DIPLOMARBEIT

Waveletmethoden für PDE-basierte elliptische Kontrollprobleme mit Dirichlet-Randkontrolle

(Wavelet Methods for PDE Constrained Elliptic Control Problems with Dirichlet Boundary Control)

Angefertigt am
Institut für Angewandte Mathematik

Vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Dezember 2005

von

Roland Pabel

aus

Köln

Preface

[This work] may well win me the Nobel Prize.
In what field?
I don't care - they all pay the same.
(Prof. Hubert Farnsworth and Turanga Leela,
Futurama, Episode "Mars University")

This diploma thesis is concerned with wavelet methods applied to control problems constrained by an elliptic partial differential equation with Dirichlet boundary control. Particular emphasis is placed on quick and efficient solution strategy of the resulting linear system of optimality equations.

I express my sincere gratitude to Prof. Dr. Angela Kunoth for her trust to enabling me to work at the Institut für Angewandte Mathematik (IAM) and the Institut für Numerische Simulation (INS), which now culminates in this diploma thesis. I am indebted for her everlasting support and committed supervision, her comments and suggestions, which shaped not only this work but greatly influenced my studies as a whole. I am also very grateful for the possibility to work in the Collaborative Research Center (SFB) 611, "Singular Phenomena and Scaling in Mathematical Models", and for financial support I received during the past years. The time spent on the internal group seminars of the students and (post-)doctoral students of Prof. Dr. Michael Griebel and Prof. Kunoth in Blankenheim (Eifel), the Netherlands and Xanten were both enjoyable and encouraging. I wish to thank Prof. Dr. Rolf Krause for his willingness to be the co-expert on this thesis.

A debt of gratitude goes to Dr. Carsten Burstedde for his help on the intricacies of operators in wavelet discretizations and the possibility to cross check our code and the computed results. I also like to thank Miriam Primbs (Universität Duisburg-Essen) for giving me access to her latest research on wavelet construction and the permission to use it for my work.

I hereby apologize to all my colleagues, coworkers and superiors at the IAM and INS for any inconvenience caused by my seemingly endless computations on their computers and workstations. Hopefully, I fixed their computers (when they broke down) as often as I rendered them unusable (with my program) ;-).

I thank my friends for diverting me from my work from time to time. My special thanks go to my former colleague Julian Krumsdorf for his support and interest in my work. Last but not least, I thank my parents for their support during all these years and Jürgen and Ralf for keeping up the brotherly competition we have indulged in since our childhood.

Bonn, December 2005 Roland Pabel

Contents

Zusammenfassung[1]

Optimale Kontrollprobleme sind seit Jahren Gegenstand umfangreicher Forschungen und numerischer Studien. In dieser Arbeit wenden wir Wavelets auf Kontrollprobleme an, welche auf linearen elliptischen partiellen Differentialgleichungen (PDE) basieren sowie auf Kostenfunktionalen mit nicht-ganzzahligen Sobolevnormen. Dabei setzen wir den Schwerpunkt auf die effiziente numerische Behandlung der PDE-Nebenbedingungen und die exakte Behandlung der rationalen Sobolevnormen. Diese Ziele erreichen wir durch die Verwendung von Waveletmethoden in Verbindung mit Normäquivalenzen. Waveletvorkonditionierungsmethoden sind asymptotisch **optimal**: die resultierenden diskretisierten linearen Operatoren besitzen gleichmäßig beschränkte spektrale Kondition. Folglich können die resultierenden linearen Gleichungssysteme iterativ mit einem Aufwand an arithmetischen Operationen gelöst werden, der **proportional** zur Anzahl der Unbekannten ist.

In der jüngeren Vergangenheit haben sich Wavelets bei der Lösung von elliptischen Differentialgleichungen bewährt. Die Stärken von Wavelets kommen besonders in der Numerischen Analysis durch das theoretische Grundgerüst, insbesondere hier durch die **Multiresolution Analysis** (**MRA**), zum Tragen. Die Riesz-Basis-Eigenschaft der Waveletbasen erlaubt es, Anwendungen mit Normen von gebrochen rationalen Sobolevräumen zu bearbeiten; ein Vorteil gegenüber Finite-Elemente-Methoden (**FEM**), welche nur Berechnungen mit ganzzahligen Sobolevindizes zulassen. Ein weiterer Vorteil ist die Verfügbarkeit von biorthogonalen Waveletbasen, d.h. die Existenz primaler und dualer Waveletbasen, $\mathbf{\Psi}, \widetilde{\mathbf{\Psi}}$ mit der Eigenschaft $\langle \mathbf{\Psi}, \widetilde{\mathbf{\Psi}} \rangle = \mathbf{1}$. Diese können unabhängig voneinander gewählt werden um **Momentenbedingungen** beliebiger Ordnung und **Normäquivalenzen** für beliebig glatte Sobolevräume zu erfüllen. Genauer gesagt erzeugen Konstruktionen aktuell verfügbarer primaler Wavelets stückweise polynomiale Waveletbasen mit kompaktem Träger, welche in expliziter Form angegeben werden können. Die dualen Waveletbasen werden anhand der aus der Problemstellung benötigten Regularität in den **Jackson-** und **Bernsteinungleichungen** konstruiert und sind in der Regel nur implizit gegeben. Die daraus resultierenden Normäquivalenzen ermöglichen die präzise Behandlung von Kontrollproblemen mit gebrochenen Sobolevnormen der Kontrolle und des Zustands mit dem gleichen Aufwand wie im L_2-Fall. Dies ist sogar noch für den Fall von dualen (gebrochenen) Sobolevnormen durchführbar (vgl. [12]).

