform, or to reduce the number of nonzero entries in the transformation matrices. In both constructions of Chapter 3, we will propose suitable matrices $\mathbf{C}_j$ and $\check{\mathbf{K}}_j$ with exactly these effects.

Transformation of the Wavelets

The freedom in the choice of $\mathbf{L}_j$ and $\mathbf{K}_j$ has been used to implement the biorthogonal projection onto the spaces W_j, see (2.3.10). Using a uniformly bounded invertible matrix $\check{\mathbf{K}}_j$, we introduce the transformation

$$\mathbf{L}_j \mapsto \mathbf{L}_j\check{\mathbf{K}}_j\,, \qquad \mathbf{K}_j \mapsto \mathbf{K}_j\check{\mathbf{K}}_j\,. \tag{2.3.17}$$

This results in

$$\mathbf{M}_j \mapsto (\mathbf{M}_{j,0},\, \mathbf{M}_{j,1}\check{\mathbf{K}}_j) \qquad \text{and} \qquad \mathbf{M}_j^{-1} \mapsto \begin{pmatrix} \tilde{\mathbf{M}}_{j,0}^T \\ \check{\mathbf{K}}_j^{-1}\check{\mathbf{G}}_{j,1} \end{pmatrix}\,. \tag{2.3.18}$$

An equivalent interpretation is to change the initial stable completion,

$$\check{\mathbf{M}}_{j,1} \mapsto \check{\mathbf{M}}_{j,1}\check{\mathbf{K}}_j\,. \tag{2.3.19}$$

In order not to impair the sparseness of the whole construction, $\check{\mathbf{K}}_j$ needs to be chosen such that it and its inverse are uniformly sparse. In view of (2.3.14) and (2.3.18), the modified multi-scale basis reads

$$\Psi \mapsto \Phi_{j_0} \bigcup_{j=j_0}^{\infty} \check{\mathbf{K}}_j^T \Psi_j\,. \tag{2.3.20}$$

We have thus transformed only the wavelets, while the single-scale basis Φ_{j_0} did not change. Biorthogonality is preserved by the dual transformation

$$\tilde{\Psi} \mapsto \tilde{\Phi}_{j_0} \bigcup_{j=j_0}^{\infty} \check{\mathbf{K}}_j^{-1} \tilde{\Psi}_j\,. \tag{2.3.21}$$

The effect on the multi-scale transformation $\mathbf{W}_J$ is the following: On each level, the two-scale matrix $\mathbf{W}_{J,j}$ now contains the modified $\mathbf{M}_{j,1}$ which is multiplied from the right with the matrix $\check{\mathbf{K}}_j$.

Transformation of the Generator Basis

It is equally well possible to change the generator basis. Consider the transformation

$$\Phi_j \mapsto \mathbf{C}_j^T \Phi_j \tag{2.3.22}$$

with a uniformly bounded invertible matrix $\mathbf{C}_j$. In order to guarantee efficient numerical schemes, the application of $\mathbf{C}_j$ and $\mathbf{C}_j^{-1}$ should be possible in linear time. To ensure biorthogonality, we define analogously

$$\tilde{\Phi}_j \mapsto \tilde{\mathbf{C}}_j^T \tilde{\Phi}_j \qquad \text{with} \qquad \tilde{\mathbf{C}}_j := \mathbf{C}_j^{-T}\,. \tag{2.3.23}$$

Considering (2.3.7), (2.3.9) and using the above definition of $\tilde{\mathbf{C}}_j$, this leads to the simultaneous transformations

$$\mathbf{M}_{j,0} \mapsto \mathbf{C}_{j+1}^{-1}\mathbf{M}_{j,0}\mathbf{C}_j\,, \qquad \tilde{\mathbf{M}}_{j,0} \mapsto \tilde{\mathbf{C}}_{j+1}^{-1}\tilde{\mathbf{M}}_{j,0}\tilde{\mathbf{C}}_j \tag{2.3.24}$$

and

$$\check{\mathbf{M}}_{j,1} \mapsto \mathbf{C}_{j+1}^{-1}\check{\mathbf{M}}_{j,1}\,, \qquad \mathbf{M}_{j,1} \mapsto \mathbf{C}_{j+1}^{-1}\mathbf{M}_{j,1}\,, \qquad \tilde{\mathbf{M}}_{j,1} \mapsto \tilde{\mathbf{C}}_{j+1}^{-1}\tilde{\mathbf{M}}_{j,1}\,. \tag{2.3.25}$$

Thus, the transformed two-scale matrices are

$$\mathbf{M}_j \mapsto \mathbf{C}_{j+1}^{-1}(\mathbf{M}_{j,0}\mathbf{C}_j,\, \mathbf{M}_{j,1})\,, \qquad \tilde{\mathbf{M}}_j \mapsto \tilde{\mathbf{C}}_{j+1}^{-1}(\tilde{\mathbf{M}}_{j,0}\tilde{\mathbf{C}}_j,\, \tilde{\mathbf{M}}_{j,1})\,. \tag{2.3.26}$$

Consequently, the resulting biorthogonal multi-scale basis is given by

$$\Psi \mapsto \mathbf{C}_{j_0}^T \Phi_{j_0} \bigcup_{j=j_0}^{\infty} \Psi_j\,, \qquad \tilde{\Psi} \mapsto \tilde{\mathbf{C}}_{j_0}^T \tilde{\Phi}_{j_0} \bigcup_{j=j_0}^{\infty} \tilde{\Psi}_j\,. \tag{2.3.27}$$

We now study the effect on the wavelet transform. By inserting the new two-level matrices (2.3.26) into the multi-scale transformations $\mathbf{W}_J$ (2.2.20) and $\tilde{\mathbf{W}}_J$ (2.3.16), the matrices $\mathbf{C}_j$, $\tilde{\mathbf{C}}_j$ cancel out on all intermediate levels. It only remains to apply them on the left and right of the *complete* multi-level transform,

$$\mathbf{W}_J \mapsto \mathbf{C}_J^{-1}\mathbf{W}_J \begin{pmatrix} \mathbf{C}_0 & 0 \\ 0 & \mathbf{I} \end{pmatrix}\,, \qquad \tilde{\mathbf{W}}_J \mapsto \tilde{\mathbf{C}}_J^{-1}\tilde{\mathbf{W}}_J \begin{pmatrix} \tilde{\mathbf{C}}_0 & 0 \\ 0 & \mathbf{I} \end{pmatrix}\,. \tag{2.3.28}$$

In view of the requirements on $\mathbf{C}_j$ and its inverse, this is a cheap operation which does not interfere with the calculations on intermediate levels.

2.3.3 Multivariate Wavelet Bases

Up to this point, we have used the framework of general Hilbert spaces $H = H(\Omega)$. The central Theorem 2.3 is formulated without explicit reference to the dimension of the domain $\Omega \subset \mathbb{R}^n$. Indeed, there exist explicit constructions of multivariate wavelets on arbitrary triangulations [54,128], satisfying locality (L), the norm equivalence (R) for H^s and the cancellation property (CP). The numerical properties of one such construction have been examined in [100].

A systematic way to construct multivariate wavelets on the unit cube is by building tensor products of univariate wavelet bases on the unit interval. In particular, stability and locality of the wavelets and the Riesz basis property are inherited from the univariate case.

We begin with the one-dimensional single-scale basis Φ_j which has been introduced in (2.2.3), spanning the space $S_j = S(\Phi_j)$ over the domain $\Omega = \mathbb{R}$ or $\Omega = (0,1)$. We reuse this basis for each spatial dimension $l = 1, \ldots, n$ by defining the multivariate collection

$$\phi_{j,\mathbf{k}}(\mathbf{x}) \coloneqq \phi_{j;k_1,\ldots,k_n}(x_1,\ldots,x_n) \coloneqq \prod_{l=1}^{n} \phi_{j,k_l}(x_l)\,, \qquad \Phi_j^n \coloneqq \{\phi_{j,\mathbf{k}} : k_l \in \Delta_j\}\,. \tag{2.3.29}$$

It forms a basis of the space $S_j^n \coloneqq S(\Phi_j^n)$ over the n-dimensional domain Ω^n. The indices of the new collection belong to the set $\mathbf{k} \in \Delta_j^n \coloneqq (\Delta_j)^{\otimes n}$. The support of each basis function is approximately a hypercube (up to boundary effects).

As hinted above, there is more than one way of deriving multivariate wavelet bases from this origin, using only the building blocks from Section 2.3.1. The different approaches can be classified by the shape of the support of the resulting tensor product wavelets. We present the procedure here for the primal wavelets, as the dual wavelets are treated in perfect analogy.

Anisotropic Construction

The simplest approach directly combines the univariate wavelet bases Ψ (2.2.16) by tensor products. The resulting multivariate basis functions are then indexed by vectors in the following manner,

$$\psi_{\mathbf{j},\mathbf{k}}(\mathbf{x}) \coloneqq \prod_{l=1}^{n} \psi_{j_l,k_l}(x_l)\,, \qquad \Psi^{\mathrm{ani}} \coloneqq \{\psi_{\mathbf{j},\mathbf{k}} : j_l \geq j_0 - 1, k_l \in \nabla_{j_l}\}\,. \tag{2.3.30}$$

Here functions on different scales j_l are coupled. The combined functions generally have rectangular support, which explains the notion *anisotropic* construction. The norm equivalences (2.2.51) can be established analogously to the univariate case [70]. To this end, the definition of the Riesz basis for H^s via a diagonal matrix (2.2.48) is generalised to

$$\Psi^{\mathrm{ani},s} \coloneqq \mathbf{D}^{-s}\Psi^{\mathrm{ani}} = \{2^{-\|\mathbf{j}\|_\infty s}\psi_{\mathbf{j},\mathbf{k}}\}_{j_l \geq j_0-1, k_l \in \nabla_{j_l}}\,. \tag{2.3.31}$$

The restriction of the multivariate wavelet basis to a fixed level J is denoted by

$$\Psi^{\mathrm{ani}}_{(J)} \coloneqq \{\psi_{\mathbf{j},\mathbf{k}} : j_l = j_0 - 1, \ldots, J-1, k_l \in \nabla_{j_l}\}\,, \tag{2.3.32}$$

which can be abbreviated as

$$\Psi^{\mathrm{ani}}_{(J)} = \bigotimes_{l=1}^{n} \Psi_{(J)}\,. \tag{2.3.33}$$

Just as Φ^n_J, this is a basis for S^n_J. In strict analogy to (2.2.19), the multi-scale transformation follows as the tensor product of the univariate transformation,

$$(\Psi^{\mathrm{ani}}_{(J)})^T = (\Phi^n_J)^T \mathbf{W}^{\mathrm{ani}}_J \qquad \text{with} \qquad \mathbf{W}^{\mathrm{ani}}_J \coloneqq \bigotimes_{l=1}^{n} \mathbf{W}_J\,. \tag{2.3.34}$$

It can be expanded similarly to (2.2.20) as

$$\mathbf{W}^{\mathrm{ani}}_J = \mathbf{W}^{\mathrm{ani}}_{J,J-1} \cdots \mathbf{W}^{\mathrm{ani}}_{J,j_0} \qquad \text{with} \qquad \mathbf{W}^{\mathrm{ani}}_{J,j} \coloneqq \bigotimes_{l=1}^{n} \begin{pmatrix} \mathbf{M}_j & 0 \\ 0 & \mathbf{I} \end{pmatrix}\,. \tag{2.3.35}$$

We see that the multiplicative cascading structure is preserved, while the two-level transformations on each level are generalised to the tensor product. Therefore, additional transformations for the univariate basis in the spirit of Section 2.3.2 can be integrated into the multivariate setting without modification.

The anisotropic combination of functions provides the basis for the construction of *sparse grids* [27]. Restricting the set (2.3.32) by the additional constraint $\|\mathbf{j}\|_1 < J + n$, the number of coefficients drops to $\mathcal{O}(N_J \log(N_J)^{n-1})$. This becomes increasingly advantageous for higher spatial dimension n. On the other hand, sparse grid bases require the stronger H^s_{mix} regularity for the same order of approximation as provided by the full basis [70]. Since our main interest is the solution of an optimal control problem in moderate spatial dimension $n \leq 3$, which is formulated in terms of standard Sobolev spaces H^s, we do not investigate this approach any further here.

Isotropic Construction

A different way of combination which leads to an approximately square support of the tensor product functions is the following. Define for $e \in E \coloneqq \{0, 1\}$,

$$\psi_{j,k,e}(x) \coloneqq \begin{cases} \phi_{j,k}(x) & \text{for } e = 0,\ k \in \Delta_j, \\ \psi_{j,k}(x) & \text{for } e = 1,\ k \in \nabla_j, \end{cases} \qquad \nabla_{j,e} \coloneqq \begin{cases} \Delta_j & \text{for } e = 0, \\ \nabla_j & \text{for } e = 1. \end{cases} \tag{2.3.36}$$

For the multivariate case let $\mathbf{e} \in E^n$ and define

$$\psi_{j,\mathbf{k},\mathbf{e}}(\mathbf{x}) \coloneqq \prod_{l=1}^{n} \psi_{j,k_l,e_l}(x_l)\,. \tag{2.3.37}$$

The index $\mathbf{e}$ describes the newly introduced *type* of the wavelet. Setting $\mathbf{e} = \mathbf{0}$ identifies a composition of single-scale generators $\phi_{j,k}$ only. All other values of $\mathbf{e}$ select at least one wavelet component $\psi_{j,k}$. Thus, the types of the composite wavelets are indexed by the set $E^* := E^n \setminus \{\mathbf{0}\}$ with cardinality $2^n - 1$. For fixed j and $\mathbf{e}$, the location is indexed over

$$\mathbf{k} \in \nabla^n_{j,\mathbf{e}} \coloneqq \nabla_{j,e_1} \otimes \ldots \otimes \nabla_{j,e_n}\,. \tag{2.3.38}$$

The differences to the anisotropic construction are twofold: Firstly, the composed function carries only one scale index j, which corresponds to an approximately square support. Secondly, in contrast to generators from the coarsest scale only, generator functions $\phi_{j,k}$from all levels can occur as tensor product component.

Analogously to the univariate case, these sets of basis functions are arranged level-wise into the multi-scale basis $\Psi^{\text{iso}}_{(J)}$, cf. (2.2.16),

$$\Psi^{\text{iso}}_{(J)} \coloneqq \{\psi_{j_0,\mathbf{k},\mathbf{e}=\mathbf{0}}\} \cup \{\psi_{j_0,\mathbf{k},\mathbf{e}\in E^*}\} \cup \ldots \cup \{\psi_{J-1,\mathbf{k},\mathbf{e}\in E^*}\}\,, \tag{2.3.39}$$

which is also a basis for S^n_J. The index $\mathbf{e} = \mathbf{0}$ only occurs on the lowest level j_0. The index sets for the location $\mathbf{k}$ depend on the type $\mathbf{e}$ as in (2.3.38).

To formulate the multivariate two-level transform, we define its rectangular building blocks as

$$\mathbf{M}^n_{j,\mathbf{e}} \coloneqq \bigotimes_{l=1}^{n} \mathbf{M}_{j,e_l} \tag{2.3.40}$$

and construct the full isotropic two-level transform $\mathbf{M}^{\text{iso}}_j : \ell_2(\Delta^n_{j+1}) \to \ell_2(\Delta^n_{j+1})$ according to

$$\mathbf{M}^{\text{iso}}_j \coloneqq (\mathbf{M}^n_{j,(0,\ldots,0)}, \mathbf{M}^n_{j,(0,\ldots,0,1)}, \mathbf{M}^n_{j,(0,\ldots,0,1,0)}, \ldots, \mathbf{M}^n_{j,(1,\ldots,1)})\,. \tag{2.3.41}$$

In words, we count $\mathbf{e}$ in the binary system from 0 to $2^n - 1$ and concatenate the rectangular block matrices $\mathbf{M}^n_{j,\mathbf{e}}$. The result is the square matrix $\mathbf{M}^{\text{iso}}_j$. It is inherently n-dimensional and enters the multi-scale transformation $\mathbf{W}^{\text{iso}}_J$ directly,

$$(\Psi^{\text{iso}}_{(J)})^T = (\Phi^n_J)^T \mathbf{W}^{\text{iso}}_J \tag{2.3.42}$$

with

$$\mathbf{W}^{\text{iso}}_J \coloneqq \mathbf{W}^{\text{iso}}_{J,J-1} \cdots \mathbf{W}^{\text{iso}}_{J,j_0} \qquad \text{and} \qquad \mathbf{W}^{\text{iso}}_{J,j} \coloneqq \begin{pmatrix} \mathbf{M}^{\text{iso}}_j & 0 \\ 0 & \mathbf{I} \end{pmatrix}. \tag{2.3.43}$$

In contrast to (2.3.35), the isotropic multi-level transform is not a direct tensor product combination. Instead, its structure is parallel to the univariate case (2.2.20) since $\mathbf{M}_j$ is transparently replaced by $\mathbf{M}^{\text{iso}}_j$. Also the standard diagonal scaling $\mathbf{D} = \{2^j\}$ can be used as opposed to the more complicated form in (2.3.31).

We conclude to work with isotropic wavelets which are used in most approaches so far to apply wavelet discretisations to operator equations [46]. In view of their square support and the structure of the multi-level transformation (2.3.43), they maintain closer similarity to the univariate setting than anisotropic wavelets.

Chapter 3

Two Constructions on the Interval

3.1 Introduction

In the last chapter, we introduced the definition of biorthogonal wavelets, and described the important construction principle of stable completions. While we condensed results from many years of recent research to establish a theoretical foundation, we deliberately provided the most general formulation.

Here we will actuate this abstract framework exemplarily for two different concrete constructions of wavelet bases. Since our main objective is the development of a fast numerical algorithm based on wavelets, the constructions are on the one hand illustrative from the practical point of view. On the other hand, we incorporate several novel optimisations with respect to structure and conditioning, which exploit the theoretical framework in a more subtle way.

Specifically, we introduce the so-called *finite element wavelets* and *biorthogonal B-spline wavelets.* (We will shortly write *spline wavelets* for the latter, stressing that we do not mean the spline prewavelets from [35].) Both constructions yield compactly supported, biorthogonal wavelets on the interval $\Omega = (0, 1)$, and allow for various orders of polynomial exactness. We will include explicit construction details for primal and dual exactness of $d = 2$ and $\tilde{d} = 4$, respectively, where the primal wavelet basis Ψ consists of piecewise linear functions. This special case provides sufficient regularity for the discretisation of second order differential equations, while being most effective computationally in the sense of a small size of the support of the wavelets and possibly few nonzero coefficients in the transformation matrices.

There are two main differences between these two constructions. Firstly, the spline wavelets are translation invariant away from the boundary, while the finite elements consist of repeating blocks containing four functions each. Secondly, the dual finite element wavelets are given by an explicit functional expression as piecewise polynomials, while the dual spline wavelets are only implicitly defined by recursion formulas.

The construction of finite element wavelets is based on polynomial interpolation, and is thus rather intuitive and self-contained. The main ideas for the optimisation of the numerical efficiency and the condition numbers of wavelets which we develop in this thesis are motivated and carried out first in this context. Biorthogonal B-Spline wavelets require a more elaborate theoretical background, since they are built upon an existing multiresolution analysis in L_2, which is constructed using Fourier techniques. In view of the envisaged application to PDEs however, they are more adequate because of the translational invariance in the interior and the flexibility of boundary conditions.

We deal with different types of boundary conditions here. The term *free* or *inhomogeneous boundary conditions* applies to a basis which can represent functions with arbitrary function values at the ends of the interval, $f(0),\ f(1) \in \mathbb{R}$. Specifying *homogeneous boundary conditions* means that only functions with $f(0) = f(1) = 0$ can be represented. Since biorthogonal bases consist of a primal and a dual set of functions, both sets may conform to different boundary conditions. We will indicate this where appropriate.

Concerning the optimisations for numerical purposes, recall that wavelets have been characterised in Chapter 2 as local bases which satisfy the Riesz basis property (R) for a range of Sobolev spaces H^s. The special case for L_2 with an explicit specification of the constants reads

$$c\|\mathbf{v}\| \leq \|v\|_{L_2} \leq C\|\mathbf{v}\| \qquad \text{for} \qquad L_2 \ni v = \mathbf{v}^T \Psi \,. \tag{3.1.1}$$

While the ratio of the constants c and C, the so-called the *condition number* of the basis, is not relevant for the abstract introduction of wavelets, it is of great practical importance for applications in numerical analysis. Notably, smaller values generally lead to faster convergence of iterative algorithms.

Definition 3.1. *The condition number of the basis Ψ is defined as the ratio of the constants in* (3.1.1)*,*

$$\kappa(\Psi) := \frac{C}{c} \,. \tag{3.1.2}$$

An orthogonal basis satisfies $c = C$, which yields the optimal condition number of 1. For a biorthogonal basis it holds generally that $c < C$, which implies that $\kappa(\Psi) > 1$. To compute the condition number numerically, we insert the expansion of v into the L_2 norm,

$$\|v\|_{L_2}^2 = (v,v)_{L_2} = (\mathbf{v}^T\Psi, \mathbf{v}^T\Psi)_{L_2} = \mathbf{v}^T\mathbf{M}\mathbf{v} \qquad \text{with} \qquad \mathbf{M} := \langle\Psi,\Psi\rangle = \big(\langle\psi_\lambda,\psi_\mu\rangle\big)_{\lambda,\mu\in I\!I} \,. \tag{3.1.3}$$

The Gramian matrix $\mathbf{M}$ is called the *mass matrix* of the basis Ψ. It is symmetric and positive definite. By comparing (3.1.1) and (3.1.3), we obtain

$$\kappa(\Psi)^2 = \kappa(\mathbf{M}) := \frac{\lambda_{\max}(\mathbf{M})}{\lambda_{\min}(\mathbf{M})} \,. \tag{3.1.4}$$

Thus, the condition number of any given basis Ψ follows from the condition number of the corresponding mass matrix.

In addition to the condition number, the absolute count of arithmetic operations and the structure of a program with respect to the nesting and the fragmentation of loops determine the performance of a numerical algorithm. Therefore, we formulate the following criteria for a numerically optimised wavelet basis.

- The condition number $\kappa(\Psi)$ should be small.
- The wavelet transformation matrices $\mathbf{W}_J$ as in (2.2.19) should have few nonzero entries.
- The pattern of nonzero entries of $\mathbf{W}_J$ should be of simple structure.

To improve the following two constructions in this sense, we employ the concept of stable completions which we presented in Section 2.3.1, and use the change of bases via matrices $\mathbf{C}_j$ and $\check{\mathbf{K}}_j$ as introduced in Section 2.3.2. Finally, we obtain the optimised refinement matrices in an exact representation using rational numbers.

3.2 Finite Element Wavelets

In the following, we will cover our first example of a wavelet construction on the unit interval. We adopt the derivation from [128] and add a few proofs of our own to clarify some details of the construction and to support the development of our optimisations.

This construction comes from the family of so-called finite element wavelets [54,128]. The name is chosen due to the fact that they are defined on dyadic refinements of a reference simplex in arbitrary spatial dimensions. The construction is based on interpolating polynomials.

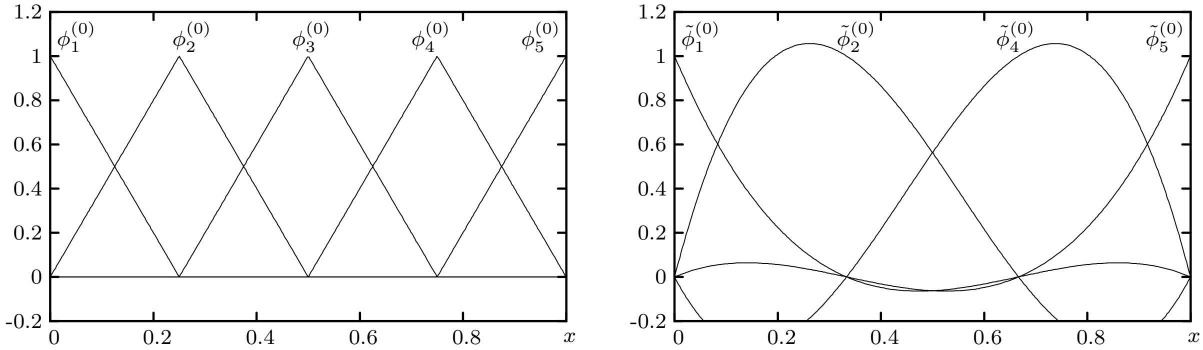


Figure 3.1: The first stage of primal and dual generators. The left picture shows a piecewise linear nodal basis for an interval which is uniformly subdivided into four parts. The right picture shows the cubic Lagrangian interpolation polynomials on the nodes $\{0, 1/3, 2/3, 1\}$.

3.2.1 The Basic Setting

The first step in this construction of biorthogonal wavelets is the design of biorthogonal generator bases, $\Phi_j \subset S_j$, $\tilde\Phi_j \subset \tilde S_j$ with $\langle \Phi_j, \tilde\Phi_j\rangle = \mathbf{I}$. We will assemble these bases from linear combinations of basic piecewise polynomial functions.

The starting point are preliminary primal and dual nodal bases on the unit interval. Let

$$\Delta^{(d,m)} \coloneqq \left\{\delta_k^{(d,m)}(x) : k \in \{0,\dots,2^m(d-1)\}\right\} \tag{3.2.1}$$

be the collection of piecewise polynomials of order d which satisfy the following interpolation property on an m-fold recursive subdivision of the unit interval,

$$\delta_k^{(d,m)}\left(\frac{2^{-m}i}{d-1}\right) = \begin{cases} 1 & \text{for } i = k\,, \\ 0 & \text{for } i \neq k\,, \end{cases} \tag{3.2.2}$$

where $i, k \in \{0,\dots,2^m(d-1)\}$. These sets induce nested function spaces $S(\Delta^{(d,m)}) \subset S(\Delta^{(d,m+1)})$, where we introduced the abbreviation $S(\Delta) \coloneqq \operatorname{span}\Delta$ for any finite set of functions Δ. We fix the lowest level $j_0 \coloneqq 2$ and define the first stages of primal and dual single-scale bases as

$$\Phi_{j_0}^{(0)} \coloneqq \{\phi_1^{(0)},\dots,\phi_5^{(0)}\} \coloneqq \Delta^{(2,2)} \qquad \text{and} \qquad \tilde\Phi_{j_0}^{(0)} \coloneqq \{\tilde\phi_1^{(0)}, \tilde\phi_2^{(0)}, \tilde\phi_4^{(0)}, \tilde\phi_5^{(0)}\} \coloneqq \Delta^{(4,0)}\,. \tag{3.2.3}$$

In words, the initial set on the primal side consists of five hat functions, and the dual set contains four cubic Lagrangian interpolation polynomials. This choice is depicted in Figure 3.1. To satisfy biorthogonality, both sets must eventually contain the same number of functions. Therefore we still need to specify the missing function $\tilde\phi_3^{(0)}$ which has been omitted in (3.2.3).

As we use the framework of multiresolution, we demand the existence of uniform refinement relations, expressed by (2.3.6). The function $\tilde\phi_3^{(0)}$ must therefore be representable by functions of the next higher level $\Delta^{(4,1)}$. At the same time, it should be zero at the boundary in order not to introduce overlap to neighbouring intervals, so we restrict the basis set to

$$\Delta_0^{(4,1)} \coloneqq \Delta^{(4,1)} \setminus \{\delta_0^{(4,1)}, \delta_6^{(4,1)}\}\,. \tag{3.2.4}$$

Additionally demanding symmetry with respect to $x = 1/2$ leads to the choice

$$\tilde\phi_3^{(0)} \in \operatorname{span}\{h_1, h_2, h_3\}\,, \tag{3.2.5}$$

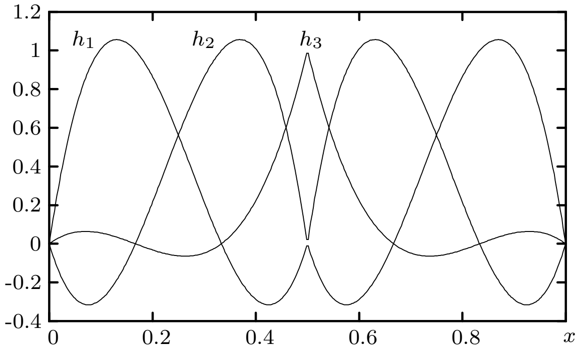


Figure 3.2: This picture shows the functions from the set defined in (3.2.6)*. All three are symmetric with respect to $x = 1/2$, which is also the point where they are not differentiable.*

with

$$h_1 \coloneqq \delta_1^{(4,1)} + \delta_5^{(4,1)}\,, \tag{3.2.6a}$$

$$h_2 \coloneqq \delta_2^{(4,1)} + \delta_4^{(4,1)}\,, \tag{3.2.6b}$$

$$h_3 \coloneqq \delta_3^{(4,1)}\,. \tag{3.2.6c}$$

These three functions are shown in Figure 3.2. Not only $\tilde{\phi}_3^{(0)}$ but the whole set of first stage basis functions is then symmetric with respect to $x = 1/2$ in the following sense,

$$\phi_i^{(0)}(x) = \phi_{6-i}^{(0)}(1-x)\,, \qquad \tilde{\phi}_i^{(0)}(x) = \tilde{\phi}_{6-i}^{(0)}(1-x)\,, \qquad i = 1, \ldots, 5\,. \tag{3.2.7}$$

In order to orthogonalise the pair of preliminary basis functions $\Phi_{j_0}^{(0)}, \tilde{\Phi}_{j_0}^{(0)}$, we perform the following steps [128].

(i) Accept $\phi_i \coloneqq \phi_i^{(0)}$ for $i = 2, 3, 4$.

(ii) Choose ϕ_1 to be orthogonal to $\tilde{\phi}_2^{(0)}$, $\tilde{\phi}_4^{(0)}$ and $\tilde{\phi}_5^{(0)}$. ϕ_5 is then defined by symmetry (3.2.7).

(iii) Choose $\tilde{\phi}_3^{(0)}$ from (3.2.5) to be orthogonal to ϕ_1. There is some freedom in this operation.

(iv) Biorthogonalise the functions of $\tilde{\Phi}_{j_0}^{(0)}$ by inversion of the matrix $\langle \Phi_{j_0}, \tilde{\Phi}_{j_0}^{(0)} \rangle$.

All transformations preserve symmetry, the uniform locality of the bases and the sparse structure of the refinement matrices. We will cover the last three steps of the construction in detail in the next section.

3.2.2 Construction of the Single-Scale Basis

We specify ϕ_1 as a linear combination of the functions $\phi_1^{(0)}$ to $\phi_5^{(0)}$. In order to preserve locality, we demand that $\phi_1(1) = 0$ and consequently exclude the fifth function with $\phi_5^{(0)}(1) \neq 0$. The freedom in normalisation is used to fix the value at the left boundary, $\phi_1(0) \coloneqq \phi_1^{(0)}(0) = 1$. Considering (i), we set

$$\phi_1 = \phi_1^{(0)} + \sum_{i=2}^{4} a_i \phi_i\,. \tag{3.2.8}$$

By taking the scalar product of this equation with functions $\tilde{\phi}_k^{(0)}$ from the dual side, and enforcing the biorthogonality condition (ii), we obtain

$$0 = \langle \phi_1, \tilde{\phi}_k^{(0)} \rangle = \langle \phi_1^{(0)}, \tilde{\phi}_k^{(0)} \rangle + \sum_{i=2}^{4} a_i \langle \phi_i, \tilde{\phi}_k^{(0)} \rangle \,, \qquad k = 2, 4, 5\,. \tag{3.2.9}$$

This leads to a 3×3 linear system which has a unique solution for (a_2, a_3, a_4), which determine ϕ_1 and ϕ_5.

It remains to find the function $\tilde{\phi}_3^{(0)}$ which is orthogonal to ϕ_1. This in turn guarantees orthogonality to ϕ_5 by symmetry. Considering (3.2.5), we need to determine the parameters b_i,

$$\tilde{\phi}_3^{(0)} = \sum_{i=1}^{3} b_i h_i \,. \tag{3.2.10}$$

Normalisation and orthogonality to ϕ_1 provide two conditions for the three unknowns b_1 to b_3. We will show in the following that no loss of information occurs by arbitrarily specifying the third condition as say $b_3 = 0$. To this end, it is sufficient to show that h_3 can be linearly combined from h_1 and h_2 and the functions from $S(\Delta_0^{4,1})$ which are already present in the dual basis, namely $\tilde{\phi}_2^{(0)}$ and $\tilde{\phi}_4^{(0)}$.

Lemma 3.2. *The set of functions* $\{q \coloneqq \tilde{\phi}_2^{(0)} + \tilde{\phi}_4^{(0)}, h_1, h_2, h_3\}$ *is linearly dependent.*

Proof. We will first reason that $\{h_1, h_2, h_3\}$ is a basis for the space of functions $p(x)$ with the following properties,

$$p|_{[0,1/2]}, p|_{[1/2,1]} \in \Pi_4 \,, \tag{3.2.11a}$$

$$p(0) = p(1) = 0 \,, \tag{3.2.11b}$$

$$p(x) = p(1-x) \,. \tag{3.2.11c}$$

Since $h_i \in \operatorname{span} \Delta_0^{(4,1)}$, the functions $h_i|_{[0,1/2]}$ form a basis for all cubic polynomials on $[0, 1/2]$ with $p(0) = 0$ (3.2.6). This establishes the first halves of (3.2.11a) and (3.2.11b). Their second halves and (3.2.11c) follow by the symmetric construction of the h_i. Finally, note that (3.2.11c) implies that p is continuous at $x = 1/2$.

By (3.2.3) and (3.2.7), q fulfils all conditions (3.2.11). This means that

$$q \in \operatorname{span}\{h_1, h_2, h_3\} \,, \tag{3.2.12}$$

and the proof is finished, implying in particular that also $h_3 \in \operatorname{span}\{h_1, h_2, q\}$. □

Thus, no matter how we choose the third condition on the b_i, and consequently alter the third dual basis function $\tilde{\phi}_3^{(0)}$, the same space is spanned by $\tilde{\Phi}_{j_0}^{(0)}$. The ambiguity in the third dual basis function is eliminated by the inversion of $\langle \Phi_{j_0}, \tilde{\Phi}_{j_0}^{(0)} \rangle$, which yields a uniquely defined final dual basis $\tilde{\Phi}_{j_0}$. This matrix has the following structure,

$$\langle \Phi_{j_0}, \tilde{\Phi}_{j_0}^{(0)} \rangle = \begin{pmatrix} * & 0 & 0 & 0 & 0 \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ 0 & 0 & 0 & 0 & * \end{pmatrix} \,, \tag{3.2.13}$$

which is inherited by its inverse. The transformation

$$\tilde{\Phi}_{j_0}^T \coloneqq (\tilde{\Phi}_{j_0}^{(0)})^T \langle \Phi_{j_0}, \tilde{\Phi}_{j_0}^{(0)} \rangle^{-1} \tag{3.2.14}$$

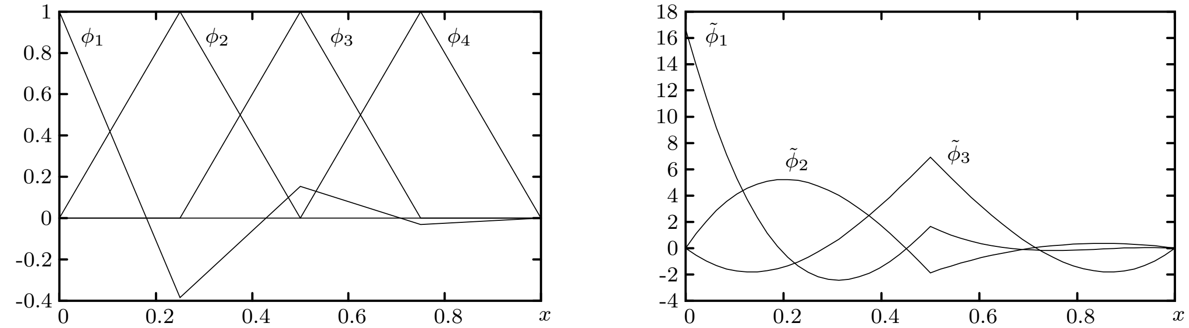


Figure 3.3: The biorthogonal generators $\Phi_j, \tilde{\Phi}_j$ *for the construction of finite element wavelets on the coarsest level* $j = j_0 = 2$. Φ_j *consists of piecewise linear functions, while each* $\tilde{\phi}_i$ *is composed of two cubic polynomials which are continuously glued together at* $x = 1/2$. *For clarity not all functions are shown. The missing ones can be inferred by symmetry* (3.2.7).

x	$\phi_1(x)$	$\phi_5(x)$
0	1	0
$\frac{1}{4}$	$-\frac{23}{60}$	$-\frac{3}{100}$
$\frac{1}{2}$	$\frac{23}{150}$	$\frac{23}{150}$
$\frac{3}{4}$	$-\frac{3}{100}$	$-\frac{23}{60}$
1	0	1

$\sum_{i=0}^{3} c_i x^i$	$c_3, \ldots, c_0$ for $[0, \frac{1}{2}]$				$c_3, \ldots, c_0$ for $[\frac{1}{2}, 1]$			
$\tilde{\phi}_1(x)$	-160	296	-138	$\frac{50}{3}$	$-\frac{160}{3}$	136	-114	$\frac{94}{3}$
$\tilde{\phi}_2(x)$	$\frac{1504}{15}$	$-\frac{4248}{25}$	$\frac{1404}{25}$	0	$\frac{32}{15}$	$-\frac{568}{25}$	$\frac{852}{25}$	$-\frac{1012}{75}$
$\tilde{\phi}_3(x)$	$-\frac{1088}{15}$	$\frac{3024}{25}$	$-\frac{712}{25}$	0	$\frac{1088}{15}$	$-\frac{2416}{25}$	$\frac{104}{25}$	$\frac{1496}{75}$
$\tilde{\phi}_4(x)$	$-\frac{32}{15}$	$-\frac{408}{25}$	$\frac{124}{25}$	0	$-\frac{1504}{15}$	$\frac{3272}{25}$	$-\frac{428}{25}$	$-\frac{1012}{75}$
$\tilde{\phi}_5(x)$	$\frac{160}{3}$	-24	2	0	160	-184	26	$\frac{44}{3}$

Table 3.1: This table contains the exact representations of the primal and dual wavelets for $j = j_0 = 2$. *For the primal basis, the values of the functions* ϕ_1 *and* ϕ_5 *are given at the nodes of the interval. For the dual basis, the coefficients of the cubic polynomials are given, once for the first half of the interval and then for the second. Technically, half of this information is redundant as it can be obtained by algebraic computations using the symmetry* (3.2.7).

$$
\begin{array}{|ccccc|cccc|}
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{2611}{3750} & \frac{1}{2} & \frac{3}{100} & 0 & -\frac{13}{1250} & 1 & 0 & 0 & 0 \\
-\frac{3151}{5625} & 1 & -\frac{23}{150} & 0 & -\frac{301}{5625} & 0 & 0 & 0 & 0 \\
-\frac{59}{2250} & \frac{1}{2} & \frac{53}{60} & 0 & \frac{271}{2250} & 0 & 1 & 0 & 0 \\
\frac{23}{150} & 0 & 1 & 0 & \frac{23}{150} & 0 & 0 & 0 & 0 \\
\frac{271}{2250} & 0 & \frac{53}{60} & \frac{1}{2} & -\frac{59}{2250} & 0 & 0 & 1 & 0 \\
-\frac{301}{5625} & 0 & -\frac{23}{150} & 1 & -\frac{3151}{5625} & 0 & 0 & 0 & 0 \\
-\frac{13}{1250} & 0 & \frac{3}{100} & \frac{1}{2} & \frac{2611}{3750} & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
\qquad
\begin{array}{|ccccc|}
\hline
1 & 0 & 0 & 0 & 0 \\
\frac{139}{128} & \frac{3373}{3200} & -\frac{627}{1600} & \frac{253}{3200} & -\frac{1}{128} \\
-\frac{11}{32} & \frac{237}{200} & -\frac{13}{100} & \frac{7}{200} & -\frac{1}{32} \\
-\frac{51}{128} & \frac{1843}{3200} & \frac{1043}{1600} & -\frac{477}{3200} & \frac{9}{128} \\
\frac{1}{5} & -\frac{28}{125} & \frac{104}{125} & -\frac{28}{125} & \frac{1}{5} \\
\frac{9}{128} & -\frac{477}{3200} & \frac{1043}{1600} & \frac{1843}{3200} & -\frac{51}{128} \\
-\frac{1}{32} & \frac{7}{200} & -\frac{13}{100} & \frac{237}{200} & -\frac{11}{32} \\
-\frac{1}{128} & \frac{253}{3200} & -\frac{627}{1600} & \frac{3373}{3200} & \frac{139}{128} \\
0 & 0 & 0 & 0 & 1 \\
\hline
\end{array}
$$

Table 3.2: This table shows the initial primal two-level transform $\sqrt{2}\check{\mathbf{M}}_j = \sqrt{2}(\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1})$ for the coarsest level $j = j_0 = 2$ on the left, using the simple guess of the stable completion. On the right hand side, we display the refinement matrix $\sqrt{2}\tilde{\mathbf{M}}_{j,0}$.

finally yields the biorthogonal pair satisfying $\langle \Phi_{j_0}, \tilde{\Phi}_{j_0} \rangle = \mathbf{I}$. The fact that only the diagonal element in the first and last row of the inverse is nonzero guarantees to end up with uniformly local basis functions. The two sets of piecewise polynomial functions are shown in Figure 3.3, where we have computed the coefficients with rational numbers. The exact representations are listed in Table 3.1.

Now that we have constructed a biorthogonal generator basis for the coarsest level $j_0 = 2$, we need to derive generator bases for higher levels. The basic technique is a transformation of the coarse basis functions,

$$\phi(x) \mapsto \sqrt{2}\,\phi\,(2x)\ . \tag{3.2.15}$$

The complete process affects both the primal and the dual bases as follows.

(i) Append two copies of the basis to each other, forming a set of functions on $(0, 2)$.

(ii) Merge the functions with do not vanish at $x = 1$ (one from the left and one from the right interval) into one function crossing the border whose support is now doubled in length (and thus is its norm). Multiply the function on the dual side by $\frac{1}{2}$ to keep the set of functions biorthonormal.

(iii) Apply the transformation (3.2.15) to all functions. This yields again a basis on $(0, 1)$ and ensures uniform normalisation, $\|\phi_{j,k}\|_{L_2}, \|\tilde{\phi}_{j,k}\|_{L_2} \sim 1$.

This procedure is repeated recursively, creating a hierarchy of sets of functions satisfying

$$\#\Phi_j = \#\tilde{\Phi}_j = 2^j + 1\,, \qquad \langle \Phi_j, \tilde{\Phi}_j \rangle = \mathbf{I}\,. \tag{3.2.16}$$

3.2.3 A Standard Stable Completion

Having constructed a biorthogonal single-scale basis which obeys stability and locality conditions (2.2.5), (2.2.7), it remains to set up the primal and dual refinement matrices $\mathbf{M}_{j,0}$, $\tilde{\mathbf{M}}_{j,0}$ and an initial stable completion $\check{\mathbf{M}}_{j,1}$ such that the inverse of $\check{\mathbf{M}}_j = (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1})$ is sparse.

The refinement matrices can be obtained by elementary calculations. In [128], the stable completion is guessed by the simplest ansatz possible, that is by putting ones on the slanted diagonal and zeros elsewhere, which corresponds to the hierarchical basis ansatz. The matrices $\check{\mathbf{M}}_j$ and $\tilde{\mathbf{M}}_{j,0}$ are shown in Table 3.2. The symmetry properties of the functions are expressed by

$$\mathbf{M}_{j,0} = \mathbf{M}_{j,0}^{\updownarrow} \qquad \text{and} \qquad \tilde{\mathbf{M}}_{j,0} = \tilde{\mathbf{M}}_{j,0}^{\updownarrow}\,. \tag{3.2.17}$$

Here the expression $\mathbf{M}^{\updownarrow}$ denotes the matrix $\mathbf{M}$ with reversed order of rows and columns.

$$\begin{array}{|ccccc|cccc|}\hline 1 & 0 & 0 & 0 & 0 & -\frac{139}{256} & \frac{51}{256} & -\frac{9}{256} & \frac{1}{256} \\ \frac{2611}{3750} & \frac{1}{2} & \frac{3}{100} & 0 & -\frac{13}{1250} & \frac{3497}{9600} & -\frac{47}{3200} & \frac{3}{3200} & -\frac{53}{9600} \\ -\frac{3151}{5625} & 1 & -\frac{23}{150} & 0 & -\frac{301}{5625} & -\frac{81}{320} & -\frac{1669}{4800} & \frac{641}{4800} & -\frac{41}{960} \\ -\frac{59}{2250} & \frac{1}{2} & \frac{53}{60} & 0 & \frac{271}{2250} & -\frac{727}{9600} & \frac{5363}{9600} & -\frac{2167}{9600} & \frac{281}{3200} \\ \frac{23}{150} & 0 & 1 & 0 & \frac{23}{150} & \frac{29}{256} & -\frac{77}{256} & -\frac{77}{256} & \frac{29}{256} \\ \frac{271}{2250} & 0 & \frac{53}{60} & \frac{1}{2} & -\frac{59}{2250} & \frac{281}{3200} & -\frac{2167}{9600} & \frac{5363}{9600} & -\frac{727}{9600} \\ -\frac{301}{5625} & 0 & -\frac{23}{150} & 1 & -\frac{3151}{5625} & -\frac{41}{960} & \frac{641}{4800} & -\frac{1669}{4800} & -\frac{81}{320} \\ -\frac{13}{1250} & 0 & \frac{3}{100} & \frac{1}{2} & \frac{2611}{3750} & -\frac{53}{9600} & \frac{3}{3200} & -\frac{47}{3200} & \frac{3497}{9600} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{256} & -\frac{9}{256} & \frac{51}{256} & -\frac{139}{256} \\ \hline \end{array}$$

$$\begin{array}{|ccccc|cccc|}\hline 1 & 0 & 0 & 0 & 0 & -\frac{48}{25} & -\frac{16}{75} & 0 & 0 \\ \frac{139}{128} & \frac{3373}{3200} & -\frac{627}{1600} & \frac{253}{3200} & -\frac{1}{128} & 2 & 0 & 0 & 0 \\ -\frac{11}{32} & \frac{237}{200} & -\frac{13}{100} & \frac{7}{200} & -\frac{1}{32} & -1 & -1 & 0 & 0 \\ -\frac{51}{128} & \frac{1843}{3200} & \frac{1043}{1600} & -\frac{477}{3200} & \frac{9}{128} & 0 & 2 & 0 & 0 \\ \frac{1}{5} & -\frac{28}{125} & \frac{104}{125} & -\frac{28}{125} & \frac{1}{5} & -\frac{16}{75} & -\frac{48}{25} & -\frac{48}{25} & -\frac{16}{75} \\ \frac{9}{128} & -\frac{477}{3200} & \frac{1043}{1600} & \frac{1843}{3200} & -\frac{51}{128} & 0 & 0 & 2 & 0 \\ -\frac{1}{32} & \frac{7}{200} & -\frac{13}{100} & \frac{237}{200} & -\frac{11}{32} & 0 & 0 & -1 & -1 \\ -\frac{1}{128} & \frac{253}{3200} & -\frac{627}{1600} & \frac{3373}{3200} & \frac{139}{128} & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -\frac{16}{75} & -\frac{48}{25} \\ \hline \end{array}$$

Table 3.3: We show the primal and dual two-level transformation matrices $\sqrt{2}\mathbf{M}_j$ *and* $\sqrt{2}\tilde{\mathbf{M}}_j$ *derived from the simple stable completion from Table 3.2 on the coarsest level* $j = j_0 = 2$*. They satisfy the biorthogonality condition* $\mathbf{M}_j = \tilde{\mathbf{M}}_j^{-T}$.

The right parts of the final primal and dual two-level transforms are then obtained by (2.3.9) and (2.3.11). Their representation with rational numbers on the coarsest level $j_0 = 2$ is given in Table 3.3. We observe that the right block of the primal matrix, $\mathbf{M}_{j_0,1}$, is fully populated, while the remaining three blocks exhibit a block-banded structure. An examination of the next higher level $j = j_0 + 1 = 3$, displayed in Table 3.4, shows that there also the primal right block contains zeros in the top right and bottom left corners, hinting at a block-banded structure. This can indeed be verified for all levels $j \geq 3$ (we do not print the matrices for higher levels here, since the entries are the same as for $j = 3$). Consequently, all two-level transformations are uniformly sparse.

3.2.4 Improvements to the Stable Completion

In view of the almost fully populated part of the right half of $\check{\mathbf{M}}_j$ for $j = 3$ in Table 3.4 and the large absolute values in the denominators of the nonzero entries, we construct an improved stable completion. To this end, we inspect the relation (2.3.9) and explicitly compare the involved matrices on the levels $j = j_0$ and $j = j_0 + 1$. We find that the number of nonzero entries of $\mathbf{I} - \mathbf{M}_{j,0}\tilde{\mathbf{M}}_{j,0}^T$ can be significantly reduced by taking appropriate linear combinations of columns. This corresponds to a multiplication from the right with a matrix $\check{\mathbf{L}}_j$, yielding the modified expression

$$\mathbf{M}_{j,1} = (\mathbf{I} - \mathbf{M}_{j,0}\tilde{\mathbf{M}}_{j,0}^T)\check{\mathbf{L}}_j\check{\mathbf{M}}_{j,1}\,, \tag{3.2.18}$$

which corresponds to the substitution

$$\check{\mathbf{M}}_{j,1} \mapsto \check{\mathbf{M}}'_{j,1} := \check{\mathbf{L}}_j\check{\mathbf{M}}_{j,1}\,. \tag{3.2.19}$$

To show that this additional matrix fits into the framework of Section 2.3.1, we formulate the following

Lemma 3.3. *Given an initial stable completion* $\check{\mathbf{M}}_{j,1}$ *and a target completion* $\check{\mathbf{M}}'_{j,1}$*, the transformation matrices from* (2.3.1) *are determined by*

$$\mathbf{L}_j^{(1)} := \check{\mathbf{G}}_{j,0}\check{\mathbf{M}}'_{j,1}\,, \qquad \mathbf{K}_j^{(1)} := \check{\mathbf{G}}_{j,1}\check{\mathbf{M}}'_{j,1}\,. \tag{3.2.20}$$

$\check{\mathbf{M}}'_{j,1}$ *is again a stable completion if and only if* $\mathbf{L}_j^{(1)}$ *and* $\mathbf{K}_j^{(1)}$ *fulfil the assumptions of Theorem 2.13.*

Proof. According to (2.3.2) and the definitions of Theorem 2.13, we have

$$\check{\mathbf{M}}'_{j,1} = \mathbf{M}_{j,0}\mathbf{L}_j^{(1)} + \check{\mathbf{M}}_{j,1}\mathbf{K}_j^{(1)} = (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1})\begin{pmatrix}\mathbf{L}_j^{(1)}\\ \mathbf{K}_j^{(1)}\end{pmatrix} = \check{\mathbf{M}}_j\begin{pmatrix}\mathbf{L}_j^{(1)}\\ \mathbf{K}_j^{(1)}\end{pmatrix}. \tag{3.2.21}$$

Multiplication with $\check{\mathbf{G}}_j = \check{\mathbf{M}}_j^{-1}$ leads to

$$\begin{pmatrix}\mathbf{L}_j^{(1)}\\ \mathbf{K}_j^{(1)}\end{pmatrix} = \check{\mathbf{G}}_j\check{\mathbf{M}}'_{j,1}\,, \tag{3.2.22}$$

which is equivalent to (3.2.20). □

The improved initial stable completion $\check{\mathbf{M}}'_{j,1}$ is presented in Table 3.5, together with the matrices $\mathbf{K}_j^{(1)}$ and $\mathbf{L}_j^{(1)}$ from (3.2.20). These transformations are applied prior to the biorthogonalisation, which is then performed by the matrices from (2.3.10),

$$\mathbf{L}_j^{(2)} := -\tilde{\mathbf{M}}_{j,0}^T\check{\mathbf{M}}'_{j,1}\,, \qquad \mathbf{K}_j^{(2)} := \mathbf{I}\,. \tag{3.2.23}$$

Thus, we deal with a two-stage process here. To formalise the concatenation of two such transformations, we derive the following

$$
\left[\begin{array}{ccccccccc|cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{139}{256} & \frac{51}{256} & -\frac{9}{256} & \frac{1}{256} & 0 & 0 & 0 & 0 \\
\frac{2611}{3750} & \frac{1}{2} & \frac{3}{100} & 0 & -\frac{13}{1250} & 0 & 0 & 0 & 0 & \frac{699439}{1920000} & -\frac{9517}{640000} & \frac{1263}{640000} & -\frac{16021}{1920000} & \frac{1807}{640000} & -\frac{663}{640000} & \frac{117}{640000} & -\frac{13}{640000} \\
-\frac{3151}{5625} & 1 & -\frac{23}{150} & 0 & -\frac{301}{5625} & 0 & 0 & 0 & 0 & -\frac{728699}{2880000} & -\frac{334703}{960000} & \frac{44439}{320000} & -\frac{164839}{2880000} & \frac{41839}{2880000} & -\frac{5117}{960000} & \frac{301}{320000} & -\frac{301}{2880000} \\
-\frac{59}{2250} & \frac{1}{2} & \frac{53}{60} & 0 & \frac{271}{2250} & 0 & 0 & 0 & 0 & -\frac{87511}{1152000} & \frac{215333}{384000} & -\frac{30429}{128000} & \frac{138829}{1152000} & -\frac{37669}{1152000} & \frac{4607}{384000} & -\frac{271}{128000} & \frac{271}{1152000} \\
\frac{23}{150} & 0 & 1 & 0 & \frac{23}{150} & 0 & 0 & 0 & 0 & \frac{8677}{76800} & -\frac{7631}{25600} & -\frac{8091}{25600} & \frac{11897}{76800} & -\frac{3197}{76800} & \frac{391}{25600} & -\frac{69}{25600} & \frac{23}{76800} \\
\frac{271}{2250} & 0 & \frac{53}{60} & \frac{1}{2} & -\frac{59}{2250} & 0 & 0 & 0 & 0 & \frac{101219}{1152000} & -\frac{86857}{384000} & \frac{71841}{128000} & -\frac{95441}{1152000} & \frac{8201}{1152000} & -\frac{1003}{384000} & \frac{59}{128000} & -\frac{59}{1152000} \\
-\frac{301}{5625} & 0 & -\frac{23}{150} & 1 & -\frac{3151}{5625} & 0 & 0 & 0 & 0 & -\frac{119849}{2880000} & \frac{118747}{960000} & -\frac{93411}{320000} & -\frac{1166989}{2880000} & \frac{437989}{2880000} & -\frac{53567}{960000} & \frac{3151}{320000} & -\frac{3151}{2880000} \\
-\frac{13}{1250} & 0 & \frac{3}{100} & \frac{1}{2} & \frac{2611}{3750} & 0 & 0 & 0 & 0 & -\frac{13211}{1920000} & \frac{8433}{640000} & -\frac{53787}{640000} & \frac{1062329}{1920000} & -\frac{362929}{1920000} & \frac{44387}{640000} & -\frac{7833}{640000} & \frac{2611}{1920000} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{1}{512} & -\frac{9}{512} & \frac{51}{512} & -\frac{139}{512} & -\frac{139}{512} & \frac{51}{512} & -\frac{9}{512} & \frac{1}{512} \\
0 & 0 & 0 & 0 & \frac{2611}{3750} & \frac{1}{2} & \frac{3}{100} & 0 & -\frac{13}{1250} & \frac{2611}{1920000} & -\frac{7833}{640000} & \frac{44387}{640000} & -\frac{362929}{1920000} & \frac{1062329}{1920000} & -\frac{53787}{640000} & \frac{8433}{640000} & -\frac{13211}{1920000} \\
0 & 0 & 0 & 0 & -\frac{3151}{5625} & 1 & -\frac{23}{150} & 0 & -\frac{301}{5625} & -\frac{3151}{2880000} & \frac{3151}{320000} & -\frac{53567}{960000} & \frac{437989}{2880000} & -\frac{1166989}{2880000} & -\frac{93411}{320000} & \frac{118747}{960000} & -\frac{119849}{2880000} \\
0 & 0 & 0 & 0 & -\frac{59}{2250} & \frac{1}{2} & \frac{53}{60} & 0 & \frac{271}{2250} & -\frac{59}{1152000} & \frac{59}{128000} & -\frac{1003}{384000} & \frac{8201}{1152000} & -\frac{95441}{1152000} & \frac{71841}{128000} & -\frac{86857}{384000} & \frac{101219}{1152000} \\
0 & 0 & 0 & 0 & \frac{23}{150} & 0 & 1 & 0 & \frac{23}{150} & \frac{23}{76800} & -\frac{69}{25600} & \frac{391}{25600} & -\frac{3197}{76800} & \frac{11897}{76800} & -\frac{8091}{25600} & -\frac{7631}{25600} & \frac{8677}{76800} \\
0 & 0 & 0 & 0 & \frac{271}{2250} & 0 & \frac{53}{60} & \frac{1}{2} & -\frac{59}{2250} & \frac{271}{1152000} & -\frac{271}{128000} & \frac{4607}{384000} & -\frac{37669}{1152000} & \frac{138829}{1152000} & -\frac{30429}{128000} & \frac{215333}{384000} & -\frac{87511}{1152000} \\
0 & 0 & 0 & 0 & -\frac{301}{5625} & 0 & -\frac{23}{150} & 1 & -\frac{3151}{5625} & -\frac{301}{2880000} & \frac{301}{320000} & -\frac{5117}{960000} & \frac{41839}{2880000} & -\frac{164839}{2880000} & \frac{44439}{320000} & -\frac{334703}{960000} & -\frac{728699}{2880000} \\
0 & 0 & 0 & 0 & -\frac{13}{1250} & 0 & \frac{3}{100} & \frac{1}{2} & \frac{2611}{3750} & -\frac{13}{640000} & \frac{117}{640000} & -\frac{663}{640000} & \frac{1807}{640000} & -\frac{16021}{1920000} & \frac{1263}{640000} & -\frac{9517}{640000} & \frac{699439}{1920000} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{1}{256} & -\frac{9}{256} & \frac{51}{256} & -\frac{139}{256}
\end{array}\right]
$$

Table 3.4: The primal two-level transform $\sqrt{2}\mathbf{M}_j$ at level $j = 3$, for the simple stable completion.

0	0	0	0
1	0	0	0
$-\frac{1}{4}$	$-\frac{3}{2}$	0	0
0	1	0	0
0	$-\frac{75}{128}$	$-\frac{75}{128}$	0
0	0	1	0
0	0	$-\frac{3}{2}$	$-\frac{1}{4}$
0	0	0	1
0	0	0	0

$\frac{9}{8}$	$\frac{13}{16}$	$\frac{1}{16}$	0
$\frac{1}{8}$	$\frac{37}{16}$	$\frac{9}{16}$	0
0	$\frac{9}{16}$	$\frac{37}{16}$	$\frac{1}{8}$
0	$\frac{1}{16}$	$\frac{13}{16}$	$\frac{9}{8}$

0	0	0	0
$-\frac{1}{4}$	$-\frac{407}{256}$	$-\frac{23}{256}$	0
0	$-\frac{75}{128}$	$-\frac{75}{128}$	0
0	$-\frac{23}{256}$	$-\frac{407}{256}$	$-\frac{1}{4}$
0	0	0	0

Table 3.5: The left table shows the new proposal for a modified stable completion $\sqrt{2}\check{\mathbf{M}}_{j,1}$ on the coarsest level $j = 2$. It contains off-diagonal entries which induce linear combinations of the columns of $\mathbf{M}_{j,1}$. It is trivially extended to higher levels by repetition. In the middle and on the right we display the associated transformation matrices $\mathbf{K}_j^{(1)}$ and $\mathbf{L}_j^{(1)}$.

Lemma 3.4. *Let $\mathbf{K}_j^{(i)}$ and $\mathbf{L}_j^{(i)}$, $i = 1, 2$, denote the transformation matrices for two successive stable completions. These two steps can be written as a single step with the matrices*

$$\mathbf{L}_j = \mathbf{L}_j^{(2)} + \mathbf{L}_j^{(1)}\mathbf{K}_j^{(2)}\,, \qquad \mathbf{K}_j = \mathbf{K}_j^{(1)}\mathbf{K}_j^{(2)}\,. \tag{3.2.24}$$

Proof. We combine (2.3.2) and the left part of (3.2.21) and obtain

$$\mathbf{M}_{j,1} = \mathbf{M}_{j,0}\mathbf{L}_j^{(2)} + \check{\mathbf{M}}'_{j,1}\mathbf{K}_j^{(2)} = \mathbf{M}_{j,0}\mathbf{L}_j^{(2)} + \left(\mathbf{M}_{j,0}\mathbf{L}_j^{(1)} + \check{\mathbf{M}}_{j,1}\mathbf{K}_j^{(1)}\right)\mathbf{K}_j^{(2)}\,. \tag{3.2.25}$$

The claim is verified by comparison with (3.2.24). □

Using this result, we can now determine the structure of the complete transformation.

Theorem 3.5. *With the definitions of Theorem 2.13, the transformation matrices for the stable completion modified according to* (3.2.19) *are given by*

$$\mathbf{L}_j = \left(-\tilde{\mathbf{M}}_{j,0}^T + \check{\mathbf{G}}_{j,0}\right)\check{\mathbf{L}}_j\check{\mathbf{M}}_{j,1}\,, \qquad \mathbf{K}_j = \check{\mathbf{G}}_{j,1}\check{\mathbf{L}}_j\check{\mathbf{M}}_{j,1}\,. \tag{3.2.26}$$

By carrying out these calculations in our example we indeed achieve a reduction of the quantity of nonzero entries in the right part of $\mathbf{M}_j$ by about a factor of 2. The improved transformation matrices on level $j = 2$ are shown in Table 3.6. In contrast to the simple stable completion from Table 3.3, the block-banded structure can be seen in all four parts of the matrices. The full effect only becomes apparent on the next higher level $j = 3$. The corresponding primal matrix $\mathbf{M}_j$ is shown in Table 3.7, and the dual matrix $\tilde{\mathbf{M}}_j$ in Table 3.8. There are several benefits of the new construction.

- The number of arithmetic operations in the forward transformation (which is given by the number of nonzero entries in the matrix) is reduced by a factor of almost two.
- The size of the support of the wavelets is reduced by a factor of two to four.
- The denominators in the fractions are significantly smaller, indicating less irrational numbers.
- The transformation matrices at level $j = 2$ contain the same entries as the matrices on higher levels which facilitates the implementation.
- The pattern of nonzero entries is similar for the primal and dual matrices.

$$
\begin{array}{|ccccc|cccc|}
\hline
1 & 0 & 0 & 0 & 0 & -\frac{75}{128} & 0 & 0 & 0 \\
\frac{2611}{3750} & \frac{1}{2} & \frac{3}{100} & 0 & -\frac{13}{1250} & \frac{2611}{6400} & \frac{839}{3200} & \frac{39}{3200} & -\frac{39}{6400} \\
-\frac{3151}{5625} & 1 & -\frac{23}{150} & 0 & -\frac{301}{5625} & -\frac{3151}{9600} & -\frac{4499}{4800} & \frac{301}{4800} & -\frac{301}{9600} \\
-\frac{59}{2250} & \frac{1}{2} & \frac{53}{60} & 0 & \frac{271}{2250} & -\frac{59}{3840} & \frac{2129}{1920} & -\frac{271}{1920} & \frac{271}{3840} \\
\frac{23}{150} & 0 & 1 & 0 & \frac{23}{150} & \frac{23}{256} & -\frac{49}{64} & -\frac{49}{64} & \frac{23}{256} \\
\frac{271}{2250} & 0 & \frac{53}{60} & \frac{1}{2} & -\frac{59}{2250} & \frac{271}{3840} & -\frac{271}{1920} & \frac{2129}{1920} & -\frac{59}{3840} \\
-\frac{301}{5625} & 0 & -\frac{23}{150} & 1 & -\frac{3151}{5625} & -\frac{301}{9600} & \frac{301}{4800} & -\frac{4499}{4800} & -\frac{3151}{9600} \\
-\frac{13}{1250} & 0 & \frac{3}{100} & \frac{1}{2} & \frac{2611}{3750} & -\frac{39}{6400} & \frac{39}{3200} & \frac{839}{3200} & \frac{2611}{6400} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\frac{75}{128} \\
\hline
\end{array}
$$

$$
\begin{array}{|ccccc|cccc|}
\hline
1 & 0 & 0 & 0 & 0 & -\frac{128}{75} & 0 & 0 & 0 \\
\frac{139}{128} & \frac{3373}{3200} & -\frac{627}{1600} & \frac{253}{3200} & -\frac{1}{128} & \frac{139}{75} & -\frac{8}{75} & \frac{2}{75} & -\frac{1}{75} \\
-\frac{11}{32} & \frac{237}{200} & -\frac{13}{100} & \frac{7}{200} & -\frac{1}{32} & -\frac{44}{75} & -\frac{32}{75} & \frac{8}{75} & -\frac{4}{75} \\
-\frac{51}{128} & \frac{1843}{3200} & \frac{1043}{1600} & -\frac{477}{3200} & \frac{9}{128} & -\frac{17}{25} & \frac{24}{25} & -\frac{6}{25} & \frac{3}{25} \\
\frac{1}{5} & -\frac{28}{125} & \frac{104}{125} & -\frac{28}{125} & \frac{1}{5} & \frac{128}{375} & -\frac{256}{375} & -\frac{256}{375} & \frac{128}{375} \\
\frac{9}{128} & -\frac{477}{3200} & \frac{1043}{1600} & \frac{1843}{3200} & -\frac{51}{128} & \frac{3}{25} & -\frac{6}{25} & \frac{24}{25} & -\frac{17}{25} \\
-\frac{1}{32} & \frac{7}{200} & -\frac{13}{100} & \frac{237}{200} & -\frac{11}{32} & -\frac{4}{75} & \frac{8}{75} & -\frac{32}{75} & -\frac{44}{75} \\
-\frac{1}{128} & \frac{253}{3200} & -\frac{627}{1600} & \frac{3373}{3200} & \frac{139}{128} & -\frac{1}{75} & \frac{2}{75} & -\frac{8}{75} & \frac{139}{75} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\frac{128}{75} \\
\hline
\end{array}
$$

Table 3.6: We show the new proposal for the primal and dual two-level transformation matrices on the coarsest level, $\sqrt{2}\mathbf{M}_j$ *and* $\sqrt{2}\tilde{\mathbf{M}}_j$*, derived from the modified stable completion from Table 3.5.*

1	0	0	0	0	0	0	0	0	$-\frac{75}{128}$	0	0	0	0	0	0	0
$\frac{2611}{3750}$	$\frac{1}{2}$	$\frac{3}{100}$	0	$-\frac{13}{1250}$	0	0	0	0	$\frac{2611}{6400}$	$\frac{839}{3200}$	$\frac{39}{3200}$	$-\frac{117}{12800}$	$\frac{39}{12800}$	0	0	0
$-\frac{3151}{5625}$	1	$-\frac{23}{150}$	0	$-\frac{301}{5625}$	0	0	0	0	$-\frac{3151}{9600}$	$-\frac{4499}{4800}$	$\frac{301}{4800}$	$-\frac{301}{6400}$	$\frac{301}{19200}$	0	0	0
$-\frac{59}{2250}$	$\frac{1}{2}$	$\frac{53}{60}$	0	$\frac{271}{2250}$	0	0	0	0	$-\frac{59}{3840}$	$\frac{2129}{1920}$	$-\frac{271}{1920}$	$\frac{271}{2560}$	$-\frac{271}{7680}$	0	0	0
$\frac{23}{150}$	0	1	0	$\frac{23}{150}$	0	0	0	0	$\frac{23}{256}$	$-\frac{49}{64}$	$-\frac{49}{64}$	$\frac{69}{512}$	$-\frac{23}{512}$	0	0	0
$\frac{271}{2250}$	0	$\frac{53}{60}$	$\frac{1}{2}$	$-\frac{59}{2250}$	0	0	0	0	$\frac{271}{3840}$	$-\frac{271}{1920}$	$\frac{2129}{1920}$	$-\frac{59}{2560}$	$\frac{59}{7680}$	0	0	0
$-\frac{301}{5625}$	0	$-\frac{23}{150}$	1	$-\frac{3151}{5625}$	0	0	0	0	$-\frac{301}{9600}$	$\frac{301}{4800}$	$-\frac{4499}{4800}$	$-\frac{3151}{6400}$	$\frac{3151}{19200}$	0	0	0
$-\frac{13}{1250}$	0	$\frac{3}{100}$	$\frac{1}{2}$	$\frac{2611}{3750}$	0	0	0	0	$-\frac{39}{6400}$	$\frac{39}{3200}$	$\frac{839}{3200}$	$\frac{7833}{12800}$	$-\frac{2611}{12800}$	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	$-\frac{75}{256}$	$-\frac{75}{256}$	0	0	0
0	0	0	0	$\frac{2611}{3750}$	$\frac{1}{2}$	$\frac{3}{100}$	0	$-\frac{13}{1250}$	0	0	0	$-\frac{2611}{12800}$	$\frac{7833}{12800}$	$\frac{839}{3200}$	$\frac{39}{3200}$	$-\frac{39}{6400}$
0	0	0	0	$-\frac{3151}{5625}$	1	$-\frac{23}{150}$	0	$-\frac{301}{5625}$	0	0	0	$\frac{3151}{19200}$	$-\frac{3151}{6400}$	$-\frac{4499}{4800}$	$\frac{301}{4800}$	$-\frac{301}{9600}$
0	0	0	0	$-\frac{59}{2250}$	$\frac{1}{2}$	$\frac{53}{60}$	0	$\frac{271}{2250}$	0	0	0	$\frac{59}{7680}$	$-\frac{59}{2560}$	$\frac{2129}{1920}$	$-\frac{271}{1920}$	$\frac{271}{3840}$
0	0	0	0	$\frac{23}{150}$	0	1	0	$\frac{23}{150}$	0	0	0	$-\frac{23}{512}$	$\frac{69}{512}$	$-\frac{49}{64}$	$-\frac{49}{64}$	$\frac{23}{256}$
0	0	0	0	$\frac{271}{2250}$	0	$\frac{53}{60}$	$\frac{1}{2}$	$-\frac{59}{2250}$	0	0	0	$-\frac{271}{7680}$	$\frac{271}{2560}$	$-\frac{271}{1920}$	$\frac{2129}{1920}$	$-\frac{59}{3840}$
0	0	0	0	$-\frac{301}{5625}$	0	$-\frac{23}{150}$	1	$-\frac{3151}{5625}$	0	0	0	$\frac{301}{19200}$	$-\frac{301}{6400}$	$\frac{301}{4800}$	$-\frac{4499}{4800}$	$-\frac{3151}{9600}$
0	0	0	0	$-\frac{13}{1250}$	0	$\frac{3}{100}$	$\frac{1}{2}$	$\frac{2611}{3750}$	0	0	0	$\frac{39}{12800}$	$-\frac{117}{12800}$	$\frac{39}{3200}$	$\frac{839}{3200}$	$\frac{2611}{6400}$
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	$-\frac{75}{128}$

Table 3.7: The primal two-level transform $\sqrt{2}\mathbf{M}_j$ *at level* $j = 3$ *constructed with our modified stable completion.*

1	0	0	0	0	0	0	0	0	$-\frac{128}{75}$	0	0	0	0	0	0	0
$\frac{139}{128}$	$\frac{3373}{3200}$	$-\frac{627}{1600}$	$\frac{253}{3200}$	$-\frac{1}{256}$	0	0	0	0	$\frac{139}{75}$	$-\frac{8}{75}$	$\frac{2}{75}$	$-\frac{1}{75}$	0	0	0	0
$-\frac{11}{32}$	$\frac{237}{200}$	$-\frac{13}{100}$	$\frac{7}{200}$	$-\frac{1}{64}$	0	0	0	0	$-\frac{44}{75}$	$-\frac{32}{75}$	$\frac{8}{75}$	$-\frac{4}{75}$	0	0	0	0
$-\frac{51}{128}$	$\frac{1843}{3200}$	$\frac{1043}{1600}$	$-\frac{477}{3200}$	$\frac{9}{256}$	0	0	0	0	$-\frac{17}{25}$	$\frac{24}{25}$	$-\frac{6}{25}$	$\frac{3}{25}$	0	0	0	0
$\frac{1}{5}$	$-\frac{28}{125}$	$\frac{104}{125}$	$-\frac{28}{125}$	$\frac{1}{10}$	0	0	0	0	$\frac{128}{375}$	$-\frac{256}{375}$	$-\frac{256}{375}$	$\frac{128}{375}$	0	0	0	0
$\frac{9}{128}$	$-\frac{477}{3200}$	$\frac{1043}{1600}$	$\frac{1843}{3200}$	$-\frac{51}{256}$	0	0	0	0	$\frac{3}{25}$	$-\frac{6}{25}$	$\frac{24}{25}$	$-\frac{17}{25}$	0	0	0	0
$-\frac{1}{32}$	$\frac{7}{200}$	$-\frac{13}{100}$	$\frac{237}{200}$	$-\frac{11}{64}$	0	0	0	0	$-\frac{4}{75}$	$\frac{8}{75}$	$-\frac{32}{75}$	$-\frac{44}{75}$	0	0	0	0
$-\frac{1}{128}$	$\frac{253}{3200}$	$-\frac{627}{1600}$	$\frac{3373}{3200}$	$\frac{139}{256}$	0	0	0	0	$-\frac{1}{75}$	$\frac{2}{75}$	$-\frac{8}{75}$	$\frac{139}{75}$	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	$-\frac{128}{75}$	$-\frac{128}{75}$	0	0	0
0	0	0	0	$\frac{139}{256}$	$\frac{3373}{3200}$	$-\frac{627}{1600}$	$\frac{253}{3200}$	$-\frac{1}{128}$	0	0	0	0	$\frac{139}{75}$	$-\frac{8}{75}$	$\frac{2}{75}$	$-\frac{1}{75}$
0	0	0	0	$-\frac{11}{64}$	$\frac{237}{200}$	$-\frac{13}{100}$	$\frac{7}{200}$	$-\frac{1}{32}$	0	0	0	0	$-\frac{44}{75}$	$-\frac{32}{75}$	$\frac{8}{75}$	$-\frac{4}{75}$
0	0	0	0	$-\frac{51}{256}$	$\frac{1843}{3200}$	$\frac{1043}{1600}$	$-\frac{477}{3200}$	$\frac{9}{128}$	0	0	0	0	$-\frac{17}{25}$	$\frac{24}{25}$	$-\frac{6}{25}$	$\frac{3}{25}$
0	0	0	0	$\frac{1}{10}$	$-\frac{28}{125}$	$\frac{104}{125}$	$-\frac{28}{125}$	$\frac{1}{5}$	0	0	0	0	$\frac{128}{375}$	$-\frac{256}{375}$	$-\frac{256}{375}$	$\frac{128}{375}$
0	0	0	0	$\frac{9}{256}$	$-\frac{477}{3200}$	$\frac{1043}{1600}$	$\frac{1843}{3200}$	$-\frac{51}{128}$	0	0	0	0	$\frac{3}{25}$	$-\frac{6}{25}$	$\frac{24}{25}$	$-\frac{17}{25}$
0	0	0	0	$-\frac{1}{64}$	$\frac{7}{200}$	$-\frac{13}{100}$	$\frac{237}{200}$	$-\frac{11}{32}$	0	0	0	0	$-\frac{4}{75}$	$\frac{8}{75}$	$-\frac{32}{75}$	$-\frac{44}{75}$
0	0	0	0	$-\frac{1}{256}$	$\frac{253}{3200}$	$-\frac{627}{1600}$	$\frac{3373}{3200}$	$\frac{139}{128}$	0	0	0	0	$-\frac{1}{75}$	$\frac{2}{75}$	$-\frac{8}{75}$	$\frac{139}{75}$
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	$-\frac{128}{75}$

Table 3.8: The dual two-level transform $\sqrt{2}\tilde{\mathbf{M}}_j$ at level $j = 3$ constructed with our modified stable completion.

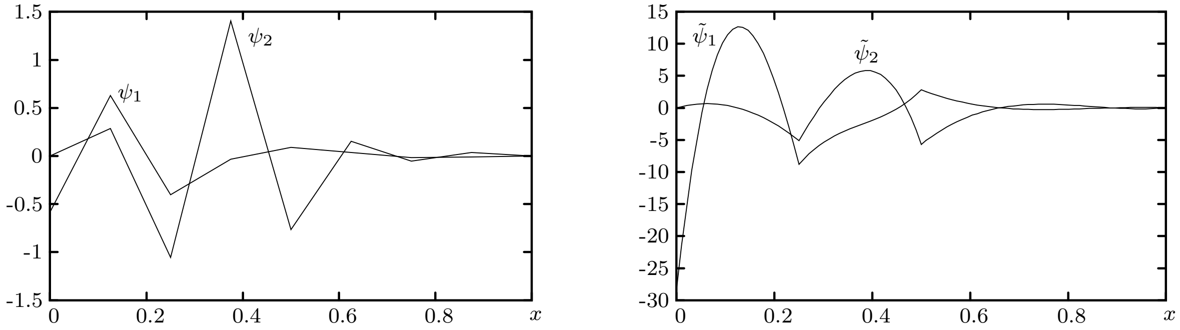


Figure 3.4: These graphs show the primal and dual wavelets for the coarsest complement space W_j (thus with $j = 2$), obtained by the modified stable completion. Only two of the four functions in each set are displayed, as the missing ones result by mirroring around $x = 1/2$. The primal wavelets on this level are piecewise linear with mesh size $1/8$, the dual wavelets are piecewise cubic with mesh size $1/4$.

$$\begin{array}{|ccccc|}\hline 1 & 0 & 0 & 0 & 0 \\ -\frac{23}{60} & 1 & 0 & 0 & -\frac{3}{100} \\ \frac{23}{150} & 0 & 1 & 0 & \frac{23}{150} \\ -\frac{3}{100} & 0 & 0 & 1 & -\frac{23}{60} \\ 0 & 0 & 0 & 0 & 1 \\ \hline\end{array} \qquad \begin{array}{|ccccc|}\hline 1 & 0 & 0 & 0 & 0 \\ \frac{23}{60} & 1 & 0 & 0 & \frac{3}{100} \\ -\frac{23}{150} & 0 & 1 & 0 & -\frac{23}{150} \\ \frac{3}{100} & 0 & 0 & 1 & \frac{23}{60} \\ 0 & 0 & 0 & 0 & 1 \\ \hline\end{array}$$

Table 3.9: This table shows the forward and backward transformation matrices between the single-scale basis Φ_j and the nodal basis for $j = j_0 = 2$. $\mathbf{C}_j^{-1}$ is shown on the left (note that the values in the first and last columns correspond to those from Table 3.1). $\mathbf{C}_j$ is shown on the right. Due to the special structure of these matrices, the inversion reduces to a change in sign of the off-diagonal entries.

The only disadvantage lies in the fact that the dual transformation has gained in the number of nonzero entries. Since the dual transform is rarely used in the context of partial differential equations, this is only a minor issue which is outweighed by far by the positive effects. The wavelets for the primal and dual spaces are finally shown in Figure 3.4.

The construction has so far been carried out for free boundary conditions. However, it is a straightforward procedure to obtain a biorthogonal wavelet basis with homogeneous boundary conditions on both the primal and dual side by simply deleting the two functions with nonzero value on the boundary. The corresponding rows and columns are then removed from the transformation matrices, which conserves biorthogonality.

3.2.5 Transformation to the Nodal Basis

The primal single-scale basis as shown in Figure 3.3 is identical to the nodal basis except for the functions which cross interval boundaries. These interrupt the translational invariance of the nodal basis and have larger support by a factor of 4. Both of these drawbacks can be resolved by an appropriate backward transformation of the single-scale basis as described in Section 2.3.2. More precisely, we aim at transformation matrices $\mathbf{C}_j$ as in (2.3.22), which transform Φ_j back into the nodal basis.

The necessary information has already been collected along the way in Section 3.2.2. The off-diagonal entries of $\mathbf{C}_{j_0}^{-1}$ are precisely the coefficients a_i from (3.2.8). We display the transformation matrix and its inverse in Figure 3.9. The matrices $\mathbf{C}_j$ and $\mathbf{C}_j^{-1}$ for higher levels contain exactly the same additional entries in every fourth column. As these matrices are uniformly sparse, their application does not spoil the operation count which is linear in the number of unknowns. The wavelet transform is corrected according to (2.3.28).

3.3 Spline Wavelets

In the previous section, we covered a construction of a wavelet basis where both primal and dual wavelets are piecewise polynomials. We started from scratch with the construction of a biorthogonal single-scale basis on the interval and concluded with proposals for optimisations.

The construction of *B-spline wavelets* in this section is conceptually different from the above since it is assumed here that a biorthogonal single-scale basis of $L_2(\mathbb{R})$ is already available. Such a basis has been constructed in [42] using Fourier techniques, where the primal basis is composed of B-splines and the functions on the dual side are given implicitly.

Wavelets on the interval are then derived in several steps. The original basis is restricted to the interval, corrections are introduced at the boundary to recover biorthogonality, and finally a stable completion is constructed.

As in the case of finite element wavelets, we establish the necessary definitions and notation for the construction, which we carry out explicitly with rational numbers, and subsequently spend some thoughts on improvements. Based on the insights gained in the previous section, we again propose a transformation to the nodal basis on the primal side to improve the condition number of the basis.

3.3.1 A Biorthogonal B-Spline Multiresolution on the Interval

We begin with the description of a B-spline multiresolution on the real line. A biorthogonal multiresolution on the unit interval $(0, 1)$ is then derived by restriction and subsequent modifications at the boundary [49].

Biorthogonal B-Spline Multiresolution of $L_2(\mathbb{R})$

The construction of B-spline wavelets which we describe here is based on the concept of *refinable functions*. A function $\phi \in L_2(\mathbb{R})$ is called refinable with *mask* $\mathbf{a} = \{a_k\}_{k\in\mathbb{Z}}$ if

$$\phi(x) = \sum_{k\in\mathbb{Z}} a_k \phi(2x - k)\,. \tag{3.3.1}$$

We say that two refinable functions ϕ, and $\tilde{\phi}$ with mask $\tilde{\mathbf{a}} = \{\tilde{a}_k\}_{k\in\mathbb{Z}}$ form a *dual pair* if

$$\big(\phi, \tilde{\phi}(\cdot - k)\big)_{L_2} = \delta_{0,k}\,, \qquad k \in \mathbb{Z}\,. \tag{3.3.2}$$

We assume in the following that ϕ and $\tilde{\phi}$ are normalised,

$$\int_{\mathbb{R}} \phi(x)\,\mathrm{d}x = \int_{\mathbb{R}} \tilde{\phi}(x)\,\mathrm{d}x = 1\,. \tag{3.3.3}$$

The refinable function ϕ is used to generate the family of functions

$$\phi_{[j,k]} := 2^{j/2}\phi(2^j \cdot - k)\,, \qquad j, k \in \mathbb{Z}\,. \tag{3.3.4}$$

Defining

$$\Phi_j := \{\phi_{[j,k]} : k \in \mathbb{Z}\} \qquad \text{and} \qquad \tilde{\Phi}_j := \{\tilde{\phi}_{[j,k]} : k \in \mathbb{Z}\}\,, \tag{3.3.5}$$

we obtain a multiresolution basis for $L_2(\mathbb{R})$ as introduced in Section 2.2.1. If ϕ and $\tilde{\phi}$ have compact support, it follows that the bases Φ_j and $\tilde{\Phi}_j$ are uniformly stable, and also that the number of nonzero entries of the masks $\mathbf{a}$ and $\tilde{\mathbf{a}}$ is finite.

The approximation power of the spaces $S_j := S(\Phi_j)$ and $\tilde{S}_j := S(\tilde{\Phi}_j)$ depends on their polynomial exactness. We say that ϕ is *exact of order d* if all polynomials of degree of at most $d-1$ can be expressed as a linear combination of the integer translates $\phi(\cdot - k)$. Likewise, the dual order of exactness is denoted by $\tilde{d}$. These properties are equivalent to the existence of the following representations of the monomials x^r,

$$x^r = \sum_{k\in\mathbb{Z}} \tilde{\alpha}_{k,r}\phi(x-k)\,, \qquad r = 0,\dots,d-1\,, \tag{3.3.6a}$$

$$x^r = \sum_{k\in\mathbb{Z}} \alpha_{k,r}\tilde{\phi}(x-k)\,, \qquad r = 0,\dots,\tilde{d}-1\,, \tag{3.3.6b}$$

with primal and dual expansion coefficients $\tilde{\alpha}_{k,r}$ and $\alpha_{k,r}$.

We will choose the *cardinal B-spline of order d* as generator for the primal multiresolution. To this end, we shortly review the associated definitions. Let $[t_0,\dots,t_d]f$ denote the d-th order divided difference of $f \in C^d(\mathbb{R})$ at the nodes $t_0 \le \dots \le t_d$. With the setting $x_+^d := (\max\{0,x\})^d$, the cardinal B-spline ϕ^d of order $d \in \mathbb{N}$ is defined as

$$\phi^d(x) := d[0,\dots,d]\left(\cdot - x - \left\lfloor \frac{d}{2} \right\rfloor\right)_+^{d-1}. \tag{3.3.7}$$

Hence, ϕ^d is symmetric around $\frac{\mu(d)}{2}$ with $\mu(d) := d \mod 2$,

$$\phi^d(x + \mu(d)) = \phi^d(-x)\,, \qquad x \in \mathbb{R}\,, \tag{3.3.8}$$

and its support is given by

$$\operatorname{supp}\phi^d = [\ell_1,\ell_2] \qquad \text{with} \qquad \ell_1 := -\left\lfloor \frac{d}{2} \right\rfloor, \qquad \ell_2 := \left\lceil \frac{d}{2} \right\rceil. \tag{3.3.9}$$

Thus, it is centred around $x = 0$ for even orders d and around $x = \frac{1}{2}$ for odd d. Note the identities

$$d = \ell_2 - \ell_1 \qquad \text{and} \qquad \mu(d) = \ell_1 + \ell_2\,. \tag{3.3.10}$$

The B-spline ϕ^d is an example of a refinable function according to (3.3.1) with mask

$$a_k = 2^{1-d}\binom{d}{k + \lfloor \frac{d}{2} \rfloor}\,, \qquad k = \ell_1,\dots,\ell_2\,. \tag{3.3.11}$$

Furthermore, ϕ^d is exact of order d. Thus, the generator for the primal multiresolution can be specified using standard B-spline theory. However, the construction of a dual generator which satisfies (3.3.2) is non-trivial. Such a result has been first obtained in [42] based on Fourier decompositions.

Theorem 3.6. *For each d, and $\tilde{d} \ge d$, $\tilde{d} \in \mathbb{N}$, with $d + \tilde{d}$ even, there exists a function $\tilde{\phi}^{d,\tilde{d}} \in L_2(\mathbb{R})$ with the following properties.*

- *The support is given by*

$$\operatorname{supp}\tilde{\phi}^{d,\tilde{d}} = [\tilde{\ell}_1,\tilde{\ell}_2] \qquad \textit{with} \qquad \tilde{\ell}_1 := \ell_1 - (\tilde{d}-1)\,, \qquad \tilde{\ell}_2 := \ell_2 + (\tilde{d}-1)\,. \tag{3.3.12}$$

- *$\tilde{\phi}^{d,\tilde{d}}$ is refinable with a finitely supported mask $\tilde{\mathbf{a}} = \{\tilde{a}_k\}_{k=\tilde{\ell}_1}^{\tilde{\ell}_2}$.*

- *$\tilde{\phi}^{d,\tilde{d}}$ has the same symmetry properties as ϕ^d, cf.* (3.3.8)*.*
- *The functions ϕ^d and $\tilde{\phi}^{d,\tilde{d}}$ form a dual pair, cf.* (3.3.2)*.*
- *$\tilde{\phi}^{d,\tilde{d}}$ is exact of order $\tilde{d}$, i.e., all polynomials of degree less than $\tilde{d}$ can be represented as linear combinations of the translates $\tilde{\phi}^{d,\tilde{d}}(\cdot - k)$, $k \in \mathbb{Z}$, see* (3.3.6b)*.*

In summary, primal and dual generators $\phi := \phi^d$ and $\tilde{\phi} := \tilde{\phi}^{d,\tilde{d}}$ with respective finite masks $\mathbf{a}$ and $\tilde{\mathbf{a}}$ and polynomial exactness d and $\tilde{d}$ do exist. To explicitly construct a wavelet basis for $L_2(\mathbb{R})$, it remains to calculate the coefficients $\tilde{\alpha}_{k,r}$ and $\alpha_{k,r}$, and to specify the complement bases Ψ_j and $\tilde{\Psi}_j$.

We infer from the biorthogonality condition (3.3.2) that the expansion coefficients from (3.3.6) have the explicit form

$$\tilde{\alpha}_{y,r} = \big((\cdot)^r, \tilde{\phi}(\cdot - y)\big)_{L_2}\,, \qquad r = 0, \ldots, d-1\,. \tag{3.3.13}$$

Using the normalisation (3.3.3), we derive the coefficients for the case $r = 0$ as

$$\tilde{\alpha}_{y,0} = 1\,. \tag{3.3.14}$$

The translated coefficients of higher order $r > 0$ for arbitrary y can be reduced to $y = 0$ via

$$\tilde{\alpha}_{y,r} = \int_{\mathbb{R}} (x+y)^r \tilde{\phi}(x)\,\mathrm{d}x = \sum_{i=0}^{r} \binom{r}{i} y^i \tilde{\alpha}_{0,r-i}\,, \qquad r = 0, \ldots, d-1\,. \tag{3.3.15}$$

Finally, $\tilde{\alpha}_{0,r}$ can be determined recursively. We refer to [50,132] for the full derivation and only state the result,

$$\tilde{\alpha}_{0,r} = (2^{r+1} - 2)^{-1} \sum_{k=\tilde{\ell}_1}^{\tilde{\ell}_2} \tilde{a}_k \sum_{s=0}^{r-1} \binom{r}{s} k^{r-s} \tilde{\alpha}_{0,s}\,. \tag{3.3.16}$$

The identities for the dual coefficients $\alpha_{y,r}$ are analogous for $r = 0, \ldots, \tilde{d} - 1$.

In [42], suitable bases Ψ_j and $\tilde{\Psi}_j$ for the complement spaces W_j and $\tilde{W}_j$, cf. (2.2.10), are derived from generator functions ψ and $\tilde{\psi}$,

$$\Psi_j := \{\psi_{[j,k]} : k \in \mathbb{Z}\} \qquad \text{and} \qquad \tilde{\Psi}_j := \{\tilde{\psi}_{[j,k]} : k \in \mathbb{Z}\}\,. \tag{3.3.17}$$

If the generators have the form

$$\psi(x) := \sum_{k\in\mathbb{Z}} b_k \phi(2x - k)\,, \qquad \tilde{\psi}(x) := \sum_{k\in\mathbb{Z}} \tilde{b}_k \tilde{\phi}(2x - k) \tag{3.3.18}$$

with masks $\mathbf{b} = \{b_k\}_{k\in\mathbb{Z}}$ and $\tilde{\mathbf{b}} = \{\tilde{b}_k\}_{k\in\mathbb{Z}}$,

$$b_k := (-1)^k \tilde{a}_{1-k}\,, \qquad \tilde{b}_k := (-1)^k a_{1-k}\,, \qquad k \in \mathbb{Z}\,, \tag{3.3.19}$$

it follows that the biorthogonality conditions are satisfied,

$$\big(\phi, \tilde{\psi}(\cdot - k)\big)_{L_2} = \big(\tilde{\phi}, \psi(\cdot - k)\big)_{L_2} = 0\,, \qquad \big(\psi, \tilde{\psi}(\cdot - k)\big)_{L_2} = \delta_{0,k}\,, \qquad k \in \mathbb{Z}\,. \tag{3.3.20}$$

Due to the refinability of ϕ and $\tilde{\phi}$, the full wavelet bases Ψ and $\tilde{\Psi}$ as assembled in (2.2.17) are also biorthogonal. Since $\tilde{\phi}$ is exact of order $\tilde{d}$, we infer that ψ and therefore all wavelets $\psi_{[j,k]}$ have $\tilde{d}$ vanishing moments,

$$\int_{\mathbb{R}} x^r \psi_{[j,k]}(x)\,\mathrm{d}x = 0\,, \qquad r = 0, \ldots, \tilde{d} - 1\,. \tag{3.3.21}$$

Finally, the biorthogonality condition (2.2.44) and the finite masks $\mathbf{a}$, $\tilde{\mathbf{a}}$ and $\mathbf{b}$, $\tilde{\mathbf{b}}$ entail that Ψ and $\tilde{\Psi}$ are Riesz bases for $L_2(\mathbb{R})$.

Restriction to the Interval

The restriction of the collections Φ_j, $\tilde{\Phi}_j$ to the interval $(0,1)$ poses two essential difficulties.

- Basis functions which contain $x=0$ or $x=1$ in their support are truncated, with the result that the respective scalar products change their values.
- As the generators ϕ and $\tilde{\phi}$ generally have supports of different lengths, the cardinalities of Φ_j and $\tilde{\Phi}_j$ are now finite but not equal.

As a result, biorthogonality is destroyed. In [49], this issue has been resolved by taking linear combinations of the truncated original functions in such a way that the new functions are identical to monomials near the boundary. We will describe this ansatz in the following.

Since the support of ϕ is bounded according to (3.3.9), the nonzero part of the mask is given by $\mathbf{a} = \{a_k\}_{k=\ell_1}^{\ell_2}$. For an arbitrary but fixed parameter $\ell \geq -\ell_1$, we introduce the following function on the left boundary,

$$\phi^{\mathrm{L}}_{j,\ell-d+r} \coloneqq \sum_{m=-\ell_2+1}^{\ell-1} \tilde{\alpha}_{m,r}\phi_{[j,m]}\Big|_{\mathbb{R}_+}, \qquad r=0,\ldots,d-1\,. \tag{3.3.22}$$

It follows from (3.3.6a) that this function is identical to the monomial x^r on the interval $2^j x \in [0, \ell+\ell_1]$ and then declines to zero. It is known from [43] that this function is refinable. The precise form of the refinement relation reads

$$\begin{aligned} \phi^{\mathrm{L}}_{j,\ell-d+r} = 2^{-(r+\frac{1}{2})} &\left(\phi^{\mathrm{L}}_{j+1,\ell-d+r} + \sum_{m=\ell}^{2\ell+\ell_1-1} \tilde{\alpha}_{m,r}\phi_{[j+1,m]} \right) \\ &+ \sum_{m=2\ell+\ell_1}^{2\ell+\ell_2-2} \tilde{\beta}_{m,r}\phi_{[j+1,m]}\,, \qquad r=0,\ldots,d-1 \end{aligned} \tag{3.3.23}$$

with

$$\tilde{\beta}_{m,r} \coloneqq 2^{-\frac{1}{2}} \sum_{q=\lceil \frac{m-\ell_2}{2} \rceil}^{\ell-1} \tilde{\alpha}_{q,r} a_{m-2q}\,. \tag{3.3.24}$$

At the right end of the interval, we define $\phi^{\mathrm{R}}_{j,\ell-d+r}$ analogously to (3.3.22) and obtain a similar refinement relation. On the dual side, we introduce the parameter $\tilde{\ell} \geq -\tilde{\ell}_1$ and define the boundary functions $\tilde{\phi}^{\mathrm{L}}_{j,\tilde{\ell}-\tilde{d}+r}$ and $\tilde{\phi}^{\mathrm{R}}_{j,\tilde{\ell}-\tilde{d}+r}$, indexed by $r=0,\ldots,\tilde{d}-1$. With the definitions

$$\tilde{\alpha}^{\mathrm{L}}_{j,m,r} \coloneqq \tilde{\alpha}_{m,r}\,, \qquad \tilde{\alpha}^{\mathrm{R}}_{j,m,r} \coloneqq \tilde{\alpha}_{2^j-m-\mu(d),r}\,, \qquad r=0,\ldots,d-1\,, \tag{3.3.25a}$$

$$\alpha^{\mathrm{L}}_{j,m,r} \coloneqq \alpha_{m,r}\,, \qquad \alpha^{\mathrm{R}}_{j,m,r} \coloneqq \alpha_{2^j-m-\mu(d),r}\,, \qquad r=0,\ldots,\tilde{d}-1\,, \tag{3.3.25b}$$

the boundary functions read

$$\begin{aligned} \phi^{\mathrm{L}}_{j,\ell-d+r} &\coloneqq \sum_{m=-\ell_2+1}^{\ell-1} \tilde{\alpha}^{\mathrm{L}}_{j,m,r}\phi_{[j,m]}\Big|_{[0,1]} \\ \phi^{\mathrm{R}}_{j,2^j-\ell-\mu(d)+d-r} &\coloneqq \sum_{m=2^j-\ell-\mu(d)+1}^{2^j-\ell_1-1} \tilde{\alpha}^{\mathrm{R}}_{j,m,r}\phi_{[j,m]}\Big|_{[0,1]} \end{aligned} \qquad r=0,\ldots,d-1 \tag{3.3.26}$$

and

$$\left.\begin{aligned}\tilde{\phi}^{\mathrm{L}}_{j,\tilde{\ell}-\tilde{d}+r} &:= \sum_{m=-\tilde{\ell}_2+1}^{\tilde{\ell}-1} \alpha^{\mathrm{L}}_{j,m,r}\tilde{\phi}_{[j,m]}\Big|_{[0,1]}\\ \tilde{\phi}^{\mathrm{R}}_{j,2^j-\tilde{\ell}-\mu(d)+\tilde{d}-r} &:= \sum_{m=2^j-\tilde{\ell}-\mu(d)+1}^{2^j-\tilde{\ell}_1-1} \alpha^{\mathrm{R}}_{j,m,r}\tilde{\phi}_{[j,m]}\Big|_{[0,1]}\end{aligned}\right\} \quad r=0,\dots,\tilde{d}-1\,. \tag{3.3.27}$$

The indexing of these functions was carefully chosen in [49] to allow for continuous numbering of all functions left, middle and right. Assigning the indices as follows,

$$k\in\Delta^{\mathrm{L}}_j \quad \text{for} \quad \phi^{\mathrm{L}}_{j,k}\,, \qquad k\in\Delta^{0}_j \quad \text{for} \quad \phi_{[j,k]}\,, \qquad k\in\Delta^{\mathrm{R}}_j \quad \text{for} \quad \phi^{\mathrm{R}}_{j,k}\,, \tag{3.3.28a}$$

$$k\in\tilde{\Delta}^{\mathrm{L}}_j \quad \text{for} \quad \tilde{\phi}^{\mathrm{L}}_{j,k}\,, \qquad k\in\tilde{\Delta}^{0}_j \quad \text{for} \quad \tilde{\phi}_{[j,k]}\,, \qquad k\in\tilde{\Delta}^{\mathrm{R}}_j \quad \text{for} \quad \tilde{\phi}^{\mathrm{R}}_{j,k}\,, \tag{3.3.28b}$$

where

$$\Delta^{\mathrm{L}}_j := \{\ell-d,\dots,\ell-1\}\,, \tag{3.3.29a}$$

$$\Delta^{0}_j := \{\ell,\dots,2^j-\ell-\mu(d)\}\,, \tag{3.3.29b}$$

$$\Delta^{\mathrm{R}}_j := \{2^j-\ell-\mu(d)+1,\dots,2^j-\ell-\mu(d)+d\}\,, \tag{3.3.29c}$$

and likewise for the dual side with parameters $\tilde{d}$ and $\tilde{\ell}$, we can set

$$\Delta_j := \Delta^{\mathrm{L}}_j \cup \Delta^{0}_j \cup \Delta^{\mathrm{R}}_j\,, \qquad \tilde{\Delta}_j := \tilde{\Delta}^{\mathrm{L}}_j \cup \tilde{\Delta}^{0}_j \cup \tilde{\Delta}^{\mathrm{R}}_j\,, \tag{3.3.30}$$

and a first version of the single-scale bases for the spaces S_j and $\tilde{S}_j$ on the interval is given by

$$\Phi^{(0)}_j := \{\phi^{\mathrm{L}}_{j,k}\} \cup \{\phi_{[j,k]}\} \cup \{\phi^{\mathrm{R}}_{j,k}\}\,, \qquad \tilde{\Phi}^{(0)}_j := \{\tilde{\phi}^{\mathrm{L}}_{j,k}\} \cup \{\tilde{\phi}_{[j,k]}\} \cup \{\tilde{\phi}^{\mathrm{R}}_{j,k}\}\,. \tag{3.3.31}$$

Note in particular that the functions adapted to the left and right borders are refinable, cf. (3.3.23). Defining $\beta_{m,r}$ analogously to $\tilde{\beta}_{m,r}$ (3.3.24), and setting

$$\tilde{\beta}^{\mathrm{L}}_{j,m,r} := \tilde{\beta}_{m,r}\,, \qquad \tilde{\beta}^{\mathrm{R}}_{j,m,r} := \tilde{\beta}_{2^{j+1}-m-\mu(d),r}\,, \qquad r=0,\dots,d-1\,, \tag{3.3.32a}$$

$$\beta^{\mathrm{L}}_{j,m,r} := \beta_{m,r}\,, \qquad \beta^{\mathrm{R}}_{j,m,r} := \beta_{2^{j+1}-m-\mu(d),r}\,, \qquad r=0,\dots,\tilde{d}-1\,, \tag{3.3.32b}$$

we obtain the refinement relations

$$\begin{aligned}\phi^{\mathrm{L}}_{j,\ell-d+r} = 2^{-(r+\frac{1}{2})}&\left(\phi^{\mathrm{L}}_{j+1,\ell-d+r} + \sum_{m=\ell}^{2\ell+\ell_1-1} \tilde{\alpha}^{\mathrm{L}}_{j,m,r}\phi_{[j+1,m]}\right)\\ &+ \sum_{m=2\ell+\ell_1}^{2\ell+\ell_2-2} \tilde{\beta}^{\mathrm{L}}_{j,m,r}\phi_{[j+1,m]}\,, \qquad r=0,\dots,d-1\end{aligned} \tag{3.3.33a}$$

and

$$\begin{aligned}\phi^{\mathrm{R}}_{j,2^j-\ell-\mu(d)+d-r} = 2^{-(r+\frac{1}{2})}&\left(\phi^{\mathrm{R}}_{j+1,2^{j+1}-\ell-\mu(d)+d-r} + \sum_{m=2^{j+1}-2\ell-\ell_1-\mu(d)+1}^{2^{j+1}-\ell-\mu(d)} \tilde{\alpha}^{\mathrm{R}}_{j,m,r}\phi_{[j+1,m]}\right)\\ &+ \sum_{m=2^{j+1}-2\ell-\ell_2-\mu(d)+2}^{2^{j+1}-2\ell-\ell_1-\mu(d)} \tilde{\beta}^{\mathrm{R}}_{j,m,r}\phi_{[j+1,m]}\,, \qquad r=0,\dots,d-1\,.\end{aligned} \tag{3.3.33b}$$

The dual relations are defined analogously. Writing these relations in matrix form produces the boundary blocks of the refinement matrices $\mathbf{M}^{(0)}_{j,0}$ and $\tilde{\mathbf{M}}^{(0)}_{j,0}$, while their interior blocks are given by the masks $\mathbf{a}$ and $\tilde{\mathbf{a}}$, see Table 3.11.

Remark 3.7. *The number of boundary functions on the primal side is always d, independent of the length of their support which is controlled by ℓ. The number of boundary functions on the dual side is $\tilde{d}$, independent of $\tilde{\ell}$. It follows that a choice of*

$$\tilde{\ell} = \ell + (\tilde{d} - d) \tag{3.3.34}$$

leads to equal cardinality of the primal and dual bases on the interval, which is a prerequisite for biorthogonality.

Remark 3.8. *Homogeneous boundary conditions on either the primal or the dual side, or on both sides, and either on the right or the left end, or on both ends, in any combination, can be achieved by deleting the monomial of degree* 0 *(corresponding to the constant $x^0 = 1$) from the appropriate index set(s). If different boundary conditions are to be fulfilled on the left and the right end, the parameters ℓ and/or $\tilde{\ell}$ might additionally need to be chosen differently on the left and the right end. Equal cardinality of the primal and dual bases can always be reestablished by a suitable modification of* (3.3.34).

Reestablishing Biorthogonality

In the previous paragraph, we have constructed primal and dual multiresolutions restricted to the interval $(0, 1)$. We still need to confirm that the collections of functions $\Phi_j^{(0)}$ and $\tilde{\Phi}_j^{(0)}$ are respectively linearly independent. Secondly, biorthogonality has to be reestablished, which has been lost due to the restriction. The route proposed in [49] is to introduce local transformations on the left and right boundaries which lead again to a biorthogonal basis. Linear independence is then a trivial consequence.

As we have $d \leq \tilde{d}$, the primal index set Δ_j^{L} for the left boundary functions is generally smaller than the dual set $\tilde{\Delta}_j^{\mathrm{L}}$. By enlarging Δ_j^{L} with the $\tilde{d} - d$ leftmost primal functions from the inner set Δ_j^0, and repeating the procedure analogously for the right end of the interval, we can define the square matrices

$$\mathbf{\Gamma}_j^{\mathrm{L}} := \Big((\phi_{j,k}, \tilde{\phi}_{j,k'})_{[0,1]} \Big)_{k,k' \in \tilde{\Delta}_j^{\mathrm{L}}} , \qquad \mathbf{\Gamma}_j^{\mathrm{R}} := \Big((\phi_{j,k}, \tilde{\phi}_{j,k'})_{[0,1]} \Big)_{k,k' \in \tilde{\Delta}_j^{\mathrm{R}}} . \tag{3.3.35}$$

We cite from [49] the following central

Theorem 3.9. *The matrices $\mathbf{\Gamma}_j^{\mathrm{X}}$, $\mathrm{X} \in \{\mathrm{L}, \mathrm{R}\}$, are independent of j and symmetric with respect to the left and right sides,*

$$\mathbf{\Gamma}_j^{\mathrm{L}} = \mathbf{\Gamma}^{\mathrm{L}} = \mathbf{\Gamma} , \qquad \mathbf{\Gamma}_j^{\mathrm{R}} = \mathbf{\Gamma}^{\mathrm{R}} = \mathbf{\Gamma}^{\updownarrow} . \tag{3.3.36}$$

In the situation of Theorem 3.6, $\mathbf{\Gamma}$ is always nonsingular.

It follows that the bases defined by

$$\Phi_j := \Phi_j^{(0)} , \qquad \tilde{\Phi}_j := \mathbf{\Gamma}_j^{-T} \tilde{\Phi}_j^{(0)} := \begin{pmatrix} \mathbf{\Gamma}^{\mathrm{L}} & & \\ & \mathbf{I}_{\#\tilde{\Delta}_j^0} & \\ & & \mathbf{\Gamma}^{\mathrm{R}} \end{pmatrix}^{-T} \tilde{\Phi}_j^{(0)} \tag{3.3.37}$$

are biorthogonal. The primal refinement relation is not affected by this transformation, while the dual refinement matrix changes,

$$\mathbf{M}_{j,0} = \mathbf{M}_{j,0}^{(0)} , \qquad \tilde{\mathbf{M}}_{j,0} = \mathbf{\Gamma}_{j+1} \tilde{\mathbf{M}}_{j,0}^{(0)} \mathbf{\Gamma}_j^{-1} . \tag{3.3.38}$$

However, an exact calculation of the entries of $\mathbf{\Gamma}$ requires some non-trivial calculations [49]. From (3.3.26) and (3.3.27), we infer for $r = 0, \ldots, d-1$ and $s = 0, \ldots, \tilde{d} - 1$

$$\Big(\phi^{\mathrm{L}}_{\ell-d+r}, \tilde{\phi}^{\mathrm{L}}_{\tilde{\ell}-\tilde{d}+s} \Big)_{[0,1]} = \sum_{\mu=-\ell_2+1}^{\ell-1} \sum_{\nu=-\tilde{\ell}_2+1}^{\tilde{\ell}-1} \tilde{\alpha}_{\mu,r} \alpha_{\nu,s} \int_0^\infty \phi(x-\mu) \tilde{\phi}(x-\nu) \,\mathrm{d}x , \tag{3.3.39}$$

and for $r = d, \dots, \tilde{d} - 1$ and $s = 0, \dots, \tilde{d} - 1$

$$\left(\phi_{[j,\ell-d+r]}, \tilde{\phi}^{\mathrm{L}}_{j,\tilde{\ell}-\tilde{d}+s}\right)_{[0,1]} = \sum_{\nu=-\tilde{\ell}_2+1}^{\tilde{\ell}-1} \alpha_{\nu,s} \int_0^\infty \phi(x - (\ell - d + r))\tilde{\phi}(x - \nu)\,\mathrm{d}x\,. \tag{3.3.40}$$

Obviously, these expressions can be reduced to a calculation of

$$I(\mu, \nu) \coloneqq \int_0^\infty \phi(x - \mu)\tilde{\phi}(x - \nu)\,\mathrm{d}x\,. \tag{3.3.41}$$

From the length of the support of ϕ and $\tilde{\phi}$, we deduce that

$$I(\mu, \nu) = \begin{cases} 0 & \text{for} \quad \mu \le -\ell_2 \quad \text{or} \quad \nu \le -\tilde{\ell}_2\,, \\ \delta_{\mu,\nu} & \text{for} \quad -\ell_1 \le \mu \quad \text{or} \quad -\tilde{\ell}_1 \le \nu\,. \end{cases} \tag{3.3.42}$$

It remains to determine the values for $I(\mu, \nu)$ in the remaining range of μ, ν. We reformulate

$$I(\mu, \nu) = \sum_{i=0}^{\mu+\ell_2-1} \int_{\mathbb{R}} \chi_{[i,i+1]}(x)\phi(x - \mu)\tilde{\phi}(x - \nu)\,\mathrm{d}x = \sum_{i=0}^{\mu+\ell_2-1} z(\mu - i, \nu - i)\,, \tag{3.3.43}$$

where we have used the definition

$$z(\mu, \nu) \coloneqq \int_{\mathbb{R}} \chi_{[0,1]}(x)\phi(x - \mu)\tilde{\phi}(x - \nu)\,\mathrm{d}x = \int_0^1 \phi(x - \mu)\tilde{\phi}(x - \nu)\,\mathrm{d}x\,. \tag{3.3.44}$$

Inserting the refinement relations for χ, ϕ and $\tilde{\phi}$ in the form of (3.3.1), we obtain the identity

$$z(\beta) = \sum_\gamma c_\gamma z(2\beta + \gamma) = \sum_\eta c_{\beta,\eta} z(\eta) \tag{3.3.45}$$

with the definition

$$c_{\beta,\eta} \coloneqq \frac{1}{2}\left(\alpha_{\eta_1-2\beta_1}\tilde{\alpha}_{\eta_2-2\beta_2} + \alpha_{\eta_1-2\beta_1+1}\tilde{\alpha}_{\eta_2-2\beta_2+1}\right)\,, \tag{3.3.46}$$

where β, γ and η are two-dimensional indices. Due to the finite support of ϕ and $\tilde{\phi}$, the index sets are also finite. This leads to an eigenvector problem for z with eigenvalue 1, whose solution is uniquely determined by the normalisation

$$\sum_\beta z(\beta) = 1\,. \tag{3.3.47}$$

We solve this eigenvector problem for z and insert the values into the equation for $I(\mu, \nu)$ (3.3.43), which in turn can be used to compute the entries of Γ. After these steps, the biorthogonalisation is complete [50].

3.3.2 Identifying a Stable Completion

In the construction of finite element wavelets we have built a stable completion by an initial guess and suitable postprocessing. In [49], the initial stable completion $\check{\mathbf{M}}_{j,1}$ is deducted in a systematic way from the entries of $\mathbf{M}_{j,0}$. Additionally, this ansatz delivers an explicit form of the inverse $\check{\mathbf{G}}_j$. We give a short overview of this process.

The structure of the primal refinement matrix $\mathbf{M}_{j,0}$ is shown on the left of Figure 3.5. The border blocks, which are independent of j, are denoted by $\mathbf{M}_{\mathrm{L}}$ and $\mathbf{M}_{\mathrm{R}}$. Their entries can be computed from the refinement relations on the boundary (3.3.33). A closer look at these equations asserts that the left

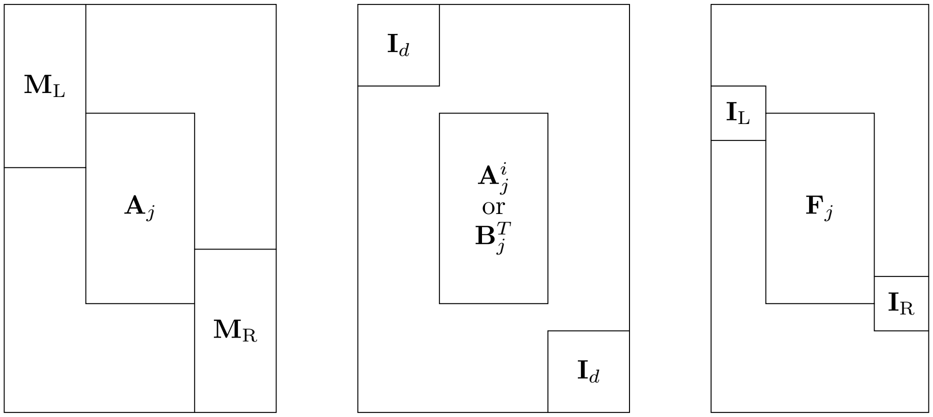

$\mathbf{M}_{\mathrm{L}}$
$\mathbf{A}_j$
$\mathbf{M}_{\mathrm{R}}$
$\mathbf{I}_d$
$\mathbf{A}_j^i$
or
$\mathbf{B}_j^T$
$\mathbf{I}_d$
$\mathbf{I}_{\mathrm{L}}$
$\mathbf{F}_j$
$\mathbf{I}_{\mathrm{R}}$

Figure 3.5: On the left, we display the block structure of the primal refinement matrix $\mathbf{M}_{j,0}$. The structure of the matrices $\hat{\mathbf{A}}_j^i$ and $\hat{\mathbf{B}}_j^T$ is shown in the middle, and $\hat{\mathbf{F}}_j$ on the right. All matrices have the dimensions $\#\Delta_{j+1} \times \#\Delta_j$ except $\hat{\mathbf{F}}_j$, which is of the size $\#\Delta_{j+1} \times \#\nabla_j$. The identity matrices inside of $\hat{\mathbf{F}}_j$ are defined as $\mathbf{I}_{\mathrm{L}} = \mathbf{I}_{\ell+\mu(d)-1}$ and $\mathbf{I}_{\mathrm{R}} = \mathbf{I}_\ell$. The interior rectangular block has the same dimensions $q(j) \times p(j)$ for all matrices.

block is split vertically into two parts. The top block is made up of a scaled identity matrix while the bottom block contains combinations of $\tilde{\alpha}^{\mathrm{L}}$ and $\tilde{\beta}^{\mathrm{L}}$. The right border block is specified by symmetry,

$$\mathbf{M}_{\mathrm{L}} = \mathbf{M}_{\mathrm{R}}^{\updownarrow}\,. \tag{3.3.48}$$

The interior part contains repeated columns of the mask vector $\mathbf{a}$,

$$(\mathbf{A}_j)_{m,k} = \frac{1}{\sqrt{2}} a_{m-2k}\,, \qquad 2\ell + \ell_1 \leq m \leq 2^{j+1} + \ell_2 - 2(\ell + \mu(d))\,, \qquad k \in \Delta_j^0\,. \tag{3.3.49}$$

The dimensions of $\mathbf{A}_j$ are $q(j) \times p(j)$ with

$$p(j) \coloneqq \#\Delta_j^0 = 2^j - 2\ell - \mu(d) + 1\,, \tag{3.3.50a}$$

$$q(j) \coloneqq 2p(j) + d - 1 = 2^{j+1} - 4\ell - 2\mu(d) + d + 1\,. \tag{3.3.50b}$$

To eventually find the initial completion $\check{\mathbf{M}}_{j,1}$ and also an explicit inverse of $\check{\mathbf{M}}_j$, we aim to find a reversible transformation to reduce $\mathbf{M}_{j,0}$ to a simple structure. The key idea is to perform Gaussian elimination on $\mathbf{A}_j \eqqcolon \mathbf{A}_j^0$, which is the interior part of $\mathbf{M}_{j,0}$, alternating from above and below, such that after d steps the matrix contains only one constant nonzero entry per column, according to

$$(\mathbf{A}_j^d)_{m,k} = b\,\delta_{m-2k,\mu(d)}\,. \tag{3.3.51}$$

This is accomplished by multiplication from the left with the cumulative product of d square transformation matrices $\mathbf{H}_j^i$ of dimension $q(j)$, which we denote by $\mathbf{H}_j \coloneqq \mathbf{H}_j^d \cdots \mathbf{H}_j^1$. Because of the special structure of $\mathbf{A}_j^d$, the matrix $\mathbf{B}_j \coloneqq b^{-2}(\mathbf{A}_j^d)^T$ satisfies

$$\mathbf{B}_j \mathbf{A}_j^d = \mathbf{I}_{p(j)}\,. \tag{3.3.52}$$

Additionally, we define the matrix $\mathbf{F}_j$ of the same dimension and pattern as $\mathbf{A}_j^d$, except that all entries are shifted up by one row,

$$(\mathbf{F}_j)_{m,k} = \delta_{m-2k,\mu(d)-1}\,. \tag{3.3.53}$$

This is done to ensure that $\mathbf{B}_j \mathbf{F}_j = 0$.

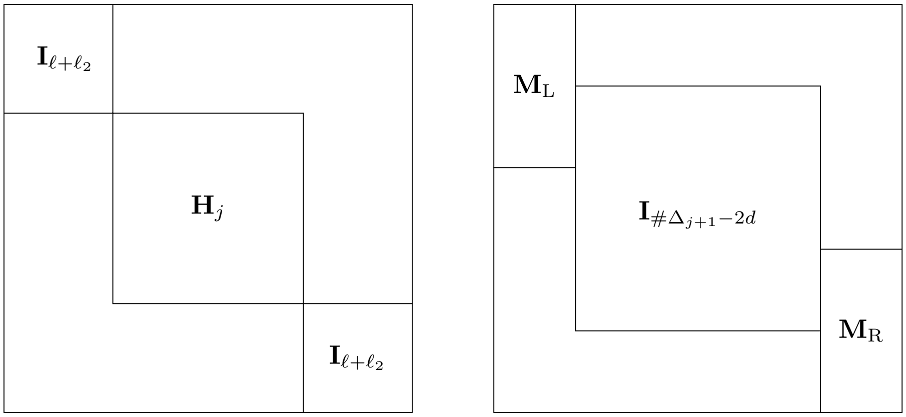

$\mathbf{I}_{\ell+\ell_2}$
$\mathbf{H}_j$
$\mathbf{I}_{\ell+\ell_2}$
$\mathbf{M}_{\mathrm{L}}$
$\mathbf{I}_{\#\Delta_{j+1}-2d}$
$\mathbf{M}_{\mathrm{R}}$

Figure 3.6: We show two square matrices of dimension $\#\Delta_{j+1}$. On the left hand side we display the extended transformation matrix $\hat{\mathbf{H}}_j$, and on the right hand side we display $\mathbf{P}_j$ which is used to reconstruct $\mathbf{M}_{j,0}$ from $\hat{\mathbf{A}}_j^0$.

Thus heaving dealt with the interior part of the refinement matrix, we extend the results to include the original border blocks. To this end, we pad the newly introduced matrices with zeros and identity blocks. As shown in the middle and on the right of Figure 3.5, we do this for

$$\mathbf{A}_j^i \mapsto \hat{\mathbf{A}}_j^i\,, \qquad \mathbf{B}_j^T \mapsto \hat{\mathbf{B}}_j^T\,, \qquad \mathbf{F}_j \mapsto \hat{\mathbf{F}}_j\,. \tag{3.3.54}$$

Similarly, the square transformation matrix $\mathbf{H}_j$ is embedded into a larger square matrix $\hat{\mathbf{H}}_j$. It is depicted on the left hand side of Figure 3.6. This allows for the formulation

$$\hat{\mathbf{A}}_j^0 = \hat{\mathbf{H}}_j^{-1}\hat{\mathbf{A}}_j^d\,. \tag{3.3.55}$$

Lemma 3.10. *With above definitions, and for any $r \neq 0$, it holds that*

$$\begin{pmatrix} \hat{\mathbf{B}}_j \\ r^{-1}\hat{\mathbf{F}}_j^T \end{pmatrix} (\hat{\mathbf{A}}_j^d, r\hat{\mathbf{F}}_j) = \begin{pmatrix} \mathbf{I}_{\#\Delta_j} & 0 \\ 0 & \mathbf{I}_{\#\nabla_j} \end{pmatrix} = \mathbf{I}_{\#\Delta_{j+1}}\,. \tag{3.3.56}$$

Compared to [49], we have introduced the parameter r which will be selected based on numerical considerations. In summary, we have completed $\hat{\mathbf{A}}_j^d$ with the matrix $r\hat{\mathbf{F}}_j$, which allows to specify an explicit inverse. We now invert the linear transformation $\mathbf{H}_j$ to return to the formulation in terms of $\check{\mathbf{M}}_j$. To this end, we define the square matrix $\mathbf{P}_j$ in such a way that

$$\mathbf{M}_{j,0} = \mathbf{P}_j\hat{\mathbf{A}}_j^0\,. \tag{3.3.57}$$

The layout of $\mathbf{P}_j$ is shown on the right of Figure 3.6. After these preparations, we define the initial stable completion by

$$\check{\mathbf{M}}_j := (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1}) := \mathbf{P}_j\hat{\mathbf{H}}_j^{-1}(\hat{\mathbf{A}}_j^d, r\hat{\mathbf{F}}_j)\,. \tag{3.3.58}$$

It follows from (3.3.56) that the inverse of $\check{\mathbf{M}}_j$ is then given by

$$\check{\mathbf{G}}_j = \begin{pmatrix} \check{\mathbf{G}}_{j,0} \\ \check{\mathbf{G}}_{j,1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{B}}_j \\ r^{-1}\hat{\mathbf{F}}_j^T \end{pmatrix} \hat{\mathbf{H}}_j\hat{\mathbf{P}}_j^{-1}\,. \tag{3.3.59}$$

All matrices involved in this construction contain an interior part which is built from staggered repetitions of one individual column, and whose entries are thus independent of the level j. It follows that the matrices

$\mathbf{H}_j^i$ and consequently $\mathbf{H}_j$ also feature this kind of periodicity. Consequently, the matrices $\check{\mathbf{M}}_j$ and $\check{\mathbf{G}}_j$ are uniformly sparse. Ultimately, we obtain

$$\|\check{\mathbf{M}}_j\|, \|\check{\mathbf{G}}_j\| \sim 1\,. \tag{3.3.60}$$

We have thus identified a stable completion satisfying all requirements of Theorem 2.13. In addition, we have obtained an explicit representation of $\check{\mathbf{G}}_{j,1}$ which is uniformly sparse.

Remark 3.11. *When d is odd, we have $\mu(d) = 1$ and $\ell_2 \neq -\ell_1$. It follows that in this case $\hat{\mathbf{F}}_j \neq \hat{\mathbf{F}}_j^{\updownarrow}$, which is inherited by $\mathbf{M}_j$ and $\mathbf{G}_j$. Consequently, the wavelets are not symmetric. It has been shown in [52] how to achieve a symmetric construction also for an odd d.*

3.3.3 Construction Details and Modifications

In the previous section, we have covered the construction of an initial stable completion with an explicitly given inverse. To complete the process of constructing biorthogonal wavelets, we apply the projection of this stable completion precisely as described in Theorem 2.13.

We deviate from the original construction in two respects. First, we choose the minimal values for ℓ, $\tilde{\ell}$ and j, in the case of free boundary conditions on the dual side and free or homogeneous boundary conditions on the primal side. We define $I_b := 0$ for free and $I_b := 1$ for homogeneous primal boundary conditions. Then the relation (3.3.34) between ℓ and $\tilde{\ell}$ is generalised to

$$\tilde{\ell} = \ell + (\tilde{d} - d) + I_b\,. \tag{3.3.61}$$

There are two conditions for the smallest value of ℓ.

(i) The support of the interior functions must be contained in $[0,1]$, which leads to

$$\ell \geq -\ell_1\,, \tag{3.3.62a}$$

$$\tilde{\ell} \geq -\tilde{\ell}_1 \qquad \Longleftrightarrow \qquad \ell \geq -\ell_1 + (d-1) - I_b\,. \tag{3.3.62b}$$

(ii) The boundary functions must be linearly independent, which means that the sum in (3.3.26) must contain enough entries,

$$\ell \geq -\ell_1 + 1 - I_b\,. \tag{3.3.63}$$

Together, this leads to the requirement

$$\ell \geq -\ell_1 + \max\{d-1, 1\} - I_b\,. \tag{3.3.64}$$

In [49], the smallest level has been set to

$$j \geq j_0' = \lceil \log_2(\tilde{\ell}_2 + \tilde{\ell} - 1) + 1 \rceil\,, \tag{3.3.65}$$

which corresponds to the obligation that the supports of the left and right boundary functions should not overlap. We propose to relax this to the extent that they may indeed overlap, but must still be contained in $[0,1]$. This means that

$$j \geq j_0 = \lceil \log_2(2\tilde{\ell} - 1 + \mu(d)) \rceil\,. \tag{3.3.66}$$

In the case of $d = 2$ and $\tilde{d} = 4$, our formula allows for a lowest level of $j_0 = 3$ in contrast to $j_0' = 4$.

Secondly, we set $r = \sqrt{2}$ in (3.3.58). This is done to account for the same factor in the definition of A_j in (3.3.49). As a side effect, it is then possible to describe all matrices with entries from $\sqrt{2}\mathbb{Q}$. As in the example of finite element matrices in Section 3.2, this leads to a more natural formulation. We will explicitly display the refinement matrices in the next section.

1	0
1	1
1	2

$\frac{1}{2}$	$\frac{1}{2}$

1	-3	$\frac{55}{6}$	$-\frac{57}{2}$
1	-2	$\frac{25}{6}$	-9
1	-1	$\frac{7}{6}$	$-\frac{3}{2}$
1	0	$\frac{1}{6}$	0
1	1	$\frac{7}{6}$	$\frac{3}{2}$
1	2	$\frac{25}{6}$	9
1	3	$\frac{55}{6}$	$\frac{57}{2}$

$\frac{61}{64}$	$\frac{29}{16}$	$\frac{1261}{384}$	$\frac{165}{32}$
$\frac{35}{32}$	$\frac{23}{8}$	$\frac{1499}{192}$	$\frac{177}{8}$
$\frac{77}{64}$	$\frac{241}{64}$	$\frac{4571}{384}$	$\frac{4851}{128}$
$\frac{1}{2}$	$\frac{51}{32}$	$\frac{485}{96}$	$\frac{1029}{64}$
$-\frac{13}{64}$	$-\frac{21}{32}$	$-\frac{805}{384}$	$-\frac{429}{64}$
$-\frac{3}{32}$	$-\frac{9}{32}$	$-\frac{55}{64}$	$-\frac{171}{64}$
$\frac{3}{64}$	$\frac{9}{64}$	$\frac{55}{128}$	$\frac{171}{128}$

Table 3.10: We show on the left the matrices $\tilde{\alpha}$ and $\sqrt{2}\tilde{\beta}$ which contain the primal coefficients for polynomial exactness from (3.3.6a)*, and refinement from* (3.3.24)*, respectively. They have $d = 2$ columns. On the right, we display the matrices α and $\sqrt{2}\beta$, which play the same role for the dual generators and consequently have $\tilde{d} = 4$ columns. The matrices $\tilde{\alpha}$ and α may exhibit more rows than strictly necessary by* (3.3.26)*,* (3.3.27)*, as their entries are also needed for the calculation of $\mathbf{\Gamma}$ in* (3.3.36)*.*

Construction with Rational Numbers

We will now provide explicit results for the case $d = 2$ and $\tilde{d} = 4$. We have calculated them for both homogeneous and free boundary conditions on the primal side. As the numbers occurring in both cases are mostly similar, we display only results for the case of free boundary conditions, i.e., $I_b = 0$.

We begin with the coefficients $\tilde{\alpha}$ and α which control the polynomial reconstruction (3.3.6), and $\tilde{\beta}$ and β which contain the information about the refinement of the boundary functions (3.3.24). These matrices are displayed in Table 3.10. The refinement matrix $\mathbf{M}_{j,0}^{(0)}$ for the primal basis is derived from the values $\tilde{\alpha}$, $\tilde{\beta}$ and a_k, while the dual matrix $\tilde{\mathbf{M}}_{j,0}^{(0)}$ is derived from α, β and $\tilde{a}_k$. We display them both in Table 3.11. Note that these refinement matrices correspond to the bases $\Phi_j^{(0)}$ and $\tilde{\Phi}_j^{(0)}$, which are not yet biorthogonal at the boundary.

In Table 3.12 we present the matrices needed for biorthogonalisation, $\mathbf{\Gamma}$ and its inverse $\mathbf{\Gamma}^{-1}$. They have been calculated according to (3.3.39) and (3.3.40). The dual refinement matrix after the biorthogonalisation (3.3.37) is shown in Table 3.13. Note that only the boundary blocks have changed relatively to Table 3.11, and the banded structure is clearly visible.

Then the initial stable completion is constructed as in (3.3.58), (3.3.59), and the lifting steps (2.3.2) and (2.3.3) are performed. This produces the final transformation matrices $\mathbf{M}_j$ and $\tilde{\mathbf{M}}_j = \mathbf{M}_j^{-T}$. We display them in Table 3.14 and Table 3.15, respectively. It becomes immediately obvious that both matrices exhibit a banded structure of repeated staggered columns. The calculation with rational numbers and our choice of r reveals a striking symmetry between the primal and the dual sides. The numbers on the left of $\mathbf{M}_j$ are identical to those on the right of $\tilde{\mathbf{M}}_j$ (irrespective of the sign), and vice versa.

3.3.4 Plotting of Representations in the Dual Basis

The dual generator functions from [42] are only given in implicit form. A function value for any given $x \in [0, 1]$ can be computed using recursion formulas. To plot the representation of a function f in the dual single-scale basis,

$$f_{(j)} = \tilde{\mathbf{c}}_j^T \tilde{\Phi}_j \,, \tag{3.3.67}$$

we need to evaluate the basis functions repeatedly on a certain grid, which is computationally expensive. Thus, we seek a way to evaluate the dual generators $\tilde{\phi}_{j,k}$ only once to a prescribed accuracy and to precalculate an approximate transformation from the dual single-scale basis to the piecewise linear nodal basis Z_j,

$$f_{(j)} \leftrightarrow \mathbf{z}_j^T Z_j \,, \qquad Z_j := \{\zeta_{j,k}(x) : k \in \Delta_j^Z\} \,, \qquad \zeta_{j,k}(2^{-j} l) = \delta_{k,l} \,. \tag{3.3.68}$$

1	0	0	0	0	0	0	0	0
0	$\frac{1}{2}$	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	$\frac{1}{2}$	0
0	0	0	0	0	0	0	0	1

1	0	0	0	0	0	0	0	0
0	$\frac{1}{2}$	0	0	0	0	0	0	0
0	0	$\frac{1}{4}$	0	0	0	0	0	0
0	0	0	$\frac{1}{8}$	0	0	0	0	0
$\frac{61}{64}$	$\frac{29}{16}$	$\frac{1261}{384}$	$\frac{165}{32}$	$\frac{3}{64}$	0	0	0	0
$\frac{35}{32}$	$\frac{23}{8}$	$\frac{1499}{192}$	$\frac{177}{8}$	$-\frac{3}{32}$	0	0	0	0
$\frac{77}{64}$	$\frac{241}{64}$	$\frac{4571}{384}$	$\frac{4851}{128}$	$-\frac{1}{4}$	$\frac{171}{128}$	$\frac{55}{128}$	$\frac{9}{64}$	$\frac{3}{64}$
$\frac{1}{2}$	$\frac{51}{32}$	$\frac{485}{96}$	$\frac{1029}{64}$	$\frac{19}{32}$	$-\frac{171}{64}$	$-\frac{55}{64}$	$-\frac{9}{32}$	$-\frac{3}{32}$
$-\frac{13}{64}$	$-\frac{21}{32}$	$-\frac{805}{384}$	$-\frac{429}{64}$	$\frac{45}{32}$	$-\frac{429}{64}$	$-\frac{805}{384}$	$-\frac{21}{32}$	$-\frac{13}{64}$
$-\frac{3}{32}$	$-\frac{9}{32}$	$-\frac{55}{64}$	$-\frac{171}{64}$	$\frac{19}{32}$	$\frac{1029}{64}$	$\frac{485}{96}$	$\frac{51}{32}$	$\frac{1}{2}$
$\frac{3}{64}$	$\frac{9}{64}$	$\frac{55}{128}$	$\frac{171}{128}$	$-\frac{1}{4}$	$\frac{4851}{128}$	$\frac{4571}{384}$	$\frac{241}{64}$	$\frac{77}{64}$
0	0	0	0	$-\frac{3}{32}$	$\frac{177}{8}$	$\frac{1499}{192}$	$\frac{23}{8}$	$\frac{35}{32}$
0	0	0	0	$\frac{3}{64}$	$\frac{165}{32}$	$\frac{1261}{384}$	$\frac{29}{16}$	$\frac{61}{64}$
0	0	0	0	0	$\frac{1}{8}$	0	0	0
0	0	0	0	0	0	$\frac{1}{4}$	0	0
0	0	0	0	0	0	0	$\frac{1}{2}$	0
0	0	0	0	0	0	0	0	1

Table 3.11: We present the refinement matrices $\sqrt{2}\mathbf{M}_{j,0}^{(0)}$ on the left and $\sqrt{2}\tilde{\mathbf{M}}_{j,0}^{(0)}$ on the right, for the smallest level $j = j_0$. They are both matrices of dimension 17×9. The effect of the inhomogeneous boundary conditions can be seen on the top left and bottom right corners of $\mathbf{M}_{j,0}$. For homogeneous boundary conditions, the outmost rows and columns of $\mathbf{M}_{j,0}^{(0)}$ must be deleted, yielding a translation invariant matrix of dimensions 15×7. The change in $\tilde{\mathbf{M}}_{j,0}^{(0)}$ is more complicated, as $\tilde{\ell}$ must also be changed.

$\frac{3}{2}$	$\frac{7}{6}$	$\frac{5}{4}$	$\frac{31}{20}$
1	1	$\frac{7}{6}$	$\frac{3}{2}$
1	2	$\frac{25}{6}$	9
1	3	$\frac{55}{6}$	$\frac{57}{2}$

$\frac{25}{6}$	$-\frac{415}{72}$	$\frac{23}{36}$	$-\frac{1}{8}$
$-\frac{35}{4}$	$\frac{755}{48}$	$-\frac{79}{24}$	$\frac{11}{16}$
5	$-\frac{119}{12}$	$\frac{19}{6}$	$-\frac{3}{4}$
$-\frac{5}{6}$	$\frac{125}{72}$	$-\frac{25}{36}$	$\frac{5}{24}$

Table 3.12: We show the Gramian matrices $\mathbf{\Gamma}^{\mathrm{L}}$ and its inverse, which are needed for biorthogonalisation after the restriction to the interval. They are of dimension $\tilde{d} = 4$.

$\frac{163}{64}$	$-\frac{1715}{768}$	$-\frac{41}{384}$	$\frac{5}{256}$	0	0	0	0	0
$\frac{35}{32}$	$-\frac{179}{384}$	$-\frac{41}{192}$	$\frac{5}{128}$	0	0	0	0	0
$-\frac{5}{16}$	$\frac{305}{192}$	$-\frac{13}{96}$	$\frac{1}{64}$	0	0	0	0	0
$-\frac{15}{32}$	$\frac{165}{128}$	$\frac{31}{64}$	$-\frac{9}{128}$	0	0	0	0	0
$\frac{15}{64}$	$-\frac{153}{256}$	$\frac{187}{128}$	$-\frac{67}{256}$	$\frac{3}{64}$	0	0	0	0
0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0	0	0
0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0
0	0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0	0
0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0
0	0	0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0
0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0
0	0	0	0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0
0	0	0	0	$\frac{3}{64}$	$-\frac{67}{256}$	$\frac{187}{128}$	$-\frac{153}{256}$	$\frac{15}{64}$
0	0	0	0	0	$-\frac{9}{128}$	$\frac{31}{64}$	$\frac{165}{128}$	$-\frac{15}{32}$
0	0	0	0	0	$\frac{1}{64}$	$-\frac{13}{96}$	$\frac{305}{192}$	$-\frac{5}{16}$
0	0	0	0	0	$\frac{5}{128}$	$-\frac{41}{192}$	$-\frac{179}{384}$	$\frac{35}{32}$
0	0	0	0	0	$\frac{5}{256}$	$-\frac{41}{384}$	$-\frac{1715}{768}$	$\frac{163}{64}$

Table 3.13: Here we display the dual refinement matrix $\sqrt{2}\tilde{\mathbf{M}}_{j,0}$ after biorthogonalisation. The middle column is the same as that from the right of Table 3.11, while the boundary blocks have changed. At this point, they have adopted the banded structure which can also be seen in the primal matrix.

1	0	0	0	0	0	0	0	0	$\frac{5}{16}$	$\frac{15}{32}$	0	0	0	0	0	0
0	$\frac{1}{2}$	0	0	0	0	0	0	0	$-\frac{305}{384}$	$-\frac{165}{256}$	$\frac{3}{64}$	0	0	0	0	0
1	1	0	0	0	0	0	0	0	$\frac{139}{192}$	$-\frac{105}{128}$	$\frac{3}{32}$	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	$-\frac{73}{128}$	$\frac{345}{256}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0	0
0	0	1	0	0	0	0	0	0	$\frac{13}{96}$	$-\frac{31}{64}$	$-\frac{19}{32}$	$\frac{3}{32}$	0	0	0	0
0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{23}{384}$	$-\frac{53}{256}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0
0	0	0	1	0	0	0	0	0	$-\frac{1}{64}$	$\frac{9}{128}$	$-\frac{19}{32}$	$-\frac{19}{32}$	$\frac{3}{32}$	0	0	0
0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$-\frac{1}{128}$	$\frac{9}{256}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0
0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{19}{32}$	$-\frac{19}{32}$	$\frac{3}{32}$	0	0
0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{9}{256}$	$-\frac{1}{128}$
0	0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{19}{32}$	$-\frac{19}{32}$	$\frac{9}{128}$	$-\frac{1}{64}$
0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{53}{256}$	$\frac{23}{384}$
0	0	0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{19}{32}$	$-\frac{31}{64}$	$\frac{13}{96}$
0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{345}{256}$	$-\frac{73}{128}$
0	0	0	0	0	0	0	1	1	0	0	0	0	0	$\frac{3}{32}$	$-\frac{105}{128}$	$\frac{139}{192}$
0	0	0	0	0	0	0	$\frac{1}{2}$	0	0	0	0	0	0	$\frac{3}{64}$	$-\frac{165}{256}$	$-\frac{305}{384}$
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	$\frac{15}{32}$	$\frac{5}{16}$

Table 3.14: We show the full primal two-level transformation matrix $\sqrt{2}\mathbf{M}_j$. The left part is identical to the refinement matrix (on the left in Table 3.11). The right part is the result of the final stable completion.

$$
\left[\begin{array}{ccccccccc|cccccccc}
\frac{163}{64} & -\frac{1715}{768} & -\frac{41}{384} & \frac{5}{256} & 0 & 0 & 0 & 0 & 0 & -1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{35}{32} & -\frac{179}{384} & -\frac{41}{192} & \frac{5}{128} & 0 & 0 & 0 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{5}{16} & \frac{305}{192} & -\frac{13}{96} & \frac{1}{64} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{15}{32} & \frac{165}{128} & \frac{31}{64} & -\frac{9}{128} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{15}{64} & -\frac{153}{256} & \frac{187}{128} & -\frac{67}{256} & \frac{3}{64} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{3}{32} & \frac{19}{32} & \frac{19}{32} & -\frac{3}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3}{64} & -\frac{1}{4} & \frac{45}{32} & -\frac{1}{4} & \frac{3}{64} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{3}{32} & \frac{19}{32} & \frac{19}{32} & -\frac{3}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{3}{64} & -\frac{1}{4} & \frac{45}{32} & -\frac{1}{4} & \frac{3}{64} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{3}{32} & \frac{19}{32} & \frac{19}{32} & -\frac{3}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{3}{64} & -\frac{1}{4} & \frac{45}{32} & -\frac{1}{4} & \frac{3}{64} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{3}{32} & \frac{19}{32} & \frac{19}{32} & -\frac{3}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{3}{64} & -\frac{67}{256} & \frac{187}{128} & -\frac{153}{256} & \frac{15}{64} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{9}{128} & \frac{31}{64} & \frac{165}{128} & -\frac{15}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{64} & -\frac{13}{96} & \frac{305}{192} & -\frac{5}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & \frac{5}{128} & -\frac{41}{192} & -\frac{179}{384} & \frac{35}{32} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -2 \\
0 & 0 & 0 & 0 & 0 & \frac{5}{256} & -\frac{41}{384} & -\frac{1715}{768} & \frac{163}{64} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -1
\end{array}\right]
$$

Table 3.15: We picture the full dual two-level transformation matrix $\sqrt{2}\tilde{\mathbf{M}}_j$*. The left part is identical to the biorthogonalised refinement matrix (shown in Table 3.13), which follows from* (2.3.12)*. The right part is similar in structure to the left part from Table 3.14.*

Here we have introduced nodal basis functions $\zeta_{j,k}$ indexed by the set Δ_j^Z. By the relation $\leftrightarrow$ we mean that equality holds on the grid $x_l = 2^{-j} l$, $l \in \Delta_j^Z$. The coefficients $\mathbf{z}_j$ in the basis Z_j then correspond to the function values of $f_{(j)}$ at the grid points and can be directly used for plotting.

Remark 3.12. *The implicitly given function $\tilde{\phi}(x)$ is approximated with piecewise linear functions. This approximation is exact at the grid points. Thus we propose an interpolation procedure. For error estimation, standard results on interpolation can be used.*

To transform the dual basis $\tilde{\Phi}_j$ into the nodal basis Z_j, we use three intermediate steps. First, we invert the biorthogonalisation (3.3.37) to return to the initial dual basis $\tilde{\Phi}_j^{(0)}$. Then we reverse the adaption to the interval boundaries from (3.3.27) to arrive at the translationally invariant basis $\{\tilde{\phi}_{[j,k]}\}$. Thus, these first two transformations have already been specified before. Finally, we transform $\{\tilde{\phi}_{[j,k]}\}$ into the nodal basis by explicitly using uniformly spaced precalculated function values of the dual generator $\tilde{\phi}$.

Theorem 3.13. *To approximately plot representations of functions in the dual single-scale basis, we may use a pipeline of three transformations,*

$$\{\tilde{\phi}_{[j,k]}\} \leftrightarrow \mathbf{V}_j^T Z_j \,, \qquad \tilde{\Phi}_j^{(0)} = \mathbf{O}_j^T \{\tilde{\phi}_{[j,k]}\} \,, \qquad \tilde{\Phi}_j = \mathbf{\Gamma}_j^{-T} \tilde{\Phi}_j^{(0)} \,. \tag{3.3.69}$$

This leads to the expansion

$$\tilde{\Phi}_j \leftrightarrow \mathbf{\Gamma}_j^{-T} \mathbf{O}_j^T \mathbf{V}_j^T Z_j \qquad \Longleftrightarrow \qquad \mathbf{z}_j = \mathbf{V}_j \mathbf{O}_j \mathbf{\Gamma}_j^{-1} \tilde{\mathbf{c}}_j \,. \tag{3.3.70}$$

Proof. First, we aim at identifying $\mathbf{V}_j$. To this end, we evaluate the dual generator $\tilde{\phi}(x)$ at the integer points $x \in \mathbb{N}$ of its support, defining the vector

$$\tilde{\mathbf{v}} := \{v_i\}_i \,, \qquad v_i := \tilde{\phi}(i) \,, \qquad i = \tilde{\ell}_1 + 1, \ldots, \tilde{\ell}_2 - 1 \,. \tag{3.3.71}$$

As the support of $\tilde{\phi}$ is larger than that of $\zeta_{0,k}$, the matrix $\mathbf{V}_j$ is rectangular. Its entries are given by

$$(\mathbf{V}_j)_{k,l} = 2^{j/2} v_{k-l} \,, \qquad k = 0, \ldots, 2^j \,, \qquad l = -\tilde{\ell}_2 + 1, \ldots, 2^j - \tilde{\ell}_1 - 1 \,. \tag{3.3.72}$$

The transformation matrix $\mathbf{O}_j$ implements the restriction to the interval as of (3.3.27) and (3.3.31). It is again rectangular, and its contents are the dual coefficients of polynomial exactness,

$$\mathbf{O}_j = \begin{pmatrix} \alpha & & \\ & \mathbf{I}_{\#\tilde{\Delta}_j^0} & \\ & & \alpha^{\updownarrow} \end{pmatrix} , \tag{3.3.73}$$

where the range of the indices of $\alpha_{m,r}$ is the same as in (3.3.27). The entries of $\mathbf{\Gamma}_j$ are already known from Section 3.3. □

To illustrate the procedure, we provide the upper left non-trivial part of the product $\mathbf{O}_j \mathbf{\Gamma}_j^{-1}$ in Table 3.16. It is a rectangular matrix whose lower part passes into the identity matrix in the interior.

Remark 3.14. *To adopt this process to homogeneous boundary conditions, suitable deletions of outmost rows and/or columns in the matrices $\mathbf{V}_j$ and $\mathbf{O}_j$ have to be executed. $\mathbf{\Gamma}_j$ already incorporates these boundary conditions by definition* (3.3.35).

100	$-\frac{580}{3}$	$\frac{178}{3}$	-15
50	$-\frac{565}{6}$	$\frac{80}{3}$	$-\frac{13}{2}$
20	$-\frac{107}{3}$	$\frac{26}{3}$	-2
5	$-\frac{89}{12}$	$\frac{7}{6}$	$-\frac{1}{4}$
0	1	0	0
0	0	1	0
0	0	0	1

100	$-\frac{280}{3}$	$\frac{178}{3}$	-15
50	$-\frac{265}{6}$	$\frac{80}{3}$	$-\frac{13}{2}$
20	$-\frac{47}{3}$	$\frac{26}{3}$	-2
5	$-\frac{29}{12}$	$\frac{7}{6}$	$-\frac{1}{4}$
0	1	0	0
0	0	1	0
0	0	0	1

Table 3.16: We display the non-trivial section of the product $\mathbf{O}_j\mathbf{\Gamma}_j^{-1}$ *for inhomogeneous boundary conditions. It can be seen that the lower part passes into the identity matrix. We provide two versions, the left matrix corresponding to our construction detailed in Section 3.3.3, while the right matrix is constructed with the additional transformations from Section 3.3.5, see below.*

3.3.5 Transformation to the Nodal Basis

As with finite element wavelets in Section 3.2, translational invariance of the primal basis can be established by a transformation to the nodal basis. In our example of $d = 2$ and $\tilde{d} = 4$, the case of homogeneous boundary conditions already yields a scaled nodal basis. For inhomogeneous boundary conditions, we have a scaled nodal basis in the interior, and only the boundary blocks have to be adapted. Again, the corresponding transformation fits into the framework of Section 2.3.2, cf. (2.3.22), with the matrix

$$\mathbf{C}_j := \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & & \ddots & & \\ & & & 1 & -1 \\ & & & & 1 \end{pmatrix}. \tag{3.3.74}$$

The effect on the refinement matrix $\mathbf{M}_j$ can be seen in the left part of Table 3.17, which is now translationally invariant. A closer inspection of the right part of $\tilde{\mathbf{M}}_j$ in Table 3.15 shows that it is translationally invariant except for the border blocks. We address this issue by devising an additional transformation of the wavelets according to (2.3.18), with the matrix

$$\check{\mathbf{K}}_j := \begin{pmatrix} -2 & & & & \\ -1 & 1 & & & \\ & & \ddots & & \\ & & & 1 & -1 \\ & & & & -2 \end{pmatrix}. \tag{3.3.75}$$

The results of the simultaneous application of these two sparse transformations are shown in Table 3.17 for the primal matrix and Table 3.18 for the dual matrix. We consider the patterns on the left of $\mathbf{M}_j$ and on the right of $\tilde{\mathbf{M}}_j$ as being of the purest form, as they are translation invariant, including the boundary functions. This is not only beneficial from the aesthetic point of view, but also advantageous numerically, as the condition numbers of the wavelet transform are substantially improved, which we will confirm in the next section. We will also see in Chapter 4 and Chapter 5 that the condition numbers of discretised elliptic operators and Riesz matrices are significantly reduced, especially for higher dimensions.

Remark 3.15. *As stated above, the use of* $\mathbf{C}_j$ *is not applicable for homogeneous boundary conditions. Nonetheless, the matrix* $\mathbf{K}_j$ *from* (3.3.75) *can be used in this case with exactly the same effect as for inhomogeneous boundary conditions.*

1	0	0	0	0	0	0	0	0	$-\frac{35}{32}$	$\frac{15}{32}$	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	0	$\frac{875}{768}$	$-\frac{45}{256}$	$\frac{3}{64}$	0	0	0	0	0
0	1	0	0	0	0	0	0	0	$-\frac{241}{384}$	$-\frac{105}{128}$	$\frac{3}{32}$	0	0	0	0	0
0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	$-\frac{53}{256}$	$\frac{345}{256}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0	0
0	0	1	0	0	0	0	0	0	$\frac{41}{192}$	$-\frac{31}{64}$	$-\frac{19}{32}$	$\frac{3}{32}$	0	0	0	0
0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{67}{768}$	$-\frac{53}{256}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0
0	0	0	1	0	0	0	0	0	$-\frac{5}{128}$	$\frac{9}{128}$	$-\frac{19}{32}$	$-\frac{19}{32}$	$\frac{3}{32}$	0	0	0
0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$-\frac{5}{256}$	$\frac{9}{256}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0
0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{19}{32}$	$-\frac{19}{32}$	$\frac{3}{32}$	0	0
0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{9}{256}$	$-\frac{5}{256}$
0	0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{19}{32}$	$-\frac{19}{32}$	$\frac{9}{128}$	$-\frac{5}{128}$
0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{53}{256}$	$\frac{67}{768}$
0	0	0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{19}{32}$	$-\frac{31}{64}$	$\frac{41}{192}$
0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{345}{256}$	$-\frac{53}{256}$
0	0	0	0	0	0	0	1	0	0	0	0	0	0	$\frac{3}{32}$	$-\frac{105}{128}$	$-\frac{241}{384}$
0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	$\frac{3}{64}$	$-\frac{45}{256}$	$\frac{875}{768}$
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	$\frac{15}{32}$	$-\frac{35}{32}$

Table 3.17: We show the full primal two-level transformation matrix $\sqrt{2}\mathbf{M}_j$ *at level* $j = j_0 = 3$ *for inhomogeneous boundary conditions, after the transformations induced by* (3.3.74) *and* (3.3.75). *We have transformed the generator basis to the nodal basis, as can be seen from the typical structure in the left part.*

$\frac{93}{64}$	$-\frac{241}{768}$	$\frac{41}{384}$	$-\frac{5}{256}$	0	0	0	0	0	$-\frac{1}{2}$	0	0	0	0	0	0	0
$\frac{35}{32}$	$\frac{241}{384}$	$-\frac{41}{192}$	$\frac{5}{128}$	0	0	0	0	0	1	0	0	0	0	0	0	0
$-\frac{5}{16}$	$\frac{245}{192}$	$-\frac{13}{96}$	$\frac{1}{64}$	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0	0
$-\frac{15}{32}$	$\frac{105}{128}$	$\frac{31}{64}$	$-\frac{9}{128}$	0	0	0	0	0	0	1	0	0	0	0	0	0
$\frac{15}{64}$	$-\frac{93}{256}$	$\frac{187}{128}$	$-\frac{67}{256}$	$\frac{3}{64}$	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0
0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0	0	0	0	0	1	0	0	0	0	0
0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0
0	0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0	0	0	0	0	1	0	0	0	0
0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0
0	0	0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0	0	0	0	0	1	0	0	0
0	0	0	$\frac{3}{64}$	$-\frac{1}{4}$	$\frac{45}{32}$	$-\frac{1}{4}$	$\frac{3}{64}$	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0
0	0	0	0	$-\frac{3}{32}$	$\frac{19}{32}$	$\frac{19}{32}$	$-\frac{3}{32}$	0	0	0	0	0	0	1	0	0
0	0	0	0	$\frac{3}{64}$	$-\frac{67}{256}$	$\frac{187}{128}$	$-\frac{93}{256}$	$\frac{15}{64}$	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0
0	0	0	0	0	$-\frac{9}{128}$	$\frac{31}{64}$	$\frac{105}{128}$	$-\frac{15}{32}$	0	0	0	0	0	0	1	0
0	0	0	0	0	$\frac{1}{64}$	$-\frac{13}{96}$	$\frac{245}{192}$	$-\frac{5}{16}$	0	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$
0	0	0	0	0	$\frac{5}{128}$	$-\frac{41}{192}$	$\frac{241}{384}$	$\frac{35}{32}$	0	0	0	0	0	0	0	1
0	0	0	0	0	$-\frac{5}{256}$	$\frac{41}{384}$	$-\frac{241}{768}$	$\frac{93}{64}$	0	0	0	0	0	0	0	$-\frac{1}{2}$

Table 3.18: We display the full dual two-level transformation matrix $\sqrt{2}\tilde{\mathbf{M}}_j$ *at level* $j = j_0 = 3$ *for inhomogeneous boundary conditions, after the transformations induced by* (3.3.74) *and* (3.3.75). *We have achieved a translation invariant structure in the right part. Note that* $\tilde{\mathbf{M}}_j = \mathbf{M}_j^{-T}$ *from Table 3.18.*

Remark 3.16. *To utilise the procedure for plotting the dual representations from Section 3.3.4, we have to take into account the transformation*

$$\mathbf{\Gamma}_j \mapsto \mathbf{C}_j^T \mathbf{\Gamma}_j \tag{3.3.76}$$

in (3.3.70). *The difference to the original version can be seen in Table 3.16.*

3.3.6 Numerical Properties

In this section we will show the results of numerical calculations of the condition numbers of the single-scale and the wavelet basis. We list these properties for one, two and three dimensions and for several levels of resolution, for both homogeneous and inhomogeneous boundary conditions. We also investigate the effect of the transformations proposed in Section 3.3.5.