Diese Diplomarbeit befaßt sich mit Waveletmethoden zur Lösung von elliptischen Kontrollproblemen mit Dirichlet-Randkontrolle. Bei dieser Klasse von Problemen erscheint die Kontrolle u als Dirichlet-Randbedingung in der partiellen Differentialgleichung, welche die Nebenbedingung des zu minimierenden Funktionals darstellt. Das abstrakte Problem ist folgendermaßen gestellt:

Gegeben seien Daten y_{Γ_y} und f. Minimiere das Funktional

$$\mathcal{J}(y,u) = \frac{1}{2}\|y - y_{\Gamma_Y}\|^2_{Y,\Gamma_Y} + \frac{\omega}{2}\|u\|^2_{Q,\Gamma}, \tag{1}$$

wobei der Zustand y und die Kontrolle u über das lineare elliptische Randwertproblem

$$\begin{aligned} -\nabla\cdot(\mathbf{a}\nabla y) + k\,y &= f && \text{in } \Omega, \\ y &= u && \text{auf } \Gamma, \\ (\mathbf{a}\nabla y)\cdot\mathbf{n} &= 0 && \text{auf } \partial\Omega\setminus\Gamma, \end{aligned} \tag{2}$$

zusammenhängen.

Die Randbedingungen im Randwertproblem (2) werden in schwacher Form durch Spuroperatoren umgesetzt und mittels **Lagrangescher Multiplikatoren** an die schwache Formulierung des elliptischen Differentialoperators angehängt. Das aus (2) resultierende Gleichungssystem besteht nur aus dem schwachen elliptischen Differentialoperator, dem Spuroperator und dem adjungierten Spuroperator,

$$L\begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ u \end{pmatrix}\ . \tag{3}$$

Durch diesem Ansatz ist das Randwertproblem als **Sattelpunktproblem** formuliert. Solch ein indefinites System ist numerisch schwieriger zu behandeln als positiv definite Systeme und erfordert deshalb spezielle Lösungsalgorithmen. Dieses Setup erlaubt auch ein einfaches Aufdatieren der Randbedingungen u, was

[1] Der folgende Text ist nach der alten deutschen Rechtschreibung verfaßt.

wiederum für die effiziente Behandlung des Kontrollproblems wichtig sein wird.
Das Sattelpunktproblem wird anschließend durch Lagrangesche Multiplikatoren an das Funktional (1) angebunden:

$$\hat{\mathcal{J}}(y,p,u,z,\mu) := \mathcal{J}(y,u) + \left\langle \begin{pmatrix} z \\ \mu \end{pmatrix}, L \begin{pmatrix} y \\ p \end{pmatrix} - \begin{pmatrix} f \\ u \end{pmatrix} \right\rangle . \tag{4}$$

Die notwendigen und hinreichenden Bedingungen für die eindeutige Lösung des Kontrollproblems sind nun durch die Optimalitätsbedingungen erster Ordnung des Funktionals (4) gegeben. Da das Funktional (1) strikt konvex ist, müssen Bedingungen höherer Ordnung nicht in Betracht gezogen werden. Dies führt zu einem gekoppeltem System von Sattelpunktproblemen in Verbindung mit einer dritten Gleichung, die Kontrolle und adjungierten Zustand koppelt.