Because of biorthogonality, the condition numbers of the primal and dual bases are the same. To calculate the condition numbers of the symmetric positive definite Gramian matrices for the respective combinations of parameters, we have used power iterations to obtain the largest eigenvalue, and inverse power iterations with an inner conjugate gradient routine to compute the smallest eigenvalue.

The results for inhomogeneous boundary conditions are shown in Table 3.19. It consists of four rows for the spatial dimensions $n = 1$ to $n = 4$. Each row contains two tables. The table on the left hand side corresponds to the basic construction specified in Section 3.3.3. The table on the right refers to the transformation (3.3.74) of the generator basis to the nodal basis as described in Section 3.3.5. In each table, we include the variant created by the additional transformation (3.3.75), denoted by a superscript $()^{\mathbf{K}}$.

While the transformation for the primal generator functions reduces the condition numbers by a factor of up to 2, the best results by far arise from the combination of both transformations. The transformation of the wavelets alone does not lead to an overall improvement. We can reduce the condition number of the wavelet basis from about 5.6^n to 2.6^n, and the condition number of the wavelet transform from about 7.3^n to 3.1^n.

The condition numbers for the homogeneous case are given in Table 3.20. While the condition number of the single-scale basis is still better than the best inhomogeneous variant, the wavelet specific condition numbers are worse. Moreover, the transformation (3.3.75) tends to even degrade the condition number in this case.

We confirm in the numerical experiments that all bases and transformations have uniformly bounded condition numbers. The case of homogeneous boundary condition does not seem to benefit from the transformation (3.3.75) on the wavelet side. Quite the contrary, the condition numbers are generally higher by a factor of up to 2 compared to the original construction. In contrast, the combination of both transformations introduced in Section 3.3.5 constitutes a significant improvement for inhomogeneous boundary conditions. The condition number of the final wavelet basis is only increased by a factor less than 2 compared to the nodal single-scale basis. The condition numbers of the wavelet transformations are also very low compared with the original constructions.

We conclude to use our transformations whenever we deal with inhomogeneous boundary conditions. We will see in forthcoming chapters of this document that also the condition numbers of discretisations of elliptic operators and Riesz operators benefit from this decision, especially for increasing spatial dimensions.

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	4.44	4.44	4.44	1.00	1.00
4	4.44	5.05	4.68	4.70	5.90
5	4.44	5.44	4.70	6.02	8.35
8	4.44	5.86	4.70	7.71	8.13
12	4.44	5.99	4.70	8.03	7.25

$n = 1$, original generators

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	1.98	1.98	1.98	1.00	1.00
4	1.99	3.74	2.45	2.83	2.10
5	2.00	4.34	2.59	3.75	2.61
8	2.00	4.84	2.67	5.27	3.17
12	2.00	4.97	2.68	5.89	3.29

$n = 1$, with $\mathbf{C}_j$ from (3.3.74)

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	19.7	19.7	19.7	1.00	1.00
4	19.7	25.5	21.9	22.1	34.8
5	19.7	28.6	22.1	34.1	64.0
8	19.7	30.9	22.1	52.4	58.7

$n = 2$, original generators

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	3.92	3.92	3.92	1.00	1.00
4	3.98	14.0	6.00	7.99	4.41
5	3.99	16.8	6.62	14.0	6.67
8	4.00	18.2	6.89	24.5	9.68

$n = 2$, with $\mathbf{C}_j$ from (3.3.74)

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	87.6	87.6	87.6	1.00	1.00
4	87.6	129	102	104	205
5	87.6	149	103	191	505
6	87.6	156	104	290	586

$n = 3$, original generators

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	7.77	7.77	7.77	1.00	1.00
4	7.93	52.4	14.7	22.6	9.24
5	7.98	66.3	16.9	52.6	17.2
6	7.99	69.0	17.4	85.1	23.5

$n = 3$, with $\mathbf{C}_j$ from (3.3.74)

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	389	389	389	1.00	1.00
4	389	649	479	488	1208
5	389	773	486	1076	4004

$n = 4$, original generators

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
3	15.4	15.4	15.4	1.00	1.00
4	15.8	196	36.0	63.9	19.5
5	15.9	266	42.9	197	44.1

$n = 4$, with $\mathbf{C}_j$ from (3.3.74)

Table 3.19: We list the condition numbers of single-scale bases Φ_j *and wavelet bases* Ψ_j *and the wavelet transformation* $\mathbf{W}_j$ (2.2.20) *for inhomogeneous boundary conditions. The spatial dimension* n *increases from top to bottom. The tables on the left hand side refer to the original construction, while the right hand side includes the transformation* $\mathbf{C}_j$ (3.3.74)*. In each table, we also give results for the transformation* $\mathbf{K}_j$ (3.3.75)*, which is denoted by the superscript* $()^{\mathbf{K}}$*. Both transformations are independent of each other. We can see that the best results stem from their combination.*

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
4	1.71	1.71	1.71	1.0	1.0
5	1.73	4.05	5.57	3.94	5.63
6	1.73	6.12	8.58	6.37	8.78
8	1.73	9.09	12.7	10.1	13.1
12	1.73	12.2	17.0	14.3	17.5

$n = 1$

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
4	2.92	2.92	2.92	1.0	1.0
5	2.98	16.4	31.0	15.5	31.7
6	2.99	33.2	59.1	37.5	66.3
8	2.99	59.0	102	78.6	127

$n = 2$

j	$\kappa(\Phi)$	$\kappa(\Psi)$	$\kappa(\Psi)^{\mathbf{K}}$	$\kappa(\mathbf{W}_j)$	$\kappa(\mathbf{W}_j)^{\mathbf{K}}$
4	5.00	5.00	5.00	1.0	1.0
5	5.14	66.3	173	61.1	179
6	5.18	183	414	224	526

$n = 3$

Table 3.20: We show the condition numbers of single-scale and wavelet bases and the wavelet transform for homogeneous boundary conditions. Here, only the transformation on the wavelets is applicable, which is again denoted by the superscript $()^{\mathbf{K}}$*. In this case, it does not improve the condition number.*

Chapter 4

Wavelet Methods for Linear Elliptic Partial Differential Equations

4.1 Introduction

In this thesis we deal with optimal control problems which are constrained by linear elliptic partial differential equations. The efficiency of a numerical algorithm for the control problem as a whole depends crucially on the efficient handling of these constraints. In fact, the character of the partial differential equation determines for the most part the type of discretisation and solution scheme of the whole problem, and its speed of convergence.

In our case, the constraints have the form of a linear elliptic PDE. Equations of this type describe several basic physical phenomena, such as stresses in elastic materials, the distribution of temperature or the concentration of chemical substances subject to diffusive processes. Furthermore, they occur as building blocks in mathematical methods for structurally more complicated problems, for example in the solution of nonlinear equations, as sub-step in the computation of fluid flow or in contact problems in medicine and engineering. Effective numerical algorithms for this class of equations are thus of highest practical relevance, and this need has distinctly and continuously shaped the research in numerical analysis, from a time even before the existence of modern computing machines until now.

The prototypical elliptic boundary value problem reads as follows: Find y such that

$$\Delta y = f \qquad \text{in } \Omega\,, \tag{4.1.1a}$$

$$y = 0 \qquad \text{on } \partial\Omega\,. \tag{4.1.1b}$$

(The mathematical objects used here will be introduced below.) The numerical approach to such an equation involves the process of discretisation which leads to a large linear system of equations. Since the direct solution would be prohibitively expensive because of its size, iterative solvers are employed, whose rate of convergence depends on the condition number of the system matrix. A solver is called *fast* or *asymptotically optimal* if it computes the solution with a number of arithmetic operations which is proportional to the number of unknowns. While finite element approaches require additional preconditioners such as the BPX scheme or the use of multigrid methods to be asymptotically optimal, wavelet discretisations of elliptic operators are inherently well-conditioned.

The present chapter is devoted to the proposition of a wavelet solver for linear elliptic partial differential equations which is fast in the above sense, and for which we show that it is competitive in terms of large scale computing. We provide an overview of the wavelet discretisation approach and a description of the solution process. The analytical derivations are then complemented with numerical results on the condition numbers of the resulting stiffness matrices. In Section 4.3.3, a new operator-adapted construction scheme is proposed which can result in significantly improved condition numbers.

4.2 Numerical Solution of Elliptic Boundary Value Problems

Many numerical methods for the solution of elliptic PDEs fall into one of two major classes, namely *finite differences* or *finite elements*. In finite difference methods, the action of differential operators on functions is approximated by taking combinations of point values [22, 86]. A famous example is the so-called *five point stencil* for the Laplace operator in two dimensions,

$$\Delta v(x) \approx \frac{v(x-he^1)+v(x-he^2)+v(x+he^1)+v(x+he^2)-4v(x)}{h^2}\,. \tag{4.2.1}$$

For this formulation, the function values of $v(x)$ must be well defined at the grid points (ih, jh), where $h \coloneqq 1/N$ and $i, j \in 0, \dots, N$, and e^1 and e^2 are the unit vectors. In contrast, finite element methods allow for relaxed regularity requirements, since they are based on the weak formulation of partial differential equations, a theory which can be formulated in terms of Lebesgue integrable functions.

A different ansatz are the so-called *spectral methods*, where the analytical framework is based on Fourier decompositions [21]. In this case general functions are approximated by linear combinations of trigonometric functions, which provide two useful features for the solution of elliptic PDEs. Firstly, trigonometric functions are eigenfunctions of the Laplace operator, which decouples the degrees of freedom of the spectral representation. Secondly, they allow to evaluate fractional Sobolev norms up to equivalence, which we regard as important in connection with optimal control problems. However, the treatment of non-periodic or non-continuous functions by such expansions is difficult, which is a severe restriction from the practical point of view.

Wavelet methods combine essential advantages of finite elements and spectral methods. They share with spectral elements the ability for the numerical evaluation of Sobolev norms. The procedure itself however is largely different, and will be discussed in detail in the next chapter. Conceptually, the wavelet method proposed here is closer to finite elements in the sense that it is also based on the modern framework of weak formulations of elliptic boundary value problems. This area of applied mathematics has been the subject of extensive studies in for the past decades, and the fundamental theorems such as the one of Lax-Milgram or Cea's Lemma are by now widely known. Because of the importance for the theoretical foundation of this work, and the finite element context in general, we will give here a short introduction to the basic notions and definitions, largely based on [22].

4.2.1 Weak Formulation

Definition 4.1. *Let H be a Hilbert space with norm $\|\cdot\|_H$. A bilinear form $a : H \times H \to \mathbb{R}$ is called continuous, if there exists $C_A > 0$ such that for any v, $w \in H$*

$$|a(v,w)| \le C_A \|v\|_H \|w\|_H\,. \tag{4.2.2}$$

A symmetric continuous bilinear form is called H-elliptic or coercive *if there exists a $c_A > 0$ such that*

$$a(v,v) \ge c_A \|v\|_H^2\,, \qquad v \in H\,. \tag{4.2.3}$$

It follows that every H-elliptic form induces a norm on H according to

$$\|v\|_a \coloneqq \sqrt{a(v,v)}\,. \tag{4.2.4}$$

This norm is called the *energy norm*, and it is equivalent to the norm of H. The bilinear form $a(\cdot,\cdot)$ induces a linear operator $A : H \to H'$ by the definition

$$\langle v, Aw\rangle := a(v,w) \qquad \text{for all } v,w \in H\,. \tag{4.2.5}$$

A common example for an elliptic form over $H_0^1(\Omega)$ or $H^1(\Omega)$, $\Omega \subset \mathbb{R}^n$ (see e.g. [3] for an introduction to these spaces), is given by the representation

$$a(v,w) := \int_\Omega \Bigl(\sum_{i,k} a_{ik}\partial_i v \partial_k w + a_0 v w\Bigr)\,\mathrm{d}x\,, \tag{4.2.6}$$

where $\mathcal{A}(x) := (a_{ik}(x))_{i,k=1}^n$ is a symmetric positive definite matrix with $\lambda_{\min}(\mathcal{A}) \geq c_A$, $\lambda_{\max}(\mathcal{A}) \leq C_A$, and $0 \leq a_0(x) \leq C_A$, almost everywhere on Ω. (For $H^1(\Omega)$, we actually need to demand $c_A \leq a_0(x) \leq C_A$.) Throughout this thesis, we always start out with such a symmetric, continuous and H-elliptic form $a(\cdot,\cdot)$. This bilinear form is associated with an elliptic differential operator L of second order according to

$$Ly := -\sum_{i,k=1}^n \partial_i(a_{ik}\partial_k y) + a_0 y\,. \tag{4.2.7}$$

With the notion of H-elliptic forms, we can establish a class of variational problems and obtain statements on existence and uniqueness of their solution.

Theorem 4.2 (Lax-Milgram). *Let $a(\cdot,\cdot)$ be an H-elliptic form. For any $f \in H'$, there exists exactly one element $y \in H$ which solves the variational problem*

$$\min_{v\in H} J(v)\,, \qquad J(v) := \frac{1}{2}a(v,v) - \langle f, v\rangle \tag{4.2.8}$$

The above statement on the minimisation of a quadratic functional is intimately connected to the concept of *weak solutions* of elliptic boundary value problems. To establish this link, we describe first two standard classical problems which differ only in the conditions specified on the boundary.

Definition 4.3 (Dirichlet problem). *A function $y \in H_0^1(\Omega)$ is called a weak solution of the elliptic boundary value problem with homogeneous Dirichlet boundary conditions specified by*

$$Ly = f \qquad \text{in } \Omega\,, \tag{4.2.9a}$$

$$y = 0 \qquad \text{on } \partial\Omega\,, \tag{4.2.9b}$$

if it holds that

$$a(y,v) = (f,v)_{L_2(\Omega)} \qquad \text{for all} \qquad v \in H_0^1(\Omega)\,. \tag{4.2.10}$$

For the second variant, we need an additional prerequisite. Let the boundary of Ω be denoted by $\Gamma = \partial\Omega$. We demand that there exists a bounded linear map $\gamma : H^1(\Omega) \to L_2(\Gamma)$ such that $\gamma v = v_{|\Gamma}$ for all $v \in C^1(\bar{\Omega})$, and that $a_0 \geq c_A$ a.e. on Ω. This can be guaranteed by a relaxed form of the trace theorem [22]. Then we may define as follows.

Definition 4.4 (Neumann problem). *A function $y \in H^1(\Omega)$ is called a weak solution of the elliptic boundary value problem with Neumann boundary conditions specified by*

$$Ly = f \qquad \text{in } \Omega\,, \tag{4.2.11a}$$

$$\sum_{i,k} \nu_i a_{ik}\partial_k y = g \qquad \text{on } \Gamma\,, \tag{4.2.11b}$$

if it holds that

$$a(y,v) = (f,v)_{L_2(\Omega)} + (g,v)_{L_2(\Gamma)} \qquad \text{for all} \qquad v \in H^1(\Omega)\,. \tag{4.2.12}$$

It turns out that weak solutions of both of these problems can be identified as solutions of an appropriate variational problem of the type described in (4.2.8). To this end, we cite the following

Theorem 4.5. *Let L be an elliptic operator as specified in* (4.2.7). *Then weak solutions to the problems* (4.2.9) *and* (4.2.11) *do always exist and are unique. They are the minimisers of the corresponding variational problems, which read*

$$\min_{v \in H_0^1(\Omega)} J(v)\,, \qquad J(v) := \frac{1}{2} a(v,v) - (f,v)_{L_2(\Omega)} \tag{4.2.13a}$$

for the Dirichlet problem and

$$\min_{v \in H^1(\Omega)} J(v)\,, \qquad J(v) := \frac{1}{2} a(v,v) - (f,v)_{L_2(\Omega)} - (g,v)_{L_2(\Gamma)} \tag{4.2.13b}$$

for the Neumann problem, respectively.

These weak formulations of elliptic boundary value problems are *well-posed* in the sense that the solution y depends continuously on the data f. This is a consequence of coercivity, which leads to the relation

$$\|y\|_H \leq c_A^{-1} \|f\|_{H'}\,. \tag{4.2.14}$$

We have thus established weak formulations of elliptic boundary value problems and provided criteria for the existence, uniqueness and boundedness of their solution $y \in H$.

4.2.2 A-priori Estimates

In the last section, we have introduced a class of variational problems over Sobolev spaces. However, for numerical computations we need to develop a strategy to cope with the finite amount of computer memory. To this end, the infinite-dimensional variational problem is transformed into an approximate but finite problem formulated in matrices and vectors. This is generally accomplished by choosing hierarchies of finite-dimensional subspaces $S_{\Lambda_j} \subset H$ indexed by a level of resolution j, and appropriate sets of basis functions. Replacing the original functions occurring in the variational problem by finite expansions in S_{Λ_j}, a reformulation is achieved in terms of the expansion coefficients of the basis functions, which leads to a linear system of equations. The general idea of this so-called *Galerkin method* is almost a century old [119]. For historical reasons, the system matrix is often called *stiffness matrix*, motivated by applications in elasticity theory [22].

It is then natural to ask how close the solution for the finite system is to the exact solution. An answer is provided by the following

Theorem 4.6 (Cea's Lemma). *Let the bilinear form $a(\cdot,\cdot)$ be H-elliptic, and let the solutions of the variational problem over the full space H and the subspace S_Λ be denoted by y and y_Λ, respectively. Then it holds that*

$$\|y - y_\Lambda\|_H \leq \frac{C_A}{c_A} \inf_{v \in S_\Lambda} \|y - v\|_H\,. \tag{4.2.15}$$

This result is of central importance, as it shows that the quality of the discrete solution can be estimated by the approximation properties of the subspaces S_{Λ_j}. These are usually designed in such a way that they contain all piecewise polynomials of order d with a mesh size $h := 2^{-j}$, and their dimension depends on j as $N_j := \dim S_{\Lambda_j} \sim 2^{nj}$. For $H = H^1$, at least piecewise linear functions ($d \geq 2$), and $y \in H^2(\Omega)$, this allows to state the following result, see e.g. [117],

$$\inf_{v \in S_{\Lambda_j}} \|y - v\|_{H^1} \lesssim 2^{-j} |y|_{H^2}\,. \tag{4.2.16}$$

Combined with (4.2.15), it follows that for $j \to \infty$ the H^1-error of the discrete solution y_{Λ_j} is proportional to h and thus of order 1.

These considerations imply that for a fixed order of approximation d, a high number of degrees of freedom N_j is needed for sufficient accuracy. The linear systems resulting from the discretisation are thus too large to employ direct solvers. Yet, the stiffness matrix has a special sparse structure which allows to evaluate the matrix-vector product in $\mathcal{O}(N_j)$ operations, which motivates the use of iterative solvers.

The convergence speed of iterative solvers like the method of conjugate gradients depends on the spectral condition number of the system matrix, which is proportional to $h^{-2} = 2^{2j}$ for standard finite element spaces. Hence the rate of convergence slows down with growing j. It is therefore essential to employ techniques for preconditioning [36, 37]. Ideally, the numerical complexity should be proportional to the number of unknowns. Most schemes make use of a multi-level approach by splitting the finite element spaces into hierarchical subspaces [24, 116, 145]. The most prominent class of solvers in the finite element setting is given by multigrid methods [23, 26, 84, 138], which have been devised to treat a manifold variety of problems [85]. Active research is undertaken on algebraic and parallel variants, see e.g. [72, 78, 122]. We will not cover these approaches here as they are discussed in great detail in the finite element literature. We will instead continue to describe the wavelet approach, which resolves the problem of preconditioning by the design of appropriate Riesz bases for H.

4.3 A Wavelet Method for Elliptic Problems

Employing the Riesz basis property (2.2.51) which is satisfied by biorthogonal wavelets, we derive that the condition number of the stiffness matrix is uniformly bounded. Thus, the wavelet approach inherently resolves the issue of preconditioning which is crucial in finite element methods.

After this, we choose a specific family of wavelets and give concrete numerical results for selected condition numbers of the stiffness matrix. Additionally, we introduce a novel technique to further improve this condition number by an operator-adapted transformation. At last, we present a nested iteration strategy to obtain discretisation error accuracy with linear computational complexity.

4.3.1 A Wavelet Galerkin Method

The wavelet framework differs from finite element techniques in the fact that the variational problem from Section 4.2 is first reformulated as an equivalent infinite-dimensional problem over the sequence space ℓ_2. This is accomplished by an expansion of the functions from the weak formulations (4.2.9) and (4.2.11) in a Riesz basis of the Hilbert space H. This reformulation in terms of expansion coefficients $\mathbf{y} \in \ell_2$ leads to an infinite linear system of equations for the discrete solution $\mathbf{y}$.

Choosing a very particular basis for this procedure, namely biorthogonal wavelets, offers the principal benefit that the condition number of the resulting system matrix is uniformly bounded, which lays the foundation for the development of a fast algorithm for the numerical solution. A detailed survey on the following facts can for example be found in [46, 102].

Let us consider either one of the problems from Section 4.2.1, that is, find $y \in H$ such that

$$a(y, v) = \langle f, v \rangle \qquad \text{for all} \qquad v \in H\,. \tag{4.3.1}$$

Recall that depending on the type of problem we have $H = H_0^1(\Omega)$ or $H = H^1(\Omega)$. Suppose now that we have a biorthogonal wavelet basis Ψ^1 of the Hilbert space H at our disposal, satisfying the norm equivalence

$$\|v\|_H = \|\mathbf{v}^T \Psi^1\|_H \sim \|\mathbf{v}\|\,, \qquad v = \mathbf{v}^T \Psi^1 \in H\,. \tag{4.3.2}$$

The equivalence constants are denoted by c_H and C_H,

$$c_H\|\mathbf{v}\| \leq \|v\|_H \leq C_H\|\mathbf{v}\| \,. \tag{4.3.3}$$

The theoretical foundations of such wavelet bases have been discussed in Chapter 2, and two specific constructions have been covered in detail in Chapter 3. Expanding the functions y and v in this wavelet basis,

$$y = \mathbf{y}^T\Psi^1 \,, \qquad \mathbf{y} = \langle y, \tilde{\Psi}^1\rangle \,, \tag{4.3.4a}$$

$$v = \mathbf{v}^T\Psi^1 \,, \qquad \mathbf{v} = \langle v, \tilde{\Psi}^1\rangle \,, \tag{4.3.4b}$$

and inserting this into (4.3.1), we obtain

$$a(\mathbf{y}^T\Psi^1, \mathbf{v}^T\Psi^1) = \langle f, \mathbf{v}^T\Psi^1\rangle \qquad \text{for all} \qquad \mathbf{v} \in \ell_2 \,. \tag{4.3.5}$$

With the expansion of the right hand side in the dual basis, $f = \mathbf{f}^T\tilde{\Psi}^1$ and $\mathbf{f} = \langle f, \Psi^1\rangle$, this is equivalent to the linear system of equations

$$\mathbf{A}\mathbf{y} = \mathbf{f} \qquad \text{with} \qquad \mathbf{A}_{\lambda,\mu} := a(\psi^1_\lambda, \psi^1_\mu) \,. \tag{4.3.6}$$

The matrix $\mathbf{A}$ is called the system or *stiffness matrix*. Technically, the wavelet basis Ψ^1 is deduced from the anchor basis Ψ for L_2 by a diagonal scaling matrix $\mathbf{D}$ (2.2.48), which leads to the derivation

$$\mathbf{A} = a(\Psi^1, \Psi^1) = \mathbf{D}^{-1}a(\Psi, \Psi)\mathbf{D}^{-1} \,. \tag{4.3.7}$$

The stiffness matrix is symmetric and positive definite, which can be seen from

$$\mathbf{v}^T\mathbf{A}\mathbf{v} = \sum_{\lambda,\mu} \mathbf{v}_\lambda \mathbf{A}_{\lambda,\mu}\mathbf{v}_\mu = a(v,v) \geq c_A\|v\|_H^2 \geq c_A c_H^2 \|\mathbf{v}\|^2 \,, \tag{4.3.8}$$

where we have used coercivity (4.2.3) and the norm equivalence (4.3.3). This relation effectively bounds the smallest eigenvalue of $\mathbf{A}$. Using the continuity (4.2.2), the largest eigenvalue is bounded by

$$\mathbf{v}^T\mathbf{A}\mathbf{v} = a(v,v) \leq C_A\|v\|_H^2 \leq C_A C_H^2\|\mathbf{v}\|^2 \,. \tag{4.3.9}$$

Combining the knowledge on the smallest and largest eigenvalues, we have therefore proved the central theorem of this section.

Theorem 4.7. *The spectral condition number of the stiffness matrix is uniformly bounded,*

$$\kappa(\mathbf{A}) \leq \frac{C_{\mathbf{A}}}{c_{\mathbf{A}}} \sim 1 \,, \tag{4.3.10}$$

with the definitions $c_{\mathbf{A}} := c_A c_H^2$ *and* $C_{\mathbf{A}} := C_A C_H^2$*.*

Inspired by the representation on the right of (4.3.7), we can also replace the diagonal scaling by the operator-adapted variant $\mathbf{D}_a$, defined as follows,

$$\mathbf{D}_a := (\operatorname{diag} a(\Psi, \Psi))^{\frac{1}{2}} \,. \tag{4.3.11}$$

This replacement changes the wavelet basis for H. The resulting stiffness matrix $\mathbf{A}_a$ has exclusively ones on the diagonal, which generally leads to a reduced condition number in comparison with $\mathbf{A}$. For completeness, we state the explicit definitions and conclusion,

$$\Psi^1_a := \mathbf{D}_a^{-1}\Psi \,, \qquad \mathbf{A}_a := a(\Psi^1_a, \Psi^1_a) = \mathbf{D}_a^{-1}a(\Psi, \Psi)\mathbf{D}_a^{-1} \,, \qquad \kappa(\mathbf{A}_a) \sim 1 \,. \tag{4.3.12}$$

Note the similarity to (4.3.7). We have thus established that the stiffness matrix in wavelet representation has a uniformly bounded condition number, with some freedom in the choice of diagonal scaling.

4.3.2 Fast Solution over Finite-Dimensional Subspaces

Up to this point, we have been working purely in the infinite-dimensional context, first discussing variational problems over Sobolev spaces and then switching to an equivalent wavelet formulation in ℓ_2. To make this formulation accessible for computations, we need to identify finite-dimensional subproblems which approximate the original problem. To this end, we present the concepts of uniform and adaptive wavelet discretisations and comment on the implications on the expansions of functions in wavelet bases and the structure of the stiffness matrix. We explain our notion of a fast solver, and demonstrate that wavelet discretisations deliver all ingredients for the conception of such a fast method.

Finite Wavelet Discretisations

The wavelet bases introduced in Chapter 2 consist of an infinite collection of hierarchically ordered functions which are indexed over $\ell_2(I\!I_H)$ and span the full space H. Finite-dimensional subspaces are selected by a reduction of the index set, i.e., by choosing a finite subset $\Lambda \subset I\!I_H$. Hence, all wavelet basis functions whose indices are not in Λ are discarded, and we obtain a finite basis Ψ_Λ spanning the finite-dimensional subspace $S_\Lambda \subset H$. Note that at this point the structure of Λ is arbitrary.

Consequently, all vectors of wavelet coefficients are truncated by deleting the entries which are not in Λ, and similarly all matrices in wavelet representation are shrunk by deleting all rows and columns which do not belong to Λ. For example, the solution y and the right hand side f are obtained as

$$\mathbf{y}_\Lambda = \langle y, \tilde{\Psi}^1_\Lambda \rangle, \qquad \mathbf{f}_\Lambda = \langle f, \Psi^1_\Lambda \rangle . \tag{4.3.13}$$

The vectors $\mathbf{y}_\Lambda$ and $\mathbf{f}_\Lambda$ have $N_\Lambda = \#\Lambda \in \mathbb{N}$ entries. The truncated stiffness matrix is denoted by $\mathbf{A}_\Lambda$, it has the dimensions $N_\Lambda \times N_\Lambda$, it is still symmetric, and it inherits the uniformly bounded condition number from the infinite-dimensional setting.

Corollary 4.8. *The condition number of the truncated stiffness matrix is bounded by the condition number of the infinite-dimensional problem,*

$$\kappa(\mathbf{A}_\Lambda) \le \kappa(\mathbf{A}) \le \frac{C_\mathbf{A}}{c_\mathbf{A}} \sim 1 . \tag{4.3.14}$$

Proof. Above relation follows from (4.3.8) and (4.3.9). □

It remains to solve the finite and well-conditioned, symmetric linear system of equations over $\mathbb{R}^{N_\Lambda}$,

$$\mathbf{A}_\Lambda \mathbf{y}_\Lambda = \mathbf{f}_\Lambda . \tag{4.3.15}$$

The structure of the matrix $\mathbf{A}_\Lambda$ depends on the strategy by which the finite number of wavelet coefficients is selected. Since wavelet bases are built over a multiresolution analysis (see Definition 2.1), we may choose a particular level of resolution j and define the subspace $S_{\Lambda_j} := S_j \subset H$ according to (2.2.3). To this space corresponds the truncated wavelet basis $\Psi_{(j)}$ (2.2.16). We refer to this approach as *uniform discretisation*, it is analogous to multi-level finite element methods.

Iterative solvers repeatedly apply the stiffness matrix $\mathbf{A}_j := \mathbf{A}_{\Lambda_j}$ to a vector. Making use of the identities (2.2.19) and (2.2.54) we obtain the representation

$$\mathbf{A}_j = \mathbf{D}^{-1} \mathbf{W}_j^T a(\Phi_j, \Phi_j) \mathbf{W}_j \mathbf{D}^{-1} . \tag{4.3.16}$$

To compute the product $\mathbf{A}_j \mathbf{v}$, we can subsequently apply the matrices on the right hand side of (4.3.16). The matrix $a(\Phi_j, \Phi_j)$ is the standard stiffness matrix in the finite element setting, which contains $\mathcal{O}(N_j)$ nonzero elements. As the multi-scale transformations $\mathbf{W}_j$ (2.2.20) for both of the wavelet constructions that we have encountered in Chapter 3 may also be applied in linear time, we conclude that $\mathbf{A}_j$ as a whole can be applied in $\mathcal{O}(N_j)$ arithmetic operations. When using the operator-adapted diagonal $\mathbf{D}_a$ instead (4.3.11), its entries can be precomputed in $\mathcal{O}(N_j)$ operations once at the beginning of the process.

While the subspaces of the uniform discretisation are selected level-wise, it is also possible to base the choice of the active wavelet indices on other criteria. In view of the norm equivalence (4.3.2), we could select only the N largest coefficients of the wavelet expansion, irrespective of their level. This is called best N-term approximation and leads to the concept of *adaptive discretisation*. For solutions which do not have the full regularity required by (4.2.16), it offers the potential to achieve the same accuracy as the uniform procedure with less coefficients, which reduces memory and time requirements. We will develop such a method in Chapter 7, including references to the necessary theory on adaptive wavelet methods.

A Fast Solver

For the solution of the finite system (4.3.15), we employ the method of conjugate gradients (CG), originally conceived in [90] as a direct solver for symmetric positive definite matrices. It has been found later that it can also be used as an efficient iterative method, see e.g. [22] for a discussion in the context of modern numerical methods. Its convergence rate depends on the condition number of the system matrix,

$$\|\mathbf{y}_\Lambda^{(k)} - \mathbf{y}_\Lambda\|_{\mathbf{A}_\Lambda} \leq 2 \left(\frac{\sqrt{\kappa(\mathbf{A}_\Lambda)} - 1}{\sqrt{\kappa(\mathbf{A}_\Lambda)} + 1} \right)^k \|\mathbf{y}_\Lambda^{(0)} - \mathbf{y}_\Lambda\|_{\mathbf{A}_\Lambda} , \tag{4.3.17}$$

where $\mathbf{y}_\Lambda$ denotes the exact solution for (4.3.15), and $\mathbf{y}_\Lambda^{(k)}$ the iterative solution in step k. The type of energy norm used here is defined as follows,

$$\|\mathbf{v}\|_{\mathbf{A}_\Lambda}^2 := \mathbf{v}^T \mathbf{A}_\Lambda \mathbf{v} \sim \|\mathbf{v}\|^2 , \tag{4.3.18}$$

where the equivalence relation on the right follows from (4.3.14) and is thus specific to the wavelet setting. This allows us to derive the following estimate in ℓ_2,

$$\|\mathbf{y}_\Lambda^{(k)} - \mathbf{y}_\Lambda\| \lesssim \rho(\mathbf{A}_\Lambda)^k \|\mathbf{y}_\Lambda^{(0)} - \mathbf{y}_\Lambda\| \qquad \text{with} \qquad \rho(\mathbf{A}_\Lambda) := \frac{\sqrt{\kappa(\mathbf{A}_\Lambda)} - 1}{\sqrt{\kappa(\mathbf{A}_\Lambda)} + 1} \leq \rho(\mathbf{A}) , \tag{4.3.19}$$

which means that the convergence rate $\rho(\mathbf{A}_\Lambda)$ is independent of the choice of the index set Λ. This guarantees that the reduction of the error by a fixed proportion η requires a constant amount K_η of iterations irrespective of the number of unknowns,

$$K_\eta \lesssim -\log(\eta) . \tag{4.3.20}$$

To determine the overall computational cost of the iterative solver, it remains to quantify the effort of one single iteration. We derive this for a uniform wavelet discretisation here, defining

$$\mathbf{y}_j := \mathbf{y}_{\Lambda_j} , \qquad \mathbf{y}_j^{(k)} := \mathbf{y}_{\Lambda_j}^{(k)} , \qquad \mathbf{A}_j := \mathbf{A}_{\Lambda_j} . \tag{4.3.21}$$

The central step of one CG iteration consists in the multiplication of the matrix $\mathbf{A}_j$ with a vector, which in view of (4.3.16) costs $\mathcal{O}(N_j)$ arithmetic operations. The memory consumption is thus proportional to N_j, and the computation time for a reduction η is proportional to $K_\eta N_j$.

Hence, it is possible to solve the discrete system for a given level j to an arbitrary high accuracy of $\mathbf{y}_j^{(k)}$ by increasing the number of iterations k. Yet we have to keep in mind that the discretisation error between the exact full solution $\mathbf{y}$ and the exact discrete solution $\mathbf{y}_j$ persists. Since the full error of the numerical scheme is composed according to

$$\|\mathbf{y} - \mathbf{y}_j^{(k)}\| \leq \|\mathbf{y} - \mathbf{y}_j\| + \|\mathbf{y}_j - \mathbf{y}_j^{(k)}\| , \tag{4.3.22}$$

```
Algorithm Nested $(\mathbf{A}, \mathbf{b}, J, \epsilon) \to \mathbf{x}$: Solves $\mathbf{A}\mathbf{x} = \mathbf{b}$ up to accuracy $\epsilon \sim 2^{-J}$.
(i) Initialisation for coarsest level
    (1) Compute start value $\mathbf{x}_{j_0} := \mathbf{A}_{j_0}^{-1}\mathbf{b}_{j_0}$ to machine precision.
    (2) Set $j := j_0$.
(ii) While $j < J$
    (1) Prolongate $\mathbf{x}_j \to \mathbf{x}_{j+1}^0$, Set $j := j + 1$.
    (2) Solve $\mathbf{A}_j\mathbf{x}_j = \mathbf{b}_j$ iteratively, using the start value $\mathbf{x}_j^0$,
        up to accuracy $2^{-(j-J)}\epsilon$.
(iii) Accept $\mathbf{x}_j \to \mathbf{x}$.
```

Algorithm 4.1: We display the nested iteration algorithm `Nested` *for the solution of an elliptic boundary value problem. The prolongation in the wavelet setting is trivially executed by padding the vector with zero coefficients. This algorithm needs $\mathcal{O}(2^J)$ operations.*

only the rightmost term tends to zero for $k \to \infty$. The left part is the *discretisation error*, which can be derived from the wavelet norm equivalence (4.3.2), Cea's Lemma (4.2.15) and (4.2.16) as

$$\|\mathbf{y} - \mathbf{y}_j\| \sim \|y - y_j\|_{H^1} \lesssim \inf_{v \in S_j} \|y - v\|_{H^1} \lesssim 2^{-j}|y|_{H^2} \,. \tag{4.3.23}$$

We conclude from this that to prescribe higher accuracies for the CG method than $\eta_j := 2^{-j}$ would be a waste of computing power. In other words, a stopping criterion of η_j for the CG method is most efficient to obtain a convergent series of discrete solutions,

$$\|\mathbf{y} - \mathbf{y}_j^{(K_{\eta_j})}\| \lesssim 2^{-j} \,. \tag{4.3.24}$$

Concluding from (4.3.20) that $K_{\eta_j} \lesssim -\log(\eta_j) = j$, we arrive at a computational cost of $\mathcal{O}(jN_j)$, which is not yet the optimal result which we ultimately aspire. To remove the logarithmic factor $j \sim \log(N_j)$, we use a strategy which is known as *nested iteration*, see e.g. [104], which works as follows (see Algorithm 4.1 for a complete listing). The system is solved to machine accuracy on the coarsest level. We then prolongate the coarse solution to the next level j and use it as a start value for the iterative solver. The solver is stopped at discretisation error accuracy 2^{-j}, and the temporary solution is prolongated again to gain a start value for the next level $j + 1$. This scheme is repeated until the highest level J is reached.

Theorem 4.9. *The nested iteration algorithm features a memory and time complexity of $\mathcal{O}(N_J)$, where J is the finest level of discretisation.*

Proof. This result essentially follows from the summation of a geometric series, see (2.2.6),

$$\mathcal{O}\Bigg(\sum_{j=j_0}^{J} N_j\Bigg) = \mathcal{O}(N_J) \,. \tag{4.3.25}$$

where we use the fact that only a constant amount of iterations per level is required for a reduction of the error by a factor of 2. □

Since prolongation and restriction are trivial operations in the wavelet setting, there are no additional difficulties with respect to the implementation. We have thus constructed an iterative solver for the wavelet discretisation of the elliptic boundary value problem which achieves optimal computational complexity.

	$(-\Delta)_0$		$(-\Delta+1)_0$		$-\Delta+1$	
j	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$
3					232	229
4	103	103	93.7	93.7	350	244
5	166	118	151	107	393	255
6	207	129	188	117	433	262
8	244	137	221	125	493	271
10	263	141	239	128	531	276
12	274	144	249	130	557	278

$n = 1$

	$(-\Delta)_0$		$-\Delta+1$	
j	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$
3			532	519
4	51.7	51.7	697	627
5	175	101	739	646
6	570	337	768	664
8	1222	738	798	681

$n = 2$

	$(-\Delta)_0$		$-\Delta+1$	
j	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$
3			1238	1103
4	34.6	34.6	3410	1917
5	2956	1015	3930	2228
6	14600	5476	4330	2459

$n = 3$

Table 4.1: We show the condition numbers $\kappa(\mathbf{A})$ (4.3.6). *We provide three tables for the spatial dimensions* $n = 1, 2, 3$. *The suffix* $()_0$ *refers to homogeneous boundary conditions, while* $\mathbf{D}$ *and* $\mathbf{D}_a$ *designate the type of diagonal scaling according to* (2.2.48) *and* (4.3.11).

Condition Numbers in Uniform Discretisation

The convergence rate of the solution of (4.3.15) depends on the condition number of the stiffness matrix. Although it is uniformly bounded and therefore independent of the level of resolution j, we are still interested in its actual values on the various levels, since smaller values generally induce shorter computation times. These values depend on the choice of wavelet basis.

We have selected the construction of biorthogonal spline wavelets detailed in Section 3.3, with $d = 2$ and $\tilde{d} = 4$. In Table 4.1, we have collected the condition numbers of the stiffness matrices for three different situations, namely for the operators $-\Delta$ and $-\Delta + 1$ with homogeneous boundary conditions and the operator $-\Delta + 1$ with inhomogeneous boundary conditions. We have devoted separate tables to the spatial dimensions 1, 2 and 3. For each combination, we list the condition numbers with either the classical diagonal scaling (2.2.48) or the operator-adapted scaling from (4.3.11).

First of all, the results confirm that the condition numbers are uniformly bounded. Secondly, we assert that the use of $\mathbf{D}_a$ is always superior to the standard diagonal scaling by a factor between about 1.6 and 2.7. Finally, the condition numbers increase exponentially with the spatial dimension as expected by the tensor product approach.

In Table 4.2, we examine the effect of the transformations to the nodal basis from (3.3.74) and (3.3.75) on the condition numbers. This transformation does not lead to reduced condition numbers with homogeneous boundary conditions. However, for inhomogeneous boundary conditions the improvements are significant. In this scenario, the condition numbers of the stiffness matrix increase only with a factor less than 2 with the spatial dimension. These results confirm the observation made in the previous chapter that the transformation to the nodal basis is generally advantagous for free boundary conditions.

	$(-\Delta)_0^{\mathbf{K}}$		$(-\Delta+1)_0^{\mathbf{K}}$		$(-\Delta+1)^{\mathbf{CK}}$	
j	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$
3					280	256
4	103	103	93.7	93.7	376	263
5	369	143	335	130	473	289
6	489	154	444	140	540	301
8	599	162	544	147	635	319
10	639	165	580	150	695	330
12	658	167	598	152	735	337

$n = 1$

	$(-\Delta)_0^{\mathbf{K}}$		$(-\Delta+1)^{\mathbf{CK}}$	
j	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$
3			312	256
4	51.7	51.7	841	308
5	1171	143	871	372
6	1371	517	895	416
8	2503	1279	927	480

$n = 2$

	$(-\Delta)_0^{\mathbf{K}}$		$(-\Delta+1)^{\mathbf{CK}}$	
j	$\mathbf{D}$	$\mathbf{D}_a$	$\mathbf{D}$	$\mathbf{D}_a$
3			337	256
4	34.6	34.6	1745	520
5	13100	1615	1897	557
6	60000	14400	1936	572

$n = 3$

	$(-\Delta+1)^{\mathbf{CK}}$	
j	$\mathbf{D}$	$\mathbf{D}_a$
3	387	256
4	3229	816
5	3728	917

$n = 4$

Table 4.2: In this table, we present condition numbers of the stiffness matrix with the transformation to the nodal basis from Section 3.3.5. The notation is the same as in Table 4.1. The transformation $\mathbf{C}_j$ (3.3.74) *is only applicable for free boundary conditions, while* $\mathbf{K}_j$ (3.3.75) *can be applied in any case.*

4.3.3 Additional Preconditioning

We have so far reasoned and confirmed numerically that the wavelet discretisation described in the previous section yields stiffness matrices with uniformly bounded condition numbers. Yet, it is tempting to further examine the multi-level structure of the wavelet discretisation to gain additional improvements with respect to conditioning.

We have developed an additional preconditioning technique which uses transformations on the coarsest level. It is based on the observation that already the functions in the space S_{j_0} give rise to a strong fraction of the full condition number. It is then only moderately enlarged further by the addition of the wavelets functions from the spaces W_j. Consequently, we propose a correction on the level j_0 which indeed provides substantial improvements. These are preserved for higher levels. By this approach we are able to reduce the condition number of the stiffness matrix by more than an order of magnitude.

We begin with an analysis of the stiffness matrix on the coarsest level, where the wavelet transformation (2.2.20) reduces to the identity,

$$\mathbf{A}_{j_0} = a(\Phi_{j_0}, \Phi_{j_0})\,. \tag{4.3.26}$$

As it is symmetric positive definite, we can diagonalise it according to

$$\mathbf{A}_{j_0} = \mathbf{U}\mathbf{S}\mathbf{U}^T\,, \tag{4.3.27}$$

with a unitarian transformation matrix $\mathbf{U}$ and a diagonal $\mathbf{S}$ containing the positive eigenvalues. The distribution of the eigenvalues of $\mathbf{S}$ (which are the same as those of $\mathbf{A}_{j_0}$) can then be analysed. Usually we encounter only a small number of them which significantly spread the spectrum towards zero or infinity. This motivates a transformation which clamps some extreme eigenvalues into a smaller range. We thus replace the matrix $\mathbf{S}$ with a diagonal matrix $\hat{\mathbf{S}}$ which exhibits a significantly smaller range of eigenvalues, and hence a smaller condition number, according to

$$\mathbf{S} \mapsto \hat{\mathbf{S}}\,, \qquad \mathbf{A}_{j_0} \mapsto \hat{\mathbf{A}}_{j_0} := a(\mathbf{C}_j^T\Phi_{j_0}, \mathbf{C}_j^T\Phi_{j_0}) = \mathbf{U}\hat{\mathbf{S}}\mathbf{U}^T\,. \tag{4.3.28}$$

diag($\mathbf{S}$)	265	254	205	142	81.1	42.9	27.6	12.1	1.14
diag($\hat{\mathbf{S}}$)	265	254	205	142	81.1	42.9	27.6	12.1	12.1

Table 4.3: We provide an example for the diagonal matrices $\mathbf{S}$ *and* $\hat{\mathbf{S}}$*. We have chosen the case* $n = 1$ *with free boundary conditions.*

	$-\Delta + 1$		$(-\Delta + 1)^{\mathbf{CK}}$	
j	0	1	0	1
3	229	22.3	256	27.1
4	244	23.9	263	27.9
5	255	25.0	289	30.6
6	262	25.7	301	31.9
8	271	26.6	319	33.9
10	276	27.1	330	35.0
12	278	27.3	337	35.8

$n = 1$

	$-\Delta + 1$				$(-\Delta + 1)^{\mathbf{CK}}$			
j	0	1	4	5	0	1	3	4
3	519	78.2	76.0	49.5	256	27.8	17.3	9.64
4	627	129	128	124	308	33.4	20.9	11.8
5	646	149	149	147	372	40.4	25.3	14.3
6	664	165	165	165	416	45.1	28.2	16.0
8	681	179	179	179	480	52.1	32.6	18.4

$n = 2$

	$-\Delta + 1$		$(-\Delta + 1)^{\mathbf{CK}}$		
j	0	9	0	1	4
3	1103	269	256	28.5	18.3
4	1917	1913	520	57.8	37.1
5	2228	2222	557	62.0	39.8
6	2459	2443	572	63.6	40.9

$n = 3$

Table 4.4: We present condition numbers of the system matrix with applied preconditioning transformation on the lowest level, as described in Section 4.3.3. The digit at the head of each column indicates the number of small eigenvalues which have been shifted upward. The number 0 *corresponds to no additional preconditioning at all and is included for reference only. We have always used the exact diagonal* $\mathbf{D}_a$*.*

As an example, we have provided the matrices $\mathbf{S}$ and $\hat{\mathbf{S}}$ for a typical situation in Table 4.3. With a shift of the smallest eigenvalue, we have improved the condition number by a factor of 10.6.

The change from $\mathbf{S}$ to $\hat{\mathbf{S}}$ defines a transformed generator basis according to (2.3.22) with the symmetric matrices

$$\mathbf{C}_{j_0} = \mathbf{U}(\mathbf{S}^{-1}\hat{\mathbf{S}})^{\frac{1}{2}}\mathbf{U}^T\,, \qquad \mathbf{C}_j = \mathbf{I} \quad \text{for} \quad j > j_0\,. \tag{4.3.29}$$

This is a recipe which allows to adjust the eigenvalues on the coarsest level at will and thus provides a means to systematically tune the condition numbers. We found that only very few small eigenvalues occur, which are grouped in clusters, such that it is sufficient to correct a small number of them.

We confirmed by numerical experiments that this procedure improves the condition numbers for all levels j. Table 4.4 shows several results for free boundary conditions. The improvement are largest when using the nodal generator basis, where we gain a factor between 10 and 25.

Remark 4.10. *The change of basis described here has side effects on other matrices, for example mass matrices and Riesz operators, which may even lead to a degradation in their respective condition numbers. In practise, this is not problematic if only a small number of eigenvalues is changed. In general however, it is advisable to monitor the condition numbers of all matrices of interest.*

Remark 4.11. *The diagonalisation of the stiffness matrix occurs on the coarsest level and costs* $\mathcal{O}(N_{j_0}^3)$ *operations. The application of the matrix* $\mathbf{C}_{j_0}$ *also affects the coarsest level only. Both cases involve a computational effort which is sublinear with respect to the highest level* N_J*. Therefore, the overall runtime* *complexity $\mathcal{O}(N_J)$ is conserved by these additional computations. Furthermore, the decomposition needs to be done only once and can be reused for all subsequent runs of the programme.*

Conclusion 4.12. *For the remainder of this thesis, we will work with B-spline wavelets of order $d = 2$ and $\tilde{d} = 4$ as constructed in Section 3.3, an internal scaling of $r = \sqrt{2}$* (3.3.58) *and free boundary conditions. In this case, the coarsest level of resolution is $j_0 = 3$. Moreover, our numerical results suggest to always use the diagonal matrix $\mathbf{D}_a$ as defined in* (4.3.11)*, and to employ the transformations* (3.3.74), (3.3.75) *and an additional preconditioning via the shift of one eigenvalue for all spatial dimensions.*

4.3.4 Pre- and Postprocessing

The complete solution process of one of the boundary value problems (4.2.9) or (4.2.11) involves additional computations apart from the iterative solution of the linear system (4.3.15). As a first step, the right hand side f and possibly the normal derivative on the boundary g need to be discretised to form the vectors $\mathbf{f}$ and $\mathbf{g}$. Depending on the nature of the wavelet basis, this must be done by interpolation or integration. The discrete solution $\mathbf{x}$ computed by the iterative scheme then needs to be reported, in general by plotting. To this end, function values need to be computed. These tasks require pre- and postprocessing steps which we adress here.

The nodal basis will play an important role in this context. It is denoted by Z_j, cf. (3.3.68). It is most convenient for interpolation, integration and plotting. The iterative solver however uses the diagonally scaled and possibly conditionally improved wavelet basis $\Psi^{s\prime}$ according to (2.2.48) and (4.3.29). Several intermediate bases are involved according to the following scheme,

$$Z_j \xleftrightarrow{\mathrm{P}} \Phi_j \xleftrightarrow{\mathrm{W}} \Psi_j \xleftrightarrow{\mathrm{D}} \Psi_j^s \xleftrightarrow{\mathrm{C}} \Psi_j^{s\prime} \,. \tag{4.3.30}$$

The transformations between these bases can be applied in linear time, so we ensure optimal complexity for the complete algorithm.

P This step corresponds to the transformation used for plotting, and its inverse. It is the only step which demands different implementations for the primal and the dual side.

For the primal basis, this is a straightforward operation, as the single-scale basis Φ_j in both constructions of wavelets from Chapter 3 is already piecewise linear. If the generator basis has not yet been transformed to the nodal basis as suggested in Section 3.2.5 or Section 3.3.5, respectively, this transformation comes in here. For higher polynomial order, a transformation to the corresponding B-spline basis may be convenient.

For the dual spline wavelets, we have described in Section 3.3.4 an approximate transformation which can be used at this point.

W Here we employ the transformation between single-scale and multi-scale basis as defined in (2.2.19), (2.2.20) and (2.3.16). The primal and dual procedures are essentially symmetric by (2.3.12).

D The diagonal scaling is the vital step which performs the shift in the Sobolev scales as described in (2.2.48), (2.2.49). It leads to a well-conditioned system, as has been discussed earlier in this chapter. The diagonal matrix can be specifically adapted to improve the condition number of the stiffness matrix (4.3.11).

C This transformation refers to the procedure developed in Section 4.3.3. It only applies to the coarsest level and does not modify the wavelet coefficients on higher levels. The goal is to reduce the constants in the equivalence relation (4.3.10), and thus to save time on the computation.

We can see from this description that the wavelet framework contains several types of transformations. They reflect the way in which Riesz bases for the Hilbert space H are constructed. The result is an inherently well-conditioned scheme of optimal computational complexity.

Part II

Fast Solution of a PDE-Constrained Optimal Control Problem in Wavelet Discretisation

Chapter 5

A Linear-Quadratic Elliptic Optimal Control Problem in Wavelet Coordinates

5.1 Introduction

For a multitude of physical phenomena, the appropriate mathematical formulation falls into the class of partial differential equations. They describe the dependence of the state of a system on driving and restraining forces, taking into account specific conditions on the boundary of the physical system. The numerical solution of such a system is often called *simulation*, in the sense that for a given set of physical conditions, the hypothetical action of reality is predicted computationally. Inherent is the idea of a strict causal relationship: The forces and boundary conditions are fixed beforehand, and they uniquely determine the state, which is unknown a priori and must be computed.

Recently, an increasingly active branch of research has emerged which follows a more general interpretation. While the PDE plays a central role, as it captures the physical principle that any given forces and boundary conditions cause an associated state, the issue is brought up which state is actually *desirable.* This question is meaningless in the context of physical causality alone. However, it immediately makes sense when two additional premises hold.

- The context of application provides criteria for the *quality* of the state. For example, an industrial product is rated "good" when it is close to a target specification within certain error bounds. Or the temperature distribution within a solid or liquid material may need to be adjusted to match a given pattern to minimise heating and cooling costs.

- The physical system offers certain possibilities of input which permit deliberate adjustment, and influence the state. Consider for example devices which can draw air through inlets in the surface of an aeroplane wing and thus affect the total drag, or water valves which open selectively to cool hot glass or steel.

The additional external influences are subsumed as unknown variable u, which control the state through the PDE. The quality of any specific pair of control u and state $y(u)$ is measured by a newly introduced objective functional $J(y, u)$, which must be minimised by finding the optimal control and its dependent state. The overall goal has shifted from simulation to *optimisation*, which brings in an additional layer of structural complexity.

In the context of partial differential equations, several physically meaningful types of control are possible. We might think of the control as determining material properties or electromagnetic influences which locally change the differential operators. Another important example are *parameter identification problems*, where u models diffusion, convection or reaction rates. It is also intuitive to control the external forces on the right hand side of the PDE, either on the whole domain or on parts of the boundary, which is the case which we cover in this document. Specifically, the PDE takes the form

$$Ay = f + Eu \tag{5.1.1}$$

with an elliptic operator A, see (4.2.5).

We consider an objective functional of *tracking type*, motivated by the engineering point of view where it is often desirable to steer the state y close to a given target y_*, while the action of the control u should be reasonably cheap and therefore small in magnitude. The general form reads

$$J(y,u) = \frac{1}{2}\|Ty - y_*\|_Z^2 + \frac{\omega}{2}\|u\|_U^2 \,, \tag{5.1.2}$$

where we will direct a special focus to the choice of the spaces Z and U. While they are commonly chosen as L_2, and sometimes the norm on U is selected as $|\cdot|_{H^1}$ to enforce higher regularity of the control, we employ the special ability of wavelet discretisations to model Sobolev norms of arbitrary smoothness in $\mathbb{R}$. We propose a novel construction here which contains the standard choices described above as special cases and thus allows for greater flexibility in modelling.

The numerical solution of optimal control problems with partial differential equations as constraints poses several major mathematical difficulties in addition to the simulation of the PDE alone. The goal of minimising a functional subject to constraints leads to necessary and sufficient conditions on the minimiser. Coupling the constraints to the functional by a Lagrangian multiplier, a coupled system of equations arises which exhibits a specific saddle point structure. Since the diversity of physical systems and approaches to modelling is far larger for the complete control problem as compared to the PDE alone, numerical solvers are usually developed for specific cases, and attempts towards a unified formulation are rare.

An overview on modern PDE-constrained optimisation is given in [11]. The optimal control of fluid flow seems to be the practically most relevant area today. A general survey and a discussion of different technical approaches via the optimality system, the analysis of sensitivities or adjoint-based methods is provided in [80]. Generally, optimal control problems motivated from industrial applications involve partial differential equations which are meant to describe realistic phenomena. This may involve large amounts of data and necessitate parallel approaches [13, 14]. Moreover, these types of problem require a relatively complex mathematical framework, and the control problem becomes even more difficult to handle. Consequently, even fundamental questions such as existence and uniqueness of solutions often remain unanswered.

Thus it makes sense to study algorithmic concepts for reasonably complex model problems. An overview on several problem classes and techniques is given in [136]. The general discussion is very active, see e.g. [89, 92, 135] for variants of the SQP method, [93, 98] for primal-dual active set strategies, and various other examples [81, 82, 88, 124, 126]. Some applications to the Navier Stokes equations are discussed in [67,83,95], and reaction diffusion systems have been treated in [17]. The perhaps most general adaptive algorithm we know of uses error estimators based on duality for the Navier Stokes equations [7,8]. While a fairly large collection of mathematical tools and numerical algorithms exists for this class of control problems, error estimators are often heuristic, and convergence results are usually not rigorously derived.

Solid mathematical results on error bounds and convergence rates have so far been obtained only for the most basic model problems. Linear elliptic partial differential equations as constraints and an optimisation functional which is quadratic in y and u have been theoretically studied in [109] in the context of L_2

spaces. This scenario can be summarised under the term *linear-quadratic elliptic* control problem. A multigrid scheme for this restricted case has been devised in [84]. Recently, an adaptive wavelet approach has been specified which allows for more general norms in the objective functional [48]. Still, large scale numerical examples have not been presented in either case.

In this thesis we develop a general numerical wavelet scheme for a linear-quadratic optimal control problem. It integrates the fast wavelet solver of elliptic PDEs described in the last chapter with the specific demands of the optimisation process, providing asymptotically optimal complexity. We then provide extensive numerical results computed with a newly written large scale implementation, for uniform and adaptive discretisations.

Specifically, we introduce the following new mathematical and algorithmic concepts.

- We follow the wavelet framework from [48, 106]. This allows general Sobolev spaces instead of L_2 for the state and the control, which permits to treat all functions in their respective natural spaces originating from the weak formulation. Additionally, the norms of the spaces used for the observation of the state and the control can be tuned, which allows for greater freedom in modelling. We explicitly include Sobolev spaces of fractional smoothness.

- We develop a numerical scheme which converges with a numerical effort proportional to the number of unknowns. This can be derived from a generalisation of the ideas known for wavelet methods for elliptic partial differential equations. In [48], two nested Richardson iterations have been proposed. We accelerate this significantly by the introduction of two nested conjugate gradient schemes [28].

- In [28, 48], the original functional is transformed into an equivalent functional in an infinite-dimensional wavelet representation over ℓ_2. Thus, the numerical solution does generally not converge towards the exact solution, but to a different function. The magnitude of this deviation, which determines the quality of the numerical solution, depends of the constants occurring in the wavelet norm equivalences.

  We propose a scheme which exactly reproduces Sobolev norms for integral smoothness indices. Consequently, we have *equality* instead of equivalence for the discrete functional, which resolves the discrepancy between the computed and the exact optimal control. For fractional smoothness, we interpolate between integer cases in a continuous fashion, which again yields equivalence, with presumably improved constants.

- We make use of the full adaptive potential of wavelets as proposed in [48], and apply it to our numerical scheme which incorporates all modifications which we have just listed here.

5.2 Mathematical Formulation

We begin with the concrete mathematical definition of the specific class of optimal control problems considered here. We mainly follow the notation from [28, 48].

5.2.1 Definitions and Notation

Let Y denote the state space and U the control space. They are assumed to be closed subspaces of Hilbert spaces, with topological duals Y', U' and associated dual forms $\langle \cdot, \cdot \rangle_Y$ and $\langle \cdot, \cdot \rangle_U$. This is abbreviated by $\langle \cdot, \cdot \rangle$ when there is no risk of confusion. The norm in a Hilbert space X is defined as usual by $\|\cdot\|_X := \sqrt{(\cdot, \cdot)_X}$ where $(\cdot, \cdot)_X$ is the standard scalar product of X.

The objective functional $J(y, u)$ (5.1.2) consists of two parts, namely the *tracking* term, which measures the distance of the state y from a predefined target y_*, and the *regularisation* term, which ensures the well-posedness of the problem and is weighted with a parameter $\omega > 0$. This functional is differentiable, which will be used to derive the conditions for minimisation, also called *optimality conditions*.

The mathematical modelling envisages some kind of measurement process T in the control functional, which operates on the state y and produces an observed value $Ty \in Z$. The *observation space* Z is again a (closed subspace of) a Hilbert space with an associated dual form. The mapping from the state space to the observation space is performed by the continuous linear operator $T : Y \to Z$,

$$\|Ty\|_Z \lesssim \|y\|_Y \,. \tag{5.2.1}$$

A typical situation is the case of *observation on the boundary* where the state y is defined on a domain Ω and it is measured on the boundary $\partial\Omega$ or a part thereof. Then the observation is realised by a *trace operator*, hence the operator has been named T. We adhere to this convention throughout this document. Another special case is given when Y is embedded in Z, and it holds $T = \mathrm{id}$ with the continuous embedding

$$\|v\|_Z \lesssim \|v\|_Y \,, \qquad v \in Y \subset Z \,. \tag{5.2.2}$$

This is the setting which will be investigated more closely numerically in the forthcoming chapters.

As outlined above, the control $u \in U$ enters on the right hand side of the partial differential equation (5.1.1). As the source term is in the dual space of Y (which is denoted by Q),

$$f \in Y' = Q \,, \tag{5.2.3}$$

we require a continuous linear operator $E : U \to Q$ to transport the control u into the space Q,

$$\|Ew\|_Q \lesssim \|w\|_U \,. \tag{5.2.4}$$

In the case of *boundary control*, the control is defined on $\partial\Omega$, and E is an *extension operator*, e.g. the adjoint of a trace operator. There exists also the possibility of *distributed control* which is modelled by $E = \mathrm{id}$ and the continuous embedding

$$\|w\|_Q \lesssim \|w\|_U \,, \qquad w \in U \subset Q \,. \tag{5.2.5}$$

Our numerical experiments will cover the distributed case.

We consistently keep the operators T and E in all formulas even in cases where they reduce to the identity. There are several reasons to do so.

- The flexibility obtained by assuming general T and E allows for a unified mathematical framework which covers all relevant cases.
- The operators are used to distinguish functions in the space Y from those in Z on the one hand and functions in U from those in Q on the other.
- When either operator is the identity, but its preimage and image are Sobolev spaces of different regularity on the same domain, the operator in wavelet discretisation becomes a diagonal scaling matrix which is naturally unequal to the identity matrix. This observation is of fundamental importance for the numerical solution process.

To formulate the elliptic constraints in weak form, let $a(v, w) : Y \times Y \to \mathbb{R}$ be a continuous and Y-elliptic bilinear form as described in Chapter 4, i.e.,

$$a(v, v) \sim \|v\|_Y^2 \,, \qquad v \in Y \,. \tag{5.2.6}$$

This defines a linear operator $A : Y \to Y'$ by $\langle v, Aw \rangle = a(v, w)$. Relation (5.2.6) implies that A is boundedly invertible as an operator from Y to Y', i.e.,

$$\|Av\|_{Y'} \sim \|v\|_Y \,, \qquad v \in Y \,. \tag{5.2.7}$$

We consider the following variational formulation of the PDE,

$$a(y,v) = \langle f + Eu, v\rangle \qquad \text{for all } v \in Y\,, \tag{5.2.8}$$

whose operator form is given by

$$Ay = f + Eu\,. \tag{5.2.9}$$

The abstract control problem is then formulated as follows.

Problem 5.1 (ACP). *For a given target observation $y_* \in Z$, right hand side $f \in Y'$ and weight parameter $\omega > 0$, minimise the functional*

$$J(y,u) \coloneqq \frac{1}{2}\|Ty - y_*\|_Z^2 + \frac{\omega}{2}\|u\|_U^2 \tag{5.2.10}$$

over $(y,u) \in Y \times U$ subject to the linear operator equation (5.2.9).

Remark 5.2. *If the data f is compatible to the target observation y_*, i.e,*

$$y_* = TA^{-1}f\,, \tag{5.2.11}$$

then the functional can be minimised trivially by setting $y = A^{-1}f$ and $u = 0$, yielding $J(y,u) = 0$.

The norms occuring in the objective functional (5.2.10) can be expressed as

$$\|v\|_V^2 = \langle v, R_V v\rangle_V\,, \qquad v \in V\,, \tag{5.2.12}$$

using a self-adjoint Riesz operator $R_V : V \to V'$. A standard way to reformulate Problem 5.1 is to append the constraints (5.2.8) to the functional via a Langrangian multiplier. To this end, we introduce the variable $p \in Y$ and define the Lagrangian functional as

$$L(y,u,p) \coloneqq \frac{1}{2}\langle Ty - y_*, R_Z(Ty - y_*)\rangle_Z + \frac{\omega}{2}\langle u, R_U u\rangle_U + a(p,y) - \langle p, f + Eu\rangle_Y\,. \tag{5.2.13}$$

The function p is also called the *adjoint* variable or the *costate*. Differentiating with respect to p, y and u and setting the derivatives to zero, we obtain the three equations

$$\begin{aligned}
\partial_p L(v) &= a(v,y) - \langle v, f + Eu\rangle_Y = 0\,, && \text{for all } v \in Y\,, && \text{(5.2.14a)}\\
\partial_y L(w) &= \langle w, T'R_Z(Ty - y_*) + a(p,w)\rangle_Y = 0\,, && \text{for all } w \in Y\,, && \text{(5.2.14b)}\\
\partial_u L(q) &= \langle q, \omega R_U u - E'p\rangle_U = 0\,, && \text{for all } q \in U\,. && \text{(5.2.14c)}
\end{aligned}$$

Here (5.2.14a) recovers the state equation (5.2.8), (5.2.14b) is denoted the adjoint equation and (5.2.14c) the *design* equation. These are the first order necessary conditions for the solution of Problem 5.1.

5.2.2 Example Problems

We now provide several examples which are all covered by the framework introduced above. Let $\Omega \subset \mathbb{R}^n$ always denote a bounded Lipschitz domain with boundary $\partial\Omega$ as in Chapter 4. Choosing $Z = U = L_2(\Omega)$ in the objective functional is the classical case treated in [109]. In the wavelet framework however, it is possible to employ Sobolev or Besov norms on Ω or (parts of) its boundary $\partial\Omega$. Thus, we may use norms which may be termed *natural* with regard to the underlying variational formulation. Examples are the norms $\|\cdot\|_Y$, $\|\cdot\|_Q$ and fractional trace norms in boundary observation and/or control [48].

The problems with distributed control or observation may be of less practical importance, however they serve as good illustrations for the essential mechanisms and are particularly suited to study numerical effects. Dirichlet boundary controls treated by saddle point formulations are investigated in [103,104,118]. Various combinations of boundary or distributed observations and controls under different boundary conditions have been listed in [84] for the L_2 case.

Dirichlet Problem with Distributed Control

The prototype for a Dirichlet problem with distributed control appears for

$$a(v,w) := \int_\Omega \nabla v \cdot \nabla w \,\mathrm{d}x\,, \qquad Y := H_0^1(\Omega)\,, \qquad Q := Y' = H^{-1}(\Omega) = (H_0^1(\Omega))', \tag{5.2.15}$$

which corresponds to the standard weak form of the elliptic boundary value problem

$$-\Delta y = f + u \qquad \text{in } \Omega\,, \tag{5.2.16a}$$
$$y = 0 \qquad \text{on } \partial\Omega\,. \tag{5.2.16b}$$

Choices for Z, U in (5.2.10) which are admissible according to (5.2.2), (5.2.5) are

$$Z := H_{00}^s(\Omega)\,, \qquad 0 \le s \le 1\,, \tag{5.2.17a}$$
$$U := H^{-t}(\Omega) = (H_{00}^t(\Omega))'\,, \qquad 0 \le t \le 1\,, \tag{5.2.17b}$$

where $H_{00}^s(\Omega)$ is defined as the intersection of $H^s(\Omega)$ with the set of functions whose trivial extension by zero is in $H^s(\mathbb{R}^n)$. The case $s = 1$ or $t = 1$ corresponds to choosing a natural norm for the observation or control space while setting $s = t = 0$ leads to the classical case of norms on $L_2(\Omega)$.

Neumann Problem with Distributed Control

Define

$$a(v,w) := \int_\Omega (\nabla v \cdot \nabla w + vw)\,\mathrm{d}x\,, \qquad Y := H^1(\Omega)\,, \qquad Q := Y' = (H^1(\Omega))'\,, \tag{5.2.18}$$

and consider as constraint for given data $\tilde f \in (H^1(\Omega))'$, $g \in H^{-1/2}(\partial\Omega)$,

$$a(y,v) := \langle \tilde f, v\rangle + \int_{\partial\Omega} gv \,\mathrm{d}s + \langle u, v\rangle \qquad \text{for all } v \in Y\,. \tag{5.2.19}$$

This can be derived from the strong form of the standard non-homogeneous Neumann problem with distributed control,

$$-\Delta y + y = \tilde f + u \qquad \text{in } \Omega\,, \tag{5.2.20a}$$
$$\frac{\partial y}{\partial n} = g \qquad \text{on } \partial\Omega\,, \tag{5.2.20b}$$

where $\frac{\partial}{\partial n}$ denotes the normal derivative in the direction of the outward normal. Abbreviating the data f by $\langle f, v\rangle := \langle \tilde f, v\rangle + \int_{\partial\Omega} gv\,\mathrm{d}s$, the constraints (5.2.19) can then be formulated as an operator equation,

$$Ay = f + u\,, \tag{5.2.21}$$

where A is indeed an isomorphism from Y to Y'. Analogously to (5.2.17) we can take here

$$Z := H^s(\Omega)\,, \qquad 0 \le s \le 1\,, \tag{5.2.22a}$$
$$U := (H^t(\Omega))'\,, \qquad 0 \le t \le 1\,, \tag{5.2.22b}$$

where again $s = t = 1$ corresponds to choosing the natural norms for y and u.

So far, both examples were characterised by $T = \mathrm{id}$ and $E = \mathrm{id}$. This may be contrasted by an observation on the boundary $\partial\Omega$ whereupon the natural observation space is $Z = H^{1/2}(\partial\Omega)$. Then $T : H^1(\Omega) \to H^{1/2}(\partial\Omega)$ coincides with the trace operator, and the control acts towards the match of y with y_* only on the boundary.

Neumann Problem with Boundary Control

Next we consider the Neumann problem from above, but this time with $E \neq \mathrm{id}$. To this end, let the boundary $\Gamma := \partial\Omega$ be decomposed into two parts $\Gamma = \overline{\Gamma_{\mathrm{N}}} \cup \overline{\Gamma_{\mathrm{C}}}$, the Neumann and the control boundary, where $\overline{\Gamma_{\mathrm{C}}}$ has non-vanishing $n-1$ dimensional measure. While the observation space Z is the same as in the previous example, the constraints change to

$$a(y, v) := \langle \tilde{f}, v\rangle + \int_{\Gamma_{\mathrm{C}}} g T_{\mathrm{C}}(v)\,\mathrm{d}s + \int_{\Gamma_{\mathrm{C}}} u T_{\mathrm{C}}(v)\,\mathrm{d}s \qquad \text{for all } v \in Y = H^1(\Omega) \tag{5.2.23}$$

for given $\tilde{f} \in Y'$, $g \in (H^{1/2}(\Gamma_{\mathrm{C}}))'$. The strong form is given by

$$-\Delta y + y = \tilde{f} \qquad \text{in } \Omega\,, \tag{5.2.24a}$$

$$\frac{\partial y}{\partial n} = \begin{cases} 0 & \text{on } \Gamma_{\mathrm{N}}\,, \\ g + u & \text{on } \Gamma_{\mathrm{C}}\,. \end{cases} \tag{5.2.24b}$$

The necessary condition for the right hand side of (5.2.23) to be well-defined demands that $u \in U = (H^t(\Gamma_{\mathrm{C}}))' \subset (H^{1/2}(\Gamma_{\mathrm{C}}))'$. The extension operator E is then the adjoint of the trace operator $T_{\mathrm{C}} : H^1(\Omega) \to H^t(\Gamma_{\mathrm{C}})$, $t \leq \frac{1}{2}$, to the control boundary Γ_{C}. It is defined as

$$\langle Eu, w\rangle_{(H^1(\Omega))' \times H^1(\Omega)} = \int_{\Gamma_{\mathrm{C}}} u T_{\mathrm{C}}(w)\,\mathrm{d}s\,. \tag{5.2.25}$$

The strong form of the constraint then reads

$$Ay = f + Eu\,. \tag{5.2.26}$$

Remark 5.3 (Dirichlet boundary control). *For linear-quadratic elliptic problems with Dirichlet boundary control, a standard way to handle the constraints are saddle point formulations [103]. These do no longer satisfy the ellipticity condition* (5.2.6). *It is nonetheless possible to apply the methods described in this document. For a discussion of this case, we refer to [40].*

Remark 5.4 (Observation on the boundary). *All cases described so far have used* distributed observation. *Alternatively, the observation space can be chosen as a space defined on an observation boundary $\Gamma_{\mathrm{O}} \subset \partial\Omega$ with strictly positive measure. The natural space in this case is $H^{1/2}(\Gamma_{\mathrm{O}})$ which could be relaxed by the more general ansatz $Z = H^s(\Gamma_{\mathrm{O}})$. The observation mapping is then given by a trace operator with additional norm shift $T : H^1(\Omega) \to H^s(\Gamma_{\mathrm{O}})$. The natural norm is chosen by $s = \frac{1}{2}$. The classical case corresponds to the choice $\Gamma_{\mathrm{O}} = \partial\Omega$ and $s = 0$, which has been treated with adaptive finite elements [8]. The case of general s is studied in [105, 118].*

5.3 Numerical Evaluation of Sobolev Norms

To evaluate the norms occurring in the functional $J(y, u)$ (5.2.10) in the wavelet context, we introduce *Riesz matrices* $\mathbf{R}_V$ for Sobolev spaces V according to

$$\|v\|_V^2 = \mathbf{v}^T \mathbf{R}_V \mathbf{v}\,, \qquad V \ni v = \mathbf{v}^T \Psi_V\,. \tag{5.3.1}$$

They are related to the Riesz operators from (5.2.12) by

$$\mathbf{R}_V = \langle \Psi_V, R_V \Psi_V\rangle_V\,. \tag{5.3.2}$$

When Ψ_V is a Riesz basis for V, it follows that $\kappa(\mathbf{R}_V) \sim 1$. While for example the Riesz matrix for $V = L_2$ can be specified exactly, the general matrix $\mathbf{R}_{H^s}$ for arbitrary $s \in \mathbb{R}$ is usually not accessible to a numerical evaluation. For non-integer cases it has to be replaced by an approximate equivalent version which is actually numerically computable.

We now examine to what extent these evaluations can be optimised in the framework of biorthogonal wavelets. As there is no unique way of defining fractional Sobolev norms, equivalence is all that we should reasonably expect anyway. However, we may demand that a discrete norm of fractional smoothness is equal to the discrete L_2 norm for all constant functions. For integral norms, exactness is a well-defined notion, and consequently the scale of integers for which a given discrete norm is exact is a direct measure of its practicability for numerical purposes. Our goal is thus a formulation of a discrete norm which fulfils the following criteria.

- The discrete norm is equivalent to Sobolev norms on H^s for all $s \in \mathbb{R}$.
- The discrete norm is equal to the Sobolev norm on H^s for $s \in \mathbb{Z}$.
- The discrete norm for all $s \in \mathbb{R}$ is equal to the L_2 norm for all constant functions.
- The discrete norm is computationally efficient for all $s \in \mathbb{R}$.

Clearly the last requirement is guided by striving for numerical efficiency. It means that in the context of a finite wavelet discretisation, the discrete norm can be evaluated with a computational cost which is proportional to the number of unknowns. In this section, we propose a framework which satisfies all of the above requirements. To our knowledge, this has not been formulated before.

In the following, we consider various expansions of different functions in different spaces. To simplify notation, we call all of these functions v, and their coefficient vectors $\mathbf{v}$. The same holds for different trial functions which are all called w, with coefficient vectors $\mathbf{w}$. It should become clear from the context which functions are meant at any time.

5.3.1 Evaluation of Norms on L_2 and H^1

Let a function be expanded in an unscaled wavelet basis according to $L_2 \ni v = \mathbf{v}^T\Psi$. We understand by $(\cdot,\cdot)$ the inner product in the L_2 sense. The L_2 norm can be evaluated by

$$\|v\|_0^2 = (\mathbf{v}^T\Psi, \mathbf{v}^T\Psi) = \mathbf{v}^T(\Psi,\Psi)\mathbf{v} = \mathbf{v}^T\mathbf{M}\mathbf{v}\,, \tag{5.3.3}$$

where the *mass matrix* $\mathbf{M}$ is just the Riesz matrix $\mathbf{R}_{L_2}$ from (5.3.1), defined as

$$\mathbf{M} := (\Psi,\Psi)\,. \tag{5.3.4}$$

Turning to the norm in H^1, we can write

$$\|v\|_1^2 = \|v\|_0^2 + |v|_1^2 = \mathbf{v}^T\mathbf{M}\mathbf{v} + (\nabla\mathbf{v}^T\Psi, \nabla\mathbf{v}^T\Psi) = \mathbf{v}^T(\mathbf{M}+\mathbf{L})\mathbf{v} \tag{5.3.5}$$

with the definition of the *Laplace matrix*

$$\mathbf{L} := (\nabla\Psi, \nabla\Psi)\,. \tag{5.3.6}$$

When we expand v in the natural wavelet basis for H^1 according to (2.2.48), $H^1 \ni v = \mathbf{v}^T\Psi^1 = \mathbf{v}^T\mathbf{D}^{-1}\Psi$, we can write the H^1 norm as

$$\|v\|_1^2 = \mathbf{v}^T\mathbf{D}^{-1}(\mathbf{M}+\mathbf{L})\mathbf{D}^{-1}\mathbf{v} =: \mathbf{v}^T\mathbf{B}_1\mathbf{v}\,, \tag{5.3.7}$$

where we have hidden the definition of $\mathbf{B}_1$, the Riesz matrix in the natural wavelet basis. Note that due to Corollary 2.10, $\mathbf{B}_1$ is uniformly well-conditioned. Furthermore, $\mathbf{M}$, $\mathbf{L}$ and $\mathbf{B}_1$ are symmetric and positive definite.

5.3.2 Evaluation of Dual Norms

In the framework of biorthogonal wavelets, the evaluation of dual norms is also done with relative ease. As an example of the main technique, we first assure that L_2 can be identified with its dual. To this end, we apply the general definition of dual norms,

$$\|v\|_{V'} := \sup_{0 \neq w \in V} \frac{\langle v, w \rangle}{\|w\|_V}\,, \tag{5.3.8}$$

to the expansion ${L_2}' \ni v = \mathbf{v}^T \tilde{\Psi} =: \mathbf{v}_0^T \Psi$. For any $L_2 \ni w = \mathbf{w}^T \Psi$, we may write

$$\|v\|_{L_2{}'} = \sup_{0 \neq w \in L_2} \frac{\langle v, w \rangle}{\|w\|_{L_2}} = \sup_{0 \neq \mathbf{w} \in \ell_2} \frac{\langle \mathbf{v}^T \tilde{\Psi}, \mathbf{w}^T \Psi \rangle}{(\mathbf{w}^T \mathbf{M} \mathbf{w})^{\frac{1}{2}}} = \sup_{0 \neq \mathbf{w} \in \ell_2} \frac{\mathbf{v}^T \langle \tilde{\Psi}, \Psi \rangle \mathbf{w}}{(\mathbf{w}^T \mathbf{M} \mathbf{w})^{\frac{1}{2}}} = \sup_{0 \neq \mathbf{w} \in \ell_2} \frac{\mathbf{v}^T \mathbf{w}}{(\mathbf{w}^T \mathbf{M} \mathbf{w})^{\frac{1}{2}}}\,. \tag{5.3.9}$$

Here we have used (5.3.3) and the biorthogonality condition (2.2.44). After all, the expression is formulated in vectors and matrices on ℓ_2 and can be reduced further by the substitution $\mathbf{g} = \mathbf{M}^{\frac{1}{2}} \mathbf{w}$ as follows,

$$\|v\|_{L_2{}'} = \sup_{0 \neq \mathbf{g} \in \ell_2} \frac{(\mathbf{M}^{-\frac{1}{2}} \mathbf{v})^T \mathbf{g}}{(\mathbf{g}^T \mathbf{g})^{\frac{1}{2}}} = (\mathbf{v}^T \mathbf{M}^{-1} \mathbf{v})^{\frac{1}{2}} = (\mathbf{v}_0^T \mathbf{M} \mathbf{v}_0)^{\frac{1}{2}} = \|v\|_{L_2}\,. \tag{5.3.10}$$

From this identity, we can derive the following conclusions.

- Biorthogonal wavelet expansions on L_2 are consistent with the fact that L_2 can be identified with its dual, and the norms are identical.
- Even if the expansion $v = \mathbf{v}_0^T \Psi$ is not available, the dual norm can be efficiently computed, as $\mathbf{M}$ has uniformly bounded condition number, and its inverse can be easily applied.
- In the case of L_2, these results hold for any other uniformly stable biorthogonal basis. The multiscale nature only becomes important for general Sobolev spaces.

After these preparations, we can discuss the norm of $H^{-1} = (H^1)'$. We expand the function $H^{-1} \ni v = \mathbf{v}^T \tilde{\Psi}^1$, and the trial function $H^1 \ni w = \mathbf{w}^T \Psi^1$. The numerator of (5.3.8) is derived analogously to (5.3.9),

$$\langle v, w \rangle = \langle \mathbf{v}^T \tilde{\Psi}^1, \mathbf{w}^T \Psi^1 \rangle = \mathbf{v}^T \langle \tilde{\Psi}^1, \Psi^1 \rangle \mathbf{w} = \mathbf{v}^T \mathbf{w}\,. \tag{5.3.11}$$

We have used here the general biorthogonality relation (2.2.50). With a similar substitution as in the derivation of (5.3.10), we arrive at

$$\|v\|_{-1}^2 = \mathbf{v}^T \mathbf{B}_1^{-1} \mathbf{v}\,. \tag{5.3.12}$$

Remark 5.5. *Due to the uniformly bounded condition number of* $\mathbf{B}_1$*, the norm* $\|v\|_{H^{-1}}$ *can be evaluated efficiently in the wavelet setting. In contrast, a standard finite element ansatz would lead to some critical problems.*

- *As the Riesz basis property is in general not satisfied for finite elements, the supremum over* ℓ_2 *is not equal to the supremum over* L_2*, and* (5.3.9) *does not hold in this form.*
- *The condition number of the finite element analogon to* $\mathbf{B}_1$ *according to* (5.3.7) *is not bounded. Additional preconditioning or multigrid techniques would need to be employed here.*
- *If a biorthogonal basis is not available, the numerator in* (5.3.9) *does not simplify, which would necessitate the additional application of a mass matrix.*

The framework of biorthogonal wavelets is thus particularly suited for the evaluation of positive and negative integral Sobolev norms. Norms of higher order can be processed in the same fashion as exercised here for H^{-1}*,* H^0 *and* H^1*, provided that the ansatz functions are sufficiently regular.*

5.3.3 Evaluation of Fractional Sobolev Norms

While the evaluation of integral norms is in principle also possible in the finite element context, the realm of fractional norms is presently only accessible numerically by wavelet methods. We exclude spectral elements here as we demand compact support of all basis functions and deal with generally non-periodic boundary conditions. We have already demonstrated in Chapter 2 that wavelets can be used to construct Riesz bases for Sobolev spaces of arbitrary smoothness. The evaluation of norms in this more general context has to our knowledge not yet been systematically studied. This motivates us to propose a novel unified formulation for the numerical evaluation of both integral and fractional Sobolev norms in the wavelet framework.

State of the Art

We give here a very short review of the state of the art of the evaluation of general Sobolev norms in the wavelet framework. We consider the expansion of a function v in the unscaled wavelet basis, $v = \mathbf{v}^T\Psi$. In [48], the norm used is

$$\|v\|_{H^s}^{(1)} := (\mathbf{v}^T\mathbf{D}^{2s}\mathbf{v})^{\frac{1}{2}} . \tag{5.3.13}$$

This norm is always equivalent to $\|\cdot\|_{H^s}$, but never exact for $s \in \mathbb{Z}$. It corresponds to the simplest approximation $\mathbf{R}_V := \mathbf{I}$. In [28], we have chosen

$$\|v\|_{H^s}^{(2)} := (\mathbf{v}^T\mathbf{D}_a^s\mathbf{M}\mathbf{D}_a^s\mathbf{v})^{\frac{1}{2}} , \tag{5.3.14}$$

which is always equivalent, but only exact for $s = 0$ (the L_2 case). This conforms with the approximation $\mathbf{R}_V := \mathbf{R}_{L_2}$ and can be seen as a qualitative improvement compared to (5.3.13). For the sake of improved preconditioning, the diagonal scaling derived from the stiffness matrix (4.3.11) has been used. Anyway, both variants permit the evaluation of the corresponding dual norms up to equivalence, and they satisfy the fundamental requirement of computational efficiency.

A New, Unified Method

In this section, we develop an interpolation technique which is *exact* for integral smoothness indices and *equivalent* for fractional smoothness, and is at the same time computationally efficient. Note that this is in no way trivial. Consider for example the ansatz

$$\|v\|_{H^s}^{(3)} := \left(\mathbf{v}^T\mathbf{M}^{\frac{1-s}{2}}(\mathbf{M}+\mathbf{L})^s\mathbf{M}^{\frac{1-s}{2}}\mathbf{v}\right)^{\frac{1}{2}} . \tag{5.3.15}$$

This expression is motivated by the analogous definition for spectral elements, where it yields the norm in the sense of a Fourier decomposition for all $s \in \mathbb{R}$. In the context of wavelets, it is exact for $s = 0$ and $s = 1$, and equivalent in between. However, the fractional powers of matrices require a singular value decomposition, which is too expensive in practise. The exponentiation of a matrix is only fast for diagonal matrices. Thus, we believe that *multiplicative* interpolation with an exponent s is not sufficient in this case.

Consequently, we also include *additive* interpolation. To this end, we start with a unified notation. Let us assume that for any $i \in \mathbb{N}_0$ with $0 \leq i \leq S \in \mathbb{N}_0$, there exists an exact representation written as

$$H^i \ni v = \mathbf{v}^T\Psi^i , \qquad \|v\|_i^2 = \mathbf{v}^T\mathbf{B}_i\mathbf{v} . \tag{5.3.16}$$

Examples are $\mathbf{B}_0 = \mathbf{M}$ from (5.3.4), and $\mathbf{B}_1$ as defined in (5.3.7). Higher orders can be derived from the standard definition of Sobolev norms. We know that all $\mathbf{B}_i$ are symmetric positive definite and spectrally equivalent to the identity matrix. In the unscaled representation, these Riesz matrices transform according to

$$H^i \ni v = \mathbf{v}^T \Psi\,, \qquad \|v\|_i^2 = \mathbf{v}^T \mathbf{D}^i \mathbf{B}_i \mathbf{D}^i \mathbf{v}\,. \tag{5.3.17}$$

With the definition of the hat function $h_i(x)$ centred at $x = i$,

$$h_i(x) \coloneqq \begin{cases} x - (i-1) & \text{for } i-1 \le x < i\,, \\ (i+1) - x & \text{for } i \le x \le i+1\,, \\ 0 & \text{else,} \end{cases} \tag{5.3.18}$$

we can formulate our first proposal for a unified Riesz matrix.

Theorem 5.6. *For $s \in \mathbb{R}$ with $0 \le s \le S$, the norm defined by*

$$H^s \ni v = \mathbf{v}^T \Psi\,, \qquad \|v\|_s^2 \coloneqq \sum_{i=0}^{S} h_i(s)\, \mathbf{v}^T \mathbf{D}^s \mathbf{B}_i \mathbf{D}^s \mathbf{v}\,, \tag{5.3.19}$$

or alternatively written in the scaled wavelet basis as

$$H^s \ni v = \mathbf{v}^T \Psi^s\,, \qquad \|v\|_s^2 = \sum_{i=0}^{S} h_i(s)\, \mathbf{v}^T \mathbf{B}_i \mathbf{v} = \mathbf{v}^T \left(\sum_{i=0}^{S} h_i(s) \mathbf{B}_i \right) \mathbf{v}\,, \tag{5.3.20}$$

is equal to the standard Sobolev norms for integral s and equivalent for fractional s. It can be computed in linear time.

Proof. For $s \in \mathbb{N}_0$, only one summand survives with $i = s$, and it reduces to the expression given in (5.3.16) or (5.3.17), respectively. For fractional s, the formulation on the right of (5.3.20) is a convex combination of two summands which are both spectrally equivalent to the identity matrix. Consequently, the whole sum is spectrally equivalent to the identity matrix, and the expression yields a norm which is equivalent to the corresponding fractional Sobolev norm. Likewise, the computational efficiency is inherited as a feature of the matrices $\mathbf{B}_i$. □

In the following, we denote our choice of Riesz matrix in the natural basis as

$$\mathbf{R}_s \coloneqq \sum_{i=0}^{S} h_i(s) \mathbf{B}_i\,, \qquad 0 \le s \le S\,. \tag{5.3.21}$$

Just as the matrices $\mathbf{B}_i$, the Riesz matrix $\mathbf{R}_s$ is spectrally equivalent to the identity matrix, which we denote by $\mathbf{R}_s \sim \mathbf{I}$, and thus uniformly well-conditioned. This implies the existence of constants c_s and C_s such that

$$c_s \|\mathbf{v}\|^2 \le \mathbf{v}^T \mathbf{R}_s \mathbf{v} \le C_s \|\mathbf{v}\|^2\,. \tag{5.3.22}$$

Corollary 5.7. *For an expansion in the natural dual basis of H^{-s}, $s \in \mathbb{R}_{\ge 0}$, the dual norm can be computed by inverting $\mathbf{R}_s$, i.e.,*

$$H^{-s} \ni v = \mathbf{v}^T \tilde{\Psi}^s\,, \qquad \|v\|_{-s}^2 = \mathbf{v}^T \mathbf{R}_s^{-1} \mathbf{v}\,. \tag{5.3.23}$$

This expression can also be evaluated in linear time, as the condition number of $\mathbf{R}_s$ is uniformly bounded.

It is important not to confuse the diagonal matrix used in the norm equivalence in unscaled form,

$$H^s \ni v = \mathbf{v}^T \Psi\,, \qquad \|v\|_s \sim \|\mathbf{D}^s \mathbf{v}\|\,, \tag{5.3.24}$$

with the diagonal matrix from (2.2.48) which may be exchanged to improve the condition numbers of operators in wavelet representation. The norm equivalence is part of the model and cannot be altered without changing the results for fractional smoothness, while preconditioning must never change the results of any given computation. Let us assume that we use the matrix $\mathbf{D}_a$ from (4.3.11) in (2.2.48). Carefully employing the appropriate semantics, we obtain the more general formulation

$$\mathbf{R}_{a,s} = \sum_{i=0}^{S} h_i(s)\mathbf{P}_a^{s-i}\mathbf{B}_i\mathbf{P}_a^{s-i} \qquad \text{with} \qquad \mathbf{P}_a := \mathbf{D}\mathbf{D}_a^{-1}\,. \tag{5.3.25}$$

When this representation is transformed into the unscaled basis Ψ, all occurrences of $\mathbf{D}_a$ vanish, which confirms the independence of the result from the choice of preconditioning. Furthermore, it is identical to the special case from (5.3.21) for $s \in \mathbb{Z}$.

For finite wavelet discretisations, all components of the proposed Riesz matrix are symmetric, positive definite and uniformly well-conditioned by inheritance from the infinite-dimensional formulation as reasoned in Section 4.3.2. Thus, the Riesz matrix as a whole can be applied and inverted with a computational effort which is proportional to the number of unknowns.

We have computed the condition numbers of several Riesz matrices for a uniform discretisation with biorthogonal spline wavelets for the range $0 \leq s \leq 1$. They are listed in Table 5.1. At the extreme points $s = 0$ and $s = 1$, the condition numbers coincide with those of the mass and stiffness matrices. Again, the transformations to the nodal basis constitutes a significant improvement for free boundary conditions, especially in more than one dimension.

5.3.4 Normalisation by Means of Constant Functions

The construction of the discrete Riesz operator for general Sobolev norms which we have developed in the previous section yields a discrete norm which is equivalent to the continuous Sobolev norm. For integral smoothness, it is exact by construction. For fractional smoothness, we do not have estimates of the constants in the discrete norm equivalence. We could in principle multiply the Riesz operator with any positive real number q_s, as long as $q_i = 1$ for $i \in \mathbb{Z}$, and still satisfy the norm equivalence.

In this section, we choose a normalisation for our Riesz operator in such a way that the discrete norm is exact for constant functions, for any smoothness $s \in \mathbb{R}$. This is possible because all Sobolev norms for constant functions are equal to the L_2 norm. The normalisation factor q_s emerges from the comparison of the Riesz matrix in present form with the exact Riesz matrix for constant functions.

As all derivatives vanish for constant functions, the integral Riesz matrices in natural representation simplify to

$$\bar{\mathbf{B}}_i := \mathbf{D}^{-i}\mathbf{M}\mathbf{D}^{-i}\,. \tag{5.3.26}$$

We also know that the wavelet coefficients for constant functions are zero. Thus, the diagonal matrix reduces to the scaling for the lowest level only,

$$\bar{\mathbf{D}} := 2^{j_0}\mathbf{I}\,. \tag{5.3.27}$$

Applying these two changes to the general expression (5.3.21), our Riesz matrix for constant functions becomes

$$\bar{\mathbf{R}}_s^{(1)} := \left(\sum_{i=0}^{S} h_i(s) 2^{-2j_0 i}\right)\mathbf{M}\,. \tag{5.3.28}$$

We can compare this expression to the exact norm for constant functions, expanded in the natural wavelet basis for H^s,

$$\Pi_1 \ni v = \mathbf{v}^T\Psi^s\,, \qquad \|v\|_{H^s}^2 = \|v\|_{L_2}^2 = \mathbf{v}^T\mathbf{D}^{-s}\mathbf{M}\mathbf{D}^{-s}\mathbf{v}\,, \tag{5.3.29}$$

s	0.0	0.2	0.4	0.6	0.8	0.9	0.95	1.0
$\mathbf{R}_s$	150	62.4	69.3	75.9	81.2	83.4	112	249
$\mathbf{R}_{a,s}$	150	57.1	59.1	61.1	64.2	66.1	67.0	130
$\mathbf{R}^{\mathbf{K}}_{a,s}$	288	75.1	62.9	60.5	63.9	68.5	71.5	151

homogeneous boundary conditions, $n = 1$, $j = 12$

s	0.0	0.2	0.4	0.6	0.8	0.9	0.95	1.0
$\mathbf{R}_s$	35.9	16.0	19.1	22.5	35.9	70.9	129	557
$\mathbf{R}_{a,s}$	35.9	15.7	15.9	16.3	19.3	37.0	66.1	278
$\mathbf{R}^{\mathbf{CK}}_{a,s}$	7.17	4.62	6.23	10.5	22.5	43.9	79.2	336

free boundary conditions, $n = 1$, $j = 12$

s	0.0	0.2	0.4	0.6	0.8	0.9	0.95	1.0
$\mathbf{R}_s$	958	265	238	237	238	239	240	798
$\mathbf{R}_{a,s}$	958	240	198	185	180	179	179	681
$\mathbf{R}^{\mathbf{CK}}_{a,s}$	47.4	16.2	12.3	13.4	30.6	61	112	480

free boundary conditions, $n = 2$, $j = 8$

s	0.0	0.2	0.4	0.6	0.8	0.9	0.95	1.0
$\mathbf{R}_s$	24300	5770	4780	4505	4390	4356	4342	4330
$\mathbf{R}_{a,s}$	24300	4555	3270	2793	2552	2488	2469	2459
$\mathbf{R}^{\mathbf{CK}}_{a,s}$	304	91.2	60.4	44.4	43.2	79.5	139	572

free boundary conditions, $n = 3$, $j = 6$

s	0.0	0.2	0.4	0.6	0.8	0.9	0.95	1.0
$\mathbf{R}^{\mathbf{CK}}_{a,s}$	1839	509	310	211	154	135	225	917

free boundary conditions, $n = 4$, $j = 5$

Table 5.1: We show the condition numbers of Riesz matrices according to (5.3.21) *and* (5.3.25) *for* $n = 1$ *and homogeneous boundary conditions, and for all dimensions from* $n = 1$ *to* 4 *for free boundary conditions. The level of resolution is denoted by* j. *The smoothness index* s *of the Riesz matrix is listed in the first row of each table. We provide results for three variants of wavelet bases, namely the classical choice from* (2.2.48)*, the diagonally adapted case according to* (4.3.11)*, and additionally for the nodal generator basis with transformed wavelets denoted by the superscripts* $()^{\mathbf{C}}$ *for the transformation* (3.3.74) *and* $()^{\mathbf{K}}$ *for* (3.3.75)*.*

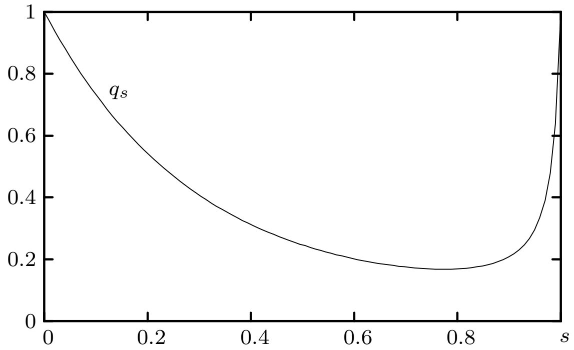


Figure 5.1: We display a graph of the normalisation factor q_s over the range $s \in [0,1]$. As required by the theory, it attains the value $q_i = 1$ at the endpoints of the interval. Its values are repeated periodically on the whole of $\mathbb{R}$.

which corresponds to the exact Riesz matrix

$$\bar{\mathbf{R}}_s^{(2)} := 2^{-2j_0 s}\mathbf{M}\,. \tag{5.3.30}$$

After these preparations, we can formulate the central result of this section.

Theorem 5.8. *Let the normalisation constant q_s, $s \geq 0$, be defined by*

$$q_s := \frac{2^{-2j_0 s}}{\sum_{i=0}^{S} h_i(s) 2^{-2j_0 i}}\,. \tag{5.3.31}$$

Then the Riesz matrix $q_s\mathbf{R}_s$ yields the exact norm for constant functions for any $s \in \mathbb{R}$. Furthermore, it inherits from $\mathbf{R}_s$ the fundamental properties of linear complexity, exactness for $s \in \mathbb{Z}$ and equivalence for $s \in \mathbb{R}$.

Proof. The numerator of (5.3.31) is given by the factor in (5.3.30), while the denominator comes from (5.3.28). Therefore, the multiplication with q_s exactly compensates for the error which we introduced by the use of $\bar{\mathbf{R}}^{(1)}$ instead of $\bar{\mathbf{R}}^{(2)}$. The inheritance of linear complexity and equivalence is trivial, while the exactness for integral smoothness follows from the fact that $q_i = 1$ for all $i \in \mathbb{Z}$. ☐

Remark 5.9. *Note that the condition numbers of $\mathbf{R}_s$ do not change by this scalar multiplication, thus Table 5.1 remains accurate. From this point on, we always silently include the normalisation factor q_s when we write $\mathbf{R}_s$.*

We show a graph of the normalisation factor q_s in Figure 5.1. Since it is periodic in s with period 1, it is sufficient to draw it on the unit interval, where it is a convex function which attains its minimum of about 0.17 at roughly $s = 0.78$.

To summarise our results, we have constructed a family of Riesz matrices which can be used to evaluate primal and dual norms for arbitrary smoothness indices with linear complexity. In case of integral smoothness indices, these norms coincide exactly with standard Sobolev norms, while they are equivalent for fractional smoothness s. In the special case of constant functions, the norms are also exact for all $s \in \mathbb{R}$. This result is useful for the accurate treatment of problems in optimal control with wavelet methods. Only when norms are evaluated exactly, the functional in wavelet coordinates is equal to the original functional.

5.4 Wavelet Representation

The abstract control problem formulated in Problem 5.1 is now transformed into wavelet coordinates. We emphasise that the result is an equivalent, infinite-dimensional formulation. In contrast to finite element discretisations, we do not restrict the space Y to a finite element space Y_h, but keep all degrees of freedom in this step. The general concept is derived according to [28]. We begin with the functional analytic setting and end up with expressions containing wavelet coordinates in ℓ_2. The key mechanism is the discrete formulation of norms which was proposed in the previous section.

As we are dealing with functions in different Sobolev spaces, care must be taken to expand each function in the proper wavelet basis belonging exactly to this space. This is denoted by subscripts on the wavelet bases, e.g., Ψ_V, $\tilde{\Psi}_W$ for the primal and dual basis of V and W, respectively. The corresponding numbering conventions for the associated sequence spaces are denoted by $\ell_2(I\!I_V)$, $\ell_2(I\!I_W)$, etc., when the assignment does not already become clear from the context.

For the expansion of operators, we use the following representation. Let V and W be Hilbert spaces with wavelet bases Ψ_V, Ψ_W and corresponding dual bases $\tilde{\Psi}_V$, $\tilde{\Psi}_W$. Suppose that $B : V \to W'$ is a linear operator with adjoint $B' : W \to V'$ defined by $\langle v, B'w' \rangle := \langle Bv, w' \rangle$ for all $v \in V$, $w' \in W$. Then $Bv = w \in W'$ can be represented in the equivalent formulation $\mathbf{Bv} = \mathbf{w}$ in terms of the wavelet coefficients $\mathbf{v}$ for v (expanded in Ψ_V) and $\mathbf{w}$ (in terms of $\tilde{\Psi}_W$), where

$$\mathbf{B} := \langle \Psi_W, B\Psi_V \rangle \,. \tag{5.4.1}$$

The infinite matrix $\mathbf{B}$ is referred to as the *standard representation* of B with respect to the underlying wavelet bases Ψ_V and Ψ_W.

The Riesz matrix $\mathbf{R}_V$ introduced in (5.3.1) is thus the standard representation of the Riesz operator $R_V : V \to V'$ from (5.2.12) according to

$$\|v\|_V^2 = \langle v, R_V v \rangle \qquad \text{and} \qquad \mathbf{R_V} = \langle \Psi_V, R_V \Psi_V \rangle = (\Psi_V, \Psi_V)_V \,. \tag{5.4.2}$$

Another example is the discretisation of the elliptic operator $A : Y \to Y'$ derived from (5.2.6). The corresponding matrix in wavelet representation is

$$\mathbf{A} = \langle \Psi_Y, A\Psi_Y \rangle = a(\Psi_Y, \Psi_Y) \,. \tag{5.4.3}$$

Embeddings and Traces

As it has been presumed that E and T are continuous linear operators, cf. (5.2.1) and (5.2.4), we can derive that their discrete counterparts in wavelet representation are uniformly bounded,

$$\|\mathbf{T}\| \lesssim 1 \,, \qquad \|\mathbf{E}\| \lesssim 1 \,. \tag{5.4.4}$$

We now demonstrate their expansion in wavelet coordinates by means of two common examples which have already been motivated earlier.

The first case arises in distributed observation by the definitions $Y = H^1(\Omega)$, $Z = H^s(\Omega)$, $0 \leq s \leq 1$, $T = \mathrm{id}$, i.e., T acts as an embedding. For the definition of the wavelet bases, we refer back to (2.2.48), (2.2.49). Following the standard expansion technique (5.4.1), we infer

$$\mathbf{T} = \langle \tilde{\Psi}^s, T\Psi^1 \rangle = \langle \mathbf{D}^s \tilde{\Psi}, T\mathbf{D}^{-1}\Psi \rangle = \mathbf{D}^s \langle \tilde{\Psi}, \mathrm{id}\,\Psi \rangle \mathbf{D}^{-1} = \mathbf{D}^{s-1} \,. \tag{5.4.5}$$

Analogous results are obtained for E in the case of distributed control.

The second case appears when T is a trace operator. Consider for example the quadratic domain $\Omega = (0,1)^2$, where the lower edge Γ is defined by $y = 0$. The matrix representation $\mathbf{T}_0$ is rectangular, since it is indexed by univariate functions on the left and bivariate functions on the right. For tensor products of biorthogonal wavelet bases Ψ, $\tilde{\Psi}$, which are used to construct the wavelet bases for Ω and Γ, the discretised trace operator is given by

$$(\mathbf{T}_0)_{i,kl} := \langle \tilde{\Psi}_\Gamma, T\Psi_\Omega \rangle_{i,kl} = \int_0^1 \tilde{\psi}_i(x) \Big[\psi_k(x)\psi_l(y) \Big]_{y=0} \mathrm{d}x = \delta_{ik}\psi_l(0)\,. \tag{5.4.6}$$

The representation over the natural spaces $T : H^1(\Omega) \to H^{\frac{1}{2}}(\Gamma)$ can be derived by the use of the properly scaled wavelet bases,

$$\mathbf{T} = \langle \tilde{\Psi}_\Gamma^{\frac{1}{2}}, T\Psi_\Omega^1 \rangle = \langle \mathbf{D}_\Gamma^{\frac{1}{2}} \tilde{\Psi}_\Gamma, T\mathbf{D}_\Omega^{-1}\Psi_\Omega \rangle = \mathbf{D}_\Gamma^{\frac{1}{2}} \mathbf{T}_0 \mathbf{D}_\Omega^{-1}\,. \tag{5.4.7}$$

The exponents $\frac{1}{2}$ and 1 provided here may of course be replaced by other values as appropriate. Thus, trace operators in wavelet discretisation on the unit cube are built from diagonal scaling matrices and pointwise evaluations of wavelet basis functions on the boundary.

Remark 5.10. *Embedding and trace operators in wavelet discretisation contain diagonal scaling matrices. These are invertible, but the norm of the inverse is generally unbounded. The diagonals cancel with the matrices from the wavelet basis* (2.2.48) *only in the situation of natural norms.*

System and Riesz Matrices

By the Riesz basis property (2.2.42) and the ellipticity of the bilinear form $a(\cdot,\cdot)$ (5.2.6), we have already established that the stiffness matrix $\mathbf{A}$ has the property

$$\|\mathbf{A}\mathbf{v}\| \sim \|\mathbf{v}\|\,. \tag{5.4.8}$$

The matrix $\mathbf{A}$ is thus uniformly well-conditioned in the wavelet representation, cf. (4.3.10). Choosing biorthogonal wavelet bases Ψ_Y for Y and Ψ_U for U, we obtain the expansions

$$y = \mathbf{y}^T \Psi_Y\,, \qquad \mathbf{y} = \langle \tilde{\Psi}_Y, y \rangle\,, \tag{5.4.9a}$$

$$f = \mathbf{f}^T \tilde{\Psi}_Y\,, \qquad \mathbf{f} = \langle \Psi_Y, f \rangle\,, \tag{5.4.9b}$$

$$u = \mathbf{u}^T \Psi_U\,, \qquad \mathbf{u} = \langle \tilde{\Psi}_U, u \rangle\,, \tag{5.4.9c}$$

and rewrite the constraint equation (5.2.9) as an equation over $\ell_2(I\!I_Y)$,

$$\mathbf{A}\mathbf{y} = \mathbf{f} + \mathbf{E}\mathbf{u}\,. \tag{5.4.10}$$

Here we have again used the standard representation for the operators A and E. Note that U is a space of non-positive Sobolev regularity, so Ψ_U has regularity not above L_2, while $\tilde{\Psi}_U$ is a basis for a possibly smoother space than L_2. We use real, non-negative indices s and t to specify the observation and control space,

$$Z = H^s(\Omega) \quad \text{or} \quad Z = H^s(\Gamma_{\mathrm{O}})\,, \qquad 0 \le s \le 1\,, \tag{5.4.11a}$$

$$U = (H^t(\Omega))' \quad \text{or} \quad U = (H^t(\Gamma_{\mathrm{C}}))'\,, \qquad 0 \le t \le 1\,, \tag{5.4.11b}$$

depending on the choice of measuring or controlling on the whole domain or on a part of the boundary, respectively. The target observation $y_* = \mathbf{y}_*^T \Psi_Z$ is discretised in the same manner as the other variables in (5.4.9). As we are dealing with non-positive norms for the space U, we take the inverse of the Riesz operator. This is motivated by (5.3.10) and concretised in (5.3.23). The result is summarised as

$$\|v\|_Z^2 \mapsto \mathbf{v}^T \mathbf{R}_s \mathbf{v}\,, \qquad \|w\|_U^2 \mapsto \mathbf{w}^T \mathbf{R}_t^{-1} \mathbf{w}\,, \tag{5.4.12}$$

where $\mathbf{v}$ and $\mathbf{w}$ are the coefficient vectors of the functions v and w in their respective natural wavelet basis. We know from Section 5.3 that this substitution is exact for $s, t \in \{0, 1\}$. For fractional indices, we obtain equivalent norms.

The Objective Functional in Wavelet Coordinates

With these tools at hand, we can now reformulate the control functional (5.2.10) as a functional again over ℓ_2,

$$\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) := \frac{1}{2}\|\mathbf{R}_s^{1/2}(\mathbf{T}\mathbf{y}-\mathbf{y}_*)\|^2 + \frac{\omega}{2}\|\mathbf{R}_t^{-1/2}\mathbf{u}\|^2 \,. \tag{5.4.13}$$

The discretised control problem is then formulated as follows.

Problem 5.11 (DCP). *For a given target vector* $\mathbf{y}_* \in \ell_2(I\!I_Z)$*, right hand side* $\mathbf{f} \in \ell_2(I\!I_Y)$ *and weight parameter* $\omega > 0$*, minimise the functional* (5.4.13) *over* $(\mathbf{y},\mathbf{u}) \in \ell_2(I\!I_Y) \times \ell_2(I\!I_U)$ *subject to the linear equation* (5.4.10)*.*

Remark 5.12. *The connection between Problem 5.1 and Problem 5.11 is established in the following sense, where we have to differentiate two cases.*

- *(i) Observation and control space are of integral smoothness. In this case, the functionals* (5.2.10) *and* (5.4.13) *are equal, and the minimiser of the discrete functional converges to the minimiser of the original functional with increasing resolution.*
- *(ii) Either norm in the functional is fractional. Then the functionals* (5.2.10) *and* (5.4.13) *are equivalent,*

  $$\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) \sim J(y,u) \,. \tag{5.4.14}$$

  This implies that in the case of compatible data, $y_* = TA^{-1}f$*, the minimisers for both versions of the functional coincide, yielding the result* $\check{\mathbf{J}}(\mathbf{y},\mathbf{u}) = J(y,u) = 0$*. Otherwise, the discrepancy between the minimisers of the two problems depends on the constants in the norm equivalences. We believe that our scheme for the evaluation of norms yields a tighter interval of these constants than the standard wavelet approach in [48].*

5.5 Optimality Conditions

In this section we reformulate the functional in wavelet coordinates (5.4.13), depending on the discrete pair of variables $\mathbf{y}$ and $\mathbf{u}$, by the elimination of the variable $\mathbf{y}$, which yields a functional in $\mathbf{u}$ alone [48]. Then we show existence and uniqueness of the minimiser of the reduced functional. The necessary and sufficient conditions result in a linear system of equations involving a symmetric positive definite matrix $\mathbf{Q}$ of uniformly bounded condition number.

By introducing a Lagrangian multiplier for the problem in wavelet coordinates, we derive a system of equations, customarily referred to as optimality system, which is equivalent to the formulation in only one variable. We use it to find a recipe to apply the matrix $\mathbf{Q}$ to a vector in the course of an iterative solver. We also establish equivalence to another two reformulations, namely by an elimination of $\mathbf{u}$ instead of $\mathbf{y}$, and the interpretation as a saddle point system.

5.5.1 Existence and Uniqueness

Once (5.4.13) is obtained, we eliminate $\mathbf{y}$ using (5.4.10) and insert this into the functional, arriving at a functional only dependent on $\mathbf{u}$,

$$\mathbf{J}(\mathbf{u}) := \frac{1}{2}\|\mathbf{R}_s^{1/2}(\mathbf{T}\mathbf{A}^{-1}\mathbf{E}\mathbf{u} - (\mathbf{y}_* - \mathbf{T}\mathbf{A}^{-1}\mathbf{f}))\|^2 + \frac{\omega}{2}\|\mathbf{R}_t^{-1/2}\mathbf{u}\|^2 \,. \tag{5.5.1}$$

Employing the abbreviations

$$\mathbf{Z} := \mathbf{R}_s^{1/2}\mathbf{T}\mathbf{A}^{-1}\mathbf{E}\,, \tag{5.5.2a}$$

$$\mathbf{G} := -\mathbf{R}_s^{1/2}(\mathbf{T}\mathbf{A}^{-1}\mathbf{f} - \mathbf{y}_*)\,, \tag{5.5.2b}$$

the functional simplifies to

$$\mathbf{J}(\mathbf{u}) = \frac{1}{2}\|\mathbf{Z}\mathbf{u} - \mathbf{G}\|^2 + \frac{\omega}{2}\|\mathbf{R}_t^{-1/2}\mathbf{u}\|^2\,, \tag{5.5.3}$$

so that the following result can be immediately established, see [48] for the original version without Riesz operators.

Theorem 5.13. *The functional* $\mathbf{J}(\mathbf{u})$ *is twice differentiable with first and second variation given by*

$$\delta\mathbf{J}(\mathbf{u}) = (\mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}_t^{-1})\mathbf{u} - \mathbf{Z}^T\mathbf{G}\,, \qquad \delta^2\mathbf{J}(\mathbf{u}) = \mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}_t^{-1}\,. \tag{5.5.4}$$

In particular, $\mathbf{J}(\mathbf{u})$ *is convex, which guarantees existence and uniqueness of the minimiser.*

Proof. First, let us define

$$\mathbf{Q} := \mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}_t^{-1}\,, \qquad \mathbf{g} := \mathbf{Z}^T\mathbf{G}\,. \tag{5.5.5}$$

The matrix $\mathbf{Q}$ is often called the *reduced system matrix*. The minimisation condition can then be written as

$$\delta\mathbf{J}(\mathbf{u}) = 0 \qquad \Longleftrightarrow \qquad \mathbf{Q}\mathbf{u} = \mathbf{g}\,. \tag{5.5.6}$$

We conclude from (5.3.22), (5.4.4) and (5.4.8) that

$$\mathbf{Z} \lesssim \mathbf{I} \qquad \text{and thus} \qquad \mathbf{Q} \sim \mathbf{I}\,, \tag{5.5.7}$$

which in particular implies that the second variation $\delta^2\mathbf{J}(\mathbf{u}) = \mathbf{Q}$ is positive definite and has a uniformly bounded condition number. ☐

Remark 5.14. *The representation of* $\mathbf{Q}$ *in wavelet variables* (5.5.5) *reveals the role of the regularisation parameter* ω. *For* $\omega > 0$, *the solution* $\mathbf{u}$ *exists and is unique even in the degenerate cases that* E *or* T *are not injective or even zero. For vanishing regularisation* $\omega = 0$, *the problem is well-posed if and only if* $\mathbf{Z}^T\mathbf{Z}$ *is uniformly well-conditioned. In view of* (5.5.2a) *and Remark 5.10, this can for example be guaranteed for natural norms in conjunction with distributed control and observation.*

Remark 5.15. *The results which we have just derived suggest to choose the method of conjugate gradients to solve* (5.5.6) *iteratively for the control* $\mathbf{Q}$. *The central element of this method is the application of the matrix* $\mathbf{Q}$ *to a vector. Equations* (5.5.5) *and* (5.5.2) *show that* $\mathbf{Q}$ *contains two occurrences of* $\mathbf{A}^{-1}$ *and one occurrence of* $\mathbf{R}_t^{-1}$. *Consequently,* $\mathbf{Q}$ *cannot be applied directly, but each application involves the solution of three elliptic systems, which are again well-conditioned.*

To substantiate the above idea of choosing the method of conjugate gradients as our numerical solver, we derive some additional relations in the following section which shed more light on the interrelations between the overall solution for $\mathbf{u}$ and the intermediate inversions of $\mathbf{A}$ and $\mathbf{R}_t$.

5.5.2 Formulation of the Optimality System

In this section, we characterise the minimisation of the control functional $\check{\mathbf{J}}(\mathbf{y}, \mathbf{u})$ (5.4.13) by a Lagrangian multiplier formulation [28, 48, 103, 104] to derive the optimality system. This can be used to develop an efficient strategy to apply the reduced matrix $\mathbf{Q}$. It also gives rise to two equivalent reformulations of the control problem.

Analogously to (5.2.13), where we have introduced the adjoint variable $p \in Y$ as a Lagrangian multiplier, we employ here its wavelet discretisation $\mathbf{p} \in \ell_2$. This leads to the functional

$$\mathbf{L}(\mathbf{y}, \mathbf{u}, \mathbf{p}) \coloneqq \check{\mathbf{J}}(\mathbf{y}, \mathbf{u}) + \mathbf{p}^T(\mathbf{A}\mathbf{y} - \mathbf{f} - \mathbf{E}\mathbf{u})\,. \tag{5.5.8}$$

The first order Euler-Lagrange equations for $\delta \mathbf{L}(\mathbf{y}, \mathbf{u}, \mathbf{p}) = 0$ follow as

$$\begin{aligned} \mathbf{A}\mathbf{y} &= \mathbf{f} + \mathbf{E}\mathbf{u}\,, && \text{(5.5.9a)}\\ \mathbf{A}^T\mathbf{p} &= -\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{y} - \mathbf{y}_*)\,, && \text{(5.5.9b)}\\ \omega\mathbf{R}_t^{-1}\mathbf{u} &= \mathbf{E}^T\mathbf{p}\,. && \text{(5.5.9c)} \end{aligned}$$

This system of equations is called *optimality system.* The three components, called the state, adjoint and design equations, are the wavelet representations of the variational formulations from (5.2.14). Note that for consistency with more general problems, we keep the notation $\mathbf{A}^T$ although in our situation $\mathbf{A}$ is symmetric.

Remark 5.16. *The order of* differentiation *and* discretisation *is much disputed in the finite element context, since the approaches differentiate-then-discretise and discretise-then-differentiate lead to different results for finite representations of functions. In the wavelet context however, where we deal with infinite-dimensional representations, this order is irrelevant. Expanding the system* (5.2.14) *in wavelet coordinates, which would correspond to differentiate-then-discretise, leads precisely to the equations from* (5.5.9)*, which have been obtained via the approach discretise-then-differentiate.*

Application of the Reduced Matrix

We now use the optimality system (5.5.9) to obtain some additional identities. These will prove helpful in the numerical realisation of the application of $\mathbf{Q}$, which is the key ingredient for the numerical procedure proposed in Remark 5.15.

Theorem 5.17. *Let* $\mathbf{u} \in \ell_2(\mathbb{I}_U)$ *be any given control vector. Inserting* $\mathbf{u}$ *into* (5.5.9a) *and solving for* $\mathbf{y}(\mathbf{u})$*, and subsequently inserting this solution* $\mathbf{y}(\mathbf{u})$ *into* (5.5.9b) *and solving for* $\mathbf{p}(\mathbf{y}) = \mathbf{p}(\mathbf{u})$*, the residual of* (5.5.6) *is*

$$\mathbf{Q}\mathbf{u} - \mathbf{g} = \omega\mathbf{R}_t^{-1}\mathbf{u} - \mathbf{E}^T\mathbf{p}(\mathbf{u})\,. \tag{5.5.10}$$

Proof. For the proof, we proceed in the opposite direction, resolving first the second equation for $\mathbf{p}(\mathbf{y})$, and then the first equation for $\mathbf{y}(\mathbf{u})$. Using the definition of $\mathbf{Z}$ and $\mathbf{g}$ from (5.5.2) and (5.5.5), we obtain

$$\begin{aligned} \mathbf{E}^T\mathbf{p}(\mathbf{u}) &= -\mathbf{E}^T\left(\mathbf{A}^{-T}\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{y}(\mathbf{u}) - \mathbf{y}_*)\right)\\ &= -\mathbf{Z}^T\mathbf{R}_s^{1/2}\left(\mathbf{T}(\mathbf{A}^{-1}\mathbf{f} + \mathbf{A}^{-1}\mathbf{E}\mathbf{u}) - \mathbf{y}_*\right)\\ &= -\mathbf{Z}^T\mathbf{Z}\mathbf{u} + \mathbf{g}\,. \end{aligned} \tag{5.5.11}$$

Using the definition of $\mathbf{Q}$ (5.5.5), the residual attains the form

$$\mathbf{Q}\mathbf{u} - \mathbf{g} = (\mathbf{Z}^T\mathbf{Z} + \omega\mathbf{R}_t^{-1})\mathbf{u} - \mathbf{g} = \omega\mathbf{R}_t^{-1}\mathbf{u} - \mathbf{E}^T\mathbf{p}(\mathbf{u})\,, \tag{5.5.12}$$

which is just the defect in the design equation (5.5.9c). □

In the proof we have inverted the relations (5.5.9a) and (5.5.9b) to arrive at an expression of the residual of (5.5.6). Concentrating on the calculation of $\mathbf{Q}\mathbf{u}$ alone, we could shift the vector $\mathbf{g}$ to the right hand side of (5.5.10). However, the result can be obtained slightly faster by the direct use of (5.5.5) and (5.5.2a).

Corollary 5.18. *Let* $\mathbf{u} \in \ell_2(I\!I_U)$ *be any given control vector, and substitute* $\mathbf{f} = 0$ *and* $\mathbf{y}_* = 0$ *into* (5.5.9a) *and* (5.5.9b) *to obtain their homogeneous forms*

$$\mathbf{A}\mathbf{y}_0(\mathbf{u}) = \mathbf{E}\mathbf{u}\,, \tag{5.5.13a}$$

$$\mathbf{A}^T\mathbf{p}_0(\mathbf{u}) = -\mathbf{T}^T\mathbf{R}_s\mathbf{T}\mathbf{y}_0(\mathbf{u})\,, \tag{5.5.13b}$$

hence defining $\mathbf{y}_0(\mathbf{u})$ *and* $\mathbf{p}_0(\mathbf{u})$*. Then it holds that*

$$\mathbf{Q}\mathbf{u} = \omega\mathbf{R}_t^{-1}\mathbf{u} - \mathbf{E}^T\mathbf{p}_0(\mathbf{u})\,. \tag{5.5.14}$$

Using this route, we save three additions and subtractions of vectors, and eliminate the dependence on the data. Equations (5.5.13) and (5.5.14) thus contain the final recipe for the computation of $\mathbf{Qu}$.

Other Equivalent Formulations

To illustrate the connections between the variables $\mathbf{y}$, $\mathbf{p}$ and $\mathbf{u}$ in more detail, we present two different but equivalent formulations of the optimality system. These give rise to other classes of numerical algorithms, which are applicable for selected special cases.

The first derivation consists in the elimination of the control. While in Section 5.5.1 we have eliminated $\mathbf{y}$ from the functional, we can also use (5.5.9c) to eliminate $\mathbf{u}$ from the optimality system. Then only two equations remain, namely

$$\mathbf{A}\mathbf{y} = \mathbf{f} + \omega^{-1}\mathbf{E}\mathbf{R}_t\mathbf{E}^T\mathbf{p}\,, \tag{5.5.15a}$$

$$\mathbf{A}^T\mathbf{p} = -\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{y} - \mathbf{y}_*)\,. \tag{5.5.15b}$$

We can rewrite these in system form according to

$$\begin{pmatrix} \mathbf{T}^T\mathbf{R}_s\mathbf{T} & \mathbf{A}^T \\ \mathbf{A} & -\omega^{-1}\mathbf{E}\mathbf{R}_t\mathbf{E}^T \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{T}^T\mathbf{R}_s\mathbf{y}_* \\ \mathbf{f} \end{pmatrix}\,. \tag{5.5.16}$$

This so called saddle point formulation [25] shows that the problem is essentially symmetric with respect to the state y and the adjoint p. However, it is no longer positive definite. Moreover, when the norms used in the objective functional are not natural, the matrices T and E introduce a diagonal scaling which lets the lowest eigenvalues of the contributions on the block diagonal go to zero.

Alternatively, we can formulate the optimality system (5.5.9) as one large block-matrix equation,

$$\begin{pmatrix} \omega\mathbf{R}_t^{-1} & 0 & -\mathbf{E}^T \\ 0 & \mathbf{T}^T\mathbf{R}_s\mathbf{T} & \mathbf{A}^T \\ -\mathbf{E} & \mathbf{A} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{y} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{T}^T\mathbf{R}_s\mathbf{y}_* \\ \mathbf{f} \end{pmatrix}\,. \tag{5.5.17}$$

The symmetric block system matrix again has the structure of a saddle point system. Defining

$$\mathbf{U} := \begin{pmatrix} \omega\mathbf{R}_t^{-1} & 0 \\ 0 & \mathbf{T}^T\mathbf{R}_s\mathbf{T} \end{pmatrix}\,, \qquad \mathbf{F} := (-\mathbf{E}, \mathbf{A})\,, \tag{5.5.18}$$

we rewrite this system as

$$\begin{pmatrix} \mathbf{U} & \mathbf{F}^T \\ \mathbf{F} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{t} \\ \mathbf{f} \end{pmatrix} \qquad \text{with} \qquad \mathbf{x} := \begin{pmatrix} \mathbf{u} \\ \mathbf{y} \end{pmatrix}\,, \qquad \mathbf{t} := \begin{pmatrix} 0 \\ \mathbf{T}^T\mathbf{R}_s\mathbf{y}_* \end{pmatrix}\,. \tag{5.5.19}$$

When $\mathbf{T}$ has full rank (which is only possible for distributed observation), $\mathbf{U}$ is invertible, and we may form the Schur complement $\mathbf{H}$,

$$\mathbf{H} := \mathbf{F}\mathbf{U}^{-1}\mathbf{F}^T = \omega^{-1}\mathbf{E}\mathbf{R}_t\mathbf{E}^T + \mathbf{A}(\mathbf{T}^T\mathbf{R}_s\mathbf{T})^{-1}\mathbf{A}^T\,. \tag{5.5.20}$$

Because of the uniformly bounded condition numbers of $\mathbf{R}_s$, $\mathbf{R}_t$ and $\mathbf{A}$, this is spectrally equivalent to

$$\mathbf{H} \sim \omega^{-1}\mathbf{E}^T\mathbf{E} + (\mathbf{T}^T\mathbf{T})^{-1} . \tag{5.5.21}$$

While the right part is bounded from below because of (5.4.4), it is only bounded from above for natural norms on the observation space, that is $s = 1$, and unbounded for $s < 1$.

In both of these equivalent formulations of the optimality system, the system matrix is not positive definite, or it does in general not have a uniformly bounded condition number. Bounded condition numbers can only guaranteed for the second example in the special case of natural norms. In contrast, the matrix $\mathbf{Q}$ from (5.5.5) is symmetric positive definite and of uniformly bounded condition number for all combinations of parameters. In this sense, an iterative method based on the application of $\mathbf{Q}$ is general and robust.

Chapter 6

A Fast Wavelet Algorithm for the Control Problem

6.1 Introduction

The purpose of this thesis is to develop and implement a fast wavelet solver for a linear-quadratic elliptic optimal control problem. In earlier chapters, we have addressed the practical importance of this class of problems and motivated a discretisation with wavelets based on two specific qualities. Firstly, the wavelet discretisation leads to well-conditioned systems of equations, which permits the fast iterative solution of the problem and allows for several millions of unknowns on a standard PC. Secondly, wavelets offer a unified framework for the numerical evaluation of Sobolev norms in the objective functional, which yields greater freedom in modelling.

After we have proved in the preceding chapters that the matrices $\mathbf{A}$, $\mathbf{R}_s$ and $\mathbf{R}_t$ occuring in the optimality system, and the reduced matrix $\mathbf{Q}$ are uniformly well-conditioned, we provide here a detailed specification of an algorithm for uniform discretisations, and a collection of systematic numerical results from one to three spatial dimensions. We confirm that all involved constants are effectively controlled, and that the algorithm is of optimal complexity in the sense that the numerical solution is computed with an effort proportional to the number of unknowns.

In addition, we examine various combinations of modelling parameters, namely the regularisation parameter ω, and the smoothness indices s and t for the observation and control spaces, see (5.4.13). We study their effects on the shape and character of the control u and the state y, and also their interplay with each other, which gives rise to a diversity of results.

As discussed in detail in the previous chapter, the necessary and sufficient conditions for an optimal discrete control $\mathbf{u}$ are formulated as the following linear system of equations,

$$\mathbf{Q}\mathbf{u} = \mathbf{g}\,, \tag{6.1.1}$$

where $\mathbf{Q}$ is symmetric positive definite. As pointed out before, the wavelet discretisation ensures that it is also well-conditioned. Therefore we use an outer loop of conjugate gradient iterations on $\mathbf{u}$, for which the central operation consists in the application of the system matrix $\mathbf{Q}$. Considering (5.5.2) and (5.5.5), one such application of the matrix $\mathbf{Q}$ involves the solution of three elliptic systems of equations over the matrices $\mathbf{A}$, $\mathbf{A}^T$ and $\mathbf{R}_t$, respectively, which are again well-conditioned in the wavelet setting. This is also accomplished by CG iterations, which constitute the inner layer of the algorithm. We have schematically lined out this structure in Algorithm 6.1.

```
Subroutine Apply-Q ($\mathbf{u}$): Computes $\mathbf{Qu}$.
(1) Perform CG method for $\mathbf{u} \to \mathbf{y}_0$ (5.5.13a).
(2) Perform CG method for $\mathbf{y}_0 \to \mathbf{p}_0$ (5.5.13b).
(3) Perform CG method for $\mathbf{u} \to \mathbf{R}_t^{-1}\mathbf{u}$.
(4) Compute $\mathbf{Qu} := \omega \mathbf{R}_t^{-1}\mathbf{u} - \mathbf{E}^T\mathbf{p}_0$ (5.5.14).
```

```
Algorithm Control-Generic ($\mathbf{f}, \mathbf{y}_*$) $\to \mathbf{u}$: Solves $\mathbf{Qu} = \mathbf{g}$.
(i) Initialisation: Compute $\mathbf{g}$ according to (5.5.5).
    (1) Perform CG method for $(\mathbf{f}, \mathbf{y}_*) \to \mathbf{G}$ (5.5.2b).
    (2) Perform CG method for $\mathbf{G} \to \mathbf{g}$ (5.5.2a), (5.5.5).
(ii) Outer CG method: Set $k := 0$, $\mathbf{u}_0 := 0$.
    (1) Call Apply-Q ($\mathbf{u}_0$) to set $\mathbf{d}_0 := -\mathbf{q}_0 := \mathbf{g} - \mathbf{Qu}_0$.
    (2) Call Apply-Q ($\mathbf{d}_k$) $\to \mathbf{h}_k$, Compute $\alpha_k := \frac{\mathbf{q}_k^T\mathbf{q}_k}{\mathbf{d}_k^T\mathbf{h}_k}$.
    (4) Compute $\mathbf{u}_{k+1} := \mathbf{u}_k + \alpha_k\mathbf{d}_k$, $\mathbf{q}_{k+1} := \mathbf{q}_k + \alpha_k\mathbf{h}_k$.
    (5) Compute $\beta_k := \frac{\mathbf{q}_{k+1}^T\mathbf{q}_{k+1}}{\mathbf{q}_k^T\mathbf{q}_k}$, $\mathbf{d}_{k+1} := -\mathbf{q}_{k+1} + \beta_k\mathbf{d}_k$.
    (6) If not ready, set $k := k + 1$, continue with (2).
(iii) Accept $\mathbf{u}_k \to \mathbf{u}$.
```

Algorithm 6.1: We provide an overview of the generic solver `Control-Generic` *for the optimal control problem in wavelet coordinates. It consists of a CG method for the reduced matrix* $\mathbf{Q}$*. Each application of* $\mathbf{Q}$ *is performed in the subroutine* `Apply-Q` *and needs three inner CG methods, two to invert* $\mathbf{A}$ *and* $\mathbf{A}^T$ *and one to invert* $\mathbf{R}_t$*. We have deliberately kept the infinite-dimensional formulation of matrices and vectors over* ℓ_2*, since it is most general and the structure for arbitrary finite subspaces is analogous.*

Recall that this numerical scheme emerges from the combination and synergy of several ingredients which have been adapted, improved or even newly developed in view of the overall goal, as discussed in the preceding chapters.

- The abstract optimal control problem under consideration has been defined in Problem 5.1 and then reformulated equivalently in terms of wavelets. Special emphasis has been put on the accurate evaluation of Sobolev norms occurring in the objective functional.
- An asymptotically optimal numerical wavelet solver for linear elliptic PDEs has been established in Chapter 4, using a nested iteration strategy and a technique to optimise the condition numbers.
- We have introduced the theoretical foundations in Chapter 2 and example constructions in Chapter 3 of suitable biorthogonal wavelet bases, including additional transformations to reduce the condition numbers.

For the detailed specification of the numerical algorithm, we need to resolve the following three remaining technical issues.

- As each outer iteration of our numerical scheme uses several inner invocations of iterative solvers for different intermediate variables, the interplay of the error bounds for these various steps needs to be analysed. The bounds on the lower and upper eigenvalues of the discretised operators come in here, which have up to this point been treated as symbolic constants. Their absolute values directly influence the choice of appropriate stopping criteria for the different instances of iterative solvers. We deduce the necessary thresholds for all subproblems to arrive at the final estimate of computational accuracy, frequently referencing the derivations from Section 5.5.
- The discrete solution $\mathbf{y}$ for the state should be computed up to discretisation error accuracy 2^{-j} for each level j. To render the convergence rate of the algorithm truly independent of the resolution, we enhance both the outer and the inner iterative solvers with dedicated nested iteration strategies.
- Since all inner iterative solvers introduce an error for their respective solution vector, the application of the system matrix $\mathbf{Q}$ in the outer iteration is perturbed. We need to discuss the effect of this perturbation on the convergence of the conjugate gradient method. Doing this, we can resort to some references on inexact Krylov subspace methods [69, 139, 140]. Generally, we can say already at this point that convergence is not impaired if the error of the inner iterations matches with the target accuracy for the outer system.

Eventually having integrated these additional considerations, we arrive at an asymptotically optimal numerical algorithm with explicitly given error bounds.

6.2 An Inexact Conjugate Gradient Method

Reviewing the schematic structure of Algorithm 6.1, we find that there are several calls to inner conjugate gradient methods. As these are iterative procedures, they in general do not deliver the exact solution of the linear systems they are supposed to solve. This is why the outer iteration must more precisely be called an *inexact* conjugate gradient method. In this section, we introduce estimates for all error bounds and stopping criteria which occur in this algorithm.

The following derivations are mainly taken from [28]. They are slightly rearranged and extended in the sense that they systematically use the general operators $\mathbf{E}$ and $\mathbf{T}$, and add the estimate for the inversion of the Riesz operator $\mathbf{R}_t$.

```
Subroutine Apply $(\mathbf{M}, \mathbf{x}, \eta) \to \mathbf{m}_\eta$: Computes $\mathbf{m}_\eta$ such that $\|\mathbf{M}\mathbf{x} - \mathbf{m}_\eta\| \leq \eta$.
```

```
Subroutine CG $(\mathbf{M}, \mathbf{b}, \epsilon) \to \mathbf{x}_\epsilon$: Computes $\mathbf{x}_\epsilon$ such that $\|\mathbf{M}\mathbf{x}_\epsilon - \mathbf{b}\| \leq \epsilon$.
(I) SET $k := 0$, SET $\mathbf{d}_0 := -\mathbf{q}_0 := \mathbf{b}$.
(II) WHILE $\|\mathbf{q}_k\| > \theta_\epsilon$
    (1) CALL Apply $(\mathbf{M}, \mathbf{d}_k, \eta_k) \to \mathbf{h}_k$, COMPUTE $\alpha_k := \frac{\mathbf{q}_k^T \mathbf{q}_k}{\mathbf{d}_k^T \mathbf{h}_k}$.
    (2) COMPUTE $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{d}_k$, $\mathbf{q}_{k+1} := \mathbf{q}_k + \alpha_k \mathbf{h}_k$.
    (3) COMPUTE $\beta_k := \frac{\mathbf{q}_{k+1}^T \mathbf{q}_{k+1}}{\mathbf{q}_k^T \mathbf{q}_k}$, $\mathbf{d}_{k+1} := -\mathbf{q}_{k+1} + \beta_k \mathbf{d}_k$.
    (4) SET $k := k + 1$.
(III) ACCEPT $\mathbf{x}_k \to \mathbf{x}_\epsilon$.
```

Algorithm 6.2: We display the basic conjugate gradient algorithm `CG`*. It solves the linear system up to residual error ϵ. In the trivial case, when $\mathbf{M}$ can be applied exactly, we set $\theta_\epsilon = \epsilon$ and $\eta_k = 0$ and recover the classical conjugate gradient method. When $\mathbf{M}$ can only be applied approximately, the error bounds θ_ϵ and η_k have to be chosen in dependence on ϵ, resulting in an inexact CG algorithm.*

6.2.1 Basic Algorithm and Error Bounds

Consider the linear system of equations

$$\mathbf{M}\mathbf{x} = \mathbf{b} \tag{6.2.1}$$

with the symmetric positive definite matrix $\mathbf{M}$, which is meant to be understood as a placeholder. Examples are the stiffness matrix $\mathbf{A}$ and its transpose $\mathbf{A}^T$, or the matrix of the reduced formulation $\mathbf{Q}$ (5.5.5). These are defined over the full infinite-dimensional space of wavelet coordinates, and we keep this setting for the derivation of the error bounds below. As shown in Section 4.3.2, the results we obtain are automatically valid for all finite submatrices created by a truncation of the index set in view of the actual computations.

In an iterative procedure, we solve each such system (6.2.1) up to a certain residual error ϵ, i.e., the approximate solution $\mathbf{x}_\epsilon$ eventually satisfies

$$\|\mathbf{M}\mathbf{x}_\epsilon - \mathbf{b}\| \leq \epsilon\,. \tag{6.2.2}$$

The tolerance ϵ should be tied to the accuracy of the discretisation, as will be discussed later. To derive the error in the solution itself, we have to examine the eigenvalues of the matrix $\mathbf{M}$ more closely. To this end, the smallest and largest eigenvalues are denoted by $c_\mathbf{M}$ and $C_\mathbf{M}$, respectively, according to the identity

$$c_\mathbf{M}\|\mathbf{x}\| \leq \|\mathbf{M}\mathbf{x}\| \leq C_\mathbf{M}\|\mathbf{x}\|\,. \tag{6.2.3}$$

The error in the solution can then be estimated by

$$\|\mathbf{x} - \mathbf{x}_\epsilon\| = \|\mathbf{M}^{-1}(\mathbf{M}\mathbf{x}_\epsilon - \mathbf{b})\| \leq \|\mathbf{M}^{-1}\|\,\|\mathbf{M}\mathbf{x}_\epsilon - \mathbf{b}\| \leq \frac{\epsilon}{c_\mathbf{M}}\,. \tag{6.2.4}$$

We show a generic conjugate gradient routine `CG` in Algorithm 6.2, which allows for approximate applications of the matrix $\mathbf{M}$. It contains error bounds η_k that control the size of the residual. We will discuss appropriate strategies to choose these error bounds in Section 6.2.2. When the matrix can be applied exactly, which is the case for the stiffness matrix and Riesz operators in uniform discretisation, the subroutine `Apply` is trivial and the bounds η_k are all set to 0.

```
Subroutine RHS $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, \zeta) \to \mathbf{g}_\zeta$: Computes $\mathbf{g}_\zeta$ such that $\|\mathbf{g} - \mathbf{g}_\zeta\| \leq \zeta$.
(1) Call CG $(\mathbf{A}, \mathbf{f}, \frac{c_\mathbf{A}}{2C_\mathbf{E}} \frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}\zeta) \to \mathbf{g}_1$.
(2) Call CG $(\mathbf{A}^T, -\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{g}_1 - \mathbf{y}_*), \frac{c_\mathbf{A}}{2C_\mathbf{E}}\zeta) \to \mathbf{g}_2$.
(3) Compute $\mathbf{E}^T\mathbf{g}_2 \to \mathbf{g}_\zeta$.
```

Algorithm 6.3: We show the routine `RHS` *for the computation of the right hand side according to* (5.5.2) *and* (5.5.5)*. It contains two calls to the conjugate gradient solver* `CG` *described in Algorithm 6.2, with appropriately chosen bounds on the residual error.*

We use this conjugate gradient method for the solutions of the primal and adjoint systems (5.5.9a) and (5.5.9b), and to invert the Riesz operator in (5.5.10). These inversions constitute the inner layer of the algorithm, and in this situation the respective matrices can be applied exactly.

The outer layer is given by the solution of the reduced equation (5.5.6). Here, we use (5.5.14) to approximately apply the matrix $\mathbf{Q}$ to a vector. Additionally, we need to compute the right hand side $\mathbf{g}$. Since this requires two inversions of $\mathbf{A}$, we do this approximately, too.

The application of $\mathbf{Q}$ and the calculation of $\mathbf{g}$ contain applications of different matrices. To derive reliable stopping criteria for the inner systems of equations which need to be solved along the way, we have to specify the bounds of all involved matrices. The estimates for the stiffness matrix and its transpose are given analogously to (6.2.3). The Riesz operators from (5.5.1) are uniformly bounded, see also (5.3.22), with constants

$$c_s\|\mathbf{x}\| \leq \|\mathbf{R}_s\mathbf{x}\| \leq C_s\|\mathbf{x}\|\,, \tag{6.2.5a}$$

$$c_t\|\mathbf{x}\| \leq \|\mathbf{R}_t\mathbf{x}\| \leq C_t\|\mathbf{x}\|\,. \tag{6.2.5b}$$

The upper bounds on the embedding and trace operators are denoted by

$$\|\mathbf{E}\mathbf{x}\| \leq C_\mathbf{E}\|\mathbf{x}\| \qquad \text{and} \qquad \|\mathbf{T}\mathbf{x}\| \leq C_\mathbf{T}\|\mathbf{x}\|\,. \tag{6.2.6}$$

We estimate the lowest eigenvalue of the reduced system matrix $\mathbf{Q}$ using (5.5.5) and (6.2.5b) as

$$c_\mathbf{Q} \geq \frac{\omega}{C_t}\,, \tag{6.2.7}$$

since $\mathbf{Z}^T\mathbf{Z}$ is generally not bounded from below.

The construction of the right hand side $\mathbf{g}$ is performed in the subroutine `RHS`, which we list in Algorithm 6.3. It outputs a vector $\mathbf{g}_\zeta$, which is accurate up to an error ζ.

Proposition 6.1. *The result* $\mathbf{g}_\zeta$ *of the subroutine* `RHS` $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, \zeta)$ *satisfies upon completion*

$$\|\mathbf{g} - \mathbf{g}_\zeta\| \leq \zeta\,. \tag{6.2.8}$$

Proof. Steps (1) and (2) assert the following relations,

$$\|\mathbf{A}\mathbf{g}_1 - \mathbf{f}\| \leq \frac{c_\mathbf{A}}{2C_\mathbf{E}}\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}\zeta\,, \tag{6.2.9a}$$

$$\|\mathbf{A}^T\mathbf{g}_2 + \mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{g}_1 - \mathbf{y}_*)\| \leq \frac{c_\mathbf{A}}{2C_\mathbf{E}}\zeta\,. \tag{6.2.9b}$$

```
Subroutine Apply $(\mathbf{Q}, \mathbf{u}, \eta) \to \mathbf{m}_\eta$: Computes $\mathbf{m}_\eta$ such that $\|\mathbf{Qu} - \mathbf{m}_\eta\| \leq \eta$.
(1) CALL CG $(\mathbf{A}, \mathbf{Eu}, \frac{c_\mathbf{A}}{3C_\mathbf{E}}\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}\eta) \to \mathbf{y}_0$.
(2) CALL CG $(\mathbf{A}^T, -\mathbf{T}^T\mathbf{R}_s\mathbf{T}\mathbf{y}_0, \frac{c_\mathbf{A}}{3C_\mathbf{E}}\eta) \to \mathbf{p}_0$.
(3) CALL CG $(\mathbf{R}_t, \mathbf{u}, \frac{c_t}{3\omega}\eta) \to \mathbf{q}_0$.
(4) COMPUTE $\omega\mathbf{q}_0 - \mathbf{E}^T\mathbf{p}_0 \to \mathbf{m}_\eta$.
```

Algorithm 6.4: This listing contains the routine `Apply` $(\mathbf{Q})$. *It calculates the approximate application of* $\mathbf{Q}$ *to a vector* $\mathbf{u}$. *The error bounds in the first two steps contain the same factors as in Algorithm 6.3.*

By the respective definitions of $\mathbf{g}$ and $\mathbf{g}_\zeta$, and the successive insertion of (6.2.9b) and (6.2.9a), we derive

$$\begin{aligned} \|\mathbf{g} - \mathbf{g}_\zeta\| &\leq C_\mathbf{E}\left\|\mathbf{A}^{-T}\left(-\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{A}^{-1}\mathbf{f} - \mathbf{y}_*) - \mathbf{A}^T\mathbf{g}_2\right)\right\| \\ &\leq \frac{C_\mathbf{E}}{c_\mathbf{A}}\left(\left\|\mathbf{T}^T\mathbf{R}_s(-\mathbf{T}\mathbf{A}^{-1}\mathbf{f} + \mathbf{T}\mathbf{g}_1)\right\| + \frac{c_\mathbf{A}}{2C_\mathbf{E}}\zeta\right) \\ &\leq \frac{C_\mathbf{E}}{c_\mathbf{A}}C_\mathbf{T}^2 C_s\|\mathbf{A}^{-1}(\mathbf{f} - \mathbf{A}\mathbf{g}_1)\| + \frac{1}{2}\zeta \leq \zeta\,, \end{aligned} \tag{6.2.10}$$

which proves the claim. □

The application of the matrix $\mathbf{Q}$ to a vector $\mathbf{u}$ is performed in the routine `Apply` from Algorithm 6.4. It computes a vector $\mathbf{m}_\eta$ with the following property.

Proposition 6.2. *The result* $\mathbf{m}_\eta$ *of the subroutine* `Apply` $(\mathbf{Q}, \mathbf{u}, \eta)$ *obeys the inequality*

$$\|\mathbf{Qu} - \mathbf{m}_\eta\| \leq \eta\,. \tag{6.2.11}$$

Proof. We first decompose the error into two terms. Using (5.5.14), we obtain

$$\begin{aligned} \|\mathbf{Qu} - \mathbf{m}_\eta\| &= \left\|\left(\omega\mathbf{R}_t^{-1}\mathbf{u} - \mathbf{E}^T\mathbf{p}\right) - \left(\omega\mathbf{q}_0 - \mathbf{E}^T\mathbf{p}_0\right)\right\| \\ &\leq \omega\|\mathbf{R}_t^{-1}\mathbf{u} - \mathbf{q}_0\| + C_\mathbf{E}\|\mathbf{p} - \mathbf{p}_0\|\,. \end{aligned} \tag{6.2.12}$$

The first term can be estimated by

$$\|\mathbf{R}_t^{-1}\mathbf{u} - \mathbf{q}_0\| = \|\mathbf{R}_t^{-1}(\mathbf{u} - \mathbf{R}_t\mathbf{q}_0)\| \leq \frac{1}{c_t}\|\mathbf{R}_t\mathbf{q}_0 - \mathbf{u}\| \leq \frac{\eta}{3\omega}\,, \tag{6.2.13}$$

where we have used the error bound $\frac{c_t}{3\omega}\eta$ from step (3). To estimate the second term, we introduce an intermediate variable $\mathbf{p}'$ as the exact solution of the equation

$$\mathbf{A}^T\mathbf{p}' = -\mathbf{T}^T\mathbf{R}_s\mathbf{T}\mathbf{y}_0\,. \tag{6.2.14}$$

By the triangle inequality, we split the remaining error again,

$$\|\mathbf{p} - \mathbf{p}_0\| \leq \|\mathbf{p} - \mathbf{p}'\| + \|\mathbf{p}' - \mathbf{p}_0\|\,, \tag{6.2.15}$$

and analyse the two contributions separately. To this end, we recall the bounds ensured by the steps (1) and (2) of the subroutine,

$$\|\mathbf{A}\mathbf{y}_0 - \mathbf{Eu}\| \leq \frac{c_\mathbf{A}}{3C_\mathbf{E}}\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}\eta\,, \tag{6.2.16a}$$

$$\|\mathbf{A}^T\mathbf{p}_0 + \mathbf{T}^T\mathbf{R}_s\mathbf{T}\mathbf{y}_0\| \leq \frac{c_\mathbf{A}}{3C_\mathbf{E}}\eta\,. \tag{6.2.16b}$$

```
Algorithm Control $(\mathbf{f}, \mathbf{y}_*, \epsilon) \to (\mathbf{y}_\epsilon, \mathbf{u}_\epsilon)$: Assures $\|\mathbf{y} - \mathbf{y}_\epsilon\| \leq \epsilon$, $\|\mathbf{u} - \mathbf{u}_\epsilon\| \leq \epsilon$.
(1) SET $\zeta := \min\{1, \frac{2}{3}\frac{c_\mathbf{A}}{C_\mathbf{E}}\}\epsilon$.
(2) CALL RHS $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, \frac{c_\mathbf{Q}}{2}\zeta) \to \mathbf{g}_\epsilon$.
(3) CALL CG $(\mathbf{Q}, \mathbf{g}_\epsilon, \frac{c_\mathbf{Q}}{2}\zeta) \to \mathbf{u}_\epsilon$.
(4) CALL CG $(\mathbf{A}, \mathbf{f} + \mathbf{E}\mathbf{u}_\epsilon, \frac{c_\mathbf{A}}{3}\epsilon) \to \mathbf{y}_\epsilon$.
```

Algorithm 6.5: We list the full algorithm which solves the optimal control problem. It computes the state $\mathbf{y}_\epsilon$ *and the control* $\mathbf{y}_\epsilon$ *up to accuracy* ϵ*. It consists of three building blocks, namely the call to the subroutine* `RHS` *to compute the reduced right hand side* $\mathbf{g}_\epsilon$*, and two calls to* `CG`*. The numerical inversion of* $\mathbf{Q}$ *in step (3) is the expensive part here, as it invokes the outer layer of inexact CG iterations using the subroutine* `Apply` $(\mathbf{Q})$ *from Algorithm 6.4.*

The first half of (6.2.15) can be processed as follows,

$$\|\mathbf{p} - \mathbf{p}'\| = \|\mathbf{A}^{-T}(\mathbf{A}^T\mathbf{p} - \mathbf{A}^T\mathbf{p}')\| \leq \frac{C_\mathbf{T}^2 C_s}{c_\mathbf{A}}\|\mathbf{y} - \mathbf{y}_0\|\,, \tag{6.2.17}$$

where we have used the definitions (5.5.13b) and (6.2.14). We can now employ (5.5.13a) and (6.2.16a) to derive

$$\|\mathbf{y} - \mathbf{y}_0\| = \|\mathbf{A}^{-1}(\mathbf{A}\mathbf{y} - \mathbf{A}\mathbf{y}_0)\| \leq \frac{1}{c_\mathbf{A}}\|\mathbf{E}\mathbf{u} - \mathbf{A}\mathbf{y}_0\| \leq \frac{1}{3C_\mathbf{E}}\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}\eta\,. \tag{6.2.18}$$

Combining this result with (6.2.17), we see that most factors cancel, and we end up with

$$\|\mathbf{p} - \mathbf{p}'\| \leq \frac{1}{3C_\mathbf{E}}\eta\,. \tag{6.2.19}$$

To handle the second half of (6.2.15), we can estimate

$$\|\mathbf{p}' - \mathbf{p}_0\| = \|\mathbf{A}^{-T}(\mathbf{A}^T\mathbf{p}' - \mathbf{A}^T\mathbf{p}_0)\| \leq \frac{1}{c_\mathbf{A}}\|{-\mathbf{T}^T\mathbf{R}_s\mathbf{T}\mathbf{y}_0} - \mathbf{A}^T\mathbf{p}_0\| \leq \frac{1}{3C_\mathbf{E}}\eta\,, \tag{6.2.20}$$

where we have used (6.2.14) and (6.2.16b). Inserting these last two results into (6.2.15), and then into (6.2.12), we confirm (6.2.11). □

Remark 6.3. *In Algorithm 6.3 and Algorithm 6.4, we have equilibrated the error into equal parts of magnitude* $\zeta/2$ *and* $\eta/3$*, respectively. To optimise the time spent on the computation, we could choose to balance these weights more specifically, depending on the ratio of the condition numbers of* $\mathbf{A}$ *and* $\mathbf{R}_t$*, and the bounds of the operators involved.*

Remark 6.4. *In Algorithm 6.3 and Algorithm 6.4, the state equation in step (1) is solved more accurately than the adjoint equation in step (2) by a factor of* $\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}$*. Step (3) in Algorithm 6.4 requires an independent accuracy of* $\frac{c_t}{3\omega}$*. These differences will lead to different iteration counts in the numerical solution of the three systems.*

The complete algorithm consists of three parts, where the first and third can be interpreted as pre- and postprocessing, and the second part contains the main computations. We display it in Algorithm 6.5. First, the right hand side $\mathbf{g}$ is computed approximately. Then, the inexact conjugate gradient routine for the matrix $\mathbf{Q}$ is executed to calculate the optimal control $\mathbf{u}_\epsilon$. Finally, we calculate the state $\mathbf{y}_\epsilon$ from $\mathbf{u}_\epsilon$.

All subroutines call various inner CG methods in turn, whose accuracy has been estimated before. To estimate the error of the complete algorithm, we need to extend Propositions 6.1 and 6.2 by an argument which takes care of the postprocessing.

Theorem 6.5. *The result* $(\mathbf{y}_\epsilon, \mathbf{u}_\epsilon)$ *of the algorithm* `Control` $(\mathbf{f}, \mathbf{y}_*, \epsilon)$ *satisfies*

$$\|\mathbf{u} - \mathbf{u}_\epsilon\| \leq \epsilon\,, \qquad \|\mathbf{y} - \mathbf{y}_\epsilon\| \leq \epsilon\,. \tag{6.2.21}$$

Proof. We can confirm the first relation by the use of the error bounds in steps (2) and (3) of `Control` as follows,

$$\|\mathbf{u} - \mathbf{u}_\epsilon\| \leq \|\mathbf{Q}^{-1}(\mathbf{Qu} - \mathbf{Qu}_\epsilon)\| \leq \frac{1}{c_\mathbf{Q}}\big(\|\mathbf{g} - \mathbf{g}_\epsilon\| + \|\mathbf{Qu}_\epsilon - \mathbf{g}_\epsilon\|\big) \leq \zeta \leq \epsilon\,. \tag{6.2.22}$$

The estimate for $\mathbf{y}_\epsilon$ is verified by combining the definition of ζ from step (1) with the error bound in step (4),

$$\|\mathbf{y} - \mathbf{y}_\epsilon\| \leq \frac{1}{c_\mathbf{A}}\|\mathbf{E}(\mathbf{u} - \mathbf{u}_\epsilon) + (\mathbf{f} + \mathbf{Eu}_\epsilon - \mathbf{Ay}_\epsilon)\| \leq \frac{1}{c_\mathbf{A}}\Big(C_\mathbf{E}\zeta + \frac{c_\mathbf{A}}{3}\epsilon\Big) \leq \frac{2}{3}\epsilon + \frac{1}{3}\epsilon = \epsilon\,, \tag{6.2.23}$$

which finishes the proof. □

Remark 6.6. *We omit the description of the computation and error estimate of the adjoint* $\mathbf{p}_\epsilon$ *here, which is essentially analogous.*

Remark 6.7. *The results derived in this section hold under the assumption that the inexact conjugate gradient method converges to the exact solution within the specified accuracy. We motivate appropriate error bounds for the inexact evaluation of* $\mathbf{Q}$ *in the following section.*

6.2.2 Inexact Outer Iterations

In the previous section, we have justified the error bounds inside the subroutines `RHS` and `Apply` $(\mathbf{Q}, \eta)$, and for the overall algorithm `Control`. They guarantee that the approximate application of $\mathbf{Q}$ to a vector $\mathbf{u}$ meets any given accuracy requirement η. We now turn to the inexact conjugate gradient routine `CG` $(\mathbf{Q}, \epsilon)$ from Algorithm 6.2 to examine the choice of the bounds θ_ϵ and η_k.