Bei der Modellierung des Kontrollproblems in Waveletkoordinaten sind zwei Details besonders zu berücksichtigen. Das Funktional (1) soll so exakt wie möglich behandelt werden. Da die Kontrolle u als Spur einer Funktion in einem ganzzahligen Sobolevraum auftritt, lebt u in einem gebrochen rationalen Sobolevraum. Die natürlichen Normen im Kostenfunktional sind daher natürlicherweise von gebrochen rationalen Sobolevnormen gegeben. Diese werden wir durch Normäquivalenzen mit Wavelets modellieren. Um die Normen möglichst exakt zu repräsentieren, werden wir **Riesz Operatoren** für diese Räume einsetzen. Die Lösung des Sattelpunktsystems (3) muß zur Lösung des Kontrollproblems mindestens einmal berechnet werden. Dies ist wegen der oben genannten Waveleteigenschaften bezüglich Vorkonditionierungsmethoden effizient möglich (vgl. [20]). Schließlich läßt sich eine unendlich-dimensionale Darstellung des Kontrollproblems in Waveletkoordinaten erreichen, die nur aus ℓ_2-Normen und -Operatoren besteht.

Die Stabilität der endlich-dimensionalen ℓ_2-Operatoren, abgeleitet aus ihren unendlich-dimensionalen Gegenstücken durch Abschneiden der Indexmenge, wird in diesem Rahmen durch die Galerkinstabilität und die Ladysenškaya-Babuška-Brezzi (**LBB**)-Bedingung garantiert. Diese Operatoren werden sich als **gleichmäßig invertierbare** ℓ_2-Automorphismen mit **gleichmäßig beschränkten** Konditionszahlen ergeben. Wir werden zwei unterschiedliche effiziente Lösungansätze zur Lösung der resultierenden gekoppelten linearen Gleichungssysteme aufzeigen.

In [46] wurde ein **vollständig iterativer Algorithmus** zur effizienten Berechnung des Zustands und der Kontrolle als Lösung der gekoppelten linearen Gleichungssysteme vorgestellt. Diese Methode entspricht einem **inexakten Gradientenverfahren** mit einer äußeren Iterationsvorschrift, welche von einem Schrittweitenparameter ρ abhängt, und zwei inneren Iterationsverfahren. In jedem (äußeren) Iterationsschritt wird jeweils das primale Sattelpunktproblem, mit der Kontrolle als rechter Seite und dem Zustand als Unbekannte, sowie das adjungierte Problem bis zu einer durch die Diskretisierung vorgegebenen Genauigkeit gelöst. Zur Lösung des Sattelpunktsystems kommen **Uzawa**-Algorithmen oder das Verfahren der konjugierten Gradienten zur Anwendung. Dem Schrittweitenparameter ρ muß dabei besondere Aufmerksamkeit gewidmet werden, da er die Konvergenz und die Konvergenzgeschwindigkeit des gesamten Verfahrens bestimmt. Alternativ können die gekoppelten Gleichungssysteme durch Variablenelimination in eine Gleichung mit einem (un)symmetrischen linearen Operator überführt werden, welche dann mit wohlbekannten iterativen oder direkten numerischen Verfahren gelöst werden können.

Die Schachtelung der endlich-dimensionalen Waveleträume ermöglicht sofort den Einsatz einer **geschachtelten Iteration**. Dabei wird die bis zur Diskretisierungsfehlergenauigkeit berechnete Lösung auf einem niedrigen Level als Startwert auf dem nächst höheren Level benutzt. Der Fehler der prolongierten Lösung muß dann nur um einen konstanten Faktor auf jedem Level reduziert werden. Beide oben genannten Lösungsverfahren besitzen **optimale Komplexität** bezüglich der Anzahl der Unbekannten. Dabei wirkt sich die Auswahl der Problemparameter wie z.B. des Schrittweitenparameters oder der geforderten Glattheit der Spuroperatoren auf den Gesamtaufwand und die Konvergenzgeschwindigkeit der Lösungsalgorithmen aus. Die Untersuchung dieser Effekte ist ein weiterer Teil dieser Arbeit. Ebenfalls werden wir in dieser Studie die Kontrolle und den Zustand z.B. bei festen rechten Seiten und variierenden Ordnungen der Spuroperatoren graphisch darstellen. Dabei wird auch der Wert des Parameters ω verändert werden, welcher die Gewichtung der einzelnen Anteile im Kostenfunktional (1) steuert. Für $\omega > 1$ wird die Kontrolle bestraft, also stärker bewertet, und sollte aus diesem Grund niedrigere Werte in der Norm annehmen.

Die Zielsetzung dieser Arbeit ist die Ausarbeitung und Präsentation der Theorie und der numerischen Ergebnisse, welche zur Behandlung der oben genannten Problemstellung benötigt werden. Dazu wird die

Theorie der Thematik linearer elliptischer Kontrollprobleme mit Dirichlet-Randkontrolle dargelegt. Wir repetieren einige elementare Ergebnisse