In [69], the convergence of an inexactly preconditioned conjugate gradient method has been discussed. It was found that the algorithm converges linearly as long as the errors η_k do not exceed a certain threshold. In fact, this threshold turned out to be fairly large, meaning that a relatively low accuracy of the inner solver is acceptable.

Our situation is more general in the sense that we do not presume a specific structure of the error in the application of $\mathbf{Q}$. Following [139], we know that the *computed* residual $\mathbf{q}_k$ for a general matrix $\mathbf{M}$ is different from the *exact* residual. It is therefore convenient to split the residual error in step k according to

$$\|\mathbf{Mx}_k - \mathbf{b}\| \leq \|\mathbf{q}_k\| + \|\mathbf{q}_k - (\mathbf{Mx}_k - \mathbf{b})\|\,. \tag{6.2.24}$$

The first term on the right is directly accessible in the algorithm, where it is bounded by θ_ϵ in the stopping criterion. The second term, which is sometimes called the *residual gap*, still needs to be estimated. This has been done for various algorithms, among others the Richardson and Chebyshev iterations, GMRES and the conjugate gradient method [139, 140].

The main result for the CG routine from Algorithm 6.2 reads

$$\|\mathbf{q}_k - (\mathbf{Mx}_k - \mathbf{b})\| \leq \sum_{i=0}^{k-1} \eta_i |\alpha_i|\,. \tag{6.2.25}$$

Setting the stopping criterion in step (II) to $\theta_\epsilon = \epsilon/2$, we can conclude from the triangle inequality that the values of η_k have to be chosen in such a way that they satisfy

$$\sum_{i=0}^{k-1} \eta_i |\alpha_i| \leq \epsilon/2\,. \tag{6.2.26}$$

Because α_k is only computed after `Apply` $(\mathbf{Q}, \eta_k)$ has been called, it seems difficult to base the choice of η_k on this estimate. However, this problem can be circumvented easily since intermediate results for $\mathbf{h}_k$ are available inside `Apply`, and the calculation of α_k can be absorbed into the subroutine.

In summary, if and when the stopping criterion is fulfilled, the inexact CG algorithm guarantees the prescribed upper bound of the residual. Convergence is confirmed in practice, as supported by numerical evidence from [19,20,139], provided that the bounds η_k are of the same order of magnitude as ϵ. Moreover, the term *relaxation strategy* has been proposed as generally only in the beginning the applications of the matrix need to be computed to this high precision, whereas the accuracy requirements can be relaxed progressively in the course of the algorithm, when the computed residual diminishes.

We have found in our experiments that setting $\eta_k = \epsilon$, also suggested in [139], yields satisfying results, even though this choice does not comply with (6.2.26). It appears as a good compromise between cheap inner iterations and a fast overall convergence of the outer algorithm.

6.2.3 Nested Iteration

Using a uniform wavelet discretisation, we work with a hierarchy of nested spaces S_j. The level j is the central parameter which specifies the resolution. The number of unknowns on a given level is $N_j \sim 2^{nj}$. Since the wavelet basis on level $j+1$ consists of the wavelet basis for level j, complemented by additional functions, the matrices $\mathbf{A}_j$ in wavelet representation are given by the upper left sub-block of size $N_j \times N_j$ of the infinite matrix $\mathbf{A}$. Hence, a matrix for a given resolution contains all smaller matrices for the lower levels. Of course, we never compute the entries of these matrices explicitly, but use (4.3.16) instead.

The process of nested iteration as described in Algorithm 4.1 starts with the exact solution of the system on the coarsest level. Then, the level of resolution is increased until it reaches the maximum predetermined level J. Every solution on intermediate levels j constitutes a more accurate representation of the infinite-dimensional problem by enlarging the system matrix, and the intermediate solutions are found in $\mathcal{O}(N_j)$ operations. The total operation count is $\mathcal{O}(N_J)$.

To integrate the principle of nested iteration with the general Algorithm 6.1, we implement the sweep from coarse to fine levels around the conjugate gradient loop by means of an enhanced routine `NICG`. The full nested iteration inexact conjugate gradient algorithm `nIIcG/2` for the control problem is presented in Algorithm 6.6. The routine `NICG` is called on the inner layer from within the routines `RHS` and `Apply` $(\mathbf{Q})$. The iterations on the outer layer are performed by calling `NICG` $(\mathbf{Q})$ in the main programme. Both layers only differ in the way the system matrix is applied.

Remark 6.8. *The nested iteration routine* `NICG` $(\mathbf{Q}, J)$ *on the outer layer indirectly calls* `Apply` $(\mathbf{Q}, j)$ *for all intermediate levels* $j_0 \leq j \leq J$*, which in turn contains three inner calls to* `NICG` (j)*. Numerical experiments confirm that this twofold recursive sweep over the levels is indeed necessary to avoid logarithmic factors in the runtime complexity. The idea of reusing start values obtained in previous outer iterations is not as robust, since the right hand sides for the inner systems change in the process.*

As reasoned in Section 4.3.2, the target accuracy of the numerical scheme should be balanced with the discretisation error 2^{-J} for piecewise linear wavelets. Here J is the finest level chosen a priori as a compromise between high accuracy on the one hand and time and memory consumption on the other. The factor ν can be used to compensate for constants which are introduced when the result needs to be processed further, or to adapt to the magnitude of the data.

To initialise the procedure on the lowest level j_0, we need exact inversions of the matrices $\mathbf{A}_{j_0}$, $(\mathbf{R}_t)_{j_0}$ and $\mathbf{Q}_{j_0}$. Since the method of conjugate gradients is an exact solver, it can also be applied on the lowest level j_0, provided that the number of iterations for each run equals N_{j_0}. We use this procedure for the stiffness matrix $\mathbf{A}_{j_0}$ and its transpose, the Riesz matrix $(\mathbf{R}_t)_{j_0}$, and also for the reduced matrix $\mathbf{Q}_{j_0}$.

```
Subroutine CG $(\mathbf{M}, \mathbf{b}, \mathbf{x}^0, j, \epsilon) \to \mathbf{x}$        (see Algorithm 6.2)
(I) SET $\mathbf{d}^0 := -\mathbf{q}^0 := \mathbf{b} - \mathbf{M}\mathbf{x}^0$, SET $k := 0$.
(II) WHILE $\|\mathbf{q}^k\| > \theta_\epsilon$
    (1) CALL Apply $(\mathbf{M}, \mathbf{d}^k, j, \eta_k) \to \mathbf{h}^k$.
    (2) COMPUTE $\alpha^k$, $\mathbf{x}^{k+1}$, $\mathbf{q}^{k+1}$.
    (3) COMPUTE $\beta^k$, $\mathbf{d}^{k+1}$, SET $k := k+1$.
(III) ACCEPT $\mathbf{x}^k \to \mathbf{x}$.
```

```
Subroutine NICG $(\mathbf{M}, \mathbf{b}, J, \epsilon) \to \mathbf{x}_J$        (see Algorithm 4.1)
(I) CALL CG $(\mathbf{M}, \mathbf{b}, \mathbf{0}, j_0, 10^{-6}\epsilon) \to \mathbf{x}_{j_0}$, SET $j := j_0$.
(II) WHILE $j < J$
    (1) PROLONGATE $\mathbf{x}_j \to \mathbf{x}_{j+1}^0$, SET $j := j+1$.
    (2) CALL CG $(\mathbf{M}, \mathbf{b}, \mathbf{x}_j^0, j, 2^{-(j-J)}\epsilon) \to \mathbf{x}_j$.
(III) ACCEPT $\mathbf{x}_j \to \mathbf{x}_J$.
```

```
Subroutine RHS $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, j, \epsilon) \to \mathbf{g}$        (see Algorithm 6.3)
(1) CALL NICG $(\mathbf{A}, \mathbf{f}, j, C_1\epsilon) \to \mathbf{g}_1$.
(2) CALL NICG $(\mathbf{A}^T, -\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{g}_1 - \mathbf{y}_*), j, C_2\epsilon) \to \mathbf{g}_2$.
(3) COMPUTE $\mathbf{E}^T\mathbf{g}_2 \to \mathbf{g}$.
```

```
Subroutine Apply $(\mathbf{Q}, \mathbf{u}, j, \epsilon) \to \mathbf{m}$        (see Algorithm 6.4)
(1) CALL NICG $(\mathbf{A}, \mathbf{E}\mathbf{u}, j, C_3\epsilon) \to \mathbf{y}_0$.
(2) CALL NICG $(\mathbf{A}^T, -\mathbf{T}^T\mathbf{R}_s\mathbf{T}\mathbf{y}_0, j, C_4\epsilon) \to \mathbf{p}_0$.
(3) CALL NICG $(\mathbf{R}_t, \mathbf{u}, j, C_5\epsilon) \to \mathbf{q}_0$.
(4) COMPUTE $\omega\mathbf{q}_0 - \mathbf{E}^T\mathbf{p}_0 \to \mathbf{m}$.
```

```
Algorithm n𝕀cG/2 $(J, \nu) \to (\mathbf{y}, \mathbf{u})$        (see Algorithm 6.5)
(1) SET $\epsilon := \nu 2^{-J}$.
(2) CALL RHS $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, J, C_6\epsilon) \to \mathbf{g}$.
(3) CALL NICG $(\mathbf{Q}, \mathbf{g}, J, C_7\epsilon) \to \mathbf{u}$.
(4) CALL NICG $(\mathbf{A}, \mathbf{f} + \mathbf{E}\mathbf{u}, J, C_8\epsilon) \to \mathbf{y}$.
```

Algorithm 6.6: We list the full two-layer nested iteration inexact conjugate gradient algorithm `n𝕀cG/2`*. All routines are called with the parameter j or J, which determines the level of resolution on which it operates. The sweeps over the hierarchy of levels are realised by the (twofold recursive) subroutine* `NICG`*.*

In the latter case, each of the N_{j_0} applications of $\mathbf{Q}_{j_0}$ requires in turn the exact inversion of the three inner systems. In practise, it is not even necessary to actually perform N_{j_0} iterations on the lowest level, which could cause a substantial impact on the overall computation time for three or more dimensions, but only a sufficient number to reduce the residual to a value below the target accuracy multiplied by a small number of e.g. 10^{-6}.

Consequently, the programme implements the exact solution on the coarsest level and the iterative solution on all higher levels in much the same way. Only the maximum number of iterations per conjugate gradient loop and the respective stopping criteria need to be adapted. This ansatz avoids redundancy and permits the reuse of code.

6.3 Numerical Results

For the numerical experiments displayed here, we have chosen the Neumann problem with distributed control as introduced in Section 5.2.2, defined on the unit cube, combined with distributed observation. The normal derivative g (5.2.20b) on the boundary is set to zero. Since the numerical scheme is general with respect to boundary conditions, we could also handle boundary control, or a Dirichlet problem with distributed control. However already the distributed Neumann problem yields a diversity of results, which we aim to discuss in detail here. Thus, it would be out of the scope of this thesis to consider numerical results for different example problems as well.

We show the effect of several choices of right hand side f and target function y_*. Additionally, we examine various combinations of the choice of observation and control spaces, characterised by the parameters s and t. The influence of the regularisation parameter ω is also studied. We have collected numerical results for one, two and three spatial dimensions and list convergence histories for various combinations of data and parameters. Graphical displays of the results are presented for $n = 1$ and $n = 2$.

The Neumann problem requires the use of wavelets with free boundary conditions according to Conclusion 4.12. The Riesz operators have been computed as proposed in (5.3.25), see Section 5.3. All newly developed techniques and optimisations covered in earlier parts of this document have thus been incorporated.

We devote the next two sections to an in-depth inspection of several one-dimensional scenarios, since they most clearly show a number of interesting features. We first confirm that the numerical method exactly recovers the analytical solutions for constant data. Then we move on to a more general example in order to inspect the influence of the smoothness parameters s and t in the objective functional. We discuss the qualitative effect of s and t, which lead to the appearance of phenomena which do not occur by the variation of ω alone. Next, we test the robustness of the method by means of two specific situations, namely the cases $\omega = 0$ and $y_* \notin Z$, for which the problem is no longer well-posed.

We conclude this chapter with examples in two and three dimensions. Due to the tensor product structure, the implementation is completely general with respect to dimension. We assess that the effects of parameters and data are similar to the one-dimensional case. Although the problem is linear, the availability of three independent modelling parameters, namely s, t and ω, in combination with arbitrary data f and y_*, gives rise to a wide variety of phenomena.

6.3.1 Conforming examples

We first explore the dependence of the state y and the control u on some basic combinations of parameters and data in one dimension, which are *conforming* in the sense that $\omega > 0$ and $f \in Y'$, $y_* \in Z$, see Section 5.2.1. The domain is $\Omega = (0, 1)$, and the accuracy multiplier for the outer iterations has been set to $\nu = \frac{1}{100}$, cf. Section 6.2.3.

s	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
c_s	0.19	0.25	0.22	0.18	0.13	0.087	0.059	0.041	0.030	0.027	0.031	0.073
C_s	8.5	5.3	3.3	2.1	1.3	0.80	0.51	0.43	0.41	0.51	0.73	2.5
$C_\mathbf{T}$	0.12	0.15	0.19	0.23	0.29	0.35	0.43	0.53	0.66	0.81	0.90	1.0

s	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
$\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}$	0.60	0.61	0.61	0.66	0.67	0.74	0.77	0.60	0.41	0.22	0.12	0.029

t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
$\frac{c_\mathbf{A}}{C_\mathbf{E}}$	0.61	0.49	0.38	0.32	0.25	0.21	0.17	0.14	0.11	0.090	0.081	0.073
$c_\mathbf{Q}$	0.12	0.19	0.30	0.48	0.77	1.3	2.0	2.3	2.4	2.0	1.4	0.40

Table 6.1: The constants appearing in the inexact conjugate gradient algorithm are listed in this table. They are computed in one dimension for the level $J = 8$. They depend on the smoothness parameters s and t. Note that $C_\mathbf{T} = C_\mathbf{T}(s)$ and $C_\mathbf{E} = C_\mathbf{E}(t)$. We have not provided explicit values for $C_\mathbf{E}$ since $C_\mathbf{E}(r) = C_\mathbf{T}(r)$. The lower bound for $\mathbf{Q}$ has been computed with $\omega = 1$, see (6.2.7).

The detailed algorithm described in Section 6.2 uses several constants to balance the accuracies of inner and outer computations. We have listed them in Table 6.1 for the applicable range of parameters, computed for $J = 8$. The most important factors are $\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}$, $\frac{c_\mathbf{A}}{C_\mathbf{E}}$ and $c_\mathbf{Q}$. The algorithm is expected to converge fastest when these are large. This indicates that the L_2 case is the most robust, while the setting of natural norms with $s = t = 1$ requires the tightest tolerances and therefore the highest count of total iterations. We will validate this observation with the numerical examples throughout the rest of this chapter.

The results of the numerical computations are structured as follows. In a first step, we present graphical displays to examine the qualitative effect of the parameters s, t and ω on the shape of the state and the control, computed with a highest level of $J = 8$. In a second step, we analyse the convergence and performance of the algorithm. To this end, we provide and interpret tables computed with $J = 16$, containing iteration counts and various measurements of numerical errors.

Example 6.1 – Constant Data

The most basic case is of course given for *constant* data f and y_*, for which the smoothness indices s and t have no effect. The control problem then simplifies to

$$y = f + u\,, \tag{6.3.1a}$$

$$p = -(y - y_*)\,, \tag{6.3.1b}$$

$$\omega u = p\,, \tag{6.3.1c}$$

which has the exact solution

$$y = \frac{y_* + \omega f}{1+\omega}\,, \qquad u = \frac{y_* - f}{1+\omega}\,. \tag{6.3.2}$$

The degenerate case of vanishing regularisation, i.e., $\omega = 0$, leads to a well-defined solution here, namely $y = y_*$ with $p = 0$. The data are compatible if and only if $f = y_*$, which is equivalent to $u = p = 0$. Moreover, this formulation gives hints for the qualitative behaviour of the state and the control when the data y_* and f are not constant, or when ω is changed.

To demonstrate that our algorithm recovers the exact solutions in the case of constant data, we vary ω and keep all other parameters unchanged. We display the corresponding results in Figure 6.1, where we examine two scenarios, namely $f = 1$, $y_* = 0$ and $f = 0$, $y_* = 1$. We observe that the results are

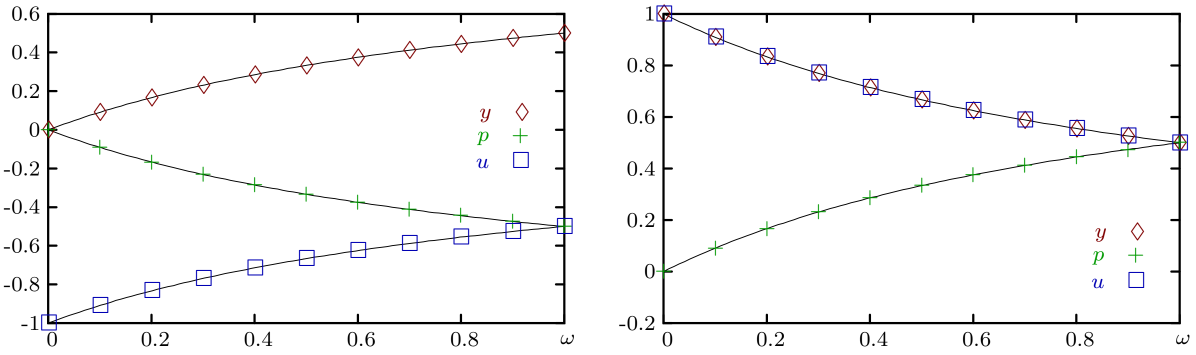


Figure 6.1: We display state, costate and control for constant data f and y_* for varying regularisation parameter ω, which is assigned to the x-axis. The exact solutions are known in this scenario, cf. (6.3.2)*, they are also constant. Their dependence on ω is drawn with black lines. For our numerical experiments, we have changed ω from* 0 *to* 1 *in steps of $\frac{1}{10}$. The left graph shows the numerical solutions for the data $f = 1$, $y_* = 0$, while the right graph corresponds to $f = 0$, $y_* = 1$. (We have selected here the norms $s = 0.6$ and $t = 0.4$, yet the results are the same for any values of these two parameters as predicted by the theory.)*

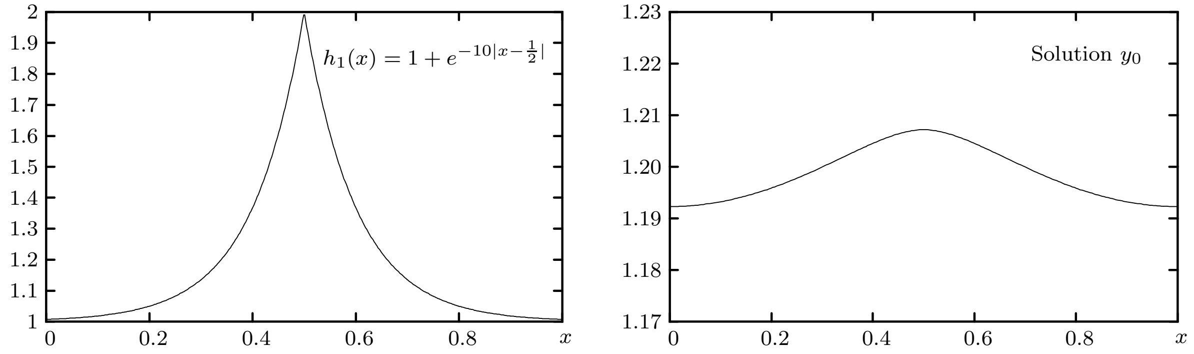


Figure 6.2: We show the right hand side $f = h_1$ from (6.3.3) *for our first set of examples in one dimension, and the corresponding solution y_0 of the elliptic PDE with zero control.*

independent of the smoothness indices s and t and agree perfectly with the theoretical predictions (6.3.1c) and (6.3.2).

Example 6.2 – Non-Constant Data

We now turn to non-trivial data, where the variation of s and t does affect the results. To this end, we choose the non-smooth right hand side $f = h_1$ with

$$h_1(x) \coloneqq 1 + e^{-10|x-\frac{1}{2}|}\,. \tag{6.3.3}$$

It exhibits a sharp peak at $x = \frac{1}{2}$. The solution y_0 of the elliptic PDE with this right hand side and zero control is smooth, with values close to $\frac{6}{5}$. We have plotted both functions in Figure 6.2.

First of all, we observe that the compatible setting $y_* = y_0$ leads to $p \equiv 0$ and $u \equiv 0$ for all combinations of s and t, as predicted in Remark 5.2. For the following experiments, we select the target state $y_* \equiv 0$, which is clearly distinct from the natural solution y_0. The weight $\omega = 1$ is held fixed for this example, and we concentrate solely on the variations of the two smoothness parameters s and t.

Although f is singular, y_0 is very smooth and still almost constant. Therefore, we can consult (6.3.2) and try to predict the outcome of the numerical simulations as follows.

- The state y will be close to $\frac{1}{2}y_0$. This is a consequence of the inversion of the differential operator, which acts as a smoothing just as in the situation with zero control. The choice of observation space H^s will have only minor effects on the solution because of the smoothness of y_0.
- Inserting the first equation from (6.3.2) into the second, we obtain an alternative representation for the control u. Comparing both formulations,

$$u^{(0)} = \frac{y_* - y}{\omega}\,, \qquad u^{(1)} = \frac{y_* - f}{1+\omega}\,, \tag{6.3.4}$$

  we can see that the control will shift between two extreme cases. In the L_2 case, we expect an almost constant $u^{(0)} \approx -\frac{1}{2}y_0$, while the natural norms will yield a sharply peaked $u^{(1)} \approx -\frac{1}{2}f$.

These heuristic arguments provide good ideas for the outcome of the numerical computations. The output of several examples is collected in Figure 6.3. We have made three series of experiments which are displayed in three rows, with the graphs for the state y on the left and the control u on the right. In the first row we keep $t = 0$ and vary s, while we vary $s = t$ simultaneously in the middle row. The last series deals with the case $t = 1$ and varying s.

The state y is indeed close in value and shape to $\frac{1}{2}y_0$ for all choices of parameters. It almost perfectly coincides in the case of natural norms $s = t = 1$, while the amplitude of its bump is enlarged by a factor of about 2 when the parameters approach $s = 0$, $t = 1$ or $s = 1$, $t = 0$. These mixed cases produce very similar states, which are also very close to the standard setting $s = t = 0$.

For $t = 0$, the control is smooth for any value of s. It is almost constant for $s = 0$ and approaches $-\frac{1}{2}y_0$ for $s = 1$. However, the picture changes completely when t is varied. For increasing t, the control develops a sharp singularity and approaches $-\frac{1}{2}f$ for $t = 1$. In the extreme cases $t = 0$ and $t = 1$, the control is smooth away from $x = \frac{1}{2}$, while it exhibits oscillations for intermediate values of t.

We interpret these oscillations of u for $0 < t < 1$, which corresponds to control spaces of fractional negative Sobolev smoothness, as artifacts induced by the inverse fractional Riesz operators from (5.3.23). Looking more closely at the graphs, we make the following observations.

- The oscillations show up visibly for about $t > \frac{1}{2}$ and attain their maximal amplitude for values of t between 0.9 and 0.95. This behaviour could be altered by a rescaling of the interpolation parameter in the definition of the Riesz operator, see (5.3.21).
- The oscillations are made up of peaks at dyadic divisions of the x-axis, i.e., they occur for $x = 2^{-j}k$, and get smaller with increasing j. The peaks are independent of the target accuracy ϵ and are also unaffected by the presence of the operator-adapted preconditioning scheme. They are thus pure effects of the space which is spanned by the choice of wavelet discretisation.
- The spikes are not punished by the weak norm in which the control u is measured. This means that high frequencies are amplified, although the ℓ_2 norm of the wavelet coefficients is bounded.
- An averaging process could be invoked to eliminate the spikes. The resulting control would span the space between the curves for $t = 0$ and $t = 1$ in a plausible manner.

We conclude that these results almost completely meet our predictions concerning shape and size of the solution and the control. For fractional values of t, we expect a control in $(H^t)'$, so the spikes do not contradict the mathematical model. On the other hand, the construction of Riesz operators for this special situation may be worth further studies.

To verify that our method has a numerical complexity of $\mathcal{O}(N)$, we present iteration histories in Table 6.2. We provide detailed characteristics of several runs of the algorithm for different sets of parameters. This

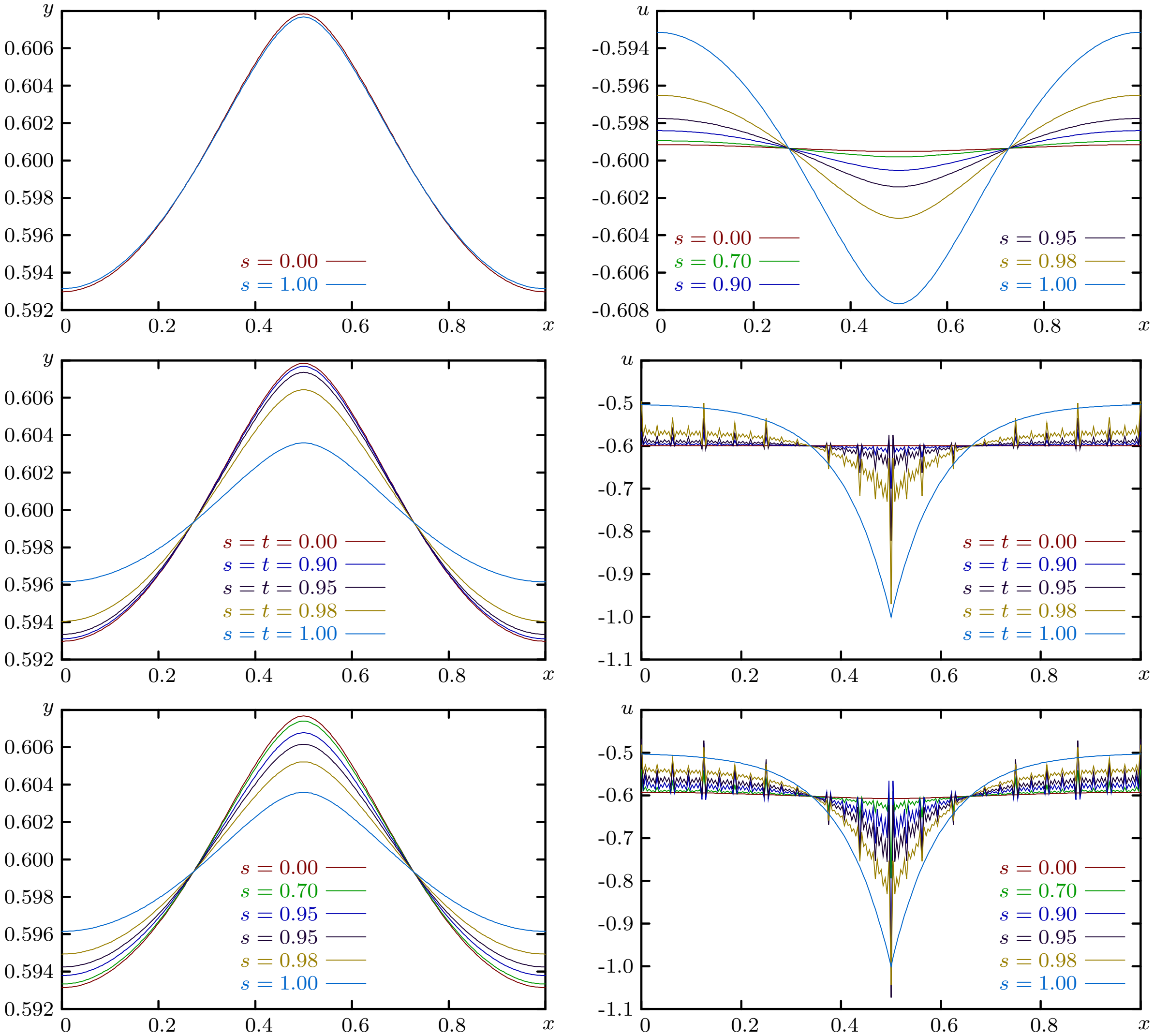


Figure 6.3: We show the results for the computations with $y_* \equiv 0$ and right hand side f as in Figure 6.2, with $\omega = 1$. Each row contains a graph for the state y on the left and the corresponding control u on the right. In the top row, we vary s between 0 and 1 while $t = 0$ is fixed. In the middle row, we vary $s = t$ simultaneously, while in the last row we fixed $t = 1$ and again only vary s.

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						3.23e-03	8.87e-03	3.98e-07	1.63e-06
4	3.72e-06	1	1	0	1	1.01e-03	4.53e-03	1.63e-06	1.63e-06
5	3.76e-06	1	2	0	3	2.77e-04	2.26e-03	1.63e-06	1.63e-06
6	1.27e-06	2	2	0	3	7.73e-05	1.12e-03	5.23e-07	5.23e-07
7	1.29e-06	1	3	0	3	2.32e-05	5.61e-04	5.23e-07	5.23e-07
8	5.51e-07	2	2	0	3	7.64e-06	2.81e-04	3.00e-07	3.00e-07
9	3.25e-07	2	2	0	3	3.80e-06	1.40e-04	1.67e-07	1.67e-07
10	1.72e-07	2	2	1	3	8.46e-07	7.01e-05	1.07e-07	1.07e-07
11	1.09e-07	2	2	1	3	5.18e-07	3.50e-05	6.58e-08	6.58e-08
12	1.50e-08	3	2	1	3	3.82e-07	1.75e-05	1.26e-08	1.26e-08
13	1.51e-08	1	5	2	3	4.08e-08	8.69e-06	1.26e-08	1.26e-08
14	1.01e-08	2	2	1	3	2.92e-08	4.24e-06	5.42e-09	5.42e-09
15	4.14e-09	2	2	1	3	2.16e-08	1.90e-06	3.98e-09	3.98e-09
16	7.80e-10	3	2	1	3	1.71e-08	1.71e-08	1.16e-09	1.16e-09

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						3.17e-03	8.69e-03	2.01e-05	1.37e-04
4	1.43e-05	4	6	2	6	9.90e-04	4.45e-03	6.71e-06	2.05e-05
5	6.03e-06	3	5	2	6	2.73e-04	2.22e-03	3.29e-06	3.95e-06
6	1.47e-06	3	5	2	5	7.62e-05	1.11e-03	1.19e-06	1.22e-06
7	1.54e-06	1	12	6	6	2.31e-05	5.52e-04	1.21e-06	1.22e-06
8	7.80e-07	2	7	3	5	7.92e-06	2.76e-04	7.77e-07	7.77e-07
9	2.68e-07	3	5	2	4	4.15e-06	1.38e-04	4.04e-07	4.04e-07
10	1.27e-07	2	7	3	4	1.17e-06	6.89e-05	3.59e-07	3.59e-07
11	8.58e-08	3	5	2	4	6.98e-07	3.44e-05	1.75e-07	1.75e-07
12	5.34e-08	3	5	2	4	5.15e-07	1.72e-05	1.17e-07	1.17e-07
13	2.55e-08	4	4	2	4	5.64e-08	8.54e-06	2.25e-08	2.25e-08
14	1.39e-08	2	7	3	5	4.59e-08	4.17e-06	1.68e-08	1.68e-08
15	5.12e-09	3	5	2	4	3.48e-08	1.86e-06	1.12e-08	1.12e-08
16	3.25e-09	4	4	2	4	1.85e-08	1.85e-08	1.66e-09	1.66e-09

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						3.23e-03	5.25e-03	1.02e-03	1.04e-03
4	2.24e-06	8	9	7	7	9.99e-04	2.42e-03	4.25e-04	4.28e-04
5	1.44e-06	9	12	9	9	2.80e-04	1.15e-03	1.32e-04	1.32e-04
6	5.97e-07	10	14	8	8	7.88e-05	5.67e-04	3.79e-05	3.79e-05
7	2.80e-07	10	14	9	9	2.25e-05	2.82e-04	1.07e-05	1.07e-05
8	1.04e-07	10	13	7	7	1.02e-05	1.41e-04	3.11e-06	3.11e-06
9	9.39e-08	9	13	7	7	2.13e-06	7.01e-05	9.57e-07	9.57e-07
10	7.80e-08	7	12	8	8	1.11e-06	3.51e-05	3.27e-07	3.27e-07
11	2.63e-08	7	11	7	7	7.70e-07	1.75e-05	1.08e-07	1.08e-07
12	2.04e-08	6	11	7	7	8.10e-08	8.75e-06	3.90e-08	3.90e-08
13	7.20e-09	6	11	7	7	4.46e-08	4.35e-06	1.38e-08	1.38e-08
14	2.46e-09	6	11	7	7	2.89e-08	2.12e-06	4.89e-09	4.89e-09
15	2.55e-09	5	11	7	7	2.29e-08	9.49e-07	1.77e-09	1.77e-09
16	8.78e-10	5	11	7	7	1.89e-08	1.89e-08	6.33e-10	6.33e-10

Table 6.2: For the case $f = h_1$ and $y_* \equiv 0$ in one dimension, we display the residuals, iteration numbers and errors. We have fixed $J = 16$ and $\omega = 1$. The topmost table contains the data for $s = t = 0$. In the middle, the parameters are $s = 1$, $t = 0$, and the last table contains values for $s = t = 1$.

includes the values for the control residual, iteration counts and error norms for all levels of resolution by comparing the results of standard runs at $\nu = \frac{1}{100}$ with precalculated solutions of high precision, namely $\nu = 10^{-5}$.

We have compiled three tables. The topmost corresponds to the values $s = t = 0$, the table in the middle to $s = 1$, $t = 0$ and the last to $s = t = 1$. Each table is organised as follows. Every row corresponds to a certain level of resolution, from the coarsest level $j_0 = 3$ to the finest level $J = 16$, which is also the level of the high accuracy solution. The columns contain first the control residual, i.e., the residual in the vector $\mathbf{u}$ after the last of the outer iterations. The total count of outer iterations for each level is provided next to it, namely in the column labelled #O. The next set of three columns contains the rounded average number of inner iterations per outer iteration for the solutions of the elliptic system (#E), the adjoint system (#A), and the inversion of the Riesz operator $\mathbf{R}_t$ (#R). As the inner iterations also employ the nested iteration strategy from the lowest to the current level, we have chosen the most conservative representation for these three columns, namely the maximum of inner iterations per level between $j_0 + 1$ and the current level. Note that for the coarsest level, the values for the control residual and the iteration numbers are not listed, since all systems are solved to high accuracy for $j = j_0 = 3$ at negligible cost. In the right part of each table, we provide the ℓ_2 norms of the error with respect to the precalculated solution of high accuracy. This is done separately for the state y and the control u. For each variable, we list the error of the current solution against the restriction of the high accuracy solution. This error is denoted by ϵ_{R}. Secondly, we provide the error of the solution at the current level j which is prolongated and compared to the full high accuracy solution, which is denoted by ϵ_{P}. It follows that $\epsilon_{\mathrm{R}} \leq \epsilon_{\mathrm{P}}$. Because of the tighter accuracy ν for the reference solution compared to the standard runs, both of these errors do not vanish at the highest level J.

The quantity ϵ_{R} measures the discrepancy between the solution at a specific level j and the ideal approximation of the solution in the space S_j. It can thus be used to judge the accuracy and stability of the algorithm. Since the solution of the control problem for u corresponds to a linear elliptic boundary value problem, and all operators have a uniformly bounded condition number, we expect by Cea's Lemma that the error ϵ_{P} against the full high accuracy solution corresponds to the approximation error at the respective level of discretisation. This error exists even for perfect accuracy of the solver for a specific level j, and depends on the smoothness of the approximated function. This reasoning carries forward to the state y, since the state is derived from the control by a single elliptic system.

We can see that the method converges as expected, with minor differences between the three runs. The first two need only very few outer iterations, generally between 1 and 4, which seems to induce some variations in the amount of inner iterations. We attribute these to the outer method of conjugate gradients which does not seem to reach the asymptotic regime with only that few iterations. Note that an inner iteration count of 0 means that the solution of the corresponding system has already been available to a sufficient precision on the lowest level. For all three examples, the amount of iterations per level stays constant with increasing resolution, which means that the computational cost is $\mathcal{O}(N_J)$. In the last example we can even see a reduction in the amount of outer iterations toward higher resolutions. The overall cost increases from $s = t = 0$ to $s = t = 1$, which appears to be the numerically most expensive case. However, this is in part due to the fact that the a-priori tolerances on the redisual in column 2 get tighter with increasing s and t. The iteration counts for the inner elliptic system are larger than for the adjoint system, since it has to be solved for higher accuracy (this has been predicted in Remark 6.4). The effort for the numerical inversion of the Riesz operator lies somewhere in between. Altogether, we have asserted that the computational complexity is linear in the number of unknowns. We will further substantiate this claim with more results in the remaining sections of this chapter.

We have included diagrams of the errors $\epsilon_{\mathrm{P}}(y)$ and $\epsilon_{\mathrm{P}}(u)$ versus the outer residual $\|\mathbf{r}_j\|$ in Figure 6.4. The errors in both y and u show a rate around 1, which matches the error of the discretisation. The error in the control is generally smaller, which can be explained by the additional solution of an elliptic system which is necessary to derive y from u. Only for natural norms ($s = t = 1$) the control exhibits an

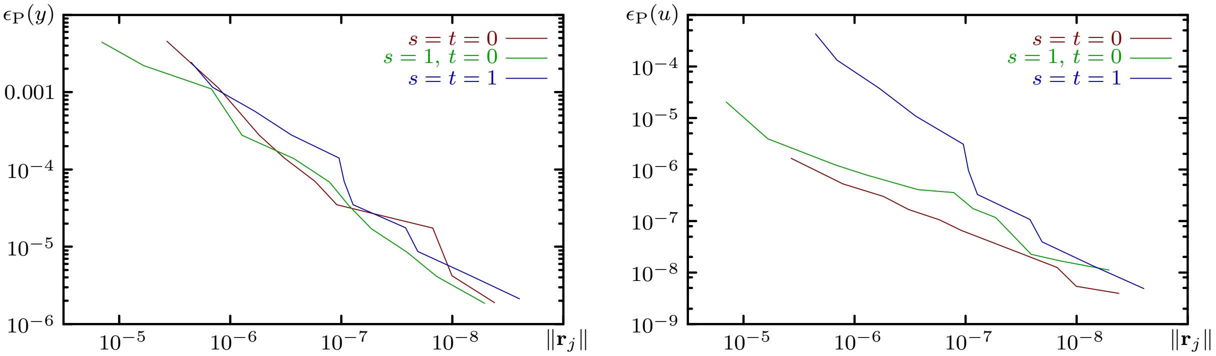


Figure 6.4: We show convergence plots for $f = h_1$ and $y_* \equiv 0$. The left hand picture contains the error $\epsilon_{\mathrm{P}}(y)$ of the state with respect to the highest level of resolution, and the right hand picture the error of the control $\epsilon_{\mathrm{P}}(u)$, both in dependence on the outer residual $\|\mathbf{r}_j\|$.

average rate of about 1.6. Considering the shape of the corresponding graph in the right hand picture, it is not quite clear if this is simply due to the steep preasymptotic reduction of $\epsilon_{\mathrm{P}}(u)$.

The runtimes on a 3.2GHz Pentium IV computer (family 15, model 4, stepping 1, with 1MB L2 Cache) are listed in the following table.

Parameters	$s = t = 0$	$s = 1,\, t = 0$	$s = t = 1$
Runtime	4.2s	5.2s	9.9s

These numbers confirm that the case of natural norms is indeed the most demanding in terms of computation time.

Summarising the results for this series of numerical experiments, we find that the method meets all expectations concerning the convergence rate and its time and memory requirements. The character and shape of the results is also consistent with our heuristic predictions. The variation of the norms in the functional, which is made possible by the wavelet discretisation, does indeed allow to study a range of phenomena which are not available in the standard L_2 case.

6.3.2 Further Characterisation of the Method

In this section, we fathom the robustness of our numerical method when the parameters or the data lie outside their allowed ranges. To this end, we examine the quality of the computed functions and convergence properties for two examples. The first deals with the limit $\omega \to 0$, while the second features a target observation $y_* \notin Z$. While discussing the latter configuration, we will also point out qualitative differences in the results between varying ω and varying the smoothness parameters s and t. We will find that the smoothness parameters indeed introduce a new and independent quality in modelling which complements the freedom already present in the variation of ω alone.

Example 6.3 – Vanishing Regularisation

First, we conduct a further test for consistency of our numerical method, namely the limit $\omega \to 0$. To this end, we continue to use the right hand side $f = h_1$. The exact results for $y_* \equiv 0$ and $\omega = 0$ are given by $y \equiv 0$ and $u = -f$. As has been pointed out in Remark 5.14, the problem is only well-posed for $s = t = 1$. The numerical results for this case are shown in Figure 6.5. It can be seen that the resulting

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						3.23e-03	3.32e-03	1.86e-03	1.90e-03
4	4.21e-07	10	9	8	6	1.01e-03	1.09e-03	7.74e-04	7.79e-04
5	6.89e-08	12	13	10	8	2.83e-04	3.48e-04	2.40e-04	2.40e-04
6	6.49e-08	13	15	11	7	7.98e-05	1.30e-04	6.90e-05	6.90e-05
7	4.45e-08	13	15	10	7	2.20e-05	5.56e-05	1.95e-05	1.95e-05
8	2.04e-08	13	15	10	6	8.79e-06	2.70e-05	5.65e-06	5.65e-06
9	1.40e-08	11	15	9	6	2.34e-06	1.30e-05	1.71e-06	1.71e-06
10	5.91e-09	11	14	8	6	1.51e-06	6.55e-06	5.41e-07	5.41e-07
11	3.20e-09	10	13	8	6	6.41e-07	3.25e-06	1.79e-07	1.79e-07
12	1.09e-09	10	12	8	6	5.19e-07	1.67e-06	6.11e-08	6.11e-08
13	3.72e-10	10	12	8	5	6.62e-08	7.93e-07	2.12e-08	2.12e-08
14	3.84e-10	8	12	8	6	2.43e-08	3.86e-07	7.53e-09	7.53e-09
15	2.65e-10	7	12	8	6	1.70e-08	1.73e-07	2.64e-09	2.64e-09
16	1.09e-10	7	12	8	5	1.45e-08	1.45e-08	1.41e-10	1.41e-10

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						3.23e-03	3.23e-03	2.03e-03	2.07e-03
4	4.36e-08	11	9	8	6	9.69e-04	9.70e-04	8.43e-04	8.49e-04
5	2.34e-08	15	14	11	6	2.79e-04	2.80e-04	2.61e-04	2.62e-04
6	1.11e-08	16	16	12	6	7.83e-05	7.91e-05	7.51e-05	7.52e-05
7	2.71e-09	17	16	12	6	2.20e-05	2.27e-05	2.13e-05	2.13e-05
8	3.36e-09	16	16	11	5	6.20e-06	6.79e-06	6.16e-06	6.16e-06
9	1.38e-09	16	16	10	5	2.28e-06	2.67e-06	1.86e-06	1.86e-06
10	8.33e-10	14	16	11	5	1.30e-06	1.47e-06	5.90e-07	5.90e-07
11	2.23e-10	14	14	9	4	1.01e-06	1.07e-06	1.95e-07	1.95e-07
12	1.91e-10	13	14	9	5	2.06e-07	2.69e-07	6.65e-08	6.65e-08
13	5.77e-11	13	13	9	4	1.44e-07	1.68e-07	2.30e-08	2.30e-08
14	2.05e-11	13	13	8	4	7.59e-08	8.67e-08	8.05e-09	8.05e-09
15	2.11e-11	11	13	9	4	6.60e-08	6.86e-08	2.73e-09	2.73e-09
16	1.24e-11	11	13	9	4	1.09e-08	1.09e-08	3.34e-11	3.34e-11

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						3.23e-03	3.23e-03	2.05e-03	2.09e-03
4	3.13e-10	12	9	9	5	9.75e-04	9.75e-04	8.51e-04	8.56e-04
5	2.29e-09	17	15	12	5	2.76e-04	2.76e-04	2.63e-04	2.64e-04
6	7.57e-10	19	17	13	5	7.57e-05	7.57e-05	7.58e-05	7.59e-05
7	4.56e-10	19	17	13	5	2.15e-05	2.15e-05	2.15e-05	2.15e-05
8	1.86e-10	19	17	12	4	6.79e-06	6.79e-06	6.21e-06	6.21e-06
9	7.20e-11	19	17	12	4	2.14e-06	2.15e-06	1.88e-06	1.88e-06
10	6.96e-11	18	17	12	4	8.93e-07	8.96e-07	5.95e-07	5.95e-07
11	2.56e-11	18	16	11	3	3.07e-07	3.09e-07	1.97e-07	1.97e-07
12	2.01e-11	16	16	11	4	1.87e-07	1.88e-07	6.71e-08	6.71e-08
13	4.87e-12	17	15	10	3	1.19e-07	1.19e-07	2.32e-08	2.32e-08
14	3.95e-12	15	15	10	3	7.72e-08	7.74e-08	8.12e-09	8.12e-09
15	1.79e-12	15	14	10	3	1.79e-08	1.80e-08	2.76e-09	2.76e-09
16	5.34e-13	15	14	9	3	1.29e-08	1.29e-08	3.19e-11	3.19e-11

Table 6.3: We show the iteration histories for $\omega \to 0$. The data are the same as in the previous experiments. We use natural norms here ($s = t = 1$) and vary ω from top to bottom in the steps 0.1, 0.01 *and* 0.001.

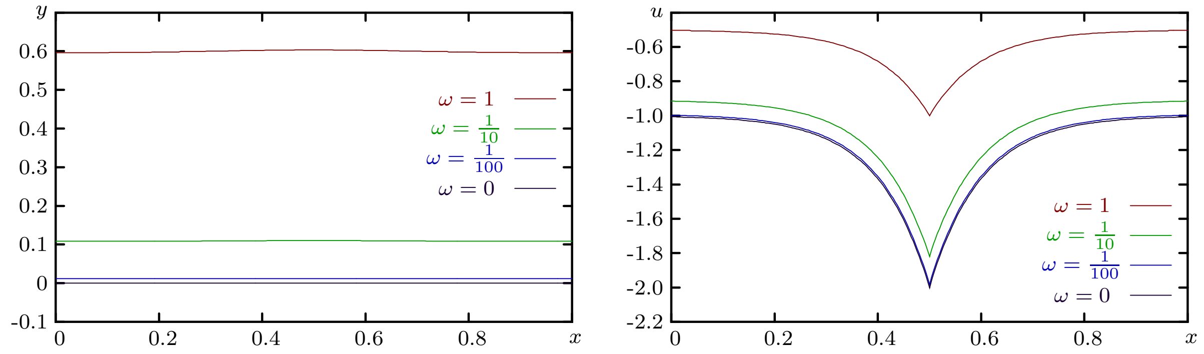


Figure 6.5: We show the graphs for y and u for the data $y_* \equiv 0$ with smoothness indices $s = t = 1$. We varied ω in four steps between 1 and 0. The curves appear well-defined and have the predicted shape. This includes the case $\omega = 0$, where the theoretical predictions $y \equiv 0$ and $u = -f$ are realised.

j	$\|\mathbf{r}_j\|$	#O	#E	#A	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3					3.23e-03	3.23e-03	2.05e-03	2.09e-03
4	2.17e-06	8	9	7	9.77e-04	9.77e-04	8.51e-04	8.56e-04
5	5.63e-06	8	12	8	2.76e-04	2.76e-04	2.64e-04	2.65e-04
6	2.29e-06	8	14	8	7.55e-05	7.55e-05	7.61e-05	7.62e-05
7	1.56e-06	8	14	8	2.11e-05	2.11e-05	2.18e-05	2.18e-05
8	8.95e-07	7	14	8	6.14e-06	6.14e-06	6.77e-06	6.78e-06
9	2.84e-07	7	13	7	1.88e-06	1.88e-06	2.06e-06	2.06e-06
10	1.81e-07	6	11	7	5.87e-07	5.87e-07	7.39e-07	7.40e-07
11	6.88e-08	6	11	8	1.93e-07	1.93e-07	2.56e-07	2.56e-07
12	4.32e-08	4	12	8	7.21e-08	7.21e-08	9.57e-08	9.57e-08
13	2.58e-08	4	11	7	2.72e-08	2.72e-08	4.19e-08	4.19e-08
14	1.23e-08	5	11	7	9.99e-09	9.99e-09	1.76e-08	1.76e-08
15	4.90e-09	5	11	7	3.50e-09	3.50e-09	7.23e-09	7.23e-09
16	3.46e-09	3	12	8	3.17e-09	3.17e-09	4.77e-09	4.77e-09

Table 6.4: These numbers correspond to the setting $\omega = 0$. As before, the smoothness indices are $s = t = 1$. In fact, the choice of t is irrelevant in this case as it drops out of the functional. The column #R disappears here for the same reason.

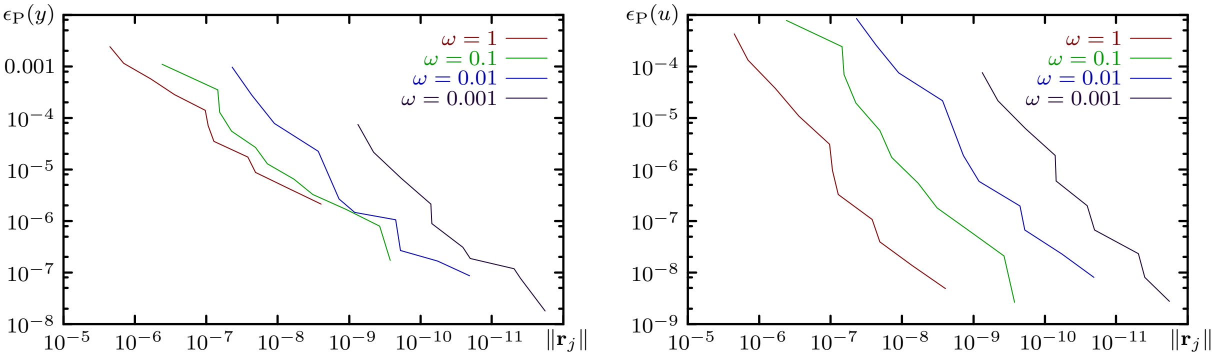


Figure 6.6: We show convergence plots for the example of vanishing regularisation with $s = t = 1$ and $\omega \to 0$.

state y and control u are again in good agreement with our expectations. The control steadily shifts from about $u = -\frac{1}{2}f$ for $\omega = 1$ to $u = -f$ for $\omega = 0$, while the state moves from about $y = \frac{1}{2}y_0$ to $y = y_* \equiv 0$.

The iteration histories are listed in Table 6.3 for the values $\omega = 0.1$ to $\omega = 0.001$, and in Table 6.4 for the special case $\omega = 0$. The iteration counts only increase slightly from $\omega = 1$ toward $\omega = 0.001$, although the bounds on the residual in column 2 get rapidly tighter for $\omega \to 0$, see (6.2.7). For the limit $\omega = 0$, we have used the ad-hoc setting $c_{\mathbf{Q}} := 1$. In this case the iterations need fewer steps than for $\omega = 1$. The occurrence of $\mathbf{R}_t$ drops out of the algorithm completely, which saves one complete inner inversion of an elliptic system and leads to a slightly simpler numerical scheme. This case also yields the highest accuracy for the state y, which appears plausible since we have chosen $y_* \equiv 0$, which becomes the exact solution in the limit $\omega = 0$.

Figure 6.6 shows graphs of the errors versus the resolution. For decreasing ω, the rate in $\epsilon_P(y)$ grows from just above 1 to almost 1.5 for $\omega = \frac{1}{1000}$. We attribute this to the fact that for $\omega = 0$ it holds $y = y_*$ (in this limit case the rate in y is 1.7), which seems to positively influence the convergence. The control features a rate around 1.6 for all runs. The generally superlinear convergence for vanishing regularisation is reflected in the steadily decreasing number of iterations with increasing resolution.

Our observations concerning the dependence of the total iteration counts on ω are reflected in the actual times needed for the computations, which are provided in the following table.

ω	1	0.1	0.01	0.001	0
Runtime	9.9s	13s	18s	22s	4.3s

Additionally, this example demonstrates the ill-posedness of the problem for $s = t = 0$. The two images in Figure 6.7 show that the computed control is not accurate in this case, but oscillating around the exact value $-f$. The images on the left have been obtained without the additional preconditioning as described in Section 4.3.3. Here the bests results are achieved in the range of $\nu \in [\frac{1}{10}, \frac{1}{100}]$. For target accuracies outside this range, no matter if larger or smaller, the computed control is completely off target. This effect is reversed with preconditioning in the right hand image, where these accuracies yield the worst results, which get better, but still not good for larger or smaller accuracy. In fact, this is the only situation which we encountered where the operator-adapted preconditioning influences the quality of the results.

Hence, these experiments confirm once more that the numerical scheme works as predicted when all parameters have allowed values, and that it can indeed produce unpredictable results when we deliberately choose an ill-posed combination.

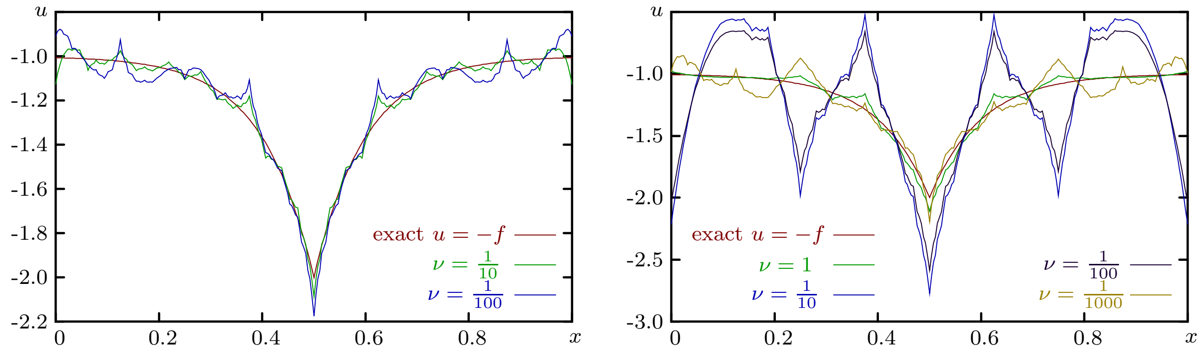


Figure 6.7: These graphics demonstrate the difficulties with $\omega = 0$ *for the parameters* $s = t = 0$*, when the problem is not well-posed. In this case, we even get different results without (left) and with (right) operator-adapted preconditioning. In the first case, for values of* ν *significantly above* $\frac{1}{10}$ *or below* $\frac{1}{100}$*, the computed control lies far off the exact function (those cases are not shown here). With preconditioning, we can see the opposite behaviour, when the results with* $\frac{1}{10}$ *and* $\frac{1}{100}$ *are worst.*

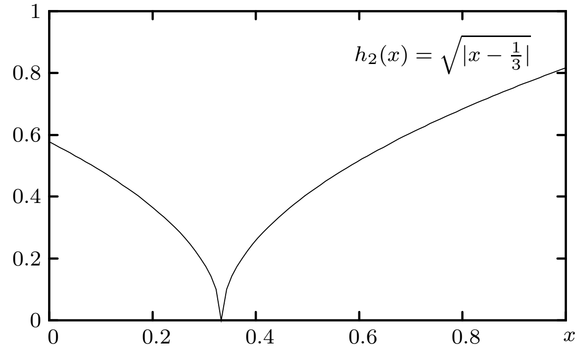


Figure 6.8: We display the target $y_* = h_2$ *as of* (6.3.5) *which is used for our non-conforming experiments.*

Example 6.4 – Non-Conforming Target Observation

The second non-conforming example deals with the constant right hand side $f \equiv 1$ and a specific non-constant target function y_*. We have chosen $y_* = h_2$ with

$$h_2(x) := \sqrt{\left|x - \frac{1}{3}\right|}. \tag{6.3.5}$$

This function is displayed in Figure 6.8. Note that this choice of y_* does not comply with homogeneous Neumann boundary conditions, and it is also not in H^1. (Thus, for natural norms it holds that $y_* \neq Z$.) Therefore, this setting is rather far away from the trivial case of compatible data. We examine the results for different norms on the spaces Z and U and also look at the interplay with the regularisation parameter ω, which is kept nonzero here.

Our first collection of experiments deals with changes in the smoothness s of the observation space Z. The control space U has been fixed to L_2, which is equivalent to $t = 0$. We provide results for three different strengths of regularisation, namely $\omega = 1, \frac{1}{10}, \frac{1}{100}$. They are shown in Figure 6.9. For each setting of ω, we created one row with two graphs, again one for the state y on the left and one for the control u on the right.

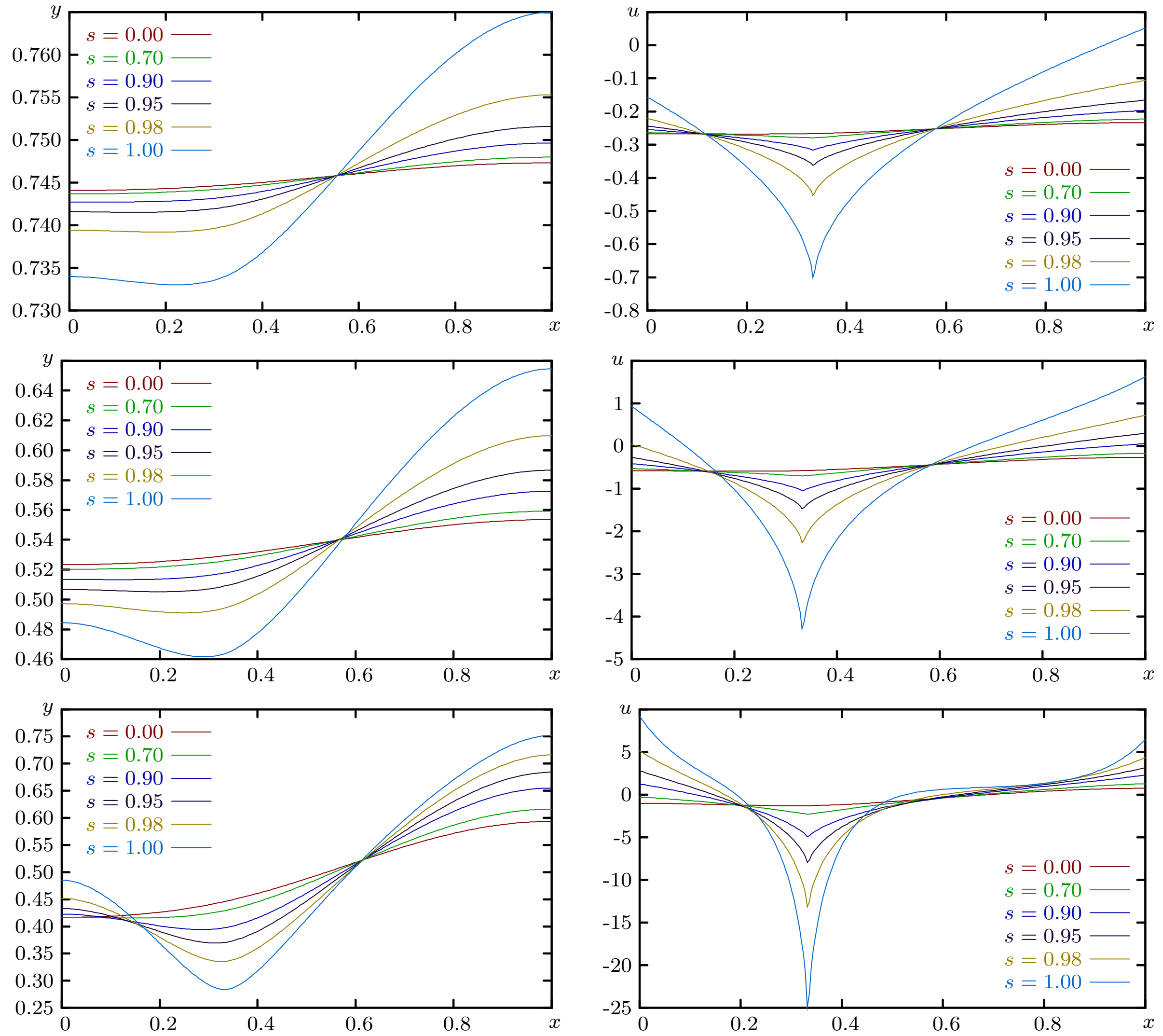


Figure 6.9: We show the results for $f \equiv 1$ and y_* as in Figure 6.8 for varying observation space $Z = H^s$ and fixed control space $U = L_2$. The smoothness index s ranges from 0 to 1, where we have chosen 6 discrete values. The state y is always shown on the left hand side, and the associated control u on the right. The three rows from top to bottom correspond to three different settings for the regularisation, namely $\omega = 1$ (top), $\omega = \frac{1}{10}$ (middle) and $\omega = \frac{1}{100}$ (bottom).

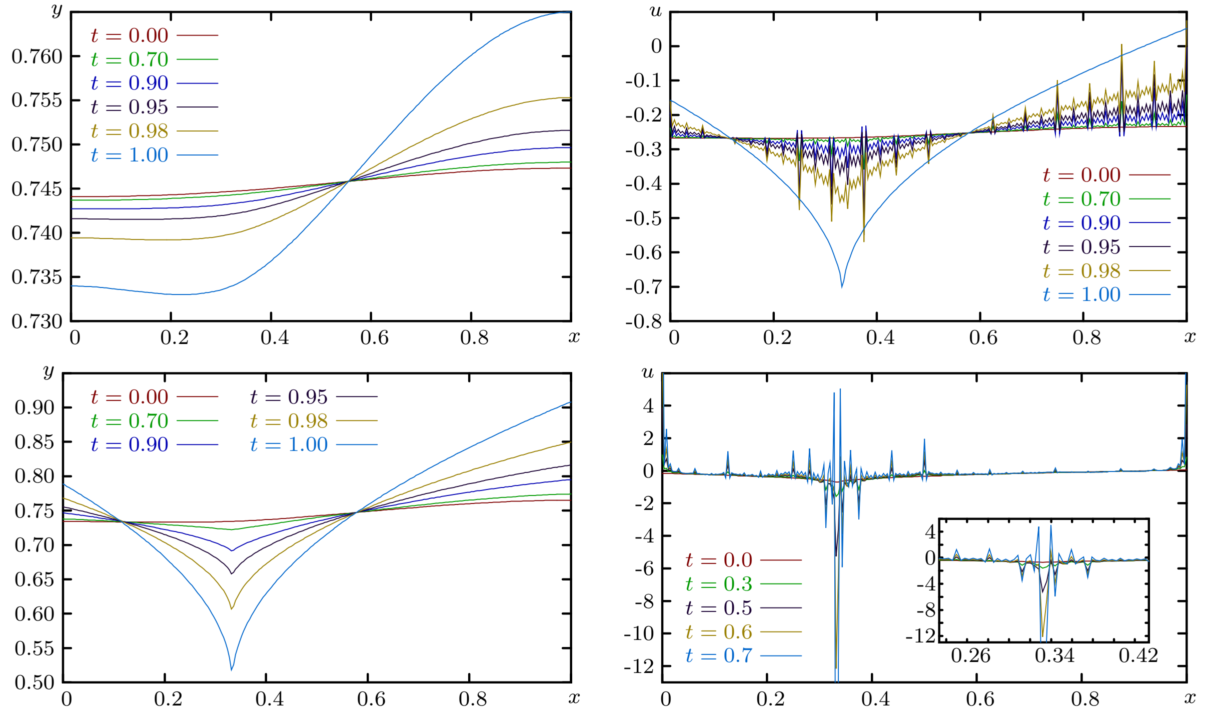


Figure 6.10: In this series of graphs we have fixed the observation space to $Z = L_2$ (top row) and $Z = H^1$ (bottom row), respectively. The control space is varied between $t = 0$ and $t = 1$ in discrete steps, while the regularisation is set to $\omega = 1$. In each row, the state y is shown on the left hand side and the control u on the right. The small picture on the bottom right shows a zoom of the region around the singularity. As the control u belongs to spaces of negative order, it is generally not smooth. The spikes occur for $t \notin \mathbb{Z}$ with $s = 0$ (top), and for $t > 0$ with $s = 1$ (bottom). The character of the oscillations is very different between the two examples, since the requirement $y_* \in Z$ is violated for the bottom row.

These experiments nicely show that there is a qualitative difference between changing s and changing ω. While for $s = 0$ a decrease in ω by several orders of magnitude is necessary to reproduce the kink towards zero present in y_*, this is quite easily accomplished by increasing the smoothness of $Z = H^s$ and keeping a comparably large ω. Moreover, u is smooth for s near zero, while the sharp singularity at $x = \frac{1}{3}$ only shows up for about $s > \frac{1}{2}$, almost independent of ω.

In a second series of experiments, we fix the observation space Z and allow for control spaces $U = (H^t)'$ with varying smoothness t. This means that the control u is generally contained in a negative order space, and we should not expect any smoothness in the classical sense. Instead, we are prepared to observe the same kind of spikes as in the earlier example of negative control smoothness. We made two runs, one for $Z = L_2$ and one for $Z = H^1$, the results of which are displayed in Figure 6.10.

The state y is smooth in the case $Z = L_2$, while it reproduces the kink for $Z = H^1$. This could not be accomplished with a control in L_2 for reasonable ω. However, the most significant difference compared to the previous examples with $U = L_2$ can be seen in the character of the control. For fractional $t \in (0, 1)$, it exhibits the dyadically arranged spikes. Again, they are largest for t very close to 1 and independent of the target accuracy ϵ of our numerical scheme. The most important observation regards the magnitude of the oscillations. The spikes in the non-conforming example with $s = 1$ are significantly larger and much less regular than in the case $s = 0$ or the example studied in the previous section. This indicates that the numerical scheme is indeed forced to work outside its range of specifications here.

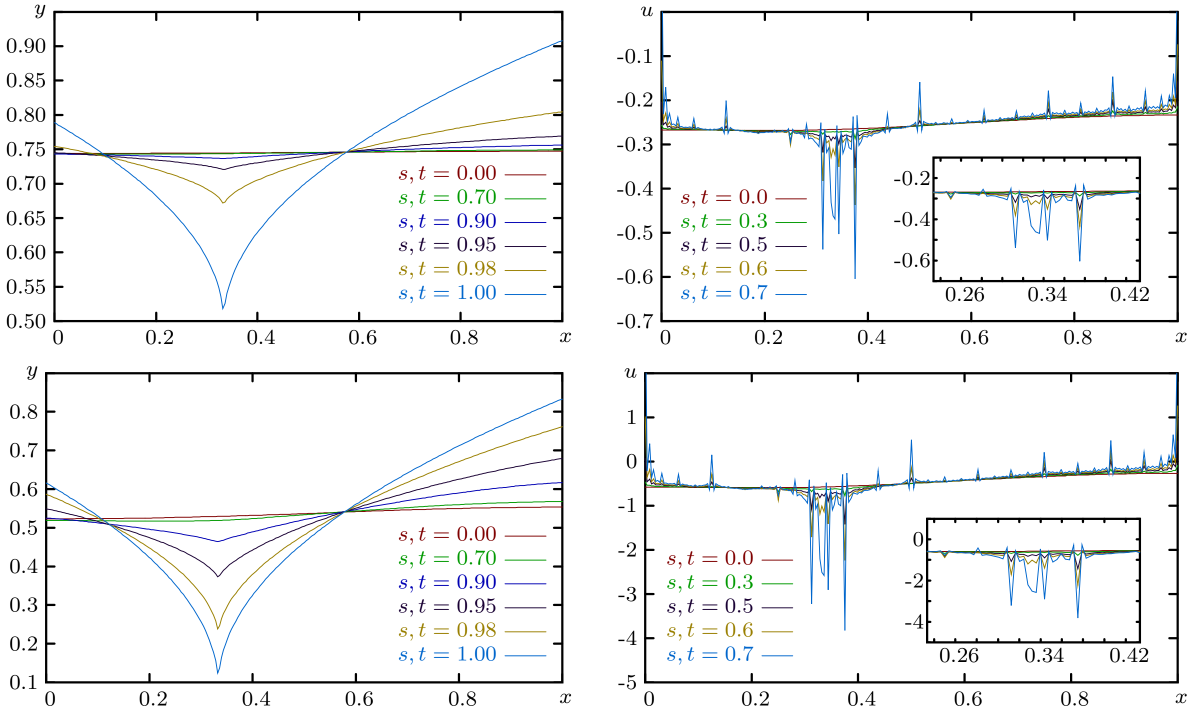


Figure 6.11: Here we have varied Z and U simultaneously, i.e., we have fixed $s = t$. We have set $\omega = 1$ in the top graph and $\omega = \frac{1}{10}$ in the bottom graph. This experiment spans the whole range from L_2 norms to the natural norms. For $s = t = 1$, the shape of y resembles very much the target observation y_*. The control can only be drawn reasonably for small to medium values of s and t, as it gets very large towards $s = t = 1$. The small image in the control graph contains a zoom around the singularity. These results demonstrate very clearly that the regularisation ω acts as a pure scaling factor and does not affect the character of the solution very much. In fact, the control around the singularity looks exactly the same for both values of ω, apart from the scaling.

A second important observation concerns the comparison of the first rows of the examples in Figure 6.9 and Figure 6.10. Both have been created with the same $\omega = 1$. In the first case, we fixed $t = 0$ and varied $s \in [0, 1]$, while the roles of s and t are reversed in the second case. The states y coincide perfectly for equal $s + t$, and the control is also the same provided that the spikes in Figure 6.10 are removed by an averaging process. This motivates the conjecture that only the difference between the smoothness indices of Z and U matters, and not the absolute value. A similar behaviour has also been observed in [118].

Finally, we combine the variations in the observation and control spaces. To this end, we set $Z = U'$ and traverse the whole spectrum $s = t = 0, \ldots, 1$. We do this for the two values $\omega = 1$ and $\omega = \frac{1}{10}$, respectively. The graphs are displayed in the rows of Figure 6.11. This approach spans between the two extreme cases, namely the classical setting with $s = t = 0$ and the natural norms with $s = t = 1$. Consequently, the results also change rather extremely, from almost constant functions for L_2 norms to a sharply kinked state y and a very irregular control u for natural norms. The highly irregular control can again be interpreted as a result of the deliberate violation of well-posedness with the selection $y_* \notin Z$.

The two graphs in Figure 6.11 for different ω are very similar in shape, and differ mostly in the scaling. Even the spikes around the singularity coincide. This illuminates the interplay between the scale-independent regularisation ω and the role of the smoothness indices s and t. While the latter parameters can be used to influence the character of state and control, variations in ω affect mostly the scaling.

We once more conclude the discussion of this series of experiments with selected iteration histories. They

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						1.62e-04	4.86e-04	1.50e-05	7.34e-05
4	8.48e-06	4	3	0	7	1.34e-04	2.61e-04	6.88e-06	1.14e-05
5	4.29e-06	3	4	2	6	2.68e-05	1.14e-04	3.22e-06	3.41e-06
6	2.90e-06	2	5	2	6	1.93e-05	5.82e-05	2.37e-06	2.37e-06
7	6.81e-07	3	3	2	5	4.07e-06	2.76e-05	4.23e-07	4.24e-07
8	6.84e-07	1	10	5	7	3.32e-06	1.40e-05	4.24e-07	4.24e-07
9	2.58e-07	2	5	2	5	2.82e-06	7.38e-06	2.27e-07	2.27e-07
10	1.78e-07	2	5	2	5	6.59e-07	3.47e-06	1.68e-07	1.68e-07
11	3.28e-08	3	3	2	4	5.54e-07	1.79e-06	5.26e-08	5.26e-08
12	3.21e-08	1	10	5	7	2.38e-07	8.82e-07	5.26e-08	5.26e-08
13	2.08e-08	2	5	2	5	2.03e-07	4.69e-07	4.58e-08	4.58e-08
14	1.20e-08	4	2	1	4	5.51e-08	2.13e-07	1.51e-08	1.51e-08
15	4.30e-09	3	3	2	4	4.47e-08	1.02e-07	1.02e-08	1.02e-08
16	2.72e-09	4	2	1	4	2.14e-08	2.14e-08	2.49e-09	2.49e-09

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						2.39e-03	7.05e-03	1.60e-03	2.81e-02
4	9.57e-06	9	10	5	9	6.92e-04	3.33e-03	9.42e-04	1.42e-02
5	4.18e-06	10	10	4	10	2.03e-04	1.63e-03	4.84e-04	7.10e-03
6	2.12e-06	10	8	3	10	6.20e-05	8.04e-04	2.42e-04	3.55e-03
7	6.83e-07	11	7	3	9	2.04e-05	4.00e-04	1.21e-04	1.78e-03
8	5.85e-07	10	6	3	9	6.78e-06	1.99e-04	6.06e-05	8.87e-04
9	2.64e-07	10	5	2	9	3.62e-06	9.96e-05	3.03e-05	4.43e-04
10	1.30e-07	10	5	2	9	8.06e-07	4.98e-05	1.51e-05	2.21e-04
11	6.47e-08	10	5	2	9	4.58e-07	2.49e-05	7.56e-06	1.10e-04
12	3.31e-08	10	4	2	9	3.05e-07	1.24e-05	3.76e-06	5.43e-05
13	1.63e-08	10	4	2	9	2.28e-07	6.17e-06	1.85e-06	2.61e-05
14	8.16e-09	10	4	2	9	1.44e-08	3.01e-06	8.37e-07	1.16e-05
15	3.62e-09	10	4	2	9	1.08e-08	1.35e-06	3.68e-07	4.53e-06
16	1.61e-09	10	4	2	9	7.64e-09	7.64e-09	1.19e-09	1.19e-09

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						5.18e-08	1.41e-00	1.34e-01	8.13e-01
4	2.39e-06	15	15	13	13	1.35e-04	1.35e-00	1.26e-01	7.78e-01
5	1.87e-06	18	19	15	15	5.72e-05	1.29e-00	1.19e-01	7.44e-01
6	1.21e-06	19	20	16	16	2.09e-05	1.22e-00	1.19e-01	7.10e-01
7	5.57e-07	21	21	16	16	5.78e-06	1.15e-00	1.19e-01	6.74e-01
8	2.79e-07	22	21	16	16	2.23e-06	1.08e-00	1.19e-01	6.37e-01
9	1.31e-07	23	21	16	16	9.91e-07	1.00e-00	1.19e-01	5.97e-01
10	8.81e-08	24	21	16	16	3.86e-07	9.21e-01	1.19e-01	5.55e-01
11	3.36e-08	25	21	16	16	1.60e-07	8.28e-01	1.18e-01	5.08e-01
12	1.58e-08	25	21	16	16	5.85e-08	7.24e-01	1.18e-01	4.58e-01
13	4.93e-09	28	21	15	15	4.54e-08	6.05e-01	1.15e-01	4.00e-01
14	4.47e-09	28	21	16	16	2.11e-08	4.61e-01	1.08e-01	3.31e-01
15	1.39e-09	29	21	16	16	2.89e-08	3.03e-01	7.90e-02	2.35e-01
16	8.57e-10	29	21	16	16	1.52e-08	1.52e-08	4.26e-10	4.26e-10

Table 6.5: We display the residuals, iteration numbers and errors for the non-conforming example $f \equiv 1$ and $y_* = h_2$ in one dimension. We have fixed $\omega = 1$. The topmost table contains the data for $s = t = 0$. In the middle, the parameters are $s = 1$, $t = 0$, and the last table contains values for $s = t = 1$.

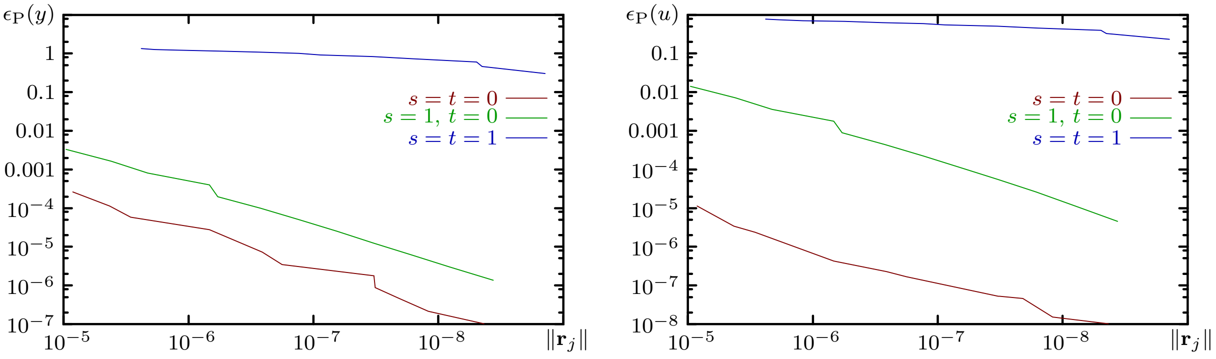


Figure 6.12: These convergence plots correspond to $f \equiv 1$ and $y_* = h_2$. The examples with $t = 0$ show a rate of 1, while the example with $s = t = 1$ exhibits very slow convergence.

are listed in Table 6.5, for the choices (from top to bottom) $s = t = 0$, $s = 1$ and $t = 0$ and finally $s = t = 1$. The first observation is that the L_2 case is cheapest in terms of computing time, while the case of natural norms is most demanding. We have presumed this at the beginning, and it has been confirmed by all examples we have encountered so far.

The second important observation deals with the convergence rates. The first two tables show that the number of iterations is indeed independent of the level of resolution, and even somewhat decreasing in the first run, just as in all previous examples. As can also be seen in the convergence plots in Figure 6.12, the errors in y and u reduce steadily with rate 1.

However the third example indicates a slowdown in convergence with increasing levels of resolution. The number of outer iterations constantly increases with j here. Moreover, the error against the high accuracy solution does not converge to zero but stays at about 0.1. The rates in both the state and the control are below 0.2. This behaviour should not come as a surprise though since the functional (5.2.10) is no longer well-defined in this particular case. Considering the shape of u for the most extreme case $s = t = 1$, the errors seem to be caused by the higher frequencies, which explains the reasonable size of the restricted error in the state $\epsilon_{\mathrm{R}}(y)$, and of all errors for the highest level $J = 16$.

We interpret these results as an indication of the robustness of the method because of the following observations.

- Already the case $s = 1$, $t = 0$ implies that the target observation is no longer in the observation space, but still the convergence rate is perfectly linear.
- Although the case $s = t = 1$ corresponds to the most extreme setting for a non-conforming y_*, the number of inner iterations remains asymptotically constant, while the increase in outer iterations is only moderate.

Our algorithm is thus capable of handling non-conforming data in a rather stable manner.

Summary

From these experiments in one dimension, our main conclusions are the following.

(i) The algorithm yields exactly the predicted results for the special cases of constant data and compatible data. It also satisfies the runtime characteristics predicted by the theory for the general case, namely an overall complexity of at most $\mathcal{O}(N_J)$ in time and memory. The efficiency of the inexact conjugate gradient algorithm in conjunction with the nested iteration strategy becomes immediately obvious in all results.

- The linear complexity is reflected in the amount of inner and outer iterations on each level, which is asymptotically constant, and even slowly reducing with increasing resolution in some examples.
- The standard case with $s = t = 0$ is computationally cheaper than the case of natural norms with $s = t = 1$. The difference manifests in a constant factor only, not in the rate of convergence.
- The numerical scheme deals well with the special case $s = t = 1$ and $\omega = 0$, which is the only allowed combination of smoothness indices for vanishing ω. For combinations inducing ill-posed systems, the scheme still converges, but the results are not reliable.
- When the parameters and the data are chosen in a way which violates the prerequisites of the general control problem discussed in Chapter 5, we observe a degradation in performance with increasing level of resolution. The algorithm is robust in the sense that it still terminates with reasonable results.

(ii) The parameters s and t, which are related to the smoothness of the state and the control, act in a different manner than the regularisation ω, which acts as a pure weight which equally affects all scales. Both approaches in modelling are largely independent.

- High regularity for Z (and also low regularity for U) leads to a good accordance in shape between y and the target observation y_*. Conversely, low regularity for Z neglects most of the features of the target y_* and leads to an average tracking in the L_2 sense.
- Negative regularity for U, i.e., values for $t > 0$, introduce oscillations in the control. These are artifacts which stem from the wavelet discretisation of fractional negative order spaces. They grow with t, as they are increasingly less penalised by the norms in $(H^t)'$, with coefficients which are always in ℓ_2.
- The solution y only depends on the *difference* between the observation and control spaces, i.e., the value of $s + t$. For the control, this holds only when the oscillations are removed by an averaging process. From a practical point of view, this motivates a fixation $U = L_2$, i.e., $t = 0$, while only s is varied. Consequently, the oscillations for the control are avoided, however, the maximal difference in smoothness between the state and the control is cut in half.
- The weight ω does not seriously affect the character of the solution, but merely acts as a global scaling.

These facts demonstrate the relatively rich behaviour of the control problem, considering the fact that the model is completely linear. We expect that the results in more than one dimension will be very similar in tendency.

6.3.3 Higher Dimensions

We now discuss some results in more than one dimension. The algorithm is completely unchanged, only the approximation and plotting of functions and the application of the various matrices needs to be extended by tensor products. We also continue to work with homogeneous Neumann boundary conditions and employ a similar right hand side f as in the previous experiments. We expect that the effects of the various parameters on the character of the solution and control are similar to the results which we discussed thoroughly in one dimension.

In the computations with uniform discretisation, the amount of available memory restricts the maximum resolution of the numerical scheme. A second bottleneck is posed by the exact solution on the coarsest level. This is no issue asymptotically, but the complexity of $\mathcal{O}(N_{j_0}^3)$ becomes practically important already in three dimensions. As discussed in Section 6.2.3, this can be alleviated by a sensible stopping criterion on the conjugate gradient method in its function as an exact solver. However, even with the fully exact solution a standard PC permits simulations with several million unknowns in three dimensions, that is, with 128^3 degrees of freedom per variable.

Example 6.5 – Two Dimensions

For our calculations in two dimensions, we again choose the parameter $\nu = \frac{1}{100}$. The choice of wavelet basis is the same as in the one-dimensional case. A standard PC permits calculations up to the maximum level $J = 10$, corresponding to roughly 10^6 degrees of freedom for each function, which we use for the iteration histories. To demonstrate the qualitative behaviour and create the plots, the level $J = 5$ is accurate enough.

Tested with constant data, our experiments indeed confirm the results from Figure 6.1. Thus, the method is in perfect agreement with the theory for constant data also in two dimensions. For all subsequent experiments, we selected $f \equiv 1$ and a generalisation of the target observation $y_* = h_2$ from (6.3.5),

$$y_* = \sqrt{\left| x - \left(\frac{1}{3}, \frac{1}{3}\right)^T \right|}. \tag{6.3.6}$$

We present three collections of results. The first is meant to illustrate the general behaviour of the method in two dimensions, and contains three different examples for standard parameter values. The second and third set deal with varying ω, once for the L_2 setting with $s = t = 0$, and secondly for the parameters $s = 1$, $t = \frac{1}{2}$. In all cases, the method yields results which are analogous to the computations in one dimension.

The first set of results deals with varying observation space Z and fixed control space $U = L_2$, which corresponds to $t = 0$. We have selected the standard cases $s = 0$ and $s = 1$, each with $\omega = 1$, and also $s = 1, \omega = \frac{1}{10}$. The state and the control for each case are shown in Figure 6.13. As in the one-dimensional case, the solution y is very smooth, and it only shows a very gentle deepening towards the singularity. The kink downward in the control only shows up for $s > 0$. Again, the change in ω introduces a different scaling and does not affect the shape of the functions y and u. The homogeneous Neumann boundary conditions for the state y are reproduced.

Next, we examine different weights of the regularisation, namely $\omega = \frac{1}{10}, \frac{1}{100}, \frac{1}{1000}$. As in the one-dimensional case, we expect that effects on the shape of the state only show up for very small ω. Our first set of experiments in this respect deals with the standard setting $U = Z = L_2$, i.e., $s = t = 0$. The results are displayed in Figure 6.14. Indeed, a qualitative change in the functions y and u can only be seen for $\omega = \frac{1}{1000}$. However, even for this nearly degenerate regularisation, the singularity in the target observation y_* is not reproduced, and we can only find a subtle and smooth bend downward. This picture changes for the combination of parameters $s = 1$ and $t = \frac{1}{2}$, which is displayed in Figure 6.15. In this scenario, the singularity does appear in the state y for small ω. This comes at the cost of a sharply spiked control, which is also less regular along the boundary.

This comparison of different smoothness indices for a given range of ω demonstrates again that the shape of the state and the control strongly depends on s and t, and only weakly on ω, while the overall scaling is mainly controlled by ω. Furthermore, the measurement of u in $H^{-\frac{1}{2}}$ does not yet produce the spikes, which only appear for spaces which are closer to H^{-1}.

As in the one-dimensional case, we close the discussion of numerical results with the iteration histories for the residuals, errors and iteration counts. The data are listed in Table 6.6, with a finest level of $J = 10$. The results for the standard case, namely $s = t = 0$, are very similar to those in one dimension, showing clear $\mathcal{O}(N_J)$ behaviour. We see that the number of inner iterations for the two state and adjoint equation increases slightly, while the number of outer iterations reduces. The total computational cost given by their product is constant.

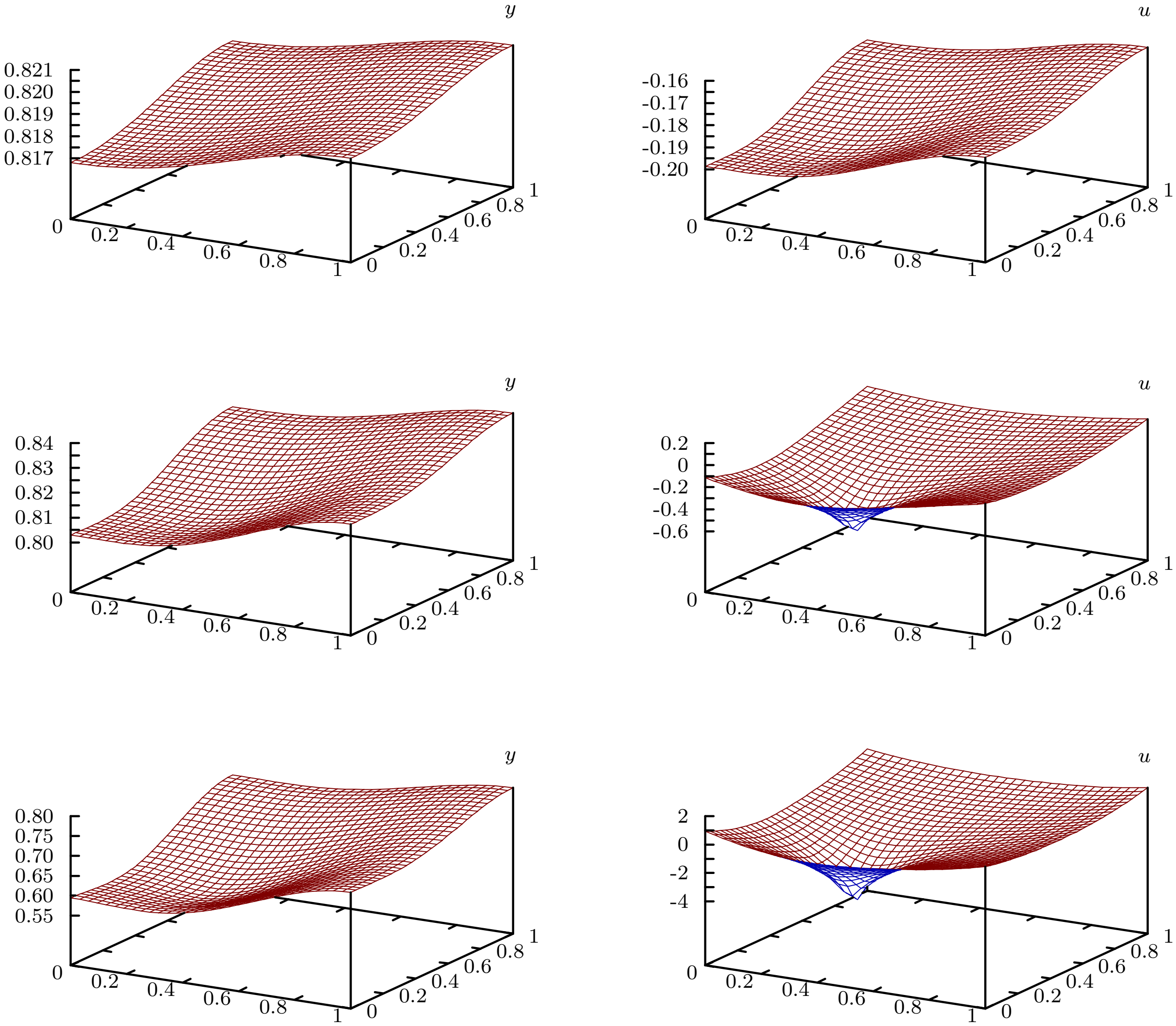


Figure 6.13: We display three sets of results in two dimensions. All were obtained with a resolution of $J = 5$, as this is most sensible for the graphical representation. The first row shows the state and the control for $s = t = 0$ and $\omega = 1$. The middle row belongs to the parameters $s = 1$, $t = 0$, $\omega = 1$, while the last row corresponds to $s = 1$, $t = 0$, $\omega = \frac{1}{10}$. These results are largely analogous to the one-dimensional situation shown in Figure 6.9.

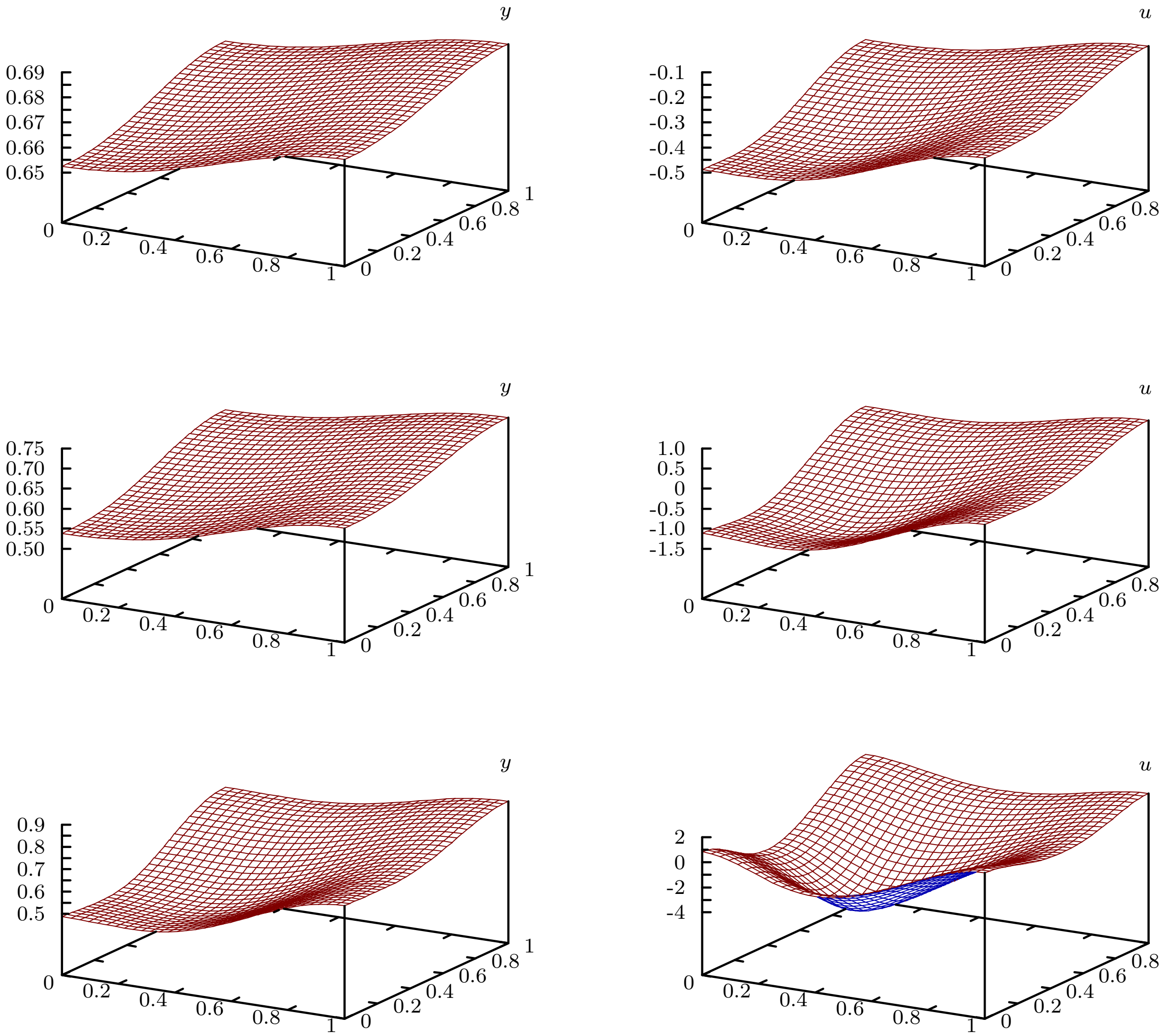


Figure 6.14: We show three sets of results for $s = t = 0$*. From top to bottom, we set* $\omega = \frac{1}{10}, \frac{1}{100}, \frac{1}{1000}$*. With decreasing* ω*, the state* y *gets larger in the* L_∞ *norm, but does not mimic the singularity present in* y_**. The control is also very smooth.*

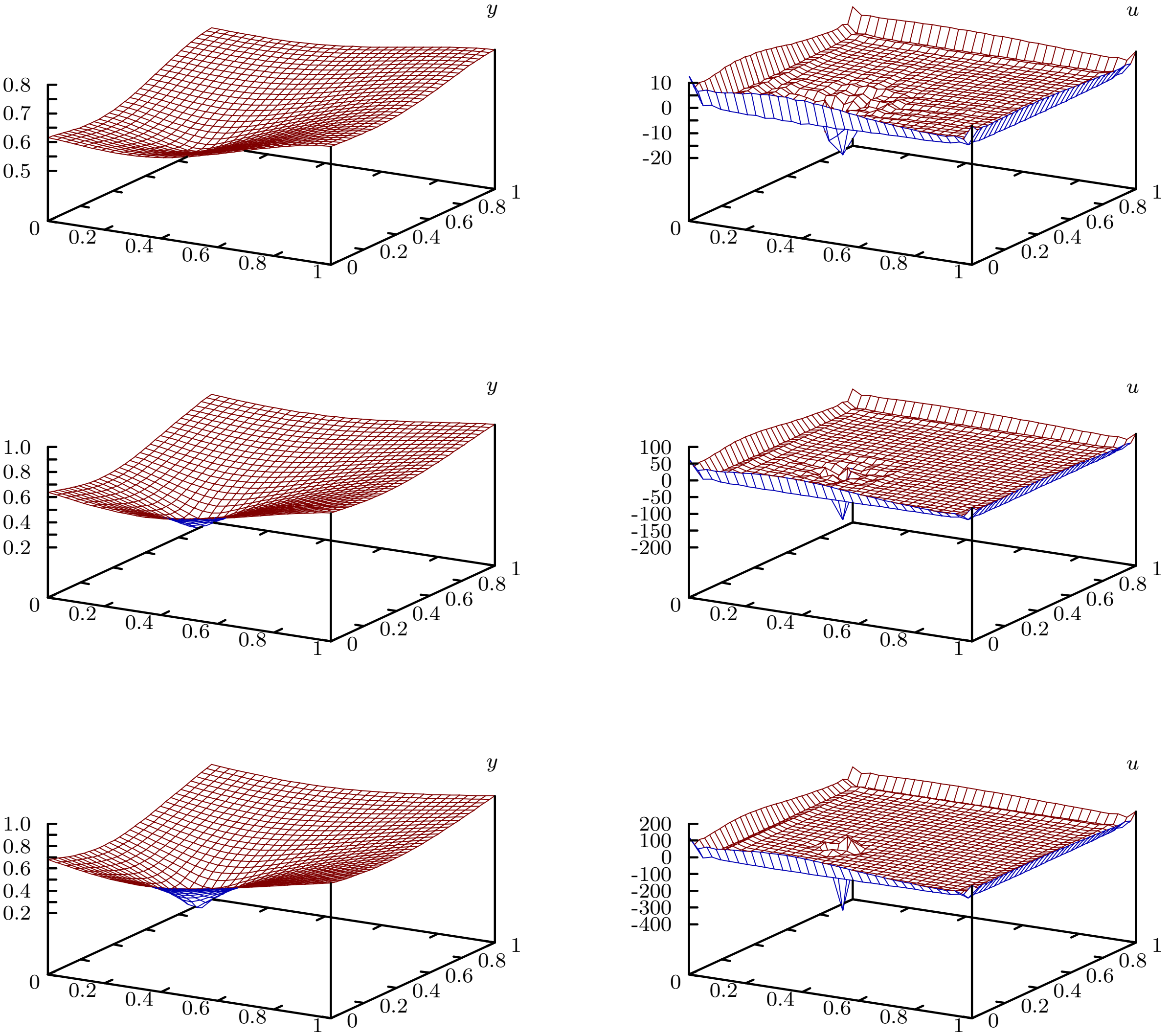


Figure 6.15: These graphs correspond to the setting $s = 1$, $t = \frac{1}{2}$. *As in Figure 6.14, the three rows have been obtained with (from top to bottom)* $\omega = \frac{1}{10}, \frac{1}{100}, \frac{1}{1000}$. *The change in the regularisation leads to controls* u *which are similar in shape, although they have different scales. The shape of the state* y *is only weakly influenced by the drastic reduction in* ω.

j	$\lVert\mathbf{r}_j\rVert$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						1.60e-04	3.73e-04	1.29e-05	3.51e-05
4	7.87e-06	6	4	1	18	1.41e-04	2.17e-04	5.11e-06	5.90e-06
5	3.89e-06	5	4	1	19	1.67e-05	8.34e-05	1.77e-06	1.79e-06
6	2.02e-06	4	4	2	17	1.43e-05	4.30e-05	7.68e-07	7.69e-07
7	1.13e-06	2	7	3	16	1.30e-05	2.39e-05	7.19e-07	7.19e-07
8	4.76e-07	5	3	1	15	1.42e-06	9.90e-06	1.85e-07	1.85e-07
9	2.25e-07	3	5	2	16	1.12e-06	4.51e-06	1.33e-07	1.33e-07
10	1.32e-07	3	5	2	16	9.48e-07	9.48e-07	1.10e-07	1.10e-07

j	$\lVert\mathbf{r}_j\rVert$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						2.32e-03	6.95e-03	2.56e-03	1.70e-02
4	4.00e-05	11	19	10	11	7.27e-04	3.39e-03	1.48e-03	1.04e-02
5	1.26e-05	12	18	9	12	2.55e-04	1.74e-03	7.56e-04	6.94e-03
6	6.71e-06	12	18	9	12	1.06e-04	9.12e-04	3.77e-04	4.64e-03
7	4.22e-06	12	17	9	13	5.17e-05	4.79e-04	1.86e-04	3.12e-03
8	1.95e-06	13	16	8	13	2.46e-05	2.45e-04	8.82e-05	2.02e-03
9	7.40e-07	14	15	8	13	1.19e-05	1.12e-04	3.44e-05	1.16e-03
10	5.98e-07	14	14	7	13	4.75e-07	4.75e-07	1.70e-07	1.70e-07

Table 6.6: We display the residuals, iteration numbers and errors for two dimensions. We have set $J = 10$, $\omega = 1$ and chosen $f \equiv 1$ and $y_* = h_2$ from (6.3.6). The upper table contains the data for $s = t = 0$. The lower table corresponds to the parameters $s = 1$, $t = \frac{1}{2}$.

The second setting with $s = 1$ and $t = \frac{1}{2}$ has been chosen right in between the selections $t = 0$ and $t = 1$ which have been examined in one dimension. As a result, the state is still computed with about the same accuracy as for the standard situation, while the error of the computed control u is larger with values around 10^{-3}. The numbers of outer iterations grow slightly in this example, indicating that the regularity of the solutions lies outside the allowed range, which we interpret as a consequence of the singular choice of y_*. The inner iteration counts are again asymptotically constant. The total amount of iterations is larger than for the standard case, which we can also see from the computation times in the following table.

Parameters	$s = t = 0$	$s = 1, t = 0.5$
Runtime	177s	1042s

Example 6.6 – Three Dimensions

We complete this chapter with iteration histories computed in three dimensions at a resolution of $J = 7$, as before with an extra accuracy of $\nu = \frac{1}{100}$. This corresponds to roughly 2 million degrees of freedom per variable. The target observation has been extended to three dimensions,

$$y_* = \sqrt{\left| x - \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T \right|}. \tag{6.3.7}$$

We find that the results resemble closely those computed in two dimensions in the previous section. The iteration histories are displayed in Table 6.7. Again, the total iteration numbers per level decrease in the standard case $s = t = 0$, while they stay constant for $s = 1$, $t = \frac{1}{2}$. Although the asymptotic regime is probably not yet reached with a highest level of 7, the complexity appears again to lie in $\mathcal{O}(N_J)$. An interesting observation which can be made here is that the number of inner iterations for the Riesz operator becomes large compared to the two other inner systems in the case $s = t = 0$. This has already

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						1.41e-04	2.92e-04	1.13e-05	2.36e-05
4	6.09e-06	10	9	1	49	1.27e-04	1.78e-04	3.46e-06	3.79e-06
5	3.25e-06	10	7	1	58	1.11e-05	6.14e-05	9.47e-07	9.53e-07
6	1.71e-06	7	6	1	57	1.00e-05	2.86e-05	5.03e-07	5.03e-07
7	8.80e-07	6	6	1	53	9.19e-06	9.19e-06	3.72e-07	3.72e-07

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	$\epsilon_{\mathrm{R}}(y)$	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{R}}(u)$	$\epsilon_{\mathrm{P}}(u)$
3						2.04e-03	4.71e-03	1.01e-03	1.28e-02
4	1.75e-05	20	26	15	24	6.28e-04	2.31e-03	4.19e-04	8.48e-03
5	7.12e-06	22	27	14	29	2.03e-04	1.18e-03	1.49e-04	5.53e-03
6	4.77e-06	21	28	14	32	7.53e-05	5.42e-04	4.15e-05	3.20e-03
7	1.98e-06	23	26	14	31	5.88e-06	5.88e-06	4.50e-07	4.50e-07

Table 6.7: We list the iteration histories in three dimensions. The parameters were $J = 7$ and $\omega = 1$ for both runs. The top table corresponds to the standard case with $s = t = 0$, while the bottom table contains the results for $s = 1$, $t = \frac{1}{2}$.

been visible to some extent in the two-dimensional case. We attribute this to dimension-dependent shifts in the eigenvalues of the operators.

It seems that also in the three-dimensional case the ℓ_2 errors in the state and the control decrease with order 1. As in all previous examples, the L_2 setting is the cheapest in terms of computing time, whereas the non-conforming case is much more expensive, as can be inferred from the numbers in the following table.

Parameters	$s = t = 0$	$s = 1$, $t = 0.5$
Runtime	3502s	12261s

Summary

We conclude with the perception that our results are largely independent of the dimension, both in quality and in asymptotic behaviour. Due to the slight increase of the condition numbers of the matrices with the dimension, the iteration counts get larger while the essential characteristics of solution and control remain unchanged. We find independently of the dimension that the regime of optimal linear complexity covers the full space of allowed parameters. In situations when the data are less regular than demanded by the norms occurring in the functional, the convergence becomes slightly slower, and the accuracy of the control degrades in some cases.

The outer-inner inexact conjugate gradient scheme in combination with the nested iteration approach delivers optimal $\mathcal{O}(N_J)$ performance in all cases conforming to the theory, and appears robust beyond. We find in our experiments that the inexactness of the conjugate gradient solvers does not affect the convergence rate which is strictly linear.

The introduction of fractional Sobolev norms in the target functional which is made possible by the wavelet approach offers an additional freedom in modelling. The parameters s and t indeed act scale-dependently and allow for a variation in the character of the tracking of the state and the smoothness of the control. The standard L_2 tracking enforces only a match on average which neglects most of the features of the state. In contrast, increasing the smoothness in the norm of observation tends to influence the shape of the solution much more effectively. The weight parameter ω clearly represents a frequency-independent scaling without much influence on the shape of the state and the control.

The computational work is lowest for the standard case $s = t = 0$ and grows with increasing s and t. This can be explained by the fact that the operators $\mathbf{T}$ and $\mathbf{E}$ act as smoothers for small s and t, and gradually transform into the identity operator in the limit $s = 1$ or $t = 1$, respectively. In other words, the case of natural norms can be interpreted as the most extreme in the sense that the full range between the space Y and its dual Y' is exploited.

Enabling negative fractional norms for the control space introduces artifacts in the control in the form of dyadically arranged spikes. When removed by an averaging process, we find that the smoothness indices act merely via their sum $s + t$. Apart from these side effects of the construction of Riesz operators for negative fractional order spaces, the method completely meets our expectations concerning the shape and character of the results and the computational complexity. Our modifications and improvements with respect to the construction of wavelets, the conditioning of operators and the numerical evaluation of norms which we laid out in earlier parts of this document have proved highly beneficial with respect to the quality of the results and led to a considerable reduction in overall computing time.

Chapter 7

An Adaptive Wavelet Method for the Control Problem

7.1 Introduction

In the previous chapters we have established a rigorous theoretical basis for the numerical solution of linear-quadratic elliptic optimal control problems in uniform wavelet discretisation. We have then devised an implementation and performed numerical computations for different sets of parameters and data. The effort concerning computation time and memory usage is proportional to the number of degrees of freedom, which can be two million for each of the state, the adjoint and the control variable on a standard PC.

On the one hand, developing a numerical scheme with these characteristics is challenging, and the amount of data which can be processed by such a fast algorithm is impressive. On the other hand, the question arises whether this large amount of computational variables is really necessary to produce results of a certain accuracy. This motivates the investigation of adaptive strategies, which aim to use as few degrees of freedom as possible, potentially selected non-uniformly over the space of coefficients, while obtaining a sufficiently precise numerical solution. Thus adaptivity offers the perspective to save a large fraction of computational resources without sacrificing accuracy.

In the finite element context, adaptive methods are based on the local refinement of triangles or rectangles in two dimensions, depending on the choice of finite element basis, and tetrahedrons or cubes in three dimensions. Typically, a-posteriori error estimators are used to determine which elements are refined in each iteration of an outer loop. These should be *reliable* in the sense that they do not miss important contributions to the error, and *efficient* in order not to refine where it is not needed. For a general survey on this subject we refer to [141]. Examples from current research are estimators for parabolic PDEs [110] and parameter identification problems [9] and a framework for the Navier Stokes equations based on duality [7,8]. Obstacle problems and mixed finite element discretisations have been discussed in [96,97]. A conceptually different ansatz is pursued in meshless methods, where the degrees of freedom may be arbitrarily distributed over the domain, see e.g. [76,77,127]. When the polynomial degree of the basis functions is also varied adaptively, the so-called *hp*-methods come into play [75,91,123].

While adaptive finite element methods for the solution of partial differential equations have been investigated and used for many years, analytical results on *convergence* have been obtained only recently for the case of a single elliptic PDE. Namely, algorithms have been proposed which guarantee that each cycle of solution, estimation and local refinement actually reduces the error by a constant factor [62,114,115]. These convergence rates have been quantified in [12,130], using ideas from nonlinear approximation.

Wavelets inherently offer a rigorous approach to adaptivity which provides reliability, efficiency and convergence in a unified framework, solidly based on nonlinear approximation theory. An adaptive wavelet Galerkin method of this type has been proposed for elliptic problems [39], which has also been verified to work in practise [5,6]. It has been extended to saddle point formulations [40] and a large class of nonlinear variational problems [41].

The concept of adaptive wavelet approximations is rather flexible. It has been used in various numerical procedures, for example for adaptive fitting of scattered data [33,34], which in turn can be employed for the numerical solution of hyperbolic conservation laws [32]. A wavelet-based finite difference method for the Navier Stokes equations has been developed in [71,99].

A particular advantage of adaptive wavelet Galerkin methods for stationary variational problems is their strong background in nonlinear approximation theory. This allows to prove convergence of the adaptive algorithm, and to specify convergence rates. Moreover, these rates are provably *optimal* in the sense that the effort for the numerical solution of the whole variational problem is proportional to the number of degrees of freedom needed for the approximation of the solution. The rate of approximation, and hence the rate of convergence is controlled by the Besov smoothness of the solution. Since Besov spaces are larger than Sobolev spaces of equal smoothness, optimality of these adaptive wavelet methods can be proved for a larger class of solutions, including strongly singular functions, than with uniform discretisations. We call this the *ideal* adaptive wavelet approach.

Recently, the adaptive wavelet framework has been extended in [48] to the class of linear-quadratic optimal control problems introduced in Chapter 5, providing proofs of optimal convergence rates. This approach is unique in the sense that the three principal variables for the state, the adjoint and the control are resolved with independent sets of coefficients. In particular, the adaptive finite element method for the control of the Navies Stokes equations from [8] uses the same grid for all variables, and does not contain statements on convergence or convergence rates.

In this chapter, we present a first realisation of a fast and fully adaptive method for such optimal control problems. We achieve this by incorporating the algorithmic ideas from adaptive wavelet methods for elliptic PDEs and control problems into our existing algorithm `nIIcG/2` in uniform discretisation. Instead of using residual-based error estimators in the outermost loop, we use the wavelet norm equivalences at several crucial points in the innermost layer of the algorithm to control the errors of truncated wavelet representations. In particular, we resolve different variables with different sets of coefficients. We retain the two-layer conjugate gradient solver from Chapter 6, as it converges substantially faster than the Richardson iterations which are used in the ideal adaptive wavelet strategies from [39, 48]. Since at present the convergence rate of adaptive wavelet methods involving Krylov subspace methods is not known theoretically, we undertake systematic numerical studies and examine the convergence behaviour and the adaptive efficiency. We find experimentally that the resulting algorithm δ-`AnIIcG/2` significantly reduces the memory requirements compared to the uniform scheme, and converges with a superior rate for several combinations of parameters and data.

7.2 Nonlinear Wavelet Approximation

The wavelet representation of a function is represented by a vector over an infinite-dimensional index set. In numerical calculations, only a finite subset of this vector can be stored. Adaptivity in the wavelet context is realised by choosing an index set of given cardinality N in such a way that the approximation error is minimised. This is called (wavelet-best) N*-term approximation*, a technique from nonlinear approximation theory which has drawn considerable attention recently. We provide a short introduction here, largely following [39].

7.2.1 Weak ℓ_τ Spaces

Let $\Lambda = \{\lambda_i\}$ denote a finite-dimensional set of wavelet indices. For any coefficient vector $\mathbf{v} \in \ell_2$, the projector onto this index set is denoted by $\mathbf{P}_\Lambda$. The nonlinear space of vectors with at most N nonzero entries is defined as $\Sigma_N \coloneqq \{\ell_2(\Lambda) : \#\Lambda \leq N\}$, and the approximation error is given by

$$\mathcal{E}_N(\mathbf{v}) \coloneqq \inf_{\mathbf{w} \in \Sigma_N} \|\mathbf{v} - \mathbf{w}\| \,. \tag{7.2.1}$$

A best approximation to $\mathbf{v}$ from Σ_N is obtained by taking a set Λ_N with $\#\Lambda_N = N$ on which the absolute values of the entries $|v_\lambda|$ assume their N largest values. This set is generally not unique, but all such sets yield best approximations from Σ_N. The error can be formulated in terms of the projector,

$$\mathcal{E}_N(\mathbf{v}) = \|\mathbf{v} - \mathbf{P}_{\Lambda_N}\mathbf{v}\| \,. \tag{7.2.2}$$

It follows from the definition (7.2.1) that the error $\mathcal{E}_N(\mathbf{v})$ reduces as N gets larger. To quantify the rate of approximation, we introduce the exponent $\sigma \geq 0$ and let $\mathcal{A}^\sigma$ denote the set of all vectors $\mathbf{v} \in \ell_2$ such that the norm

$$\|\mathbf{v}\|_{\mathcal{A}^\sigma} \coloneqq \sup_{N \geq 0} (N+1)^\sigma \mathcal{E}_N(\mathbf{v}) \tag{7.2.3}$$

is finite, with the additional definition $\mathcal{E}_0(\mathbf{v}) \coloneqq \|\mathbf{v}\|$.

The set $\mathcal{A}^\sigma$ consists of all vectors which can be approximated with order $\mathcal{O}(N^{-\sigma})$ by the elements of Σ_N. It can be characterised by the *decreasing rearrangement* $\mathbf{v}^*$ of $\mathbf{v}$. For each $N \geq 1$, let v_N^* be the N-th largest of the entries $|v_\lambda|$, and let $\mathbf{v}^* := (v_N^*)_{N=1}^\infty$. Introducing a new parameter $0 < \tau < 2$, we let ℓ_τ^{w} denote the collection of all vectors $\mathbf{v} \in \ell_2$ for which the expression

$$|\mathbf{v}|_{\ell_\tau^{\mathrm{w}}} \coloneqq \sup_{N \geq 1} N^{\frac{1}{\tau}} v_N^* \tag{7.2.4}$$

is finite. In other words, we have $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$ if and only if $v_N^* \lesssim N^{-\frac{1}{\tau}}$ for all $N \geq 1$, which means that the entries of $\mathbf{v}$, sorted by size, decay with $\mathcal{O}(N^{-\frac{1}{\tau}})$. An alternative characterisation is given by

$$\#\{\lambda : |v_\lambda| \geq \epsilon\} \lesssim \epsilon^{-\tau} \,. \tag{7.2.5}$$

The smallest constant in these estimations is defined as $|\mathbf{v}|_{\ell_\tau^{\mathrm{w}}}^\tau$.

The space ℓ_τ^{w} is called *weak* ℓ_τ. It is a special case of a *Lorentz sequence space*. It holds that $\ell_\tau \subsetneq \ell_\tau^{\mathrm{w}} \subsetneq \ell_2$, and we may define the quasi-norm

$$\|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}} \coloneqq \|\mathbf{v}\|_{\ell_2} + |\mathbf{v}|_{\ell_\tau^{\mathrm{w}}} \,. \tag{7.2.6}$$

By trivially estimating $N(v_N^*)^\tau \leq \sum_{N \geq 1} (v_N^*)^\tau \leq \|\mathbf{v}\|_{\ell_\tau}^\tau$, we conclude that

$$\|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}} \leq 2\|\mathbf{v}\|_{\ell_\tau} \,. \tag{7.2.7}$$

The norms on ℓ_τ^{w} are connected to the norms on $\mathcal{A}^\sigma$ by the following

Proposition 7.1. *Given $\sigma > 0$, let τ be defined by*

$$\frac{1}{\tau} = \frac{1}{2} + \sigma \,. \tag{7.2.8}$$

Then the sequence $\mathbf{v}$ belongs to $\mathcal{A}^\sigma$ if and only if $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$, and we have the equivalence

$$\|\mathbf{v}\|_{\mathcal{A}^\sigma} \sim \|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}} \,. \tag{7.2.9}$$

In particular, if $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$, then

$$\mathcal{E}_N(\mathbf{v}) \lesssim \|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}} N^{-\sigma} \,, \qquad N \geq 1 \,. \tag{7.2.10}$$

The constants only depend on τ, which is linked to σ by (7.2.8).

The proof can be found in [57, 60]. We have thus provided a characterisation of vectors by their approximation properties in the space $\mathcal{A}^\sigma$, which is equivalent to the space ℓ_τ^{w}.

In the limit case $\tau = 2$, which corresponds to standard ℓ_2 spaces, we have $\sigma = 0$, which means that the approximation error does not decay at all. In other words, nonlinear approximation delivers a better rate than linear approximation if and only if the coefficient vector is in ℓ_τ^{w} for $0 < \tau < 2$.

7.2.2 Connection to Besov Spaces

So far, we have considered general vectors over subsets of ℓ_2. To specialise to wavelet representations, recall that in Section 2.2.2 we have established a norm equivalence between functions in Sobolev spaces and vectors of wavelet coefficients in ℓ_2. We now generalise this concept to Besov spaces [1, 57, 134].

Adaptive methods are especially motivated when singularities come into play. Consider for example functions which are smooth away from the singularities. These have may have very low Sobolev regularity, which restricts their rate of approximation with uniform discretisations. The introduction of Besov spaces provides a means to measure smoothness in a more general way, and to gain finer control of singularities. To this end, we define the norm

$$\|f\|_{B^\alpha_{q,p}} := \left(\|f\|^q_{L_p} + \sum_{j \geq j_0} \left(2^{\alpha j} \omega_p^m(f, 2^{-j})\right)^q \right)^{\frac{1}{q}} \tag{7.2.11}$$

with $m \in \mathbb{N}_0$, $m > \alpha$. Here, $\omega_p^m(f, \Delta)$ is the L_p modulus of smoothness of m-th order,

$$\omega_p^m(f, \Delta) := \sup_{|h| \leq \Delta} \left\| \sum_{k=0}^{m} \binom{m}{k} (-1)^k f(\cdot - kh) \right\|_{L_p} . \tag{7.2.12}$$

Note that for bounded domains Ω, we are only allowed to take the supremum of the norms over $L_p(\Omega_{m,h})$, where $\Omega_{m,h} = \{x : x + kh \in \Omega, k = 0, \ldots, m\}$. The Besov space $B^\alpha_{q,p}$ is defined as the set of all functions in L_p for which the norm (7.2.11) is finite.

Similar to the description in Section 2.2.2, there exists a generalised wavelet norm equivalence. Let $f = \mathbf{c}^T \Psi$ be the representation of a function f in an L_2-stable wavelet basis Ψ. Then we can formulate the following relation,

$$\|f\|_{B^\alpha_{q,p}} \sim \left(\sum_{j=j_0-1}^{\infty} 2^{jq(\alpha + \frac{n}{2} - \frac{n}{p})} \left(\sum_{k \in \nabla_j} |c_{j,k}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} . \tag{7.2.13}$$

It reduces to (2.2.35) for the case $p = q = 2$, i.e., $H^\alpha = B^\alpha_{2,2}$. Details on the definitions and proofs can be found in [38, 57, 58]. Setting $p = q = \tau$, $\alpha = \sigma n + \beta$ and considering (7.2.8), we obtain

$$\|f\|_{B^{\sigma n + \beta}_{\tau,\tau}} \sim \|\mathbf{D}^\beta \mathbf{c}\|_{\ell_\tau} = \|\mathbf{c}_\beta\|_{\ell_\tau} , \tag{7.2.14}$$

where $\mathbf{c}_\beta$ denotes the coefficient vector in the wavelet expansion using the scaled basis Ψ^β, $f = \mathbf{c}_\beta^T \Psi^\beta$, cf. (2.2.48). We have thus established the link between Besov spaces over L_τ and the ℓ_τ sequence spaces. The weaker condition $\mathbf{c}_\beta = \mathbf{D}^\beta \mathbf{c} \in \ell_\tau^{\mathrm{w}}$ leads to slightly larger spaces, which contain precisely the functions whose best N-term wavelet approximation in the energy norm produces an error in $\mathcal{O}(N^{-\sigma})$.

Note also that by the Sobolev embedding theorem we have the relation

$$B^{\sigma n + \beta}_{\tau,\tau} \subset H^\beta \qquad \text{for} \qquad \frac{1}{\tau} \leq \frac{1}{2} + \sigma , \tag{7.2.15}$$

Subroutine `Coarse` $(\mathbf{v},\eta) \to \mathbf{v}_\eta$: Reduces the support of $\mathbf{v}$ such that $\|\mathbf{v}-\mathbf{v}_\eta\| \le \eta$.

(1) Define $N := \#(\operatorname{supp}\mathbf{v})$ and sort the nonzero entries of $\mathbf{v}$ into decreasing order, with sorted indices $\lambda_1,\ldots,\lambda_N$.

(2) Compute $\|\mathbf{v}\|_{\ell_2}^2 := \sum_{i=1}^{N}|\mathbf{v}_{\lambda_i}|^2$.

(3) For $k=1,2,\ldots$, form the sum $S_k := \sum_{i=1}^{k}|\mathbf{v}_{\lambda_i}|^2$ and find the smallest k such that $S_k \ge \|\mathbf{v}\|_{\ell_2}^2 - \eta^2$. With this k, set $\mathbf{v}_\eta := \mathbf{P}_{\{\lambda_1,\ldots,\lambda_k\}}\mathbf{v}$.

Algorithm 7.1: This routine implements the coarsening step. It returns a vector $\mathbf{v}_\eta$ whose support is made up of the smallest subset of Λ which guarantees the error bound η. The sorting step can be replaced by a binary binning, which removes the logarithmic factor in the runtime complexity, only increasing the error by a constant factor.

which means that Besov spaces with $p=q=\tau<2$ are larger than Sobolev spaces of the same smoothness index. This permits an increasingly better resolution of singularities for smaller values of τ. In the context of adaptive wavelet methods, this means that optimal convergence rates can be guaranteed for a larger class of functions compared to uniform discretisations.

7.2.3 Coarsening Strategy

At this point, we can formulate the first ingredient for an adaptive wavelet algorithm, namely a routine to reduce the number of nonzero coefficients in an adaptive wavelet representation in a controlled fashion. Because of (7.2.14), the error made in the coarsening of the vector is equivalent to the error in the corresponding function.

We propose the routine `Coarse`, originally described in [39], which utilises the concept of decreasing rearrangement of a vector. It is displayed in Algorithm 7.1. Its purpose is to reduce the number of nonzero entries in the index set Λ of a vector $\mathbf{v}$ as much as possible, under the constraint that the error is bounded. In a naive implementation, the sorting operation introduces a logarithmic factor in the runtime complexity. This can be removed by performing a binary binning instead [6], which increases the error by at most a constant factor. The amount by which the support of a vector $\mathbf{v}$ can be reduced by this routine depends on the class $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$ to which it belongs. This is not only important for intermediate steps of the algorithm, but also for the processing of input data.

The numerical method for the control problem refers to two input variables, namely the right hand side of the partial differential equation f and the target observation y_*. In both cases we assume that all wavelet coefficients $\mathbf{f}$ and $\mathbf{y}_*$ are known or can be computed to any desired accuracy. Consequently, we can approximate these functions to arbitrary accuracy by finite wavelet expansions $\bar{\mathbf{f}}$ and $\bar{\mathbf{y}}_*$. This happens in the routine `Approx`, which is conceptually similar to `Coarse` and therefore not displayed here. The theory stated above implies that for any $\mathbf{f} \in \ell_\tau^{\mathrm{w}}$, $0<\tau<2$, there exists an approximate finite wavelet representation $\bar{\mathbf{f}}$ whose support is determined by the target accuracy η,

$$\|\mathbf{f}-\bar{\mathbf{f}}\|_{\ell_2} \le \eta\,, \qquad \#\operatorname{supp}\bar{\mathbf{f}} \lesssim \eta^{-1/\sigma}\|\mathbf{f}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma}\,, \tag{7.2.16}$$

where s and τ are related as in (7.2.8). An analogous relation holds for y_*.

7.2.4 Application of Quasi-Sparse Matrices

The iterative solver for the control problem relies entirely on matrix-vector products which can be computed in linear time. For a uniform discretisation, the cost is $\mathcal{O}(N)$ with $N \sim 2^{nJ}$, since all operations can be formulated as a product of uniformly sparse matrices.

To ensure optimal computational complexity for our adaptive computations we can only accept a cost of $\mathcal{O}(N_{\mathrm{ad}})$, where $N_{\mathrm{ad}} \leq N$ is the current size of the support of the adaptive wavelet representation of a function. Furthermore, the size of the support of the output vector must be controlled. To this end, special modifications of the matrix-vector multiplications are required, which make use of the following two properties.

- Stiffness and mass matrices in wavelet representation are *quasi-sparse*. They exhibit a decay of entries away from the diagonal which can be quantified a-priori.
- It is not necessary to compute the exact matrix-vector product, which would result in an infinite number of nonzero entries in the output vector. In the wavelet setting, estimates can be derived which guarantee the required accuracy for an *approximate* product.

To quantify these two statements, we note that for a large class of elliptic operators B over Sobolev spaces H^α the following decay property holds,

$$2^{-(|\lambda'|+|\lambda|)\alpha}|\langle B\psi_\lambda, \psi'_\lambda\rangle| \lesssim 2^{-||\lambda|-|\lambda'||\gamma}(1+r(\lambda,\lambda'))^{-\beta}\,, \tag{7.2.17}$$

with parameters $\gamma > \frac{n}{2}$, $\beta > n$ and

$$r(\lambda,\lambda') := 2^{\min\{|\lambda|,|\lambda'|\}}\operatorname{dist}(\operatorname{supp}(\psi_\lambda), \operatorname{supp}(\psi_{\lambda'}))\,. \tag{7.2.18}$$

This relation has been confirmed in various settings, see e.g. [10,46,133,143]. The constant γ depends on the smoothness of the wavelets, whereas β is related to the approximation order of the dual multiresolution and the order of the operator B.

A matrix $\mathbf{B} = (b_{\lambda,\lambda'})_{\lambda,\lambda'}$ is said to be quasi-sparse if it belongs to the class $\mathcal{A}_{\gamma,\beta}$, defined as

$$\mathcal{A}_{\gamma,\beta} := \left\{\mathbf{B} : |b_{\lambda,\lambda'}| \leq C(\mathbf{B})2^{-||\lambda|-|\lambda'||\gamma}(1+r(\lambda,\lambda'))^{-\beta}\right\}, \tag{7.2.19}$$

with $r(\lambda,\lambda')$ defined as in (7.2.18). It follows from (7.2.17) that $\mathbf{B} \in \mathcal{A}_{\gamma,\beta}$ holds for the expansion of an operator B in a scaled wavelet basis for H^α. Matrices from $\mathcal{A}_{\gamma,\beta}$ are bounded operators on ℓ_2. They are also *compressible*, where we refer to [10,51,125,129,143] for details on matrix compression. The reasoning from [39] is based on the following

Proposition 7.2. *For each $\gamma > \frac{n}{2}$ and $\beta > n$, let*

$$\sigma^* := \min\left\{\frac{\gamma}{n} - \frac{1}{2}, \frac{\beta}{n} - 1\right\} \tag{7.2.20}$$

and assume that $\mathbf{B} \in \mathcal{A}_{\gamma,\beta}$. Then, for any given $\sigma < \sigma^*$ and every $j \in \mathbb{N}$, there exists a matrix $\mathbf{B}_j$ which contains at most 2^j nonzero entries in each row and column and provides the approximation efficiency

$$\|\mathbf{B} - \mathbf{B}_j\| \lesssim 2^{-\sigma j}\,. \tag{7.2.21}$$

This result also holds for $\sigma = \sigma^*$ if $\gamma - \frac{n}{2} \neq \beta - n$.

Remark 7.3. *For expansions of elliptic differential operators in the natural wavelet basis, which occur in the discretisation of the control problem, the matrices $\mathbf{B}_j$ can be assembled according to*

$$(\mathbf{B}_j)_{\lambda,\lambda'} = \begin{cases} b_{\lambda,\lambda'} & \text{for } ||\lambda| - |\lambda'|| \leq \frac{j}{n}\,, \\ 0 & \text{else}\,, \end{cases} \tag{7.2.22}$$

for any $\sigma \leq \frac{\gamma}{n} - \frac{1}{2}$. This means that the parameter β is not relevant in this case.

Thus, stiffness matrices arising from wavelet discretisations of elliptic problems are compressible, and allow finite approximations $\mathbf{B}_j$ in the above sense. For an optimal adaptive matrix-vector multiplication we use the following slightly more general characterisation.

Definition 7.4. *A matrix* $\mathbf{B}$ *is said to be in the class* $\mathcal{B}_\sigma$ *if there are two positive sequences* $(\alpha_j)_{j\geq 0}$ *and* $(\beta_j)_{j\geq 0}$ *that are both summable, and for every* $j \geq 0$ *there exists a matrix* $\mathbf{B}_j$ *with at most* $2^j\alpha_j$ *nonzero entries per row and column such that*

$$\|\mathbf{B} - \mathbf{B}_j\| \leq 2^{-\sigma j}\beta_j\,. \tag{7.2.23}$$

We further define

$$\|\mathbf{B}_\sigma\|_{\mathcal{B}_\sigma} \coloneqq \min\max\left\{\sum_{j\geq 0}\alpha_j, \sum_{j\geq 0}\beta_j\right\}\,, \tag{7.2.24}$$

where the minimum is taken over all such sequences (α_j) *and* (β_j)*.*

With σ^* defined as in (7.2.20), we have $\mathcal{A}_{\gamma,\beta} \subset \mathcal{B}_\sigma$ for every $0 \leq \sigma < \sigma^*$. For $\mathbf{B} \in \mathcal{A}_{\gamma,\beta}$, the sequences (α_j), (β_j) can be chosen to decay exponentially, which means that stiffness matrices in wavelet representation belong to $\mathcal{B}_\sigma$. The main result is then formulated as follows.

Proposition 7.5. *For* $\mathbf{B} \in \mathcal{B}_\sigma$*, the matrix* $\mathbf{B}$ *is a continuous mapping over* ℓ^{w}_τ*, i.e., for any* $\mathbf{v} \in \ell^{\mathrm{w}}_\tau$ *we have*

$$\|\mathbf{B}\mathbf{v}\|_{\ell^{\mathrm{w}}_\tau} \lesssim \|\mathbf{v}\|_{\ell^{\mathrm{w}}_\tau}\,. \tag{7.2.25}$$

Furthermore, for each $\mathbf{v} \in \ell^{\mathrm{w}}_\tau$ *and* $\epsilon > 0$*, there exists a vector* $\mathbf{w}_\epsilon$ *such that*

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_\epsilon\| \leq \epsilon \tag{7.2.26}$$

and

$$\#\operatorname{supp}\mathbf{w}_\epsilon \lesssim N_\tau(\mathbf{v},\epsilon) = \epsilon^{-1/\sigma}\|\mathbf{v}\|^{1/\sigma}_{\ell^{\mathrm{w}}_\tau}\,, \tag{7.2.27}$$

with σ *and* τ *related as in* (7.2.8)*. The number of arithmetic operations to compute* $\mathbf{w}_\epsilon$ *is also bounded by a constant multiple of* $N_\tau(\mathbf{v},\epsilon)$*. In all of these estimates, the constant depends only on* $\|\mathbf{B}\|$ *and* $\|\mathbf{B}\|_{\mathcal{B}_\sigma}$*.*

It remains to explain how $\mathbf{w}_\epsilon$ can actually be computed. To this end, let $\mathbf{v}_{[j]} \in \Sigma_{2^j}$ be a best 2^j-term approximation to $\mathbf{v}$ in ℓ_2 as introduced in Section 7.2.1, set $\mathbf{v}_{[-1]} \coloneqq 0$, and let

$$\mathbf{w}_j \coloneqq \mathbf{B}_j(\mathbf{v}_{[0]} - \mathbf{v}_{[-1]}) + \cdots + \mathbf{B}_0(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]})\,. \tag{7.2.28}$$

The error is computed according to

$$\mathbf{B}\mathbf{v} - \mathbf{w}_j = \mathbf{B}(\mathbf{v} - \mathbf{v}_{[j]}) + (\mathbf{B} - \mathbf{B}_0)(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}) + \cdots + (\mathbf{B} - \mathbf{B}_j)(\mathbf{v}_{[0]} - \mathbf{v}_{[-1]})\,. \tag{7.2.29}$$

By the triangle inequality, we can estimate this sum by quantities which are known, namely the matrix norms abbreviated as $b \coloneqq \|\mathbf{B}\|$, $b_j \coloneqq \|\mathbf{B} - \mathbf{B}_j\|$ and the vector norms $v_j \coloneqq \|\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\|$ and $\bar{v}_j \coloneqq \|\mathbf{v} - \mathbf{v}_{[j]}\|$. This leads to

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_j\| \leq R_j \coloneqq b\bar{v}_j + b_0 v_j + \cdots + b_j v_0\,. \tag{7.2.30}$$

Using the estimates for the approximation of vectors in ℓ^{w}_τ and inserting (7.2.23), we eventually obtain

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_j\| \lesssim 2^{-\sigma j}\|\mathbf{v}\|_{\ell^{\mathrm{w}}_\tau}\,. \tag{7.2.31}$$

The complete derivation can be found in [39]. We can see that the ℓ_2-error of the approximate matrix-vector product decays with j if $\sigma > 0$. Algorithmically, we run through the levels j and construct the vectors $\mathbf{w}_j$ and norms $\bar{v}_j$ and v_j, starting with $j = 0$, and accept the first j for which the right hand

```
Subroutine Ad-Apply $(\mathbf{B}, \mathbf{v}, \epsilon) \to \mathbf{w}_\epsilon$: Computes $\mathbf{w}_\epsilon$ such that $\|\mathbf{B}\mathbf{v} - \mathbf{w}_\epsilon\| \leq \epsilon$.
(1) Define $N := \#(\operatorname{supp}\mathbf{v})$ and sort the nonzero entries of $\mathbf{v}$ into decreasing order. Form $\mathbf{v}_{[0]}$ and $\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}$, $j = 0, \ldots, \lceil \log(N) \rceil$.
(2) Compute $v_j := \|\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\|$, $\bar{v}_j := \|\mathbf{v} - \mathbf{v}_{[j]}\|$, $j = 0, \ldots, \lceil \log(N) \rceil$.
(3) For $j = 0, 1, \ldots$, compute the right hand side $R_j$ of (7.2.30) and find the smallest $j$ such that $R_j \leq \epsilon$.
(4) For this $j$, compute $\mathbf{w}_j$ as in (7.2.28), accept $\mathbf{w}_j \to \mathbf{w}_\epsilon$.
```

Algorithm 7.2: This subroutine realises the adaptive approximate matrix-vector multiplication for elliptic operators. It provides a runtime and memory complexity of order $N_\tau(\mathbf{v}, \epsilon)$, which is the optimal rate in the sense of ℓ^{w}_τ approximation.

side of (7.2.30) is less or equal to ϵ. The subroutine `Ad-Apply`, which is listed in Algorithm 7.2, performs exactly these actions. For efficient operation, we employ the trivial relations $\bar{v}_j^2 = \|\mathbf{v}\|^2 - \|\mathbf{v}_{[j]}\|^2$ and $\|\mathbf{v}_{[j]}\|^2 = \sum_{l=0}^{j} v_l^2$.

We have thus specified a subroutine which calculates an approximate matrix-vector product within given error bounds (7.2.26). If the input vector $\mathbf{v}$ is in ℓ^{w}_τ with $0 < \sigma < \sigma^*$ and (7.2.8) satisfied, the size of the output vector is bounded by a constant multiple of $N_\tau(\mathbf{v}, \epsilon)$. The number of arithmetic operations is bounded by $CN_\tau(\mathbf{v}, \epsilon) + 2N$, and the output vector satisfies

$$\|\mathbf{w}_\epsilon\|_{\ell^{\mathrm{w}}_\tau} \lesssim \|\mathbf{v}\|_{\ell^{\mathrm{w}}_\tau} . \tag{7.2.32}$$

The subroutines `Ad-Apply` detailed above and `Coarse` described in Section 7.2.3 constitute the two main ingredients in the conception of a fully adaptive wavelet algorithm.

7.3 An Adaptive Algorithm for the Control Problem

We build the adaptive wavelet algorithm for the optimal control problem by incorporating the adaptive concepts described in the previous section into the wavelet solver presented in Algorithm 6.6, carefully balancing errors from adaptive approximation with the errors from the inexact conjugate gradient routine. Thereby we gain a fully adaptive solver and retain the superior convergence rate of the conjugate gradient method for the uniformly well-conditioned systems.

7.3.1 Properties of the Original Richardson Algorithm

First, we shortly list the essential properties of the ideal adaptive wavelet algorithms for elliptic boundary value problems [39], and for linear-elliptic control problems as proposed in [48]. We begin with the first, simpler case. Here a H-elliptic system in natural wavelet coordinates $\mathbf{A}\mathbf{u} = \mathbf{f}$ as described in Section 4 has to be solved up to accuracy ϵ. The algorithm consists of Richardson iterations of the type

$$\mathbf{u}^{(l+1)} = \mathbf{u}^{(l)} - c(\mathbf{A}\mathbf{u}^{(l)} - \mathbf{f}) , \tag{7.3.1}$$

where the constant c is determined by the eigenvalues of $\mathbf{A}$. The application of $\mathbf{A}$ is realised by the routine `Ad-Apply` described in Section 7.2.4. The subroutine `Coarse` from Section 7.2 is called at various points in the algorithm to keep the size of the support of the involved vectors under control. The error bounds for these routines are purposefully adjusted to guarantee the following convergence result.

Theorem 7.6. *Assume that* $\mathbf{A} \in \mathcal{B}_\sigma$ *with* $0 < \sigma < \sigma^*$. *Then for any* $\epsilon > 0$ *and* $\mathbf{f} \in \ell_2$, *the computations yield an approximation* $\mathbf{u}_\epsilon$ *with* $N_\epsilon = \#\operatorname{supp}\mathbf{u}_\epsilon < \infty$ *satisfying*

$$\|\mathbf{u} - \mathbf{u}_\epsilon\|_{\ell_2} \leq \epsilon \, . \tag{7.3.2}$$

If the exact solution satisfies $\mathbf{u} \in \ell_\tau^{\mathrm{w}}$, *where* σ *and* τ *are related by* (7.2.8), *then the algorithm is optimal in the sense that*

$$N_\epsilon \lesssim \epsilon^{-1/\sigma} \|\mathbf{u}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma} \, , \tag{7.3.3}$$

and the number of arithmetic operations is also bounded by a constant multiple of N_ϵ.

Moreover, if we have $\mathbf{A} \in \mathcal{A}_{\gamma,\beta}$, $f \in H^{-\beta}$, *and the solution* u *belongs to the Besov space* $B_{\tau,\tau}^{\sigma n+\beta}$, *then the complexity can be estimated from above by*

$$N_\epsilon \lesssim \epsilon^{-1/\sigma} \|u\|_{B_{\tau,\tau}^{\sigma n+\beta}}^{1/\sigma} \, . \tag{7.3.4}$$

In the optimal control problem, we have to deal with three variables, namely the state y, the costate p and the control u. By solving the reduced equation for the control (5.5.6) with outer iterations similar to (7.3.1), which in turn involves iterations on the inner equations (5.5.9a) and (5.5.9b), an additional level of calculations is introduced which complicates the synchronisation of the various parameters for the routines `Ad-Apply` and `Coarse`. Yet, the main techniques are the same as for the single elliptic equation, and also the type of estimates used is similar. The following result on convergence can be derived [48].

Theorem 7.7. *For any* $\epsilon > 0$ *and* $\mathbf{f}, \mathbf{y}_* \in \ell_2$, *the computations yield approximations* $(\mathbf{y}_\epsilon, \mathbf{p}_\epsilon, \mathbf{u}_\epsilon)$ *with*

$$\|\mathbf{y} - \mathbf{y}_\epsilon\|_{\ell_2} \lesssim \epsilon \, , \qquad \|\mathbf{p} - \mathbf{p}_\epsilon\|_{\ell_2} \lesssim \epsilon \, , \qquad \|\mathbf{u} - \mathbf{u}_\epsilon\|_{\ell_2} \leq \epsilon \, . \tag{7.3.5}$$

If the exact solutions $(\mathbf{y}, \mathbf{p}, \mathbf{u})$ *of* (5.5.9) *all belong to the space* ℓ_τ^{w} *for* $0 < \tau < 2$, *where* σ *and* τ *are related by* (7.2.8), *it follows that*

$$\|\mathbf{y}_\epsilon\|_{\ell_\tau^{\mathrm{w}}} + \|\mathbf{p}_\epsilon\|_{\ell_\tau^{\mathrm{w}}} + \|\mathbf{u}_\epsilon\|_{\ell_\tau^{\mathrm{w}}} \lesssim \|\mathbf{y}\|_{\ell_\tau^{\mathrm{w}}} + \|\mathbf{p}\|_{\ell_\tau^{\mathrm{w}}} + \|\mathbf{u}\|_{\ell_\tau^{\mathrm{w}}} \tag{7.3.6}$$

and

$$\#\operatorname{supp}\mathbf{y}_\epsilon + \#\operatorname{supp}\mathbf{p}_\epsilon + \#\operatorname{supp}\mathbf{u}_\epsilon \lesssim \epsilon^{-1/\sigma} \left(\|\mathbf{y}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma} + \|\mathbf{p}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma} + \|\mathbf{u}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma} \right) \, . \tag{7.3.7}$$

The number of arithmetic operations is also proportional to the right hand side of (7.3.7).

In summary, both of these algorithms are asymptotically optimal in the sense that the numerical effort is proportional to the cost of approximation of the exact solutions in the ℓ_τ^{w} sense. This is made possible by investigating the properties of Richardson iterations (7.3.1) in the context of an adaptive wavelet discretisation, using the techniques of coarsening and approximate matrix-vector products.

7.3.2 An Adaptive Conjugate Gradient Method for the Control Problem

We now propose a way to enhance Algorithm 6.6 with the adaptive techniques which have been introduced in Section 7.2. This leads to an algorithm which is conceptually different from the ideal wavelet scheme quoted above. Here, the investigation of error bounds is closely parallel to Section 6.2.1, where the wavelet algorithm in uniform discretisation has been described in detail. It turns out that we can adopt its main structure almost unchanged. More precisely, we carry out the following steps to obtain the adaptive algorithm `AnIIcG/2`.

- All direct matrix-vector products previously realised in the routine `Apply` are replaced by calls to the routine `Ad-Apply`. This introduces an error which makes the inner CG methods inexact.

```
Subroutine Ad-RHS $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, \zeta) \to \mathbf{g}_\zeta$: Computes $\mathbf{g}_\zeta$ such that $\|\mathbf{g} - \mathbf{g}_\zeta\| \leq \zeta$.
(I) SET $\zeta_1 := \frac{1}{3}\frac{c_{\mathbf{A}}}{2C_{\mathbf{E}}}\frac{c_{\mathbf{A}}}{C_{\mathbf{T}}^2 C_s}\zeta$.
    (1) CALL Approx $(\mathbf{f}, \zeta_1) \to \bar{\mathbf{f}}$.
    (2) CALL CG $(\mathbf{A}, \bar{\mathbf{f}}, \zeta_1) \to \mathbf{g}_1$.
    (3) CALL Coarse $(\mathbf{g}_1, \frac{\zeta_1}{c_{\mathbf{A}}}) \to \mathbf{g}_2$.
(II) SET $\zeta_2 := \frac{1}{5}\frac{c_{\mathbf{A}}}{2C_{\mathbf{E}}}\zeta$.
    (1) CALL Approx $(\mathbf{y}_*, \frac{\zeta_2}{C_{\mathbf{T}}C_s}) \to \bar{\mathbf{y}}_*$.
    (2) CALL Ad-Apply $(\mathbf{R}_s, \mathbf{T}\mathbf{g}_2 - \bar{\mathbf{y}}_*, \frac{\zeta_2}{C_{\mathbf{T}}}) \to \mathbf{v}_1$.
    (3) CALL Coarse $(\mathbf{v}_1, \frac{\zeta_2}{C_{\mathbf{T}}}) \to \mathbf{v}_2$.
    (4) CALL CG $(\mathbf{A}^T, -\mathbf{T}^T\mathbf{v}_2, \zeta_2) \to \mathbf{g}_3$.
    (5) CALL Coarse $(\mathbf{g}_3, \frac{\zeta_2}{c_{\mathbf{A}}}) \to \mathbf{g}_4$.
(III) COMPUTE $\mathbf{E}^T\mathbf{g}_4 \to \mathbf{g}_\zeta$.
```

Algorithm 7.3: We show the adaptive algorithm `Ad-RHS` *for the computation of the right hand side. It contains two calls to the routine* `Approx` *described in Section 7.2.3. To guarantee the overall accuracy, the error bounds from the original version (Algorithm 6.3) are split for the several substeps.*

- We insert a call to `Coarse` after each call to an inner conjugate gradient method.
- The Riesz matrix $\mathbf{R}_s$ is applied by the routine `Ad-Apply`. This affects the right hand side of the adjoint equation, where the additional error must be considered.
- After each invocation of `Ad-Apply`, we insert a call to the routine `Coarse` to reduce the size of the output vector. The coarsening error adds to the error in the approximate matrix-vector product.
- The right hand side $\mathbf{g}$ needs coarse versions of the data $\mathbf{f}$ and $\mathbf{y}_*$. The corresponding errors from the routine `Approx` are taken into account in the routine `RHS`, and in the reconstruction of $\mathbf{y}_\epsilon$ at the very end.

Note that the application of the matrices $\mathbf{E}$ and $\mathbf{T}$ does not need to be reconsidered, since both have an explicitly known diagonal form.

In Section 6.2.2, we have considered the error in the application of the matrix $\mathbf{Q}$ during the outer iterations and motivated the bound $\eta_k = \epsilon$ in the inexact conjugate gradient method. We carry over this strategy to the inner level and take the same bounds for the combination of adaptive application and subsequent coarsening step. More precisely, the routine `Ad-Apply` when called with the matrices $\mathbf{A}$, $\mathbf{A}^T$ and $\mathbf{R}_t$, and the routine `Coarse` on the respective results are each given the error bound $\eta_k = \frac{\epsilon}{2}$.

We now review the subroutines from the full Algorithm 6.6 and recalculate the error bounds where necessary. We begin with the routine `RHS`, which has been described in original form in Algorithm 6.3. The adaptive version `Ad-RHS` is displayed in Algorithm 7.3. Its structure has not changed much, we have essentially split the steps (I) and (II) into substeps, preserving the error bound from Proposition 6.1.

Proposition 7.8. *The result* $\mathbf{g}_\zeta$ *of the subroutine* `Ad-RHS` $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, \zeta)$ *satisfies upon completion*

$$\|\mathbf{g} - \mathbf{g}_\zeta\| \leq \zeta\,. \tag{7.3.8}$$

```
Subroutine Ad-Apply $(\mathbf{Q}, \mathbf{u}, \eta) \to \mathbf{m}_\eta$: Computes $\mathbf{m}_\eta$ such that $\|\mathbf{Qu} - \mathbf{m}_\eta\| \leq \eta$.
(I) Set $\eta_1 := \frac{1}{2}\frac{c_\mathbf{A}}{4C_\mathbf{E}}\frac{c_\mathbf{A}}{C_\mathbf{T}^2 C_s}\eta$.
    (1) Call CG $(\mathbf{A}, \mathbf{Eu}, \eta_1) \to \mathbf{y}_1$.
    (2) Call Coarse $(\mathbf{y}_1, \frac{\eta_1}{c_\mathbf{A}}) \to \mathbf{y}_2$.
(II) Set $\eta_2 := \frac{1}{4}\frac{c_\mathbf{A}}{4C_\mathbf{E}}\eta$.
    (1) Call Ad-Apply $(\mathbf{R}_s, \mathbf{Ty}_2, \frac{\eta_2}{C_\mathbf{T}}) \to \mathbf{w}_1$.
    (2) Call Coarse $(\mathbf{w}_1, \frac{\eta_2}{C_\mathbf{T}}) \to \mathbf{w}_2$.
    (3) Call CG $(\mathbf{A}^T, -\mathbf{T}^T\mathbf{w}_2, \eta_2) \to \mathbf{p}_1$.
    (4) Call Coarse $(\mathbf{p}_1, \frac{\eta_2}{c_\mathbf{A}}) \to \mathbf{p}_2$.
(III) Set $\eta_3 := \frac{1}{2}\frac{c_t}{4\omega}\eta$.
    (1) Call CG $(\mathbf{R}_t, \mathbf{u}, \eta_3) \to \mathbf{q}_1$.
    (2) Call Coarse $(\mathbf{q}_1, \frac{\eta_3}{c_t}) \to \mathbf{q}_2$.
(IV) Call Coarse $(\omega\mathbf{q}_2 - \mathbf{E}^T\mathbf{p}_2, \frac{\eta}{4}) \to \mathbf{m}_\eta$.
```

Algorithm 7.4: This table contains the step from the adaptive version of `Apply` $(\mathbf{Q})$. *The error is divided into four equal parts which are split again in the several substeps. Note that each call to an operation of type* `Ad-Apply` *or* `CG` *is followed by a call to* `Coarse`.

Proof. Considering step (I), we calculate

$$\|\mathbf{A}^{-1}\mathbf{f} - \mathbf{g}_2\| \leq \|\mathbf{A}^{-1}(\mathbf{A}\mathbf{g}_1 - \bar{\mathbf{f}})\| + \frac{\zeta_1}{c_\mathbf{A}} \leq \frac{1}{c_\mathbf{A}}\big(\|\mathbf{A}\mathbf{g}_1 - \mathbf{f}\| + \zeta_1\big) + \frac{\zeta_1}{c_\mathbf{A}} \leq 3\frac{\zeta_1}{c_\mathbf{A}}\,, \tag{7.3.9}$$

where the estimates follow from the definition of the routines `Approx` and `CG`. Step (II) is examined in a similar way,

$$\begin{aligned}
\|\mathbf{A}^{-T}\mathbf{T}^T\mathbf{R}_s(\mathbf{T}\mathbf{g}_1 - \mathbf{y}_*) + \mathbf{g}_4\| &= \frac{1}{c_\mathbf{A}}\|\mathbf{T}^T(\mathbf{R}_s(\mathbf{T}\mathbf{g}_2 - \mathbf{y}_*) - \mathbf{v}_2) + \mathbf{T}^T\mathbf{v}_2 + \mathbf{A}^T\mathbf{g}_3\| + \frac{\zeta_2}{c_\mathbf{A}} \\
&\leq \frac{C_\mathbf{T}}{c_\mathbf{A}}\|\mathbf{R}_s(\mathbf{T}\mathbf{g}_2 - \mathbf{y}_*) - \mathbf{v}_1 + \mathbf{v}_1 - \mathbf{v}_2\| + 2\frac{\zeta_2}{c_\mathbf{A}} \\
&\leq \frac{C_\mathbf{T}}{c_\mathbf{A}}\|\mathbf{R}_s(\mathbf{T}\mathbf{g}_2 - \bar{\mathbf{y}}_* + \bar{\mathbf{y}}_* - \mathbf{y}_*) - \mathbf{v}_1\| + 3\frac{\zeta_2}{c_\mathbf{A}} \leq 5\frac{\zeta_2}{c_\mathbf{A}}\,.
\end{aligned} \tag{7.3.10}$$

After inserting the definitions of ζ_1 and ζ_2, a comparison with the tolerances from Proposition 6.1 confirms that (7.3.8) is indeed satisfied. □

We move on to the application of the reduced system matrix $\mathbf{Q}$, which is performed by the routine `Ad-Apply` $(\mathbf{Q})$ presented in Algorithm 7.4. The modifications are analogous to those in the routine `Ad-RHS`. This adaptive routine computes the same quantities as the original routine `Apply` $(\mathbf{Q})$ in uniform discretisation, to the same target accuracy, cf. Proposition 6.2.

Proposition 7.9. *The result* $\mathbf{m}_\eta$ *of the subroutine* `Ad-Apply` $(\mathbf{Q}, \mathbf{u}, \eta)$ *obeys the inequality*

$$\|\mathbf{Qu} - \mathbf{m}_\eta\| \leq \eta\,. \tag{7.3.11}$$

```
Algorithm Ad-Control $(\mathbf{f}, \mathbf{y}_*, \epsilon) \to (\mathbf{y}_\epsilon, \mathbf{u}_\epsilon)$: Assures $\|\mathbf{y} - \mathbf{y}_\epsilon\| \leq \epsilon$, $\|\mathbf{u} - \mathbf{u}_\epsilon\| \leq \epsilon$.
(i) Set $\zeta := \min\{1, \frac{2}{3}\frac{c_{\mathbf{A}}}{C_{\mathbf{E}}}\}\epsilon$.
(ii) Call Ad-RHS $(\mathbf{A}, \mathbf{f}, \mathbf{y}_*, \frac{c_{\mathbf{Q}}}{2}\zeta) \to \mathbf{g}_\epsilon$.
(iii) Call CG $(\mathbf{Q}, \mathbf{g}_\epsilon, \frac{c_{\mathbf{Q}}}{2}\zeta) \to \mathbf{u}_\epsilon$.
(iv) Set $\epsilon_1 := \frac{1}{2}\frac{c_{\mathbf{A}}}{3}\epsilon$.
    (1) Call Approx $(\mathbf{f}, \epsilon_1) \to \bar{\mathbf{f}}$.
    (2) Call CG $(\mathbf{A}, \bar{\mathbf{f}} + \mathbf{E}\mathbf{u}_\epsilon, \epsilon_1) \to \mathbf{y}_\epsilon$.
```

Algorithm 7.5: The complete adaptive algorithm is presented here. It is very similar in structure to the uniform version from Algorithm 6.5, since most adjustments occur inside subroutines. Only the last step is extended by an approximation of the right hand side f of the elliptic constraint with a finitely supported coefficient vector $\bar{\mathbf{f}}$.

The proof reuses the concepts and techniques from Proposition 7.8, and is therefore omitted. The substeps of the routines `Ad-RHS` and `Ad-Apply` are all very similar, and so is the calculation of their accuracies.

The main routine `Control` listed in Algorithm 6.5 remains essentially unchanged. All adjustments are accomplished inside its subroutines, with the exception that the final solution for the state $\mathbf{y}_\epsilon$ requires a coarse version of the right hand side $\mathbf{f}$. For completeness, we display the complete adaptive scheme in Algorithm 7.5. We drop the proof of the error bounds here, as it is analogous to (7.3.9).

With the replacement of the crucial routines of the algorithm in uniform discretisation by their adaptive extensions, we have realised a complete conversion while preserving an operation count of $\mathcal{O}(N_{\mathrm{ad}})$, where N_{ad} is the number of nonzero wavelet coefficients at any given point of the computations. The remaining operations, namely additions, subtractions and the scaling of vectors, as well as scalar products, are also of linear complexity. Having at this point completed the specification of our adaptive algorithm, we attempt a comparison to the ideal approach followed e.g. in [39, 48, 105].

Conclusion 7.10. *In our approach the level j of resolution is systematically increased in the outer loop, determining the target accuracy 2^{-j}. Only wavelet coefficients up to the current level can occur. In a sense, the level dominates the accuracy. In contrast, the ideal wavelet method runs the outer loop over the target accuracy, which is tightened from iteration to iteration. For each grade of accuracy, the wavelet coefficients largest in magnitude are added to the set Λ, irrespective of the level. Thus, the accuracy dominates the selection and hence the levels of the coefficients.*

The ideal algorithm is solely designed to execute the process of nonlinear approximation. In our case, we have two parallel and competing goals, namely the scheme of nested iteration to keep the iteration numbers independent of the level, and the routines `Coarse` and `Ad-Apply` to control the number of nonzero coefficients. This approach trades mathematical pureness for practically fast solution in the framework of the nested iteration conjugate gradient solver. Consequently, the efficiency of our method is best studied numerically. To this end, we dedicate the following section to an extensive presentation and discussion of numerical experiments.

7.4 Numerical Results

The adaptive algorithm differs from Algorithm 6.6 in the addition of the subroutines `Approx`, `Coarse` and `Ad-Apply`, and the recalculation of error bounds. When the accuracy thresholds for these adaptive routines go to zero, `Approx` reduces to the standard expansion of a function in wavelet coefficients, the routine `Coarse` transforms into the identity, and `Ad-Apply` becomes the standard matrix-vector product. Consequently, the adaptive variant of the algorithm is structurally similar to the uniform. On the other hand, the following main differences exist.

- The adaptive application of the reduced system matrix $\mathbf{Q}$ as described in Algorithm 7.4 splits the errors of the various substeps into equally sized parts. As a result, the stopping criterion for the inner conjugate gradient methods is tightened by a factor between 2 and 4. This leads to a certain amount of additional inner iterations to reach the increased accuracy.
- The inner conjugate gradient iterations, which have been exact up to roundoff error in the uniform case, are now afflicted with systematic errors from the adaptive application of the matrix and the subsequent coarsening step. This makes them inexact and thus inaccessible to standard convergence theory. Yet the results from the previous chapter on inexact Krylov methods can be reused, so we expect that the adaptive algorithm also features iteration numbers which are independent of the level of resolution.

Technically, we modify our uniform algorithm to simulate adaptivity. This means that it is semantically equivalent to the adaptive method `AnIIcG/2`, i.e., all computations are carried out with exactly the same procedure as specified in Section 7.3.2. We achieve this by setting selected wavelet coefficients to zero inside our uniform data structures, thus distinguishing active from inactive coefficients. Additionally, an adaptive wavelet method may produce active coefficients with zero value, which are eliminated by calls to the routine `Coarse`. Consequently, our simulation can correctly assess the portion of nonzero coefficients at the end of this routine, which we use to analyse the convergence properties and estimate the cost and savings compared to the algorithm in uniform discretisation.

To judge the saving of memory and computing time through the adaptive approach, recall that the algorithm solves the elliptic equation $\mathbf{Qu} = \mathbf{g}$ for the control, while state $\mathbf{y}$ and costate $\mathbf{p}$ are computed in a postprocessing step. Consequently, the three inner CG methods inside the application of $\mathbf{Q}$ consume most of the computation time. Since all involved variables have different sets of nonzero coefficients which fluctuate in every inner iteration, the number of nonzero coefficients $N_{\mathrm{ad}} \leq N_J$ can only be determined on average. The application of the system matrix and the additions and subtractions occurring in the conjugate gradient method increase this number, while it is reduced by every invocation of the coarsening routine. Moreover, the nested iteration ansatz for the solution of the inner systems of equations involves a complete traversal of levels for each outer iteration. Finally, the outer iterations work on another, different set of variables whose distribution of coefficients are also varying with each operation.

We use the following recipe to estimate the main computational work which is concentrated in the iterative solution of the three inner systems of equations. The method of conjugate gradients works with the search direction $\mathbf{d}$, its approximate product with the system matrix $\mathbf{h} = \mathbf{Md}$, the residual $\mathbf{q}$ and finally the solution $\mathbf{x}$. For these four vectors, we acquire the respective counts of nonzero coefficients, which are then averaged over all iterations per level to obtain the quantity N_{ad}. The measurements occur at the end of the respective coarsening operations. Additionally, we express the ratio N_{ad}/N_j in percent, and examine the percentage of nonzero coefficients in $\mathbf{x}$ after the final coarsening operation at the end of each inner CG loop.

All results have been obtained with the same class of wavelets as in Chapter 6, with an accuracy factor of $\nu = 1$ (see Conclusion 4.12 and Section 6.3 for details).

7.4.1 The Solution of a Single Elliptic System

The algorithm for the optimal control problem contains a rather complex interplay between inner and outer iterations, and nested sweeps from coarse to fine levels. Since even for a single elliptic PDE the convergence of inexact Krylov methods is not proved rigorously, and there exists even less theory for our adaptive variant, it is indispensable to undertake first studies of the adaptive method in this reduced

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}
4	0.00e-00	0	0.0137%	0.0137%	0.0259%	52.9%	9
5	0.00e-00	0	0.0137%	0.0137%	0.0504%	27.2%	9
6	0.00e-00	0	0.0137%	0.0137%	0.0992%	13.8%	9
7	0.00e-00	0	0.0137%	0.0137%	0.197%	6.95%	9
8	0.00e-00	0	0.0137%	0.0137%	0.392%	3.49%	9
9	0.00e-00	0	0.0137%	0.0137%	0.783%	1.75%	9
10	0.00e-00	0	0.0137%	0.0137%	1.56%	0.878%	9
11	0.00e-00	0	0.0137%	0.0137%	3.13%	0.438%	9
12	0.00e-00	0	0.0137%	0.0137%	6.25%	0.219%	9
13	0.00e-00	0	0.0137%	0.0137%	12.5%	0.110%	9
14	0.00e-00	0	0.0137%	0.0137%	25%	0.0548%	9
15	0.00e-00	0	0.0137%	0.0137%	50%	0.0274%	9
16	0.00e-00	0	0.0137%	0.0137%	100%	0.0137%	9

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.27e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.87e-03
5	6.99e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.87e-03
6	2.80e-03	1	0.0137%	0.0763%	0.0992%	76.9%	50	8.87e-03
7	1.53e-03	2	0.0137%	0.172%	0.197%	87.3%	113	8.56e-03
8	8.17e-04	3	0.0137%	0.293%	0.392%	74.7%	192	8.38e-03
9	3.85e-04	4	0.0168%	0.638%	0.783%	81.5%	418	6.10e-03
10	1.54e-04	5	0.0259%	1.20%	1.56%	76.9%	786	3.30e-03
11	1.16e-04	4	0.0519%	2.36%	3.13%	75.4%	1547	1.65e-03
12	5.63e-05	4	0.0977%	4.76%	6.25%	76.2%	3120	8.39e-04
13	2.66e-05	4	0.195%	10.5%	12.5%	84.0%	6881	4.21e-04
14	1.31e-05	4	0.389%	20.1%	25%	80.4%	13173	2.14e-04
15	6.37e-06	4	0.777%	35.4%	50%	70.8%	23200	1.09e-04
16	3.55e-06	4	1.55%	69.7%	100%	69.7%	45679	5.78e-05

Table 7.1: We show the results for two solutions of one single elliptic system with different data. The case of constant data with $f \equiv 1$ is given above. The second table displays a solution for the right hand side $f = h_1$ as in (6.3.3).

setting. To this end, we consider the solution of a single state equation with zero control, which means that we deal with one run of the CG method `CG` $(\mathbf{A}, \mathbf{f}, \epsilon)$ as specified in Algorithm 6.2, combined with our adaptation strategy and nested iteration. In the course of these experiments, we propose an additional, heuristic ingredient to the adaptive conjugate gradient method which helps to save both memory and computation time.

Example 7.1 – A Single Elliptic System

The simplest example is given by constant data $f \equiv 1$, with constant solution $y \equiv 1$. We expect in this special case that the final result is already computed accurately on the coarsest level, and that no additional wavelet coefficients are introduced on higher levels.

The results in one dimension are shown in the top half of Table 7.1. The first three columns are arranged as in the previous chapter. The first column states the level of resolution j (on the lowest level $j_0 = 3$ the system is solved exactly, so it is not listed). The second holds the computed residual at the end of the conjugate gradient loop and the third the number of iterations on this respective level. Then, the column labelled P stands for the percentage of nonzero coefficients at the end of each loop, and the column V contains the average percentage throughout the iterations which corresponds to the number of arithmetic

```
Subroutine Ad-CG $(\mathbf{M}, \mathbf{b}, \epsilon) \rightarrow \mathbf{x}_\epsilon$: Computes $\mathbf{x}_\epsilon$ such that $\|\mathbf{M}\mathbf{x}_\epsilon - \mathbf{b}\| \leq \epsilon$.
(I) SET $k \coloneqq 0$, SET $\mathbf{d}_0 \coloneqq -\mathbf{q}_0 \coloneqq \mathbf{b}$.
(II) WHILE $\|\mathbf{q}_k\| > \theta_\epsilon$
    (1) CALL Ad-Apply $(\mathbf{M}, \mathbf{d}_k, \frac{\eta_k}{2}) \rightarrow \mathbf{h}_k$, CALL Coarse $(\mathbf{h}_k, \frac{\eta_k}{2})$,
        COMPUTE $\alpha_k \coloneqq \frac{\mathbf{q}_k^T \mathbf{q}_k}{\mathbf{d}_k^T \mathbf{h}_k}$.
    (2) COMPUTE $\mathbf{x}_{k+1} \coloneqq \mathbf{x}_k + \alpha_k \mathbf{d}_k$, $\mathbf{q}_{k+1} \coloneqq \mathbf{q}_k + \alpha_k \mathbf{h}_k$.
    (3) COMPUTE $\beta_k \coloneqq \frac{\mathbf{q}_{k+1}^T \mathbf{q}_{k+1}}{\mathbf{q}_k^T \mathbf{q}_k}$, $\mathbf{d}_{k+1} \coloneqq -\mathbf{q}_{k+1} + \beta_k \mathbf{d}_k$.
        CALL Coarse $(\mathbf{x}_{k+1}, \delta \frac{\eta_k}{C_\mathbf{M}})$, Coarse $(\mathbf{q}_{k+1}, \delta \eta_k)$. Coarse $(\mathbf{d}_{k+1}, \delta \eta_k)$,
    (4) SET $k \coloneqq k + 1$.
(III) ACCEPT $\mathbf{x}_k \rightarrow \mathbf{x}_\epsilon$.
```

Algorithm 7.6: We show the modified conjugate gradient method for the adaptive algorithm δ-`An𝕀cG/2`. Compared to the original Algorithm 6.2, we replaced the matrix-vector product in step (1) with an adaptive application of the operator $\mathbf{M}$ and a subsequent coarsening step with equally divided tolerances. Furthermore, three additional coarsening steps are introduced in step (4), which are controlled by the parameter δ.

operations. Both percentages are measured with respect to the highest level. As predicted for constant data, these come out as $N_{j_0}/N_J = 0.0137\%$. The last three columns are used to express the efficiency of the adaptive method on each level. First, the column labelled M holds the percentage of coefficients compared to the highest level, N_j/N_J, which is a known quantity included for comparison. The column S displays the quotient of the average number of nonzero coefficients V to the full number M on the respective level j. A value of 100% for S corresponds to no adaptation at all, while smaller numbers indicate higher degrees of adaptive efficiency. Finally, the quantity N_{ad} denotes the average number of nonzero coefficients.

The basic situation of constant data constitutes a test case for every adaptive method. Since the problem is already solved exactly on the lowest level, no additional coefficients should be introduced on higher levels. Our algorithm `An𝕀cG/2` handles this case well, using the minimum count of degrees of freedom $N_{\text{ad}} = 9$.

For our next experiment, we select the non-smooth $f = h_1$ from (6.3.3). The corresponding results are listed in the lower half of Table 7.1. The first three columns are similar to the uniform situation, we can see that the iteration numbers are asymptotically constant over the levels. The column P shows low values up to 1.55% on the highest level, which hints at a potentially high efficiency of the adaptive method. However, the next column V, which is a more significant measure of the computational work, goes up to about 70%. The numbers in column S, which indicate the average number of nonzero coefficients per level, stay between about 70% and 85%.

We have also included the column $\epsilon_{\text{P}}(y)$ to display the error of the solution compared to a reference solution computed with uniform discretisation to high accuracy. This is the same quantity as introduced in the previous chapter. Since the errors are already small on the lower levels, convergence starts relatively late, that is only at level 9. From then convergence is linear, in agreement with the theory.

Example 7.2 – Introduction of Additional Coarsening Steps inside the CG Loop

Having established that convergence appears unharmed by the coarsening of the wavelet expansions, we presume at this point that the efficiency of the adaptive algorithm can be further improved. Specifically,

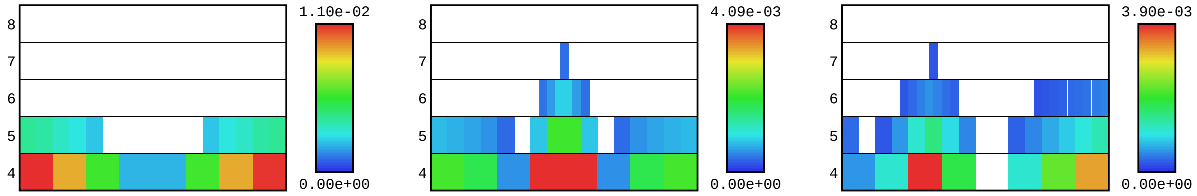


Figure 7.1: We display the distribution of wavelet coefficients of the solution y for three different choices of right hand side f. The left picture corresponds to the smooth function $f(x) = \cos(\pi x)$. In the middle and on the right, we have chosen the non-smooth functions from (6.3.3) *and* (6.3.5)*, respectively. All results have been obtained with* $J = 8$ *and* $\delta = 0.25$.

the variables $\mathbf{x}$, $\mathbf{q}$ and $\mathbf{d}$ in the CG method (see Algorithm 6.2) are so far allowed to evolve as in the uniform scheme, which motivates the introduction of additional coarsening operations. We propose a modified inner loop as given in Algorithm 7.6, where the coarsening threshold is modulated by an additional parameter $\delta \geq 0$. This leads to the algorithm δ-`AnIIcG/2`. There are two points which should be noted.

- Our basic adaptive CG method is recovered as a special case by setting $\delta = 0$.
- Values of $\delta > 0$ lead to a heuristic modification which might influence the convergence of the CG method. We examine in numerical experiments what range of values of δ is applicable.

We repeat our previous numerical simulations for $f = h_1$ with various parameters $\delta > 0$. Results for the values $\delta = \frac{1}{1000}, \frac{1}{100}$ and $\frac{1}{10}$ are provided from top to bottom in Table 7.2. Firstly, we observe that the convergence behaviour is completely unchanged. Iteration numbers, residuals and errors differ at most a few percent from the previous results in the lower half of Table 7.1. The percentage of nonzero coefficients in the solution as listed in the column P is also almost identical.

Examining column S, we see that already the small values $\delta = \frac{1}{1000}$ and $\delta = \frac{1}{100}$ gain about 10% efficiency compared to $\delta = 0$. We conclude that the auxiliary variables of the CG iterations indeed contain a substantial number of nonzero coefficients of very small absolute value, which are effectively eliminated by these small values of δ without any negative effect on the convergence. An additional 10% is saved by $\delta = \frac{1}{10}$.

These findings lead us to an examination of even larger values of δ. The results are listed in Table 7.3. We have restricted ourselves to a more condensed representation here, showing only the iteration numbers #O and the average percentage S for values of δ between 0.2 and 0.5. We observe that a reduction of S to 40% is possible without affecting the convergence. An additional drop by 10% comes at the cost of slower convergence, and the method does not converge any longer for $\delta = 0.6$. We conclude that values from about $\delta = 0.4$ on affect the CG algorithm in a critical way. Summarising our experiments with this choice of right hand side f, we can say that a sensible choice of δ almost doubles the efficiency of the adaptive method in this case, by a reduction of nonzero coefficients from about 70% to 40%.

To obtain more information on appropriate choices of δ, we repeat the above experiments with a different right hand side $f = h_2$ as given in (6.3.5). The results are very similar to our first choice of f, hence the corresponding table is omitted. Again, already very small values of δ lead to a reduction of the computational costs by 10% to 15%. By increasing δ further, up to a value of 0.3, we achieve twice the efficiency as for the original adaptive CG method, namely we come from 78% down to 37% without affecting the convergence. However, the algorithm becomes very sensitive from there, and convergence deteriorates rapidly from about $\delta = 0.4$ on.

In Figure 7.1 we present graphical representations of the solution y for three different choices of the right hand side. Each image contains a colour-coded array of wavelet coefficients, where the level $j = j_0 + 1, \ldots, J$ is mapped to the y-axis, and the coefficients for each level are arranged horizontally corresponding

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.27e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.87e-03
5	6.99e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.87e-03
6	2.80e-03	1	0.0137%	0.0694%	0.0992%	70.0%	45	8.87e-03
7	1.53e-03	2	0.0137%	0.136%	0.197%	69.0%	89	8.56e-03
8	8.17e-04	3	0.0137%	0.266%	0.392%	67.9%	174	8.38e-03
9	3.85e-04	4	0.0168%	0.568%	0.783%	72.5%	372	6.10e-03
10	1.54e-04	5	0.0259%	1.06%	1.56%	67.9%	695	3.30e-03
11	1.16e-04	4	0.0519%	2.13%	3.13%	68.1%	1396	1.65e-03
12	5.63e-05	4	0.0977%	4.37%	6.25%	69.9%	2864	8.39e-04
13	2.66e-05	4	0.195%	8.59%	12.5%	68.7%	5630	4.21e-04
14	1.31e-05	4	0.389%	19.1%	25%	76.4%	12518	2.14e-04
15	6.37e-06	4	0.777%	33.5%	50%	67.0%	21955	1.09e-04
16	3.55e-06	4	1.55%	61.4%	100%	61.4%	40240	5.79e-05

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.27e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.87e-03
5	6.99e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.87e-03
6	2.80e-03	1	0.0137%	0.0668%	0.0992%	67.3%	44	8.87e-03
7	1.54e-03	2	0.0137%	0.132%	0.197%	67.0%	87	8.56e-03
8	8.17e-04	3	0.0137%	0.253%	0.392%	64.5%	166	8.38e-03
9	3.86e-04	4	0.0168%	0.545%	0.783%	69.6%	357	6.10e-03
10	1.54e-04	5	0.0259%	1.01%	1.56%	64.7%	662	3.30e-03
11	1.16e-04	4	0.0519%	2.05%	3.13%	65.5%	1344	1.65e-03
12	5.62e-05	4	0.0977%	4.21%	6.25%	67.4%	2759	8.39e-04
13	2.66e-05	4	0.195%	8.29%	12.5%	66.3%	5433	4.21e-04
14	1.31e-05	4	0.389%	16.7%	25%	66.8%	10945	2.14e-04
15	6.37e-06	4	0.777%	32.6%	50%	65.2%	21365	1.09e-04
16	3.54e-06	4	1.55%	60.2%	100%	60.2%	39453	5.79e-05

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.27e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.87e-03
5	6.99e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.87e-03
6	2.80e-03	1	0.0137%	0.0473%	0.0992%	47.7%	31	8.87e-03
7	1.58e-03	2	0.0137%	0.0954%	0.197%	48.4%	63	8.56e-03
8	8.21e-04	3	0.0137%	0.188%	0.392%	48.0%	123	8.38e-03
9	3.95e-04	4	0.0168%	0.405%	0.783%	51.7%	265	6.10e-03
10	1.58e-04	5	0.0259%	0.745%	1.56%	47.8%	488	3.30e-03
11	1.17e-04	4	0.0519%	1.59%	3.13%	50.8%	1042	1.65e-03
12	5.70e-05	4	0.0992%	3.30%	6.25%	52.8%	2163	8.26e-04
13	2.57e-05	4	0.198%	6.40%	12.5%	51.2%	4194	4.14e-04
14	1.31e-05	4	0.397%	13.0%	25%	52.0%	8520	2.10e-04
15	6.49e-06	4	0.79%	25.7%	50%	51.4%	16843	1.08e-04
16	3.60e-06	4	1.58%	50.6%	100%	50.6%	33162	5.72e-05

Table 7.2: We list a series of experiments with $f = h_1$ *where we have varied the extra coarsening ratio* δ *from* 0.001 *(top) over* 0.01 *(middle) to* 0.1 *(bottom).*

	$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$		$\delta = 0.5$	
j	#O	S	#O	S	#O	S	#O	S
4	0	52.9%	0	52.9%	0	52.9%	0	52.9%
5	0	27.2%	0	27.2%	0	27.2%	0	27.2%
6	1	41.1%	1	34.6%	1	29.2%	1	25.0%
7	2	40.9%	2	34.9%	2	30.0%	2	26.1%
8	3	41.1%	3	35.5%	3	31.1%	3	27.8%
9	4	44.3%	4	38.2%	4	34.2%	4	30.4%
10	5	40.8%	5	39.0%	5	35.0%	6	28.5%
11	4	45.0%	4	40.6%	5	34.2%	5	30.5%
12	4	46.9%	4	42.1%	5	32.2%	5	30.6%
13	4	45.1%	4	41.2%	5	32.2%	5	30.1%
14	4	45.2%	4	42.0%	5	32.2%	5	29.2%
15	4	44.8%	4	40.8%	4	37.6%	5	29.6%
16	4	44.6%	4	40.5%	4	36.7%	5	29.2%

Table 7.3: We show selected results for a further increase in the supplemental adaptation parameter δ. For each value of δ given in the first row, we list the iteration numbers #O and the average percentage S, which are meant to be compared against the equally labelled columns in Table 7.2. As before, we have fixed $f = h_1$ from (6.3.3). *For a value of $\delta = 0.6$ the algorithm diverges.*

to the location k. Since the number of coefficients per level increases geometrically proportional to 2^{nj}, the higher levels contain narrower rectangles. We have omitted the coefficients for the generators $\phi_{j_0,k}$, since they contain no useful information for the adaptive scheme, and display the oscillatory parts instead, namely the coefficients for the wavelets $\psi_{j,k}$.

First, we have selected the smooth function $f(x) = \cos(\pi x)$. The solution contains few significant coefficients on the lowest level, and only small contributions from the second lowest, indicating that it is indeed very smooth. The second and third examples have been obtained for peaked functions f from (6.3.3) and (6.3.5), respectively. It can be seen that these right hand sides f indeed induce a locally higher resolution around their singularities at $x = \frac{1}{2}$ (left) and $x = \frac{1}{3}$ (right), which is exactly the behaviour which we expect from an adaptive numerical method.

Example 7.3 – Varying the Smoothness of the Right Hand Side

We now examine the influence of decreasing smoothness of the right hand side. To this end, we modify the function h_1 from (6.3.3) by the shift operator $D_r : H^s \to H^{s-r}$ (2.2.52). (We use a scaled version here which preserves the function values on average.) Since h_1 lies in a Sobolev space between $H^{1/2}$ and H^1, a shift of $r = \frac{3}{2}$ produces a function of a smoothness between $H^{-1/2}$ and H^{-1}. We display three versions in Figure 7.2, roughened by the exponents $r = \frac{1}{2}$, 1 and $\frac{3}{2}$.

The results for the solution of the elliptic systems with the right hand sides $f = D_{1/2}h_1$, $D_1 h_1$ and $h_3 := D_{3/2}h_1$ are listed in Table 7.4. It turns out that the decreasing smoothness of the right hand side has negligible effects on the iteration numbers, the percentages and the errors. To investigate further in what way the rough right hand side influences the solution, we compare the distributions of the wavelet coefficients of y for the two cases $f = h_1$ and $f = h_3$, shown in Figure 7.3. While the general shape of both distributions is similar, the diagram for $f = h_3$ shows a stronger concentration of coefficients around $x = \frac{1}{2}$, and higher absolute values on the levels 4 and 5. The oscillations in h_3 are thus transferred to the solution y, but with strongly reduced amplitude. Since the same target accuracy is achieved for this irregular right hand side by an adequately adjusted set of coefficients, we assert the correct performance of the adaptive scheme.

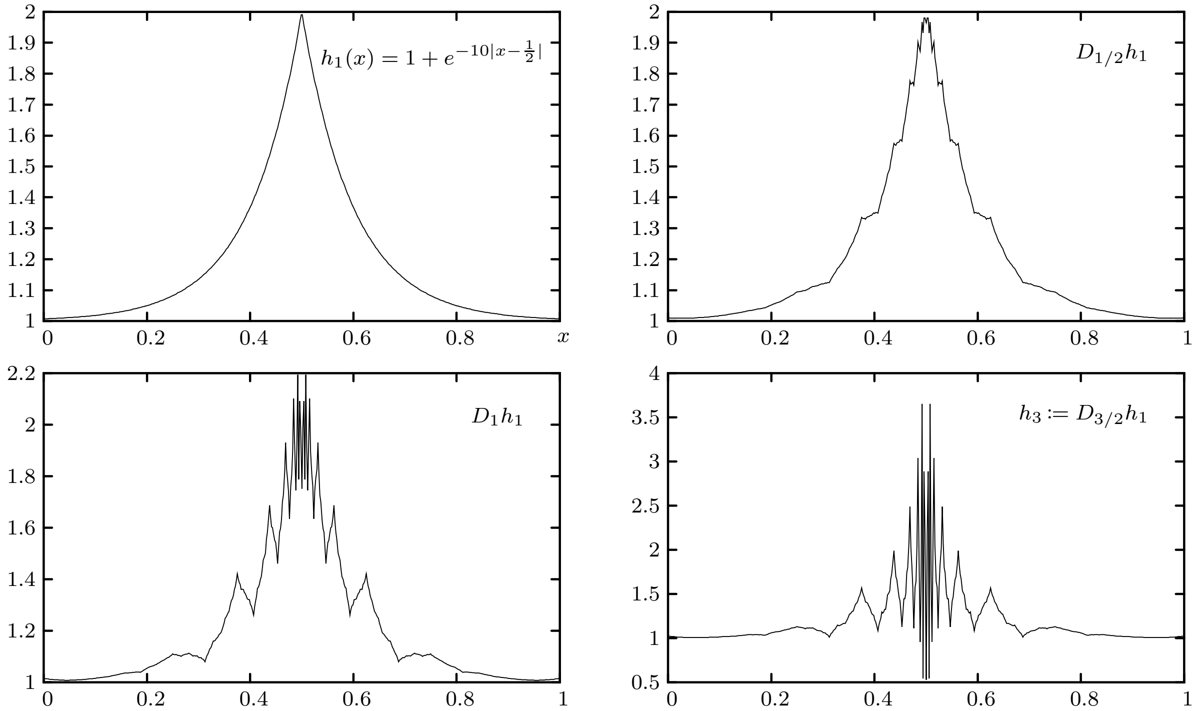


Figure 7.2: These plots show the function h_1 and three versions with successively decreased smoothness, obtained by applications of the shift operator D_r with exponents $r = \frac{1}{2}$, 1 and $\frac{3}{2}$, respectively. The resolution used here is $J = 9$.

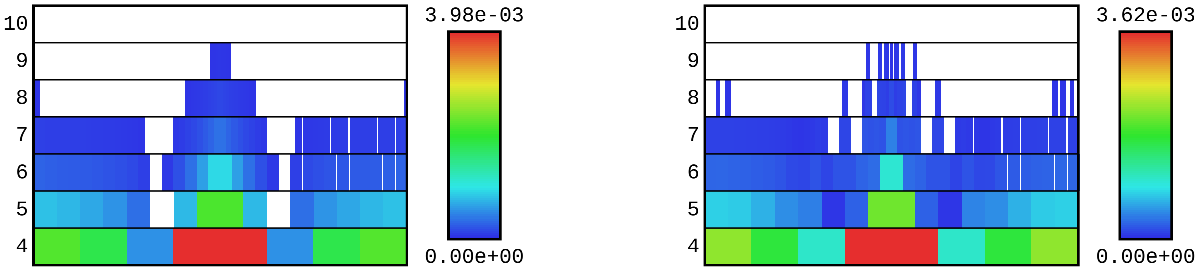


Figure 7.3: We show the solution y of the single elliptic system for the right hand sides $f = h_1$ (left) and $f = h_3$ (right). We have set $\delta = 0.2$.

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.22e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.81e-03
5	6.98e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.81e-03
6	2.83e-03	1	0.0137%	0.042%	0.0992%	42.3%	28	8.81e-03
7	1.56e-03	2	0.0137%	0.0793%	0.197%	40.3%	52	8.49e-03
8	8.03e-04	3	0.0137%	0.159%	0.392%	40.6%	104	8.32e-03
9	4.20e-04	4	0.0168%	0.347%	0.783%	44.3%	227	6.10e-03
10	1.75e-04	5	0.0259%	0.641%	1.56%	41.1%	420	3.31e-03
11	1.19e-04	4	0.0504%	1.42%	3.13%	45.4%	931	1.69e-03
12	5.72e-05	4	0.0992%	2.94%	6.25%	47.0%	1927	8.33e-04
13	2.73e-05	4	0.201%	5.78%	12.5%	46.2%	3788	4.17e-04
14	1.35e-05	4	0.398%	11.7%	25%	46.8%	7668	2.12e-04
15	6.07e-06	4	0.801%	22.6%	50%	45.2%	14811	1.07e-04
16	3.25e-06	4	1.59%	45.1%	100%	45.1%	29557	5.62e-05

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.14e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.70e-03
5	6.93e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.70e-03
6	2.81e-03	1	0.0137%	0.0412%	0.0992%	41.5%	27	8.70e-03
7	1.57e-03	2	0.0137%	0.0792%	0.197%	40.2%	52	8.39e-03
8	7.77e-04	3	0.0137%	0.162%	0.392%	41.3%	106	8.22e-03
9	4.05e-04	4	0.0168%	0.342%	0.783%	43.7%	224	6.10e-03
10	1.72e-04	5	0.0259%	0.644%	1.56%	41.3%	422	3.31e-03
11	1.20e-04	4	0.0504%	1.44%	3.13%	46.0%	944	1.69e-03
12	5.78e-05	4	0.102%	2.95%	6.25%	47.2%	1933	8.31e-04
13	2.72e-05	4	0.204%	5.82%	12.5%	46.6%	3814	4.21e-04
14	1.31e-05	4	0.412%	11.5%	25%	46.0%	7537	2.11e-04
15	6.64e-06	4	0.815%	23.1%	50%	46.2%	15139	1.08e-04
16	3.61e-06	4	1.63%	45.8%	100%	45.8%	30016	5.74e-05

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	5.02e-03	0	0.0137%	0.0137%	0.0259%	52.9%	9	8.52e-03
5	6.95e-03	0	0.0137%	0.0137%	0.0504%	27.2%	9	8.52e-03
6	2.78e-03	1	0.0137%	0.042%	0.0992%	42.3%	28	8.52e-03
7	1.54e-03	2	0.0137%	0.079%	0.197%	40.1%	52	8.21e-03
8	7.41e-04	3	0.0137%	0.161%	0.392%	41.1%	106	8.05e-03
9	3.84e-04	4	0.0153%	0.338%	0.783%	43.2%	222	7.09e-03
10	1.96e-04	5	0.0259%	0.691%	1.56%	44.3%	453	3.35e-03
11	6.98e-05	5	0.0473%	1.27%	3.13%	40.6%	832	1.69e-03
12	5.67e-05	4	0.105%	2.84%	6.25%	45.4%	1861	8.38e-04
13	2.69e-05	4	0.204%	5.85%	12.5%	46.8%	3834	4.22e-04
14	1.35e-05	4	0.415%	11.6%	25%	46.4%	7602	2.10e-04
15	5.88e-06	4	0.821%	22.6%	50%	45.2%	14811	1.06e-04
16	3.20e-06	4	1.65%	45.0%	100%	45.0%	29492	5.46e-05

Table 7.4: These listings contain results obtained with the roughened right hand sides (from top to bottom) $f = D_{1/2}h_1$, $D_1 h_1$ and $D_{3/2}h_1$, and a coarsening parameter $\delta = 0.2$.

j	$\lVert \mathbf{r}_j \rVert$	#O	P	V	M	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$
4	1.22e-02	0	0.00771%	0.00771%	0.0275%	28.0%	81	1.37e-02
5	3.48e-03	1	0.00771%	0.0602%	0.104%	57.9%	632	1.37e-02
6	2.34e-03	2	0.00771%	0.295%	0.402%	73.4%	3099	1.31e-02
7	1.22e-03	3	0.00771%	1.10%	1.58%	69.6%	11557	1.26e-02
8	7.16e-04	4	0.00771%	4.10%	6.29%	65.2%	43076	1.23e-02
9	4.16e-04	5	0.0109%	17.7%	25%	70.8%	185961	9.10e-03
10	2.05e-04	6	0.0304%	61.2%	100%	61.2%	642983	4.55e-03

$\delta = 0$

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
j	#O	S	#O	S	#O	S	#O	S	#O	S
4	0	28.0%	0	28.0%	0	28.0%	0	28.0%	0	28.0%
5	1	39.2%	1	22.0%	1	18.4%	1	14.8%	1	12.9%
6	2	52.0%	2	33.8%	2	27.4%	2	23.3%	2	20.0%
7	3	49.3%	3	32.5%	3	26.6%	3	22.2%	3	19.1%
8	4	47.4%	4	32.6%	4	27.8%	4	24.3%	4	21.1%
9	5	49.2%	5	34.3%	5	28.5%	5	24.8%	7	16.3%
10	6	44.8%	6	30.9%	6	26.3%	7	21.2%	5	26.4%

j	$\lVert \mathbf{r}_j \rVert$	#O	P	V	M	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$
4	1.17e-02	0	0.00771%	0.00771%	0.0275%	28.0%	81	1.32e-02
5	3.60e-03	1	0.00771%	0.0199%	0.104%	19.1%	209	1.32e-02
6	2.40e-03	2	0.00771%	0.108%	0.402%	26.9%	1135	1.25e-02
7	1.30e-03	3	0.00771%	0.409%	1.58%	25.9%	4297	1.21e-02
8	7.12e-04	4	0.00771%	1.74%	6.29%	27.7%	18281	1.18e-02
9	4.12e-04	5	0.00999%	7.03%	25%	28.1%	73859	9.58e-03
10	2.27e-04	6	0.0247%	27.6%	100%	27.6%	289973	4.77e-03

$\delta = 0.2$

Table 7.5: We display the results for the elliptic system in two dimensions with right hand side $f_{2,2} = h_1 \otimes h_1$ in the top and middle tables, and $D_{3/2} f_{2,2}$ in the bottom table.

Example 7.4 – Efficiency in Higher Dimensions

We complement above results in one spatial dimension with calculations in two and three dimensions, testing different scenarios with respect to the isotropic or anisotropic character and the smoothness of the right hand side.

We begin with a right hand side of $f_{2,2} := h_1 \otimes h_1$, with h_1 as in (6.3.3). The results displayed in Table 7.5 have similar structure as in the one-dimensional case, with generally reduced percentages S. Again, a parameter of $\delta = 0.2$ appears most sensible, reducing the number of nonzero coefficients by more than a factor of 2. In the bottom table, we have changed the right hand side to the rough version $D_{3/2} f_{2,2}$, which leads to almost identical percentages as in the middle table for $\delta = 0.2$.

For Table 7.6 we have selected $f_{2,1} := h_1 \otimes 1$ and observe that the adaptive method indeed reacts to the smoothness of the solution in one direction, as the number of nonzero coefficients goes down by a factor of 5 for a supplemental adaptation parameter of $\delta = 0.3$. For $\delta = 0$, the effect of the anisotropy is not nearly that strong, which confirms again that the introduction of δ is beneficial to obtain a greater adaptive efficiency. The choice of the roughened function $D_{3/2} f_{2,1}$ in the bottom table produces about the same percentages as in the middle table at $\delta = 0.2$.

Finally, experiments in three dimensions are covered in an additional set of tables, which have the same

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	6.39e-03	0	0.00771%	0.00771%	0.0275%	28.0%	81	8.04e-03
5	1.60e-03	1	0.00771%	0.0375%	0.104%	36.1%	394	8.04e-03
6	3.02e-03	1	0.00771%	0.256%	0.402%	63.7%	2690	8.04e-03
7	1.49e-03	2	0.00771%	0.653%	1.58%	41.3%	6861	7.68e-03
8	7.76e-04	3	0.00771%	2.93%	6.29%	46.6%	30783	7.40e-03
9	4.49e-04	4	0.00771%	14.3%	25%	57.2%	150239	7.23e-03
10	2.12e-04	6	0.0109%	51.9%	100%	51.9%	545274	4.52e-03

$\delta = 0$

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
j	#O	S	#O	S	#O	S	#O	S	#O	S
4	0	28.0%	0	28.0%	0	28.0%	0	28.0%	0	28.0%
5	1	11.4%	1	6.40%	1	5.36%	1	5.29%	1	5.29%
6	1	23.7%	1	17.7%	1	15.1%	1	12.6%	1	10.3%
7	2	21.0%	2	15.2%	2	12.5%	2	10.6%	2	8.67%
8	3	20.2%	3	14.4%	3	12.1%	3	10.5%	3	9.25%
9	4	21.8%	4	15.5%	4	13.2%	4	11.4%	6	7.20%
10	6	21.8%	6	14.5%	6	11.9%	6	10.2%	5	9.98%

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	6.17e-03	0	0.00771%	0.00771%	0.0275%	28.0%	81	7.73e-03
5	1.80e-03	1	0.00771%	0.00602%	0.104%	5.79%	63	7.73e-03
6	2.95e-03	1	0.00771%	0.0602%	0.402%	15.0%	632	7.73e-03
7	1.46e-03	2	0.00771%	0.201%	1.58%	12.7%	2112	7.36e-03
8	7.44e-04	3	0.00771%	0.784%	6.29%	12.5%	8237	7.10e-03
9	4.20e-04	4	0.00771%	3.21%	25%	12.8%	33725	6.94e-03
10	2.11e-04	6	0.0106%	11.4%	100%	11.4%	119771	4.65e-03

$\delta = 0.2$

Table 7.6: Here we have selected the anisotropic right hand sides $f_{2,1} = h_1 \otimes 1$ (top and middle tables) and $D_{3/2} f_{2,1}$ (bottom table). The percentages are lower than for the isotropic case.

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	1.86e-03	1	0.0336%	0.102%	0.229%	44.5%	2190	2.87e-01
5	6.48e-03	5	0.0339%	1.12%	1.67%	67.1%	24043	1.45e-01
6	3.06e-03	6	0.034%	8.10%	12.8%	63.3%	173882	1.80e-02
7	1.38e-03	3	0.034%	58.4%	100%	58.4%	1253666	1.72e-02

$\delta = 0$

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
j	#O	S	#O	S	#O	S	#O	S	#O	S
4	1	21.8%	1	10.3%	1	10.2%	1	10.2%	1	10.2%
5	5	40.4%	5	24.0%	5	19.6%	5	17.3%	5	14.5%
6	6	33.2%	6	17.3%	6	13.7%	6	11.6%	6	10.0%
7	3	33.9%	3	23.8%	3	20.8%	3	18.3%	3	16.5%

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	2.17e-03	1	0.0336%	0.0252%	0.229%	11.0%	541	2.87e-01
5	6.40e-03	5	0.0339%	0.321%	1.67%	19.2%	6891	1.44e-01
6	3.11e-03	6	0.034%	1.69%	12.8%	13.2%	36279	1.74e-02
7	1.47e-03	3	0.034%	21.0%	100%	21.0%	450805	1.66e-02

$\delta = 0.2$

Table 7.7: Here we present the three-dimensional example with a tensor product right hand side $f_{3,3} = h_1 \otimes h_1 \otimes h_1$, and all other parameters unchanged. The bottom table corresponds to the roughened right hand side $D_{3/2} f_{3,3}$.

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	1.43e-02	0	0.0335%	0.0335%	0.229%	14.6%	719	2.74e-01
5	6.56e-03	5	0.0338%	0.908%	1.67%	54.4%	19492	1.50e-01
6	3.44e-03	6	0.034%	6.67%	12.8%	52.1%	143184	1.57e-02
7	1.41e-03	3	0.034%	43.7%	100%	43.7%	938103	1.43e-02

$\delta = 0$

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
j	#O	S	#O	S	#O	S	#O	S	#O	S
4	0	14.6%	0	14.6%	0	14.6%	0	14.6%	0	14.6%
5	5	30.7%	5	19.2%	5	15.7%	5	13.3%	5	11.5%
6	6	23.9%	6	12.4%	6	9.92%	6	7.89%	6	6.96%
7	3	21.7%	3	15.0%	3	12.6%	3	10.7%	3	9.46%

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	1.43e-02	0	0.0335%	0.0335%	0.229%	14.6%	719	2.74e-01
5	6.93e-03	5	0.0338%	0.263%	1.67%	15.7%	5646	1.49e-01
6	3.45e-03	6	0.034%	1.27%	12.8%	9.92%	27263	1.52e-02
7	1.42e-03	3	0.034%	12.6%	100%	12.6%	270483	1.38e-02

$\delta = 0.2$

Table 7.8: The top and middle table contain results for the right hand side $f_{3,2} = h_1 \otimes h_1 \otimes 1$. Just as in the previous example, the bottom table corresponds to the roughened version $D_{3/2} f_{3,2}$.

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	7.80e-03	0	0.0334%	0.0334%	0.229%	14.6%	717	2.96e-01
5	5.49e-03	6	0.0338%	0.92%	1.67%	55.1%	19750	1.46e-01
6	3.51e-03	6	0.0339%	6.71%	12.8%	52.4%	144043	7.40e-02
7	1.69e-03	7	0.034%	41.0%	100%	41.0%	880142	1.08e-02

$\delta = 0$

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$	
j	#O	S	#O	S	#O	S	#O	S	#O	S
4	0	14.6%	0	14.6%	0	14.6%	0	14.6%	0	14.6%
5	6	24.7%	6	16.0%	6	13.2%	6	11.1%	6	9.64%
6	6	15.5%	6	8.52%	6	6.5%	6	5.41%	6	4.58%
7	7	10.6%	7	4.9%	7	3.7%	7	3.03%	8	2.33%

j	$\|\mathbf{r}_j\|$	#O	P	V	M	S	N_{ad}	$\epsilon_{\text{P}}(y)$
4	7.80e-03	0	0.0334%	0.0334%	0.229%	14.6%	717	2.96e-01
5	5.89e-03	6	0.0338%	0.221%	1.67%	13.2%	4744	1.46e-01
6	3.40e-03	6	0.0339%	0.832%	12.8%	6.5%	17860	7.40e-02
7	1.76e-03	7	0.034%	3.7%	100%	3.7%	79427	1.09e-02

$\delta = 0.2$

Table 7.9: Here the right hand side has been specified to $f_{3,1} = h_1 \otimes 1 \otimes 1$, which is smooth with respect to the y and z dimensions. The bottom table deals with the rough version $D_{3/2} f_{3,1}$.

layout as the tables for two dimensions. For Table 7.7, we have set the right hand side to $f_{3,3} := h_1 \otimes h_1 \otimes h_1$, which is isotropic and non-smooth in all dimensions. Table 7.8 contains results for the intermediate case $f_{3,2} := h_1 \otimes h_1 \otimes 1$, and Table 7.9 corresponds to the anisotropic case $f_{3,1} := h_1 \otimes 1 \otimes 1$, which is the smoothest of these three examples. The last table on each page lists results for the roughened version $D_{3/2} f_{3,i}$ of the respective right hand side.

These tables consistently demonstrate that the isotropic and least smooth right hand side requires the most degrees of freedom, while the smoothest and most anisotropic right hand side needs the fewest. The intermediate right hand side yields values in between the two extreme cases. This demonstrates that the adaptive strategy reacts to the smoothness in some of the three dimensions by a substantial elimination of small coefficients. Altogether, the efficiency is substantially higher than for the above examples in two dimensions.

The topmost tables on each page, which contain results for $\delta = 0$, show only moderate differences in percentage. The choice of $\delta = 0.2$ however, for both the standard and the roughened right hand sides, produces a greater variation of adaptive efficiency between the three examples. The percentage goes down from around 20% over 13% to below 4%. This last example shows a gain in efficiency of a factor 10 by the introduction of the parameter δ. It is also the only one which exhibits a significant decrease of the percentages with the level of resolution, which indicates superlinear convergence of the adaptive approximation.

Conclusions

The numerical solution of a single elliptic system with our adaptive conjugate gradient method yields the same convergence behaviour as the uniform discretisation scheme, with a significantly reduced count of nonzero coefficients. The number of iterations on each level of resolution j stays constant, exactly as in the uniform case. Convergence generally attains linear behaviour from about level $j = 9$ on, meaning that the error in H^1 is asymptotically proportional to 2^{-j}.

We have supplemented the adaptive algorithm with three additional coarsening steps in the inner loop of the conjugate gradient routine, parametrised by $\delta \geq 0$. Values around $\delta = 0.2$ have been found optimal in the sense that they lead to a significant increase of efficiency without impairing the convergence. The savings induced by this additional parameter become larger with higher spatial dimension.

We conclude that the adaptive method is most effective in two or three dimensions for several reasons. Firstly, the cost in memory and computation time of uniform discretisations grows exponentially with the dimension, making them unacceptably expensive already for relatively low levels of resolution. Secondly, the potential to gain efficiency is larger compared to one dimension, since many samples of data may be non-smooth in one direction, while exhibiting smoothness in another. Finally, the effect of the additional coarsening parameter δ also grows with the dimension, enabling possibly large additional savings in the computational cost.

7.4.2 Results for the Optimal Control Problem

After the studies of the solution of a single elliptic system with the adaptive conjugate gradient method in the previous section, we finally discuss results of the full control problem in adaptive discretisation. The supplemental adaption parameter δ is incorporated in both the inner as well as the outer elliptic solvers just as specified in Algorithm 7.6. We continue to compare the unmodified case $\delta = 0$ to the setting $\delta = 0.2$.

We examine combinations of data and parameters which have already been discussed in the context of the uniform discretisation. Thereby we can compare the convergence behaviour of the adaptive algorithm with the uniform case. The solutions of the adaptive computations are measured against the same uniform high accuracy solutions which have been used as reference solutions in the previous chapter.

Additionally, we aim to measure the rate σ of error reduction with respect to the number of degrees of freedom. In analogy to (7.3.3), it is defined by the relation

$$N_{\mathrm{ad}} \sim \epsilon^{-1/\sigma}\,, \tag{7.4.1}$$

where ϵ denotes the error in H_1. For a uniform discretisation with piecewise linear wavelets, it holds that $N \sim 2^{nj}$ and $\epsilon \sim 2^{-j}$, which yields a rate of $\sigma_n = 1/n$ for smooth functions. In the following experiments, we will determine the rate of the adaptive algorithm δ-`AnIIcG/2` numerically.

Example 7.5 – One Dimension

We begin with results for the example problem 6.2 from the last chapter. Specifically, we choose $f = h_1$ from (6.3.3), $y_* \equiv 0$ and $\omega = 1$. We compare the standard L_2 case, that is $s = t = 0$, with natural norms, defined by $s = t = 1$.

The results for $U = Z = L_2$ are given in Table 7.10. We have added two columns which refer to the average number of nonzero coefficients, namely the percentage S with respect to the uniform discretisation on each level, and the absolute number N_{ad}. As before, lower values of S stand for greater adaptive efficiency.

The iteration numbers are independent of the level, whereupon we observe slight fluctuations in the amount of inner iterations and in the balance of iterations between the inner and outer loops. Up to level 12, we have one outer iteration per level which seems to reduce the residual far below the threshold of 2^{-j}, with the effect that the residual only begins to change at level 13. The control exhibits the same behaviour, the error already sets off rather low and stagnates until it begins to decrease further at level 13. The error in the state starts at a considerably higher value, but in contrast to the outer residual and the control we can see a reduction by a factor 2 between levels almost from the coarsest level on. We

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								8.87e-03	1.63e-06
4	5.01e-06	1	0	0	0	0.0%	0	8.30e-03	1.63e-06
5	5.16e-06	1	0	0	0	0.0%	0	3.83e-03	1.63e-06
6	5.17e-06	1	0	0	0	0.0%	0	2.03e-03	1.63e-06
7	5.17e-06	1	1	0	0	54.7%	71	1.04e-03	1.63e-06
8	5.17e-06	1	1	0	1	21.9%	56	5.17e-04	1.63e-06
9	5.23e-06	1	2	0	4	66.3%	340	2.61e-04	1.63e-06
10	4.28e-06	1	4	0	4	51.7%	530	1.32e-04	1.63e-06
11	4.09e-06	1	5	0	4	52.6%	1079	6.79e-05	1.63e-06
12	4.01e-06	1	5	1	5	34.8%	1427	3.58e-05	1.63e-06
13	1.38e-06	2	4	0	4	43.0%	3521	1.66e-05	5.98e-07
14	6.42e-07	2	4	1	4	40.6%	6657	7.87e-06	3.33e-07
15	6.36e-07	1	5	3	5	28.7%	9400	4.14e-06	3.33e-07
16	8.52e-08	3	3	1	5	39.9%	26137	2.71e-06	8.45e-08

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								8.87e-03	1.63e-06
4	5.01e-06	1	0	0	0	0.0%	0	8.40e-03	1.63e-06
5	5.16e-06	1	0	0	0	0.0%	0	4.20e-03	1.63e-06
6	5.17e-06	1	0	0	0	0.0%	0	2.19e-03	1.63e-06
7	5.17e-06	1	1	0	0	21.5%	28	1.09e-03	1.63e-06
8	5.17e-06	1	1	0	1	15.9%	41	5.46e-04	1.63e-06
9	5.23e-06	1	2	0	4	11.7%	60	2.81e-04	1.63e-06
10	4.79e-06	1	4	0	4	15.1%	155	1.44e-04	1.63e-06
11	4.07e-06	1	5	0	5	15.1%	308	6.95e-05	1.63e-06
12	4.02e-06	1	5	1	5	12.8%	523	3.32e-05	1.63e-06
13	1.37e-06	2	4	0	6	6.4%	528	1.66e-05	1.60e-06
14	7.34e-07	2	4	1	5	7.3%	1196	8.27e-06	1.59e-06
15	4.81e-07	3	3	1	6	4.2%	1368	4.39e-06	1.59e-06
16	1.17e-07	4	4	1	5	3.9%	2563	2.73e-06	5.03e-07
						$\sigma \approx 1.33$			

Table 7.10: We provide results for the adaptive control problem in one dimension with highest level $J = 16$, and L_2 norms for the state and the control. We have used the data $f = h_1$ and $y_* \equiv 0$. The regularisation has been set to $\omega = 1$. The top table corresponds to $\delta = 0$, and the bottom table to $\delta = 0.2$.

attribute this behaviour to the strategy of nested iteration, which provides a sufficiently accurate solution already at coarse levels at practically no cost, requiring only one outer iteration per level.

The effect of the parameter δ is twofold. Firstly, the iteration counts increase marginally on the higher levels. Secondly, we find that the percentage of nonzero coefficients drops by a factor of 10 at the highest level which is most significant with respect to numerical performance. This corresponds to a tenfold reduction in memory and computing time compared to the unmodified version of the adaptive algorithm, which is possible here since both y and u are smooth (see Figure 6.3 for graphs of these functions). The rate σ is about $\frac{4}{3}$, which is significantly larger than the predicted value 1.

While L_2 norms lead to smooth solution and control for these data, natural norms allow for singularities in the control. The corresponding results are listed in Table 7.11. The top table uses $\omega = 1$, and the bottom table covers the case of vanishing regularisation $\omega = 0$, which was inspected in the uniform setting in Example 6.3.

The results are largely similar for both settings of ω. The iteration numbers increase in comparison to L_2 norms, notably the amount of inner iterations (this has been already observed for the uniform discretisation). The error in the control starts at about the same order of magnitude as the error in the solution, and both errors reduce in a more volatile manner than for the L_2 case, still exhibiting an average factor of reduction of 2 between levels. Only the error $\epsilon_{\mathrm{P}}(y)$ for $\omega = 0$ is close to zero, as the state y converges to $y_* \equiv 0$ here, and the coarsening operation truncates toward zero.

The introduction of the parameter δ reduces the average number of nonzero coefficients down to about 50%. For this case of natural norms, the control is less regular than for L_2 norms, and hence more wavelet coefficients on high levels are significant. The rate σ is still somewhat larger than 1, with values of about 1.1.

We have included additional results in Table 7.12, using the roughened right hand side $f = h_3$ from Figure 7.2 and a target $y_* = h_2$, with $\delta = 0.2$. The top table contains data for $s = t = 0$, and the bottom table for $s = \frac{1}{2}$, $t = 0$. The overall work in terms of iteration numbers lies between the previous two examples of L_2 norms and natural norms, respectively. As in the case for L_2 norms and zero target, the percentages reduce to values below 10%, with similar behaviour of the errors and iteration numbers. In this example, the rates of convergence are largest with values close to 1.4.

Convergence plots for the one-dimensional results which have been obtained for the optimal control problem up to this point are provided in Figure 7.4. The top two tables show the decay of the errors $\epsilon_{\mathrm{P}}(y)$ and $\epsilon_{\mathrm{P}}(u)$. Here the graph for the state y for $\omega = 0$ is not shown because it is very close to zero (see above discussion). Also excluded is the control u for L_2 norms, which already starts off at a very small error, which does not decrease further during the simulation. The remaining graphs demonstrate the theoretically predicted slope of 1. The last table displays the target error $\eta_j = 2^{-j}$ versus N_{ad}, showing the rates of adaptive approximation $\sigma > 1$.

In summary, we state that our adaptive algorithm for the optimal control problem converges with the same rate of error reduction as the algorithm in uniform discretisation. The number of inner and outer iterations are again constant on average, that is, independent of the level of resolution. The savings in computational complexity compared to the full grid range from a factor of over 20 for smooth functions and L_2 norms to about 2 in the case of natural norms and singular control. The convergence rate σ of the average number of nonzero coefficients N_{ad} is superlinear in all examples.

To illustrate the distributions of coefficients, we have collected some graphical representations. Figure 7.5 deals with our well-known example with $f = h_1$ from (6.3.3) and $y_* \equiv 0$, for the L_2 case and for natural norms. It can be seen that the peak in the right hand side at $x = \frac{1}{2}$ is reflected in the pictures. Moreover, they demonstrate that the computational cost is indeed significantly reduced compared to the uniform algorithm `nIIcG/2`, and that the wavelet approach utilises different distributions of coefficients for different variables.

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								5.25e-03	1.04e-03
4	2.94e-04	4	10	10	7	90.4%	15	4.48e-03	5.64e-04
5	2.31e-04	4	13	10	8	88.2%	29	3.66e-03	3.01e-04
6	6.23e-05	6	13	9	8	83.8%	54	1.90e-03	9.59e-05
7	4.85e-05	4	12	9	8	82.1%	106	9.72e-04	6.21e-05
8	1.33e-05	6	11	8	7	82.2%	211	4.77e-04	3.54e-05
9	1.62e-05	5	11	9	9	76.3%	391	2.41e-04	1.26e-05
10	7.56e-06	5	11	10	9	73.2%	751	1.22e-04	1.16e-05
11	2.36e-06	6	11	9	9	66.0%	1352	6.30e-05	3.81e-06
12	1.94e-06	5	10	9	9	61.5%	2519	3.14e-05	1.86e-06
13	9.90e-07	5	11	9	9	59.5%	4871	1.68e-05	1.19e-06
14	2.63e-07	6	11	9	9	57.8%	9470	7.82e-06	4.66e-07
15	2.31e-07	4	12	9	9	52.4%	17185	3.99e-06	2.21e-07
16	1.34e-07	5	11	9	9	54.7%	35832	2.29e-06	6.35e-08
						$\sigma \approx 1.08$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3							3.23e-03	2.09e-03
4	1.05e-03	2	10	8	94.1%	16	2.24e-11	1.59e-03
5	5.83e-04	3	13	10	91.1%	30	2.24e-11	8.72e-04
6	2.03e-04	4	13	8	86.0%	56	2.24e-11	3.17e-04
7	1.70e-04	3	14	8	82.1%	106	2.24e-11	1.72e-04
8	9.10e-05	3	12	9	81.0%	208	2.24e-11	1.23e-04
9	2.38e-05	5	11	7	77.5%	398	2.24e-11	2.17e-05
10	2.19e-05	1	15	12	69.1%	708	2.24e-11	2.17e-05
11	1.11e-05	3	12	10	69.8%	1430	2.24e-11	1.32e-05
12	3.19e-06	5	10	9	62.8%	2571	2.24e-11	3.37e-06
13	1.78e-06	3	11	9	59.0%	4832	2.24e-11	2.33e-06
14	1.27e-06	3	10	10	55.1%	9025	2.24e-11	1.52e-06
15	4.99e-07	4	10	9	48.5%	15877	2.24e-11	5.32e-07
16	2.81e-07	3	10	9	52.3%	34281	2.24e-11	3.67e-07
					$\sigma \approx 1.09$			

Table 7.11: These results have been obtained in the same scenario as those in the table on the previous page, but with natural norms instead ($s = t = 1$). The top table has been computed with $\omega = 1$, and the bottom table with $\omega = 0$. Both use the parameter $\delta = 0.2$.

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								8.45e-03	7.34e-05
4	1.87e-04	1	2	1	3	60.3%	10	7.91e-03	7.34e-05
5	1.68e-04	1	5	1	5	65.5%	22	4.23e-03	7.34e-05
6	1.52e-04	1	7	2	5	61.2%	40	2.10e-03	7.34e-05
7	1.48e-04	1	7	3	5	58.4%	75	1.09e-03	7.34e-05
8	5.96e-05	2	4	2	6	41.4%	106	5.59e-04	7.26e-05
9	2.85e-05	3	5	3	5	28.7%	147	2.82e-04	7.20e-05
10	1.34e-05	4	6	2	6	23.7%	243	1.46e-04	1.70e-05
11	6.38e-06	3	5	3	5	20.2%	415	7.58e-05	1.69e-05
12	2.45e-06	4	6	2	5	14.0%	574	3.80e-05	1.69e-05
13	2.45e-06	4	6	2	5	15.3%	1252	1.85e-05	3.32e-06
14	1.39e-06	3	6	3	5	14.4%	2354	9.24e-06	3.20e-06
15	5.23e-07	4	6	2	5	10.3%	3380	4.83e-06	2.12e-06
16	1.99e-07	5	6	2	5	8.2%	5343	3.03e-06	7.02e-07
						$\sigma \approx 1.36$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								8.41e-03	3.15e-04
4	3.77e-04	1	2	1	5	69.0%	12	7.87e-03	3.15e-04
5	3.84e-04	1	3	1	5	74.2%	24	4.19e-03	3.15e-04
6	2.37e-04	2	4	2	6	55.0%	36	2.09e-03	3.16e-04
7	1.41e-04	3	4	1	6	49.2%	63	1.10e-03	3.15e-04
8	6.11e-05	4	6	1	7	41.1%	106	5.58e-04	1.45e-04
9	2.21e-05	4	4	2	6	34.5%	177	2.86e-04	1.20e-04
10	2.05e-05	4	6	2	5	28.3%	290	1.47e-04	5.03e-05
11	7.55e-06	4	4	2	6	25.4%	520	7.70e-05	2.30e-05
12	4.24e-06	4	3	2	6	18.8%	769	3.78e-05	1.55e-05
13	2.55e-06	4	4	2	6	16.6%	1364	1.85e-05	8.24e-06
14	8.41e-07	5	3	2	5	10.7%	1745	9.26e-06	5.08e-06
15	5.10e-07	5	4	2	5	10.0%	3291	4.81e-06	2.46e-06
16	2.10e-07	5	4	2	5	7.7%	5038	3.08e-06	1.08e-06
						$\sigma \approx 1.39$			

Table 7.12: These tables correspond to the data $f = h_3$ and $y_* = h_2$, computed with $\delta = 0.2$. The top table contains results for $s = t = 0$, and the bottom table for $s = \frac{1}{2}$, $t = 0$.

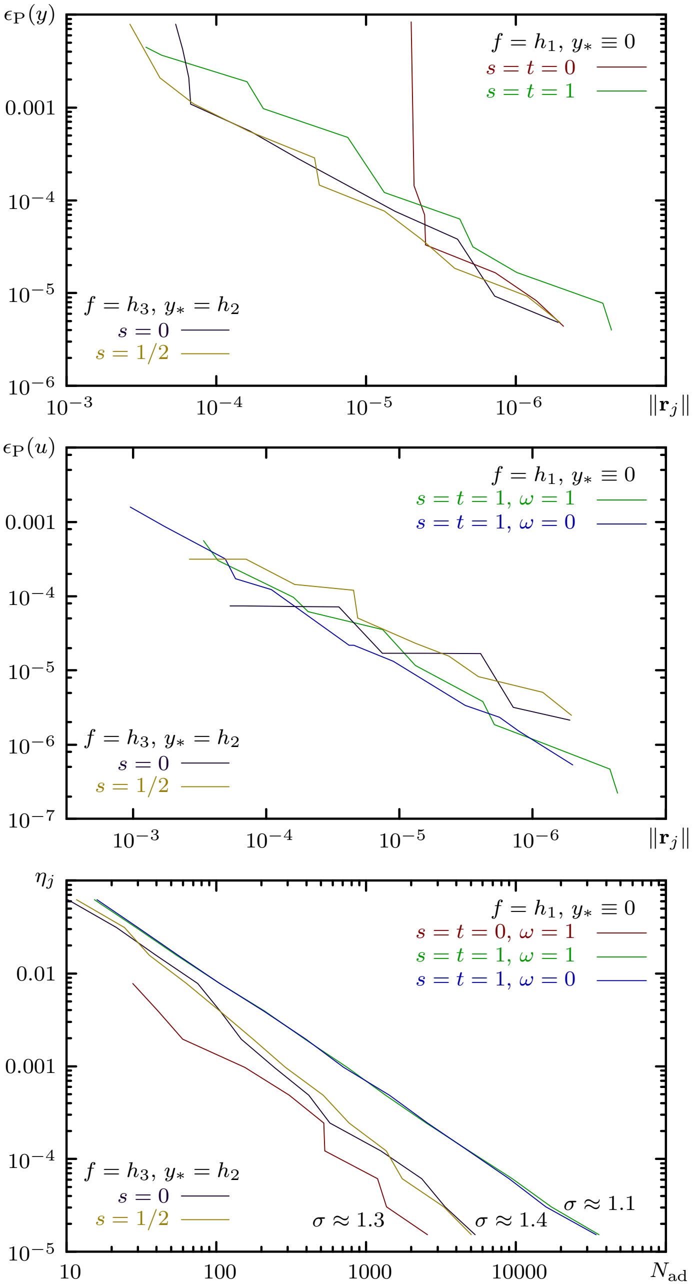


Figure 7.4: We show convergence plots for the experiments in one dimension, cf. Table 7.10, Table 7.11 and Table 7.12. The top and middle pictures contain the errors $\epsilon_{\mathrm{P}}(y)$ and $\epsilon_{\mathrm{P}}(u)$ of the state and the control, respectively. Both are plotted versus the outer residual $\|\mathbf{r}_j\|$, exhibiting a slope slightly greater than 1 for the state, and slightly less than 1 for the control. The bottom table shows the relation between the error $\eta_j = 2^{-j}$ and the average number of nonzero coefficients N_{ad}.

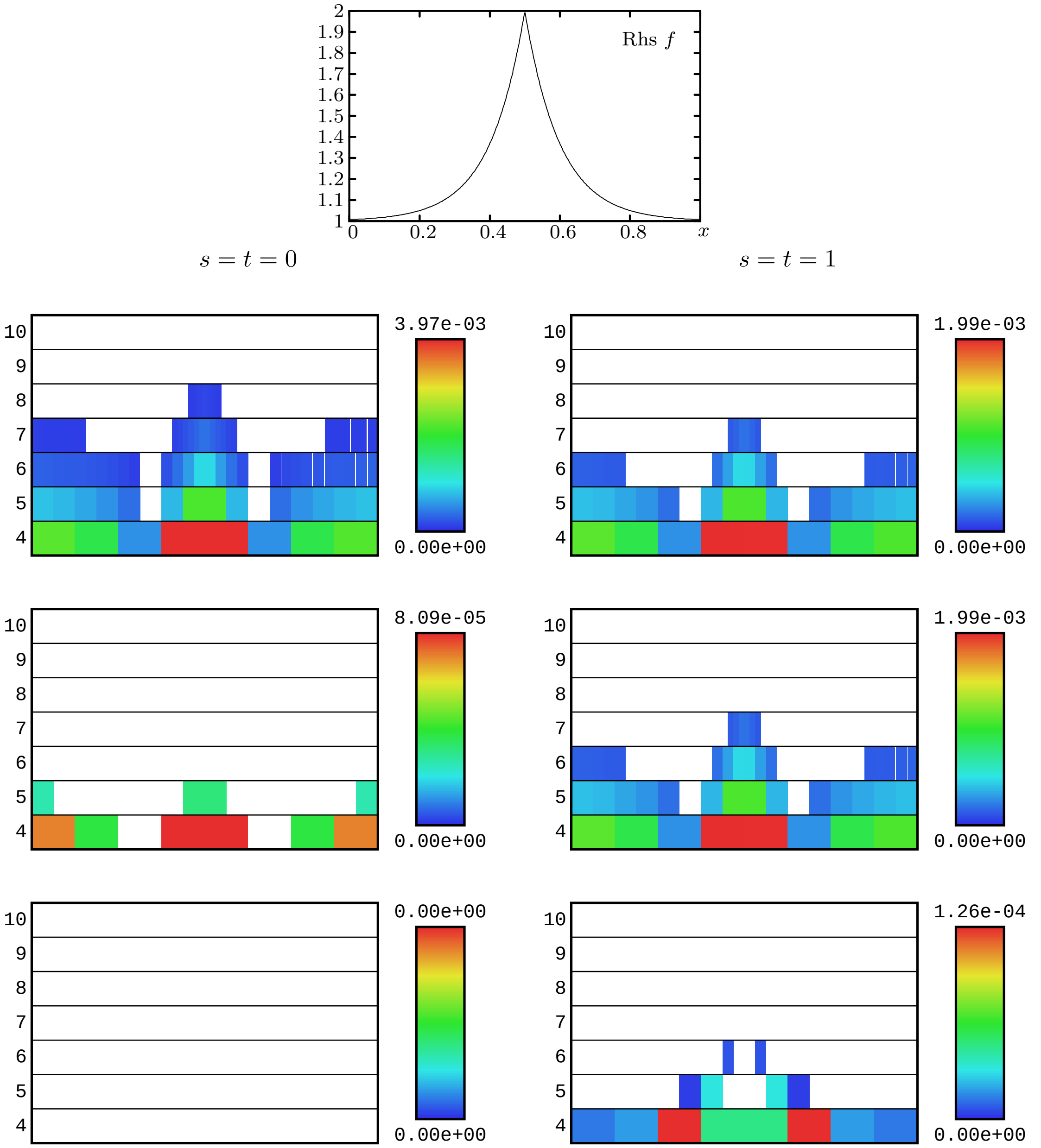


Figure 7.5: We show distributions of wavelet coefficients with $f = h_1$, shown in the topmost graph, and $y_* \equiv 0$. The left column of pictures corresponds to $s = t = 0$, and the right column to $s = t = 1$. The three rows contain from top to bottom the state y, the adjoint p and the control u. The control on the left consists purely of generator functions, hence no wavelet coefficients show up.

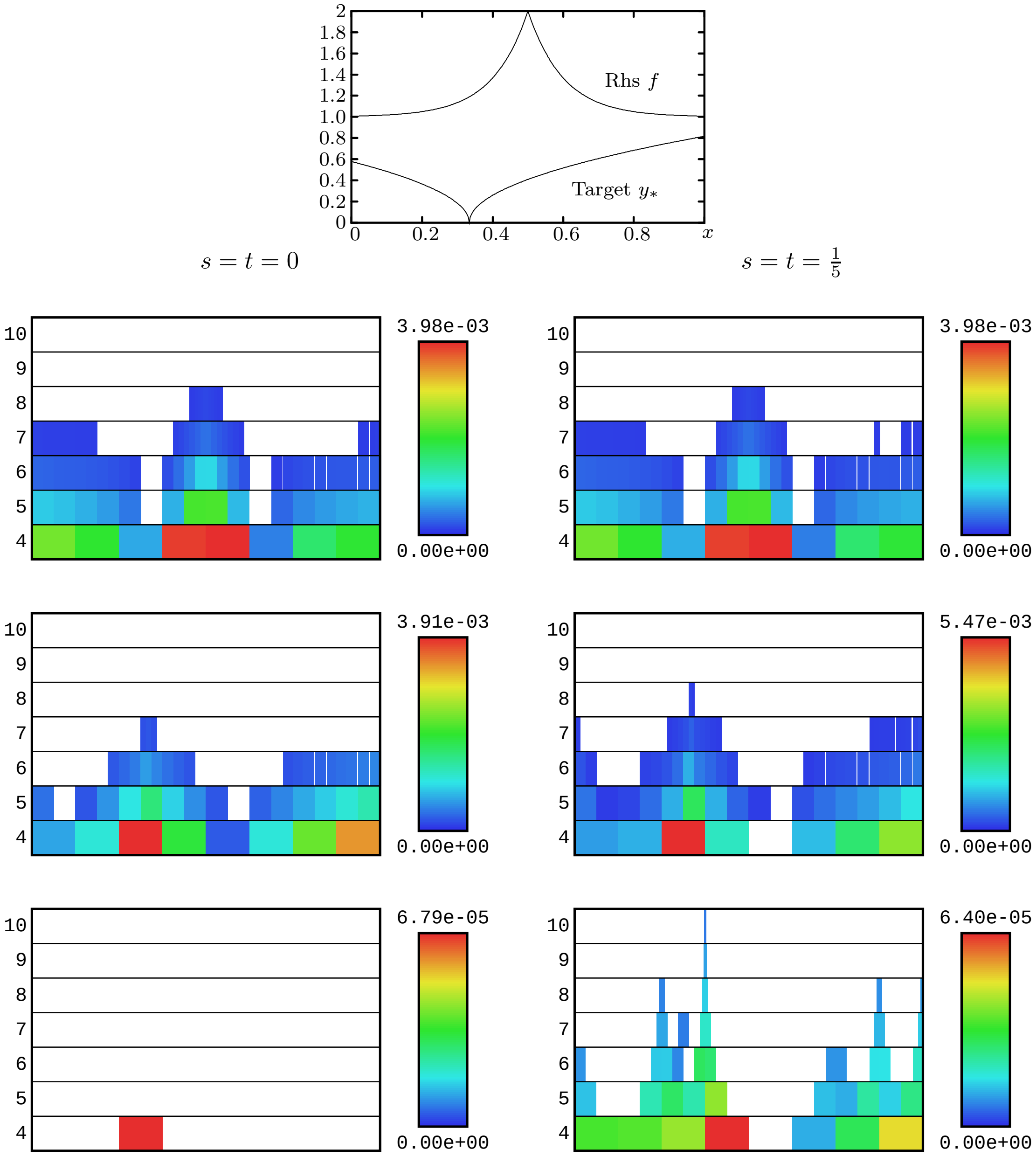


Figure 7.6: We show graphics of wavelet coefficients for the combination of $f = h_1$ *and* $y_* = h_2$ *from* (6.3.5) *as shown in the top picture. The three rows of pictures contain the state (top), the adjoint (middle) and the control (bottom), for two combinations of regularities, namely* $s = t = 0$ *for the left column and* $s = t = \frac{1}{5}$ *for the right.*

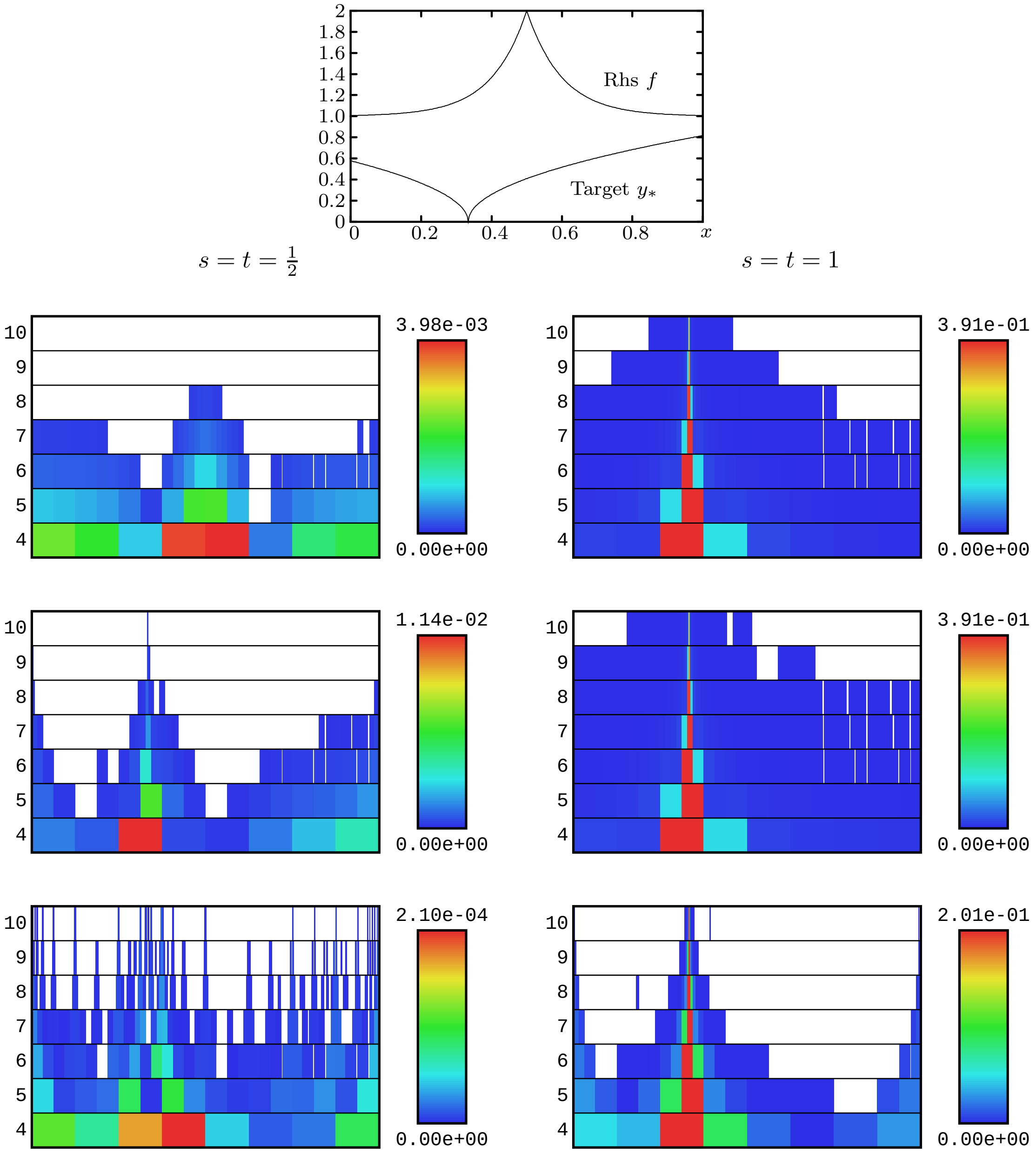


Figure 7.7: As in the previous figures, we display two sets of three images each, containing the state in the top row, the adjoint in the middle and the control in the bottom row. These have been computed for the same f and y_* as in Figure 7.6. The regularity parameters s and t have been set to $\frac{1}{2}$ (left) and 1 (right).

In Figure 7.6 and Figure 7.7, we have kept $f = h_1$ as before and added $y_* = h_2$ from (6.3.5). We present four sets of results, arranged column-wise, for the smoothness parameters $s = t = 0$, $\frac{1}{5}$, $\frac{1}{2}$ and 1. The increase of s and t has the effect that all variables develop a singular character, as has also been observed with the uniform discretisation. Moreover, the patterns of coefficients become denser as the represented functions get sharper, and spread to higher levels when s and t go to 1. The state and the adjoint feature many small coefficients (coloured blue), which we explain with a pessimistic overestimation in the postprocessing.

Furthermore we observe that the singularity of the target y_* at $x = \frac{1}{3}$ is mirrored by the coefficients of the adjoint p. The state y shows the singularity of the right hand side f at $x = \frac{1}{2}$, except for natural norms, where it is similar to p. The control acts as a sort of compromise between the state and adjoint variables. The control also shows the strongest change in character for varying smoothness. For the L_2 case, it is almost perfectly smooth, while it develops irregular patterns of coefficients around $s = t = \frac{1}{2}$ and features the single peak from y_* for natural norms.

The sets of coefficients differ characteristically, both among the three variables y, p and u of one run, and also between runs with varying regularities. Thus, the type of wavelet algorithm presented here inherently creates separate adjustments of the adaptive index sets for the different variables. To our knowledge, this feature is unique to the wavelet ansatz, since in finite element methods all variables are discretised on the same grid.

Example 7.6 – Two Dimensions

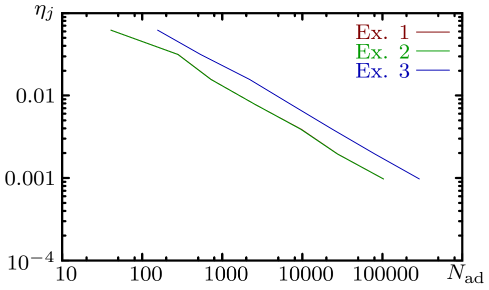
In two spatial dimensions, we first study the control problem in L_2 norms for an isotropic and an anisotropic tensor combination of the right hand side, and a target function $y_* \equiv 0$. The maximum level of resolution has been specified to $J = 10$.

The layout of the tables is the same for these and all following results, also in three dimensions. Each type of right hand side f leads to a set of three tables, which contain from top to bottom the results for the original f and $y_* \equiv 0$, then for $D_{3/2}f$ and $y_* \equiv 0$, and finally for $D_{3/2}f$ and a target y_* built from a tensor product of the function h_2 (6.3.5) analogously to f. The first two tables use $s = t = 0$, and the third $s = \frac{1}{2}$, $t = 0$. All of them are computed with $\delta = 0.2$.

The results for the isotropic right hand side $f_{2,2} := h_1 \otimes h_1$ are given in Table 7.13. The convergence behaviour is very similar to the one-dimensional case which was presented in Table 7.10. Namely, the state y exhibits a reduction of error by a factor 2 per level in agreement with the theory, while the error in the control u is small from the lowest level on. In the first two simulations, one outer iteration suffices to reduce the residual and the error in u to a value far smaller than the discretisation error. The number of inner iterations for all tables and the number of outer iterations for the last increase with the level shown here. We expect that the asymptotic regime of constant iteration counts begins around level 9 to 10, as it was the case in one dimension.

The effect of δ is also clearly visible in the first two tables, where the percentages of nonzero coefficients are reduced below 10%. As before, the rough right hand side in the second table does not influence the convergence behaviour. The case of non-smooth y_* in the third table is the most expensive, as can be seen from the iteration numbers and an end percentage of just below 30%. All tables exhibit the same adaptive rate of 0.55, which is above $1/n = 1/2$.

In Table 7.14, we have chosen the anisotropic right hand side $f_{2,1} := h_1 \otimes 1$. The convergence history is much the same as for isotropic data, where also the residual and the error in the control already begin with small values. The error in the state decays constantly by a factor of about 2 between levels. The number of inner iterations increases also in this example, hinting that the asymptotic regime has not yet been reached with $J = 10$.

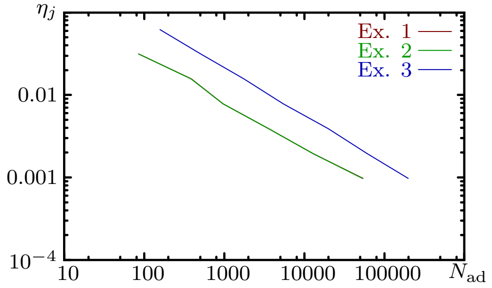


j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								1.37e-02	3.05e-06
4	8.95e-06	1	1	0	1	13.9%	40	1.03e-02	3.05e-06
5	9.24e-06	1	1	0	3	25.7%	280	5.20e-03	3.05e-06
6	9.40e-06	1	2	0	5	17.2%	725	2.58e-03	3.05e-06
7	9.44e-06	1	3	0	6	15.4%	2568	1.28e-03	3.05e-06
8	9.48e-06	1	5	0	7	14.7%	9715	6.44e-04	3.05e-06
9	8.16e-06	1	5	0	12	10.5%	27710	3.09e-04	3.05e-06
10	7.33e-06	1	6	1	12	9.8%	102851	1.18e-04	3.05e-06
						$\sigma \approx 0.55$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								1.32e-02	3.36e-06
4	8.91e-06	1	1	0	1	13.9%	40	1.02e-02	3.36e-06
5	9.20e-06	1	1	0	3	25.6%	279	5.09e-03	3.36e-06
6	9.35e-06	1	2	0	5	17.0%	718	2.56e-03	3.36e-06
7	9.40e-06	1	3	0	6	15.4%	2568	1.31e-03	3.36e-06
8	9.44e-06	1	5	0	7	14.6%	9638	6.75e-04	3.36e-06
9	8.11e-06	1	5	0	12	10.4%	27447	3.24e-04	3.36e-06
10	7.28e-06	1	6	1	12	9.8%	103182	1.25e-04	3.36e-06
						$\sigma \approx 0.55$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								1.31e-02	2.19e-04
4	3.09e-04	1	6	1	11	54.0%	156	1.02e-02	2.19e-04
5	3.55e-04	1	6	2	11	49.0%	534	5.08e-03	2.19e-04
6	1.80e-04	4	4	1	20	51.6%	2182	2.55e-03	2.19e-04
7	1.22e-04	6	6	1	21	43.1%	7169	1.31e-03	2.19e-04
8	5.61e-05	8	8	1	23	36.0%	23745	6.73e-04	2.19e-04
9	2.22e-05	10	9	1	23	30.6%	80525	3.33e-04	1.55e-04
10	1.15e-05	12	9	2	24	27.6%	289790	1.25e-04	1.07e-04
						$\sigma \approx 0.55$			

Table 7.13: Results for the adaptive algorithm for the control problem in two dimensions are given here. The right hand side for the top table has been set to the tensor product $f_{2,2} = h_1 \otimes h_1$. The two lower tables use the rough version $D_{3/2} f_{2,2}$. All tables have been computed with $y_* \equiv 0$, $s = t = 0$ and $\delta = 0.2$, with the exception of the last which deviates in the parameter $s = \frac{1}{2}$ and the non-smooth target $y_* = h_2 \otimes h_2$. The diagram above shows the convergence history with respect to N_{ad}, where examples 1 to 3 denote the three tables from top to bottom.



j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								8.04e-03	1.62e-06
4	5.31e-06	1	0	0	0	0.0%	0	7.24e-03	1.62e-06
5	5.47e-06	1	1	0	1	7.8%	85	4.42e-03	1.62e-06
6	5.48e-06	1	1	0	3	9.2%	389	2.15e-03	1.62e-06
7	5.57e-06	1	2	0	5	5.8%	960	1.07e-03	1.62e-06
8	5.64e-06	1	4	0	7	5.5%	3609	5.32e-04	1.62e-06
9	5.55e-06	1	5	0	10	4.8%	12720	2.53e-04	1.62e-06
10	4.73e-06	1	7	0	10	5.1%	53953	1.15e-04	1.62e-06
						$\sigma \approx 0.55$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								7.72e-03	1.64e-06
4	5.29e-06	1	0	0	0	0.0%	0	7.07e-03	1.64e-06
5	5.44e-06	1	1	0	1	7.8%	84	4.27e-03	1.64e-06
6	5.46e-06	1	1	0	3	9.1%	386	2.16e-03	1.64e-06
7	5.54e-06	1	2	0	5	5.8%	960	1.07e-03	1.64e-06
8	5.61e-06	1	4	0	7	5.5%	3609	5.47e-04	1.64e-06
9	5.53e-06	1	5	0	10	4.8%	12720	2.67e-04	1.64e-06
10	4.70e-06	1	7	0	10	5.1%	53335	1.26e-04	1.64e-06
						$\sigma \approx 0.55$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								7.66e-03	1.05e-04
4	2.18e-04	1	5	1	12	54.5%	158	7.02e-03	1.05e-04
5	2.12e-04	1	8	2	12	46.2%	503	4.30e-03	1.05e-04
6	2.03e-04	1	8	3	12	41.4%	1748	2.16e-03	1.05e-04
7	1.14e-04	3	5	1	18	32.6%	5431	1.06e-03	1.05e-04
8	5.08e-05	6	7	2	21	30.2%	19949	5.44e-04	1.04e-04
9	2.52e-05	8	9	1	20	23.1%	60790	2.71e-04	9.11e-05
10	1.14e-05	10	8	1	21	19.0%	199271	1.27e-04	5.37e-05
						$\sigma \approx 0.58$			

Table 7.14: We list the results of the two-dimensional control problem, this time with an anisotropic right hand side of $f_{2,1} = h_1 \otimes 1$. The target for the last run has been selected as $y_* = h_2 \otimes 1$. All other parameters and data for these three experiments are the same as in the previous table.

Because of the smoothness of f in one direction, the effect of δ is strongest here. The percentage of nonzero coefficients goes down to about 5% for this right hand side. This is only half the percentage compared to the isotropic right hand side discussed above. Also the last row of the table, corresponding to the target $y_* = h_2 \otimes 1$, shows lower percentages of about 20%. The rates are again significantly over $1/2$.

We have also created graphical representations of the arrangement of wavelet coefficients in two dimensions. For Figure 7.8, we have chosen the parameters $s = 1$, $t = 0$ and the tensor product data $f_{2,2} = h_1 \otimes h_1$ and $y_* = 1 \otimes h_2$. As discussed in Section 2.3.3, n-dimensional tensor product wavelet bases consist of $2^n - 1$ types of wavelets indexed by $\mathbf{e}$. In two dimensions, we obtain three types $\mathbf{e} = (1, 0)$, $(0, 1)$ and $(1, 1)$, which measure the oscillations in the coordinate directions x_1 and x_2 and the mixed portion x_1x_2, respectively. To demonstrate the reaction of the adaptive wavelet scheme to the tensor product structure of the data, we have displayed the type $(1, 0)$ in the left column, and the type $(0, 1)$ in the right column. As in the one-dimensional figures, we have arranged the state y, the adjoint p and the control u from top to bottom.

Because of the x_2-dependence of the target y_*, the wavelets of the type $(0, 1)$ are much more active in this example. The singular line along $x_2 = \frac{1}{3}$ can be clearly seen in all pictures in the right hand column. The dependence on the x_1 coordinate is only weakly introduced through the right hand side f, causing minor variations of the plots in the left column. The diagrams for the mixed type $(1, 1)$ (not shown here) are completely empty for all variables, an indication for the strong tensor product nature of the data. This example is especially instructive with respect to the wavelet-specific resolution of not only different functions, but also different types of wavelets for the same function via different index sets. In other words, adaptive wavelet methods automatically include a *dimension-adaptive* concept.

For the last example, shown in Figure 7.9, we have selected a rotationally symmetric target with a circular singularity. In this case, the graphs for the types $(1, 0)$ and $(0, 1)$ are identical subject to a rotation of 90 degrees. The wavelets of mixed type $(1, 1)$ for the adjoint and the control are displayed in Figure 7.10, they are both invariant under rotations of 90 degrees. Thus, the symmetry properties of the data directly control the symmetries of the solution. Moreover, the circular line of the singularity shows up in both the adjoint and the control for all types of wavelets. This is an instructive example of adaptive refinement, in the sense that singularities are resolved locally for greater accuracy, while as few coefficients as possible are spent on smooth parts of the solution.

Example 7.7 – Three Dimensions

As for the single elliptic system in three spatial dimensions, we cover the right hand sides $f_{3,3} := h_1 \otimes h_1 \otimes h_1$, $f_{3,2} := h_1 \otimes h_1 \otimes 1$ and $f_{3,1} := h_1 \otimes 1 \otimes 1$ to study the effect of anisotropy on the efficiency of the adaptive method. The layout of the corresponding listings in Table 7.15, Table 7.16 and Table 7.17 is the same as for the experiments in two dimensions. The top and middle table on each page correspond to $s = t = 0$ and $y_* \equiv 0$, and the bottom table to $s = \frac{1}{2}$, $t = 0$ and a target y_* which is constructed by a tensor combination of h_2. The middle and bottom table have been computed with the rough right hand sides $D_{3/2} f_{3,i}$, and we have used $\delta = 0.2$ throughout our simulations.

The results show the same tendencies as for the two-dimensional case. Namely, the unmodified right hand sides $f_{3,i}$ lead to very similar convergence behaviour as their rough versions, the case of non-smooth target y_* and $s = \frac{1}{2}$ is several times as expensive as the setting $y_* \equiv 0$, and smoother right hand sides generally need fewer nonzero coefficients provided that all other parameters are equal. However, the maximum level $J = 7$ in three dimensions does not allow to examine the asymptotic regime. Nonetheless we suppose in accordance with previous results that the iteration numbers per level are eventually constant and that the order of adaptive approximation is slightly higher than $1/n = 1/3$.

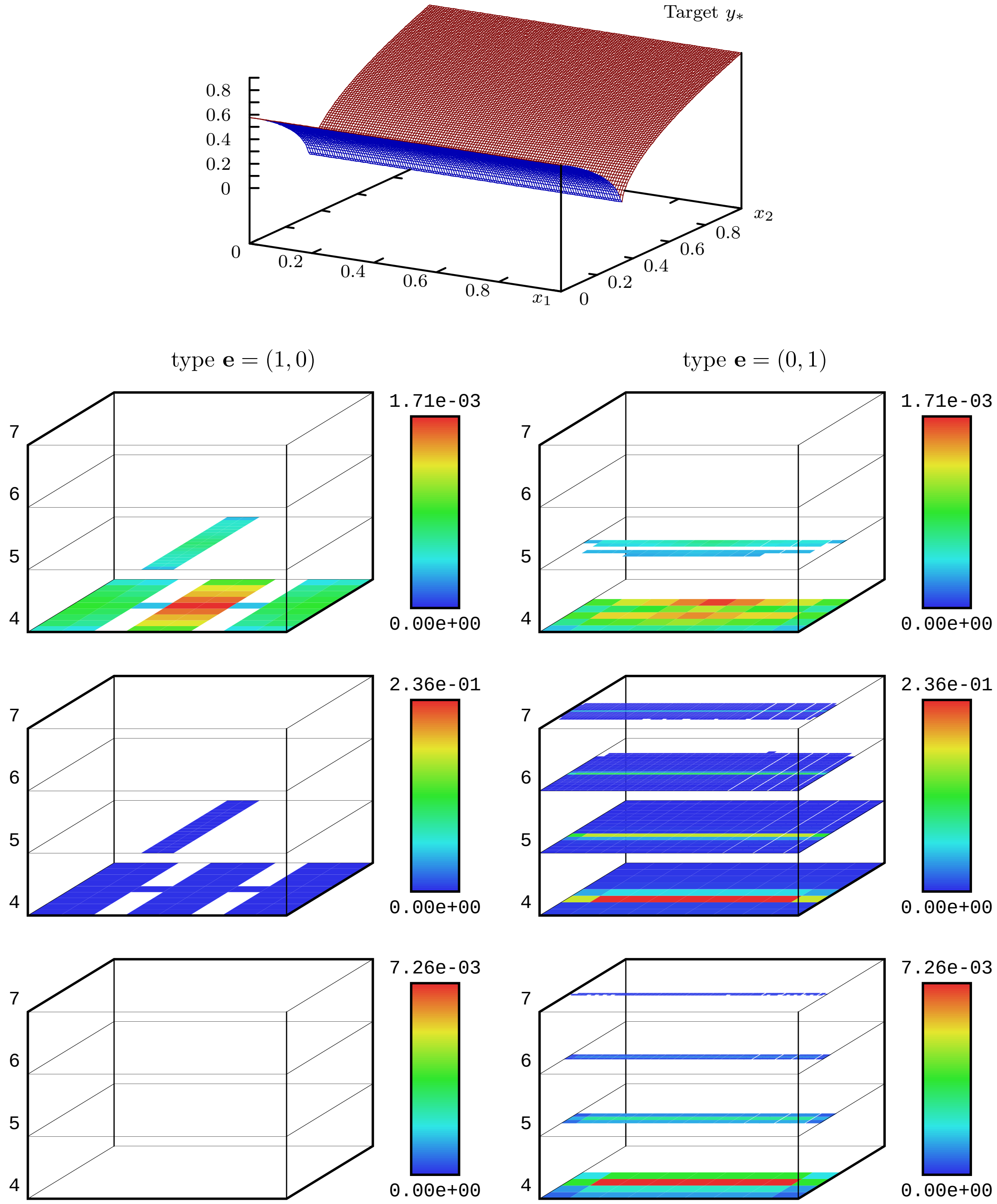


Figure 7.8: We show two-dimensional results for a right hand side $f_{2,2} = h_1 \otimes h_1$ *and an anisotropic target* $y_* = 1 \otimes h_2$ *using the one-dimensional function* h_2 *from* (6.3.5)*. The target is shown in the topmost graph. The three rows contain the state* y*, the adjoint* p *and the control* u *as in the previous plots. The left column displays the wavelets of type* $(1, 0)$*, and the right column the type* $(0, 1)$*.*

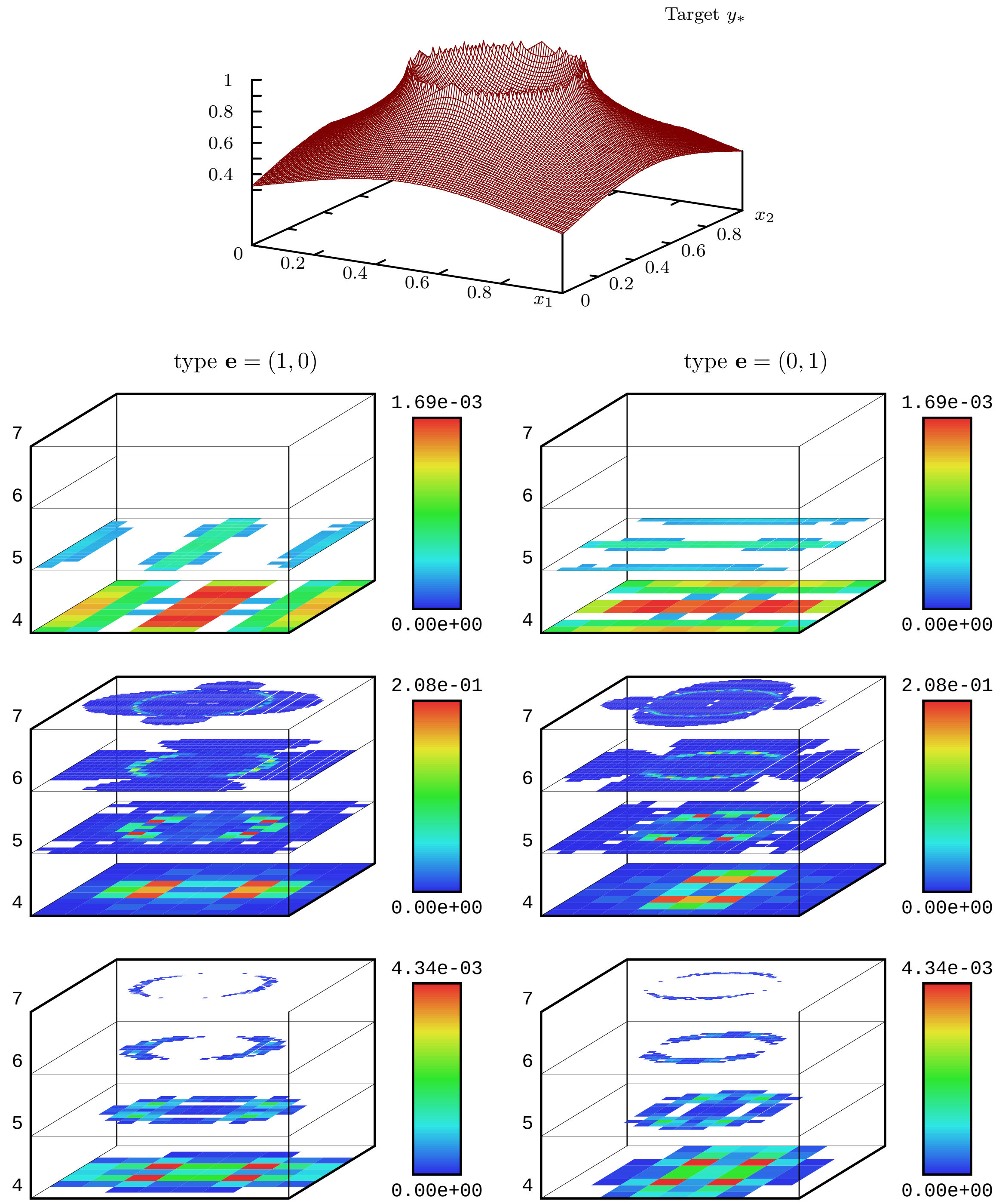
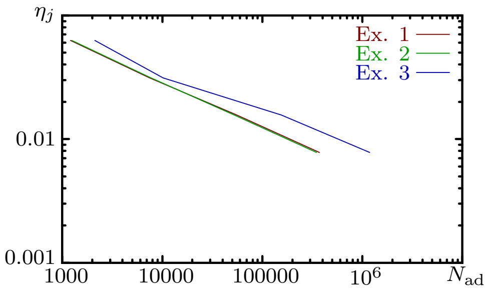


Figure 7.9: We have repeated the previous set of two-dimensional experiments with the rotationally symmetric target function y_* as shown above. Again, the rows hold the state, the adjoint and the control, and the columns show the tensor product wavelets of types $(1, 0)$ and $(0, 1)$, respectively. The circular structure of the target is mirrored in the arrangement of coefficients.

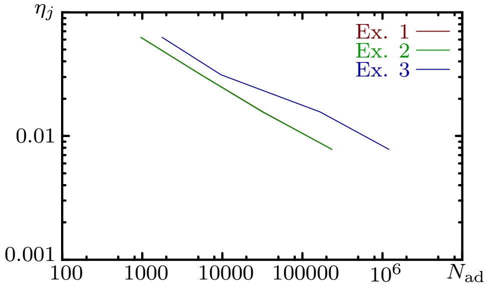


j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								1.80e-02	5.48e-06
4	1.43e-05	1	1	0	7	24.5%	1204	1.19e-02	5.48e-06
5	1.46e-05	1	5	0	14	20.7%	7450	5.82e-03	5.48e-06
6	1.46e-05	1	5	0	14	20.2%	55373	2.75e-03	5.48e-06
7	1.47e-05	1	6	0	17	17.3%	372031	1.02e-03	5.48e-06
						$\sigma \approx 0.36$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								1.73e-02	5.54e-06
4	1.43e-05	1	1	0	11	25.2%	1236	1.17e-02	5.54e-06
5	1.46e-05	1	5	0	14	21.5%	7715	5.74e-03	5.54e-06
6	1.46e-05	1	5	0	14	19.0%	52135	2.75e-03	5.54e-06
7	1.46e-05	1	6	0	17	16.2%	347670	1.02e-03	5.54e-06
						$\sigma \approx 0.37$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\mathrm{P}}(y)$	$\epsilon_{\mathrm{P}}(u)$
3								1.72e-02	1.31e-04
4	2.17e-04	1	10	1	28	42.9%	2106	1.17e-02	1.31e-04
5	2.45e-04	1	10	2	34	28.3%	10167	5.73e-03	1.31e-04
6	1.66e-04	7	10	0	72	56.2%	154473	2.75e-03	1.30e-04
7	9.46e-05	14	10	0	78	55.0%	1180143	1.02e-03	1.30e-04
						$\sigma \approx 0.32$			

Table 7.15: These results of the adaptive algorithm for the control problem in three dimensions have been obtained with the isotropic and non-smooth right hand side $f_{3,3}$. The first two tables use $s = t = 0$ and $y_* \equiv 0$, while the last table contains data for $s = \frac{1}{2}$, $t = 0$ and $y_* = h_2 \otimes h_2 \otimes h_2$. Tables two and three use the rough right hand side $D_{3/2} f_{3,3}$. As in two dimensions, the convergence history is displayed in the top figure.

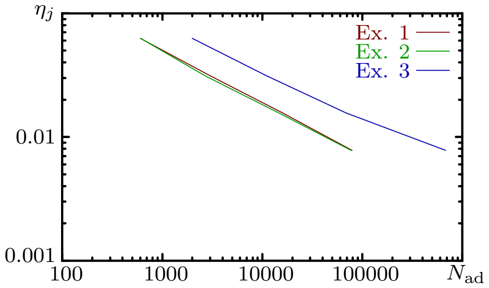


j	$\Vert\mathbf{r}_j\Vert$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								1.22e-02	4.97e-06
4	9.73e-06	1	1	0	4	19.6%	961	9.69e-03	4.97e-06
5	1.01e-05	1	2	0	11	15.1%	5424	4.94e-03	4.97e-06
6	1.00e-05	1	5	0	13	11.8%	32314	2.32e-03	4.97e-06
7	1.00e-05	1	5	0	13	10.9%	234347	9.99e-04	4.97e-06
						$\sigma \approx 0.38$			

j	$\Vert\mathbf{r}_j\Vert$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								1.17e-02	4.92e-06
4	9.69e-06	1	1	0	4	19.4%	952	9.56e-03	4.92e-06
5	9.98e-06	1	2	0	14	14.9%	5373	4.80e-03	4.92e-06
6	9.98e-06	1	5	0	14	11.6%	31813	2.35e-03	4.92e-06
7	9.97e-06	1	5	0	14	10.9%	233763	1.00e-03	4.92e-06
						$\sigma \approx 0.38$			

j	$\Vert\mathbf{r}_j\Vert$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								1.16e-02	2.11e-04
4	3.21e-04	1	10	1	35	36.0%	1768	9.55e-03	2.11e-04
5	3.80e-04	1	11	2	35	26.8%	9648	4.79e-03	2.11e-04
6	1.65e-04	11	11	0	79	61.0%	167454	2.35e-03	2.10e-04
7	8.27e-05	16	11	1	81	56.3%	1209416	1.00e-03	2.10e-04
						$\sigma \approx 0.31$			

Table 7.16: These results in three dimensions have been obtained with the right hand side $f_{3,2} = h_1 \otimes h_1 \otimes 1$. The target for the last row has been selected as $y_* = h_2 \otimes h_2 \otimes 1$. All other characteristics are the same as in the previous table. In particular, the last two tables correspond to the rough right hand side $D_{3/2} f_{3,2}$.

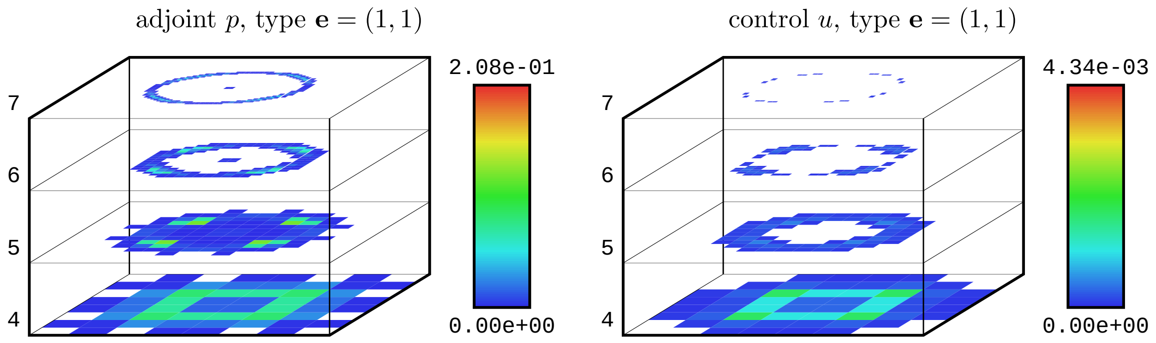


j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								7.14e-03	2.60e-06
4	5.65e-06	1	1	0	3	12.3%	603	6.24e-03	2.60e-06
5	5.85e-06	1	2	0	5	8.4%	3003	4.30e-03	2.60e-06
6	5.90e-06	1	3	0	8	5.8%	15903	2.02e-03	2.60e-06
7	5.89e-06	1	6	0	13	3.7%	78976	9.63e-04	2.60e-06
						$\sigma \approx 0.43$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								6.86e-03	2.56e-06
4	5.62e-06	1	1	0	3	12.4%	608	6.02e-03	2.56e-06
5	5.88e-06	1	2	0	8	7.4%	2674	4.27e-03	2.56e-06
6	5.88e-06	1	3	0	10	5.3%	14576	2.05e-03	2.56e-06
7	5.86e-06	1	6	0	12	3.6%	77281	9.65e-04	2.56e-06
						$\sigma \approx 0.43$			

j	$\|\mathbf{r}_j\|$	#O	#E	#A	#R	S	N_{ad}	$\epsilon_{\text{P}}(y)$	$\epsilon_{\text{P}}(u)$
3								6.82e-03	4.58e-05
4	1.72e-04	1	9	2	29	40.6%	1996	5.90e-03	4.58e-05
5	1.39e-04	1	11	2	29	30.6%	11005	4.27e-03	4.58e-05
6	1.18e-04	1	11	4	29	25.1%	69024	2.05e-03	4.58e-05
7	7.41e-05	5	11	2	60	31.8%	681610	9.45e-04	4.45e-05
						$\sigma \approx 0.36$			

Table 7.17: This table covers the anisotropic situation with right hand side $f_{3,1} = h_1 \otimes 1 \otimes 1$. The target for the first two tables is again $y_* \equiv 0$, while the last table utilises $y_* = h_2 \otimes 1 \otimes 1$. All other settings are the same as for the previous two sets of results in three dimensions. In particular, the last two tables utilise the function $D_{3/2} f_{3,1}$.


adjoint p, type $\mathbf{e} = (1,1)$

control u, type $\mathbf{e} = (1,1)$

Figure 7.10: This figure shows the distribution of coefficients of the mixed type of wavelet $\mathbf{e} = (1,1)$ *for the adjoint (left) and the control (right). The corresponding diagram for the state is not shown here, since it does not contain any coefficients for* $j \geq 4$. *These graphics belong to the experiment from Figure 7.9.*

Conclusion

In this chapter we have briefly presented the theoretical background of adaptive wavelet methods. We have then motivated the routines `Coarse` to reduce the number of coefficients of an adaptive wavelet expansion, and the routine `Ad-Apply` which computes an adaptive matrix-vector product. Both are derived from nonlinear approximation theory and provide strict error control. We have then incorporated these routines into our algorithm `nⅡcG/2` developed in the previous chapter, carefully combining the thresholds from the two-layer conjugate gradient algorithm with the error bounds from the newly introduced adaptive routines. The result is the adaptive algorithm δ-`AnⅡcG/2`, fully specified with rigorously derived tolerances.

Adaptive finite element methods generally use repeated cycles comprised of the substeps solution, estimation and refinement. Our method is different in the sense that these operations are essentially merged in a unified way. The adaptive index sets of all involved variables fluctuate during the course of the inner iterative solvers for the elliptic subproblems. For the solution of the optimal control problem, this ansatz allows us to resolve all variables with individual, different sets of coefficients. This includes the principal variables y, p and u, and also the auxiliary variables used in the method of conjugate gradients.

We found experimentally that the adaptive wavelet scheme δ-`AnⅡcG/2` retains the convergence behaviour from `nⅡcG/2`, that is, the H^1 errors of the state reduce with the level of resolution as 2^{-j}. Moreover, the numbers of inner and outer iterations are constant across all levels of resolution in one dimension. Although the asymptotic regime is not reached for two and three dimensions, the similarity of the iteration histories to the one-dimensional case suggests that the computational complexity is also linear for higher dimensions. Altogether, the results indicate that the two-layer nested iteration scheme originally conceived for the uniform algorithm works with optimal complexity also in the adaptive context.

We have estimated the effective adaptive efficiency of our method via the average percentage of nonzero coefficients with respect to the cardinality of the full uniform index set. Hereby the averaging process comprises the solution of the three elliptic systems of equations in the inner loop of the scheme, which accounts for the predominant fraction of the computational cost. We have found that this quantity could be significantly reduced by the introduction of additional routines `Coarse` for the three auxiliary variables of the inner conjugate gradient loop. The effect of these routines is controlled by the parameter $\delta \geq 0$. While setting $\delta = 0$ effectively disables the additional coarsening steps, higher values than about 0.3 to 0.5 impair the convergence of the scheme. Yet, in all situations we could find a sensible intermediate choice of δ, yielding an increased adaptive efficiency with no impact on the speed of convergence. In several non-trivial situations, the memory requirements were reduced by an order of magnitude. The convergence of the error with respect to the degrees of freedom has been found to be superlinear, with strongest evidence for this fact in one dimension.

The qualitative behaviour of our method meets all expectations which we have deduced from general wavelet theory and the experiments with the uniform algorithm from Chapter 6. Namely, constant data are automatically handled by only the single-scale coefficients, and smooth functions are represented by very few wavelet coefficients. L_2 norms in the objective functional generally lead to a smooth solution and control, requiring a small number of wavelet coefficients. When tuning the norms towards H^1 for the state and H^{-1} for the control, all variables including the adjoint exhibit locally higher resolutions around the singularities induced by the data f and y_*. Finally, we can observe that the variables y, p and u are resolved with clearly different index sets, which depend on the data and the smoothness parameters s and t. The adaptive method also resolves anisotropic singularities, where smoothness is given in one coordinate direction and not in the other. Thus, the adaptive wavelet scheme automatically and effectively exploits the potential inherent in many practical selections of data, namely, to reduce the number of coefficients without losing accuracy.

While the thoughtful combination of adaptive wavelet operations with a two-layer conjugate gradient method and a nested iteration strategy has indeed proved successful for the examples considered here, they also serve as motivation for further considerations and theoretical studies. On the one hand, the analysis of the interplay of the solution of the inner and outer systems, and their dependence on the various coarsening parameters, might lead to novel algorithmic improvements. On the other hand, the following subjects of possible future research arise not only in the context of adaptive wavelet methods for elliptic optimal control problems, but also for more general adaptive numerical schemes.

- Investigate inexact Krylov subspace methods, where an error is not only introduced in the application of the matrix, but also for one or more auxiliary variables in the inner loop.
- Study adaptive wavelet approximation in the context of Krylov methods, and examine statements on the convergence rate. In particular, establish connections between the smoothness of solutions measured a-priori in ℓ_τ^{w} and the error reduction rate $N^{-\sigma}$.
- Examine in which way and under which circumstances the convergence rates predicted by the ideal wavelet algorithm from Section 7.3.1 can be reproduced by the algorithm δ-`AnIIcG/2`.

Altogether, we have collected various results which indicate that adaptive wavelet discretisations provide a sophisticated, powerful and flexible tool for the efficient large-scale numerical solution of the type of PDE-constrained control problem considered in this thesis.

Chapter 8

Conclusion and Outlook

In this thesis we have developed a wavelet method for the fast numerical solution of linear-quadratic elliptic optimal control problems, this is, the solution is computed with a computational effort which is proportional to the number of unknowns N_J on the highest level of resolution J. We have then provided extensive numerical results in one to three spatial dimensions for various combinations of parameters and data, analysing both a uniform and an adaptive discretisation strategy.

The starting point has been the reformulation of the control problem following the wavelet paradigm. Here the key mechanism is the systematic use of norm equivalences, which lead to an equivalent, infinite-dimensional multi-scale representation of the problem in terms of vectors in ℓ_2. Specifically, we have used an optimised construction of biorthogonal B-spline wavelets. The norm equivalence guarantees uniformly bounded condition numbers of all elliptic operators, and allows the fast numerical evaluation of Sobolev norms of arbitrary smoothness $s \in \mathbb{R}$ by appropriate Riesz matrices $\mathbf{R}_s$. We have designed a novel unified construction of these Riesz matrices, which is exact for positive and negative integral orders of smoothness, and for arbitrary norms of constant functions. For the general fractional case, we interpolate the norms equivalently between the nearest integers with a continuous dependence on the smoothness s.

Based on the wavelet formulation, we have devised the two-layer nested iteration inexact conjugate gradient scheme `nIIcG/2`, shown in Algorithm 6.6. All error bounds and stopping criteria for its subroutines have been rigorously derived from the bounds of the involved operators in wavelet representation. The nested iteration approach allows to compute the solution up to discretisation error accuracy with a constant amount of iterations per level of resolution, yielding an optimal computational complexity of $\mathcal{O}(N_J)$.

We have accelerated the numerical solution process by several optimisations which considerably reduce the condition numbers of wavelet bases and elliptic operators. In particular, we have introduced transformations of the generator and wavelet bases to the nodal basis to derive a translationally invariant form of the refinement matrices $\mathbf{M}_j$, and we use the diagonal $\mathbf{D}_a$ of the stiffness matrix in wavelet representation for the scaling of the wavelets. Furthermore, we have devised an additional preconditioning technique by analysing and shifting the eigenvalues of the stiffness matrix on the coarsest level. —

The first set of results in Chapter 6 has been obtained with the numerical wavelet scheme in uniform discretisation. Out of the variety of possible combinations of the independent modelling parameters s, t and ω with different choices of data f and y_*, we have inspected several exemplary cases, for one to three spatial dimensions. First of all, we find that the theoretically predicted convergence behaviour of the inexact conjugate gradient scheme is confirmed in all situations for which the problem is well-posed. That is, we observe iteration counts which are constant across all levels of resolution. Moreover, the absolute iteration numbers have been substantially reduced by means of our optimised wavelet construction.

We also confirm that the introduction of the parameters s and t, which is made possible within the wavelet framework, leads to a much richer behaviour of the model compared to the standard approach with L_2 norms in the objective functional and the single parameter ω. While ω acts as a global *scaling* between the state y and the control u, the parameters s and t govern the *smoothness* of y and u, and thus affect the character of these functions. Varying for example s between 0 and 1, the state transforms continuously from an average tracking of the target y_* towards a qualitative match of its shape.

When the regularity of the data is less than required by the objective functional, convergence slows down slightly, but is otherwise unharmed, which indicates that the method is robust with respect to non-conforming choices of parameters and data. Only the inadmissible combination of vanishing regularisation $\omega = 0$ and non-natural norms of smoothness $s, t \neq 1$ has been observed to provoke a deviation of the solution.

The model contains the possibility of using negative fractional Sobolev norms for the control u. For these cases we observe that u develops small systematic oscillations, which we interpret as effects of the discretisation introduced by the construction of inverse Riesz operators. Applying an averaging process, the shape of the control would interpolate continuously between the situations with integer norms on L_2 and H^{-1}. —

In Chapter 7, we have designed an adaptive wavelet algorithm by incorporating techniques from adaptive wavelet methods for stationary variational problems into the algorithm `nIIcG/2`. The newly introduced adaptive routines are designed to guarantee strict error bounds for their output variables. We systematically integrate these bounds with the tolerances of the original algorithm to specify the adaptive algorithm `AnIIcG/2`. Analogously to the uniform discretisation scheme, we predict the independence of the iteration numbers on the level of resolution by reusing results on inexact Krylov methods.

Thereafter, we introduce additional coarsening operations in the inner loop of the conjugate gradient scheme, leading to the final adaptive algorithm δ-`AnIIcG/2`. On the one hand, this is the key to a significant gain in the adaptive efficiency. On the other hand, it complicates the mathematical analysis of adaptive wavelet schemes in conjunction with Krylov subspace methods, for which presently no theoretical results on convergence and convergence rates exist. To inspect the qualitative and quantitative behaviour, and to judge the practical suitability of the adaptive algorithm, it is therefore indispensable to undertake extensive numerical studies with respect to convergence and adaptive efficiency.

We observe in our accordingly designed experiments that the iteration numbers on each level are constant. This is clearly visible in one dimension, and suggested by preasymptotic evidence for two and three dimensions. Thus, the convergence behaviour of the uniform wavelet scheme is essentially reproduced. The savings in memory consumption with respect to the full uniform grid range from a factor of 2 for natural norms and irregular data to more than 10 for L_2 norms and smooth data. The convergence rate with respect to the number of degrees of freedom is found to be superlinear in several examples, most obviously in one dimension.

In contrast to adaptive finite element methods, our algorithm inherently resolves different variables with different sets of nonzero coefficients. This offers additional potential to enhance the adaptive efficiency. We observe in all simulations that the three principal variables y, p and u are indeed represented with different distributions of coefficients. Moreover, also the wavelet coefficients of each variable alone have different resolutions in the coordinate directions depending on the possibly anisotropic structure of the data. Thus, the wavelet approach is inherently dimension-adaptive. —

In summary, for the development of the algorithm δ-`AnIIcG/2` we have enhanced techniques from wavelet construction and wavelet methods for stationary variational problems. We have studied the modelling of linear-quadratic control problems and the behaviour of inexact nested iteration conjugate gradient solvers, and integrated results from nonlinear approximation theory. By shaping the wavelet framework for the uniform preconditioning of operators and the optimisation of constants, and the proposition of a general recipe for the evaluation of Sobolev norms and the design of a two-layer nested iteration strategy, we have achieved a numerical scheme of optimal computational complexity.

The wavelet algorithms `nⅡcG/2` and δ-`AnⅡcG/2` have been implemented within the newly developed programme framework `BWP`. Providing a wide collection of numerical results from one to three spatial dimensions, using a resolution of up to two million unknowns per function variable, we have demonstrated the convergence and convergence rate of our method, the richness of the modelling ansatz using the additional wavelet-specific parameters s and t and the efficiency of the adaptive discretisation scheme. These results indicate that the wavelet ansatz indeed provides a competitive tool for the computational treatment of optimal control problems constrained by elliptic PDEs.

Our results and observations motivate various directions of future research. From the practical point of view, it would be desirable to improve the construction of fractional Riesz operators of negative order to remove the oscillations. More importantly, we realise a need for dedicated theoretical studies on inexact conjugate gradient methods, notably in conjunction with adaptive wavelet discretisations. Another route towards the development of an adaptive wavelet algorithm for the control problem could use a fixed set of active coefficients during the solution of each inner system. This would circumvent the perturbation of the inner Krylov solvers by coarsening operations, and facilitate the exact measurement of the number of active coefficients. For the structural design of such an algorithm, the handling of different distributions of coefficients for the principal variables and the integration of appropriate error estimators would have to be investigated. The key ingredient would consist in accurate adaptive quadrature rules and the corresponding application of mass and stiffness matrices with respect to arbitrary distributions of wavelet indices. The realisation of such schemes in linear time is a non-trivial subject of current research, see e.g. [4].

Furthermore, the unique properties of wavelets offer promising perspectives for a wider range of applications. A first step could be to include nonlinear PDEs as constraints as well as additional conditions on the control in the form of pointwise inequalities. There exist studies of these topics in the finite element context, see e.g. [18, 93, 94, 136]. In principle, wavelets are capable of treating pointwise control constraints by local transformations to the generator basis and e.g. the use of projected gradient methods. — Time-dependent PDE constraints in two or three spatial dimensions require a large amount of degrees of freedom when discretised uniformly with finite elements. To reduce the number of unknowns which need to be stored simultaneously, checkpointing techniques may be employed. Yet, there exist at least two possible ways to tackle this problem on a more fundamental level, namely, using space-time sparse grids [66, 73] and adaptive space-time wavelet discretisations, which are currently under investigation. — The handling of state constraints is somewhat more involved mathematically since the adjoint then needs to be formulated in terms of Borel measures. Several theoretical studies have been undertaken e.g. in [30, 31, 120, 121]. This class of problems is much more difficult to handle numerically.

In conclusion, wavelets were demonstrated to provide a powerful and flexible tool for the numerical solution of optimal control problems constrained by elliptic PDEs. Supported by the results obtained in this thesis with respect to modelling, preconditioning, fast iterative solution and adaptivity, the theoretical and practical potential of wavelets for so many central numerical issues is a strong motivation to expand the wavelet ansatz to more general problems in the rapidly evolving and active field of PDE-constrained optimisation.

List of Symbols

For reference, we provide a list of the mathematical symbols used and their general meaning in the context of this document.

Table 8.1: Mathematical Symbols

n	Spatial dimension
Ω	Domain $\subset \mathbb{R}^n$
H	General Hilbert space over Ω
$d, \tilde{d}$	Primal and dual order of polynomial exactness
j_0	Lowest / coarsest level of resolution
j	Generic index for level of resolution
J	Maximal level of resolution in a particular context
S_j	Closed subspace of H
Φ_j	Single-scale basis for the space S_j
Δ_j	Index set for single-scale basis Φ_j
$\phi_{j,k}$	Single-scale basis function at location $k \in \Delta_j$
N_j	Number of degrees of freedom on level j
Ψ_j	Complement basis for the space W_j
∇_j	Index set for complement basis Ψ_j
$\psi_{j,k}$	Wavelet basis function at location $k \in \nabla_j$
λ	Combined notation for wavelet index $\lambda = (j, k)$
Λ	Set of wavelet indices $\Lambda = \{\lambda_i\}$
$\Psi_{(J)}$	Wavelet basis up to maximum resolution J
Ψ	Infinite-dimensional wavelet basis for the full Hilbert space
$Q_j, \tilde{Q}_j$	Primal and dual biorthogonal projectors
$\mathbf{I}_N$	Identity matrix of dimension $N \times N$, subscript may be omitted
$\mathbf{M}_j$	Twolevel transformation matrix between levels j and $j+1$, size $\#\Delta_{j+1} \times \#\Delta_{j+1}$
$\mathbf{M}_{j,0}$	Left half of $\mathbf{M}_j$, size $\#\Delta_{j+1} \times \#\Delta_j$
$\mathbf{M}_{j,1}$	Right half of $\mathbf{M}_j$, size $\#\Delta_{j+1} \times \#\nabla_j$
$\mathbf{G}_j$	Inverse matrix of $\mathbf{M}_j$
$\mathbf{G}_{j,0}$	Upper half of $\mathbf{G}_j$
$\mathbf{G}_{j,1}$	Lower half of $\mathbf{G}_j$
$\check{\mathbf{M}}_{j,1}$	Initial stable completion
$\check{\mathbf{M}}_j$	Initial twolevel transformation $(\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1})$

Table of Mathematical Symbols

$\check{\mathbf{G}}_j$	Inverse of $\check{\mathbf{M}}_j$
$\check{\mathbf{G}}_{j,0}$	Upper half of $\check{\mathbf{G}}_j$
$\check{\mathbf{G}}_{j,1}$	Lower half of $\check{\mathbf{G}}_j$
Ξ_j	Initial complement basis
$\mathbf{K}_j, \mathbf{L}_j$	Transformation matrices for the stable completion
$\check{\mathbf{K}}_j, \check{\mathbf{L}}_j$	Additional transformation matrices
$\mathbf{C}_j$	Transformation matrix for the single-scale basis
$\mathbf{W}_{J,j}$	Extended twolevel transformation matrix on level j
$\mathbf{W}_j$	Fast wavelet transformation up to highest level j
$\gamma, \tilde{\gamma}$	Range of smoothness for the norm equivalence
H^s	Sobolev space with smoothness index s
s, t	Smoothness indices for Sobolev spaces
D_t	Shift operator on a scale of Sobolev spaces
$\mathbf{D}$	Diagonal matrix to shift the smoothness of wavelet expansions
$\mathbf{W}_j^s$	Univariate wavelet transformation for Sobolev space H^s with $n = 1$
$\Psi_{(J)}^{\mathrm{ani}}$	Anisotropic multivariate wavelet basis up to maximum resolution J
Ψ^{ani}	Full anisotropic multivariate wavelet basis
$\Psi^{\mathrm{ani},s}$	Anisotropic multivariate wavelet basis for H^s, $n > 1$
$\mathbf{W}_J^{\mathrm{ani}}$	Anisotropic multivariate wavelet transformation
Ψ^{iso}	Isotropic multivariate wavelet basis
E	Index set $E = \{0, 1\}$ for the type of isotropic wavelet
E^*	Contains all composite wavelets, $E^* = E^n \setminus \{\mathbf{0}\}$
$\mathbf{e}$	Type of n-dimensional isotropic wavelet, $\mathbf{e} \in E^*$
ℓ_1, ℓ_2	End points of the support of the primal B-spline generator
$\tilde{\ell}_1, \tilde{\ell}_2$	End points of the support of the dual B-spline generator
$\ell, \tilde{\ell}$	Parameters for the boundary adaption of the B-spline generators
α, β	Matrices of refinement parameters for B-spline generators on the boundary
$\Psi_j^{(0)}, \tilde{\Psi}_j^{(0)}$	Primal and dual intermediate wavelet bases on level j
$\mathbf{\Gamma}$	Gramian matrix used for biorthogonalisation
$\mathbf{V}_j, \mathbf{O}_j$	Transformation matrices used for plotting of dual spline wavelet representations
Z_j	Nodal basis used for plotting, interpolation and integration
$\kappa(\Sigma)$	Condition of any collection of functions Σ
$\kappa(\mathbf{M})$	Condition number of symmetric positive definite matrix $\mathbf{M}$
Y	Hilbert space for the state
y	State variable $y \in Y$
$a(\cdot, \cdot)$	Y-elliptic bilinear form
$\mathbf{D}_a$	Diagonal matrix adapted to $a(\cdot, \cdot)$
$\mathbf{M}$	Mass matrix
$\mathbf{L}$	Laplace matrix
$\mathbf{A}$	Stiffness matrix
$\mathbf{S}$	Singular values of the stiffness matrix on the lowest level

Table of Mathematical Symbols

f	Right hand side for elliptic constraints, $f \in Y'$
g	Normal derivative for Neumann boundary conditions
Z	Observation space
y_*	Target observation, $y_* \in Z$
U	Control space
u	Control variable, $u \in U$
$T, \mathbf{T}$	Trace operator $T : Y \to Z$, and its discretised form
$E, \mathbf{E}$	Extension operator $E : U \to Y'$, and its discretised form
ω	Weight of the regularisation term in the cost functional
$J(y, u)$	Cost functional for the control problem
$\check{\mathbf{J}}(\mathbf{y}, \mathbf{u})$	Discretised cost functional
$\mathbf{J}(\mathbf{u})$	Reduced cost functional in the control variable
$\mathbf{Q}$	System matrix for the all-in-one formulation of the control problem
$\mathbf{g}$	Discretised right hand side for the all-in-one formulation
$\mathbf{p}$	Discretised Lagrangian multiplier
$\mathbf{L}(\mathbf{y}, \mathbf{u}, \mathbf{p})$	Lagrange functional incorporating the linear elliptic constraints
$\mathbf{R}_V$	Riesz matrix for any Hilbert space V
$\mathbf{B}_i$	Riesz matrix to calculate $\|\cdot\|_{H^i}$ in the natural wavelet basis, $i \in \mathbb{N}_0$
$\mathbf{R}_s$	Riesz matrix for the Hilbert space H^s in the natural wavelet basis, $s \in \mathbb{R}$
q_s	Corrective factor to normalise Riesz operator
$\sigma_N(\mathbf{v})$	Error of best N-term approximation of $\mathbf{v}$
ℓ_τ^{w}	Weak ℓ_τ sequence space
$B_{q,p}^{\alpha}$	Besov space of smoothness α over L_p, with additional index q
N_{ad}	Average number of nonzero wavelet coefficients
σ	Rate of error reduction with respect to N_{ad}

Bibliography

[1] R. Adams. *Sobolev Spaces.* Academic Press, 1975.

[2] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi. Adaptive solution of partial differential equations in multiwavelet bases. *J. Comput. Phys.*, 182:149–190, 2002.

[3] H. Alt. *Lineare Funktionalanalysis.* Springer, 2002.

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

[5] A. Barinka, T. Barsch, P. Charton, A. Cohen, S. Dahlke, W. Dahmen, and K. Urban. Adaptive wavelet schemes for elliptic problems – implementation and numerical experiments. *SIAM J. Sci. Comp.*, 23:910–939, 2001.

[6] T. Barsch. *Adaptive Multiskalenverfahren für elliptische partielle Differentialgleichungen – Realisierung, Umsetzung und numerische Ergebnisse.* Shaker, 2001.

[7] R. Becker. Adaptive finite elements for optimal control problems. Habilitationsschrift, Universität Heidelberg, 2001.

[8] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimal control of partial differential equations: Basic concept. *SIAM J. Contr. Optim.*, 39:113–132, 2000.

[9] R. Becker and B. Vexler. A posteriori error estimation for finite element discretization of parameter identification problems. *Numer. Math.*, 96(3):435–459, 2004.

[10] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, 44:141–183, 1991.

[11] L. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, editors. *Large-Scale PDE-Constrained Optimization*, volume 30 of *Lecture Notes in Computational Science and Engineering.* Springer, 2003.

[12] P. Binev, W. Dahmen, and R. DeVore. Adaptive finite element methods with convergence rates. *Numer. Math.*, 97:219–268, 2004.

[13] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part i: the Krylov-Schur solver. To appear in SIAM J. Sci. Comp.

[14] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part ii: the Lagrange-Newton solver, and its application to optimal control of steady viscous flows. To appear in SIAM J. Sci. Comp.

[15] A. Borzì. Multigrid methods for parabolic distributed optimal control problems. *J. Comp. Appl. Math*, 157:365–382, 2003.

[16] A. Borzì and G. Borzì. An algebraic multigrid method for a class of elliptic differential systems. *SIAM J. Sci. Comp.*, 25(1):302–323, 2003.

[17] A. Borzì and K. Kunisch. A multigrid method for the optimal control of time-dependent reaction diffusion processes. In K.H. Hoffmann, R. Hoppe, and V. Schulz, editors, *Fast solution of discretized optimization problems*, volume 138 of *International Series on Numerical Mathematics*. Birkhäuser, 2001.

[18] A. Borzì and K. Kunisch. A multigrid scheme for elliptic constrained optimal control problems. *Comput. Optim. Appl.*, 31:309–333, 2005.

[19] A. Bouras and V. Fraysse. A relaxation strategy for inexact matrix-vector products for Krylov methods. Technical Report TR/PA/00/15, CERFACS, France, 2000.

[20] A. Bouras, V. Fraysse, and L. Giraud. A relaxation strategy for inner-outer linear solvers in domain decomposition methods. Technical Report TR/PA/00/17, CERFACS, France, 2000.

[21] J. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Press, 2nd edition, 2000.

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

[23] J. Bramble. *Multigrid Methods*. Pitman, 1993.

[24] J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 37:1–22, 1981.

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

[26] W. Briggs, V. Henson, and S. McCormick. *A Multigrid Tutorial*. SIAM publications, 2nd edition, 2000.

[27] H. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:1–123, 2004.

[28] C. Burstedde and A. Kunoth. Fast iterative solution of elliptic control problems in wavelet discretization. Preprint No. 127, SFB 611, Universität Bonn, Dec. 2003. Revised and accepted for publication in J. Comp. Appl. Math., Aug. 2005.

[29] J. Carnicer, W. Dahmen, and J. Peña. Local decomposition of refinable spaces. *Appl. Comp. Harm. Anal.*, 3:127–153, 1996.

[30] E. Casas and F. Tröltzsch. Second order necessary optimality conditions for some state-constrained control problems of semilinear elliptic equations. *Appl. Math. Optim.*, 39:211–228, 1999.

[31] E. Casas, F. Tröltzsch, and A. Unger. Second order sufficient optimality conditions for some state-constrained control problems of semilinear elliptic equations. *SIAM J. Cont. Optim.*, 38:1369–1391, 2000.

[32] D. Castaño. *Adaptive Scattered Data Fitting with Tensor Product Spline-Wavelets*. Dissertation, Institut für Angewandte Mathematik, Universität Bonn, 2005.

[33] D. Castaño and A. Kunoth. Adaptive fitting of scattered data by spline-wavelets. In L. Schumaker et.al., editor, *Curves and Surfaces*, pages 65–78. Vanderbilt University Press, 2003.

[34] D. Castaño and A. Kunoth. Multilevel regularization of wavelet based fitting of scattered data – some experiments. *Numer. Algor.*, 39:81–96, 2005.

[35] C. Chui and E. Quak. Wavelets on a bounded interval. In D. Braess and L. Schumaker, editors, *Numerical Methods of Approximation Theory*, pages 1–24. Birkhäuser, 1992.

[36] P. Ciarlet. *The Finite Element Method for Elliptic Problems.* Studies in Mathematics and its Applications. North-Holland, 1978.

[37] P. Ciarlet and J. Lions, editors. *Handbook of Numerical Analysis II – Finite Element Methods.* North-Holland, 1991.

[38] A. Cohen. Wavelet methods in numerical analysis. In P. Ciarlet and J. Lions, editors, *Handbook of Numerical Analysis*, volume VII. Elsevier, 1991.

[39] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations – convergence rates. *Math. Comp.*, 70:27–75, 2001.

[40] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods II – beyond the elliptic case. *Found. Computat. Math.*, 2:203–245, 2002.

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

[42] A. Cohen, I. Daubechies, and J. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure an Appl. Math.*, 45:485–560, 1992.

[43] A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transforms. *Appl. Comp. Harm. Anal.*, 1:54–81, 1993.

[44] G. Corliss, C. Faure, A. Griewank, L. Hascoët, and U. Naumann. *Automatic Differentiation: from Simulation to Optimization.* Springer, 2001.

[45] W. Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl.*, 4:341–362, 1996.

[46] W. Dahmen. Wavelet and multiscale methods for operator equations. *Acta Numerica*, 6:55–228, 1997.

[47] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63:315–344, 1992.

[48] W. Dahmen and A. Kunoth. Adaptive wavelet methods for linear-quadratic elliptic control problems: convergence rates. *SIAM J. Contr. Optim.*, 43(5):1640–1675, 2005.

[49] W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal spline-wavelets on the interval – stability and moment conditions. *Appl. Comput. Harm. Anal.*, 6:132–196, 1999.

[50] W. Dahmen and C. Micchelli. Using the refinement equation for evaluating integrals of wavelets. *SIAM J. Numer. Anal.*, 30:507–537, 1993.

[51] W. Dahmen, S. Prößdorf, and R. Schneider. Multiscale methods for pseudo-differential equations on smooth manifolds. In C.K. Chui, L. Montefusco, and L. Puccio, editors, *Proceedings of the International Conference on Wavelets: Theory, Algorithms, and Applications*, pages 385–424. Academic Press, 1994.

[52] W. Dahmen and R. Schneider. Composite wavelet bases for operator equations. *Math. Comp.*, 68:1533–1567, 1999.

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

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

[55] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.

[56] I. Daubechies. Ten lectures on wavelets. *CBMS-NSF Regional Conference Series in Applied Math.*, 61, 1992.

[57] R. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[58] R. DeVore, B. Jawerth, and V. Popov. Compression of wavelet decompositions. *Amer. J. Math.*, 114:737–785, 1992.

[59] R. DeVore and B. Lucier. Wavelets. *Acta Numerica*, 1:1–56, 1992.

[60] R. DeVore and V. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Math.*, 5:173–1124, 1996.

[61] D. Donoho. Ridge functions and orthonormal ridgelets. Manuscript, 1998.

[62] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33:1106–1124, 1996.

[63] T. Dreyer, B. Maar, and V. Schulz. Multigrid optimization in applications. *J. Comp. Appl. Maths.*, 120:67–84, 2000.

[64] K. Eppler and F. Tröltzsch. Fast optimization methods in the selective cooling of steel. In M. Grötschel, S. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems*, pages 185–204. Springer, 2001.

[65] M. Fahl and E. Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In L.T. Biegler et al., editor, *Large-scale PDE-constrained optimization*, volume 30 of *Lect. Notes Comput. Sci. Eng.*, pages 268–280. Springer, 2003.

[66] C. Feuersänger. Dünngitterverfahren für hochdimensionale elliptische partielle Differentialgleichungen. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2005.

[67] A. Fursikov, M. Gunzburger, and L. Hou. Boundary value problems and optimal boundary control for the Navier-Stokes system: The two-dimensional case. *SIAM J. Cont. Optim.*, 36:852–894, 1998.

[68] J. Geronimo, D. Hardin, and P. Massopust. Fractal functions and wavelet expansions based on several scaling functions. *J. Approx. Th.*, 78:373–401, 1994.

[69] G. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Sci. Comput.*, 21:1305–1320, 2000.

[70] M. Griebel and S. Knapek. Optimized tensor-product approximation spaces. *Constr. Approx.*, 16(4):525–540, 2000.

[71] M. Griebel and F. Koster. Adaptive wavelet solvers for the unsteady incompressible Navier Stokes equations. In J. Malek, J. Necas, and M. Rokyta, editors, *Advances in Mathematical Fluid Mechanics*, volume 82 of *Lecture Notes of the Sixth International School Mathematical Theory in Fluid Mechanics*. Springer, 2000.

[72] M. Griebel, D. Oeltz, and M. A. Schweitzer. An algebraic multigrid method for linear elasticity. *SIAM J. Sci. Comp.*, 25(2):385–407, 2003.

[73] M. Griebel, D. Oeltz, and P. Vassilevski. Space-time approximation with sparse grids. Preprint UCRL-JRNL-211686, Lawrence Livermore National Laboratory, 2005. Also as Preprint No. 222, SFB 611, Universität Bonn. Submitted for publication.

[74] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numer. Math.*, 70:163–180, 1995.

[75] M. Griebel and M. A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic PDE. *SIAM J. Sci. Comp.*, 22(3):853–890, 2000.

[76] M. Griebel and M. A. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations*, volume 26 of *Lecture Notes in Computational Science and Engineering*. Springer, 2002.

[77] M. Griebel and M. A. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations II*, volume 43 of *Lecture Notes in Computational Science and Engineering*. Springer, 2005.

[78] M. Griebel and G. Zumbusch. Parallel multigrid in an adaptive PDE solver based on hashing and space-filling curves. *Parallel Computing*, 25:827–843, 1999.

[79] A. Griewank and A. Walther. Advantages of binomial checkpointing for memory-reduced adjoint calculations. In M. Feistauer et al., editor, *Proceedings of ENUMATH 2003, the 5th European conference on numerical mathematics and advanced applications, Prague, Czech Republic, August 18-22, 2003*, Numerical mathematics and advanced applications, pages 834–843. Springer, 2004.

[80] M. Gunzburger. *Perspectives in Flow Control and Optimization*. Advances in Design and Control. SIAM, 2003.

[81] M. Gunzburger and L. Hou. Finite dimensional approximation of a class of constrained nonlinear optimal control problems. *SIAM J. Cont. Opt.*, 34:1001–1043, 1996.

[82] M. Gunzburger and H. Lee. Analysis, approximation, and computation of a coupled solid/fluid temperature control problem. *Comp. Meth. Appl. Mech. Engrg.*, 118:133–152, 1994.

[83] M. Gunzburger and S. Manservisi. Analysis and approximation of the velocity tracking problem for Navier-Stokes flows with distributed control. *SIAM J. Numer. Anal.*, 37:1481–1512, 2000.

[84] W. Hackbusch. Fast solution of elliptic control problems. *J. Optim. Theory Appl.*, 31:565 – 581, 1980.

[85] W. Hackbusch. *Multigrid Methods and Applications*. Springer, 1985.

[86] W. Hackbusch. *Elliptic Differential Equations*. Springer, 1992.

[87] J. Haslinger and R. Mäkinen. *Introduction to shape optimization. Theory, approximation, and computation*, volume 7 of *Advances in Design and Control*. SIAM, 2003.

[88] M. Heinkenschloss. Time-domain decomposition iterative methods for the solution of distributed linear quadratic optimal control problems. *J. Comp. Appl. Math*, 173:169–198, 2005.

[89] M. Heinkenschloss and L. Vicente. Analysis of inexact trust-region SQP algorithms. *SIAM J. Optimization*, 12(2):283–302, 2001.

[90] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. NBS*, 49:409–436, 1952.

[91] V. Heuveline and R. Rannacher. Duality-based adaptivity in the hp-finite element method. *Numer. Math.*, 11(2):95–103, 2003.

[92] M. Hintermüller. On a globalized augmented lagrangian SQP-algorithm for nonlinear optimal control problems with box constraints. In K. Hoffmann, R. Hoppe, and V. Schulz, editors, *Fast Solution Methods for Discretized Optimization Problems*, pages 139–153. Birkhäuser, 2001.

[93] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semi-smooth Newton method. *SIAM Journal on Optimization*, 13:865–888, 2002.

[94] M. Hinze. A variational discretization concept in control constrained optimization: the linear-quadratic case. *Comput. Optim. Appl.*, 30:45–63, 2005.

[95] M. Hinze and K. Kunisch. Second order methods for boundary control of the instationary Navier-Stokes system. *ZAMM*, 84(3):171–187, 2004.

[96] R. Hoppe and R. Kornhuber. Adaptive multilevel methods for obstacle problems. *SIAM J. Numer. Anal.*, 31:301–323, 1994.

[97] R. Hoppe and B. Wohlmuth. Adaptive multilevel techniques for mixed finite element discretizations of elliptic boundary value problems. *SIAM J. Numer. Anal.*, 34:1658–1681, 1997.

[98] K. Ito and K. Kunisch. Semi-smooth Newton methods for variational inequalities of the first kind. *Mathematical Modelling and Numerical Analysis*, 37:41–62, 2002.

[99] F. Koster. *Multiskalen-basierte Finite-Differenzen-Verfahren auf adaptiven dünnen Gittern.* Dissertation, Institut für Angewandte Mathematik, Universität Bonn, 2002.

[100] J. Krumsdorf. Finite element wavelets for the numerical solution of elliptic partial differential equations on polygonal domains. Diploma thesis, Institut für Angewandte Mathematik, Universität Bonn, 2004.

[101] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2002.

[102] A. Kunoth. *Multilevel Preconditioning.* Shaker, 1994.

[103] A. Kunoth. *Wavelet Methods – Elliptic Boundary Value Problems and Control Problems.* Teubner, 2001.

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

[105] A. Kunoth. Adaptive wavelet schemes for an elliptic control problem with Dirichlet boundary control. *Numer. Algor.*, 39(1-3):199–220, 2005.

[106] A. Kunoth. Wavelet-based multiresolution methods for stationary PDEs and PDE-constrained control problems. In J. Blowey and A. Craig, editors, *Frontiers in Numerical Analysis (Durham 2004)*, Universitext, pages 1–63. Springer, 2005.

[107] A. Kunoth and J. Sahner. Wavelets on manifolds: An optimized construction. Preprint No. 163, SFB 611, Universität Bonn, July 2004, revised, Apr. 2005, to appear in Math. Comp.

[108] R. Li, W. Liu, H. Ma, and T. Tang. Adaptive finite element approximation for distributed elliptic optimal control problems. *SIAM J. Control. Optim.*, 41(5):1321–1349, 2002.

[109] J. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer, 1971.

[110] C. Makridakis and R. Nochetto. Elliptic reconstruction and a posteriori error estimates for parabolic problems. *SIAM J. Numer. Anal.*, 41(4):1585–1594, 2003.

[111] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbb{R})$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

[112] Y. Meyer. *Ondelettes et opérateurs 1–3: Ondelettes.* Hermann, 1990.

[113] B. Mohammadi. Shape optimization for 3D turbulent flows using automatic differentiation. *Int. J. Comput. Fluid Dyn.*, 11(1-2):27–50, 1998.

[114] P. Morin, R. Nochetto, and K. Siebert. Data oscillation and convergence of adaptive fem. *SIAM J. Numer. Anal.*, 38(2):466–488, 2000.

[115] P. Morin, R. Nochetto, and K. Siebert. Convergence of adaptive finite element methods. *SIAM Rev.*, 44(4):631–658, 2002.

[116] P. Oswald. On discrete norm estimates related to multilevel preconditioners in the finite element method. In K.G. Ivanov, P. Petrushev, and B. Sendov, editors, *Constructive Theory of Functions*, Proc. Int. Conf. Varna, pages 203–214, 1991.

[117] P. Oswald. *Multilevel finite element approximation: Theory and applications.* Teubner, 1994.

[118] R. Pabel. Wavelet methods for a PDE constrained control problem with Dirichlet boundary control. Diploma thesis, Institut für Angewandte Mathematik, Universität Bonn, 2005. In preparation.

[119] W. Ritz. Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. *J. reine angew. Math.*, 135:1–61, 1908.

[120] A. Rösch and F. Tröltzsch. Sufficient second order optimality conditítions for a state-constrained optimal control problem of a weakly singular integral equation. *Num. Funct. Anal. Appl.*, 23:173–193, 2002.

[121] A. Rösch and F. Tröltzsch. Sufficient second order optimality conditítions for a parabolic optimal control problem with pointwise control-state constraints. *SIAM J. Cont. and Optim.*, 42:138–154, 2003.

[122] J. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. McCormick, editor, *Multigrid methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, 1987.

[123] S. Sauter and C. Schwab. Realization of hp-Galerkin BEM in 3-d. In W. Hackbusch and G. Wittum, editors, *Boundary Elements: Implementation and Analysis of Advanced Algorithms*, volume 54 of *NNNFM*, pages 194–206. Vieweg, 1996.

[124] A. Schiela and M. Weiser. Function space interior point methods for PDE constrained optimization. *PAMM*, 4(1):43–46, 2004.

[125] R. Schneider. Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme. Habilitationsschrift, Technische Hochschule Darmstadt, 1995.

[126] V. Schulz. Simultaneous solution approaches for large optimization problems. *J. Comp. Appl. Math*, 164-165:629–641, 2004.

[127] M. A. Schweitzer. Implementation and parallelization of meshfree methods. In J. Blowey and A. Craig, editors, *Frontiers in Numerical Analysis (Durham 2004)*, Universitext, pages 195–257. Springer, 2005.

[128] R. Stevenson. Locally supported, piecewise polynomial biorthogonal wavelets on non-uniform meshes. *Constr. Approx.*, 19(4):477–508, 2003.

[129] R. Stevenson. On the compressibility of operators in wavelet coordinates. *SIAM J. Math. Anal.*, 35(5):1110–1132, 2004.

[130] R. Stevenson. An optimal adaptive finite element method. *SIAM J. Numer. Anal.*, 42(5):2188–2217, 2005.

[131] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comp. Harm. Anal.*, 3:186–200, 1996.

[132] W. Sweldens and R. Piessens. Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions. *SIAM J. Num. Anal.*, 31:2140–2164, 1994.

[133] P. Tchamitchian. Wavelets, functions, and operators. In G. Erlebacher, M.Y. Hussaini, and L. Jameson, editors, *Wavelets: Theory and Applications*, Series in Computational Science and Engineering, pages 83–181. Oxford University Press, 1996.

[134] H. Triebel. *Interpolation Theory, Function Spaces and Differential Operators.* North Holland, 1978.

[135] F. Tröltzsch. On the Lagrange-Newton-SQP method for the optimal control of semilinear parabolic equations. *SIAM J. Contr. Optim.*, 38:294–312, 1999.

[136] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen – Theorie, Verfahren und Anwendungen.* Vieweg, 2005.

[137] F. Tröltzsch and A. Unger. Fast solution of optimal control problems in the selective cooling of steel. *ZAMM*, 81:447–456, 2001.

[138] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, 2001.

[139] J. van den Eshof and G. Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM J. Matrix Anal. A.*, 26(1):125–153, 2004.

[140] J. van den Eshof, G. Sleijpen, and M. van Gijzen. Relaxation strategies for nested Krylov methods. Preprint 1268, Department of Mathematics, Utrecht University, 2003.

[141] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Wiley-Teubner, 1996.

[142] S. Volkwein. Optimal and suboptimal control of partial differential equations: Augmented Lagrange-SQP methods and reduced-order modeling with proper orthogonal decomposition. *Grazer Math. Ber.*, 343(vi):1–131, 2001.

[143] T. von Petersdorff and C. Schwab. Fully discrete multiscale Galerkin BEM. In W. Dahmen, A. Kurdila, and P. Oswald, editors, *Multiscale Wavelet Methods for PDEs*, pages 287–346. Academic Press, 1997.

[144] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34:581–613, 1992.

[145] H. Yserentant. On the multilevel splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

Danksagung

An erster Stelle danke ich Frau Professor Dr. Angela Kunoth für die Anstellung im Projekt C1 des SFB 611, und die Freiheit und ihr fortwährendes Vertrauen beim Stöbern, Forschen und Entwickeln, für konstruktive Diskussion, Rat und Unterstützung beim Erarbeiten der vorliegenden Ergebnisse, ihren Einsatz bei den Korrekturen, und die Versorgung mit letztendlich immer auf sinnvolle und gewinnbringende Art erfüllbaren Herausforderungen.

Professor Dr. Michael Griebel danke ich für die Übernahme des Zweitgutachtens und weitere unabhängige, hilfreiche Einschätzungen.

Peter Orbanz verdanke ich den Hinweis, die Universität Bonn zum Promovieren in Betracht zu ziehen, und auch sonst den einen oder anderen Impuls.

Daniel Castaño Díez danke ich für offene und freundschaftliche Kollegialität in wissenschaftlichen und persönlichen Fragen, die gelegentlich notwendige gemeinsame Herstellung eines gewissen Abstands zum aktuellen Geschehen, und allgemeinen Optimismus in bewegten Zeiten.

Roland Pabel danke ich für den Austausch von Erfahrungen und Ideen und die gegenseitige Verifikation beim Implementieren unserer beiden voneinander unabhängigen Waveletcodes.

Ich bedanke mich bei den aktuellen und ehemaligen Mitarbeitern, Mitdoktoranden und Studenten am Institut für Angewandte Mathematik und am Institut für Numerische Simulation, die mir als freundlich, hilfsbereit und fair in Erinnerung bleiben werden. —

Der B dankt den beiden Os für universelle Freundschaft, und überhaupt verdammt viel. Pimpmobile für den Rock. Alten Freunden. Und dem zusammengewürfelten Haufen aus Musikern, Chaoten und Lebenskünstlern für die Bevölkerung seiner alternativen Realität. Nicht zuletzt dankt er den Menschen, die die Bücher geschrieben, die er in dieser Zeit gelesen, und denen, die die Musik gespielt, die er in dieser Zeit gehört.

Meine Eltern und meine Geschwister haben mir den nötigen Rückhalt, Sicherheit und Zuversicht gegeben, längst nicht nur, aber auch um mich immer wieder in diese Arbeit vertiefen zu können.

Mit ganzem Herzen danke ich der lieben, großartigen Katja, die so lange wunderbar auf meiner Seite war.

Und ich danke Marie. —

I have benefited enormously, in the context of this thesis and for other projects, from the existence of (in alphabetical order) CVS, Debian, Emacs, gcc, glib/GTK+, Gnuplot, LaTeX, libpng, Linux, Octave, OpenGL, Perl, Postscript, Povray, Python. Thanks also to Clive Barker, id Software, Stanley Kubrick and the developers of Painkiller and Unreal.

Without judgement what would we do?
We would be forced to look
At ourselves emerged in lost time
Assuming what may be
Without judgement
Perception would increase a million times

C. Schuldiner

Aber es gehört mehr Muth *dazu, ein Ende zu machen, als einen neuen Vers: das wissen alle Ärzte und Dichter.*

F. Nietzsche

Lebenslauf

Dipl.-Phys. Carsten Burstedde

Persönliche Daten

1975	Geboren in Köln
1982 – 1986	Besuch der Grundschule Frankenforst in Bergisch Gladbach, Bensberg
1986 – 1994	Besuch des Otto-Hahn-Gymnasiums in Bensberg
17.05.1994	Abitur (Note 1.0)
Okt. 1994 – Sep. 1995	Grundwehrdienst
Okt. 1995	Beginn des Studiums der Physik an der Universität zu Köln
Sep. 1997	Vordiplome in Physik und Mathematik (sehr gut)
Okt. 1997 – Juni 1998	Auslandsjahr an der University of Edinburgh, Schottland
29.03.2001	Diplom der Physik an der Universität zu Köln (mit Auszeichnung)
seit Juli 2001	Doktorand von Frau Prof. Dr. Angela Kunoth und wissenschaftlicher Mitarbeiter des SFB 611 (DFG) am Institut für Angewandte Mathematik der Universität Bonn

Stipendien

1995 – 2000	Stipendiat der Studienstiftung des Deutschen Volkes
1997 – 1998	Auslandsstipendium des Deutschen Akademischen Austauschdienstes

Auszeichnungen

1994	Einer der fünf Bundessieger der 25. Internationalen Physikolympiade (Schülerwettbewerb) und Mitglied der deutschen Mannschaft
1994	Silbermedaille und Sonderpreis für beste Lösung einer Aufgabe bei der 25. Internationalen Physikolympiade in Peking, China
1995	Schülerpreis der Deutschen Physikalischen Gesellschaft
2003	Hausdorffpreis der Mathematisch-Naturwissenschaftlichen Fakultät, Universität Bonn

Veröffentlichungen

[1] C. Burstedde. Simulation von Fußgängerverhalten mittels zweidimensionaler zellulärer Automaten. Diplomarbeit, Universität zu Köln, März 2001.

[2] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A*, 295:507–525, 2001.

[3] C. Burstedde, A. Kirchner, K. Klauck, A. Schadschneider, and J. Zittartz. Cellular automaton approach to pedestrian dynamics – applications. In M. Schreckenberg and S. Sharma, editors, *Pedestrian and Evacuation Dynamics*. Springer, 2002.

[4] M. Mützel, U. Rasbach, D. Meschede, C. Burstedde, J. Braun, A. Kunoth, K. Peithmann, and K. Buse. Atomic nanofabrication with complex light fields. *Appl. Phys. B*, 77:1–9, 2003.

[5] C. Burstedde and A. Kunoth. Fast iterative solution of elliptic control problems in wavelet discretization. Preprint Nr. 127, SFB 611, Universität Bonn, Dez. 2003. Überarbeitet und angenommen zur Veröffentlichung in J. Comp. Appl. Math., Aug. 2005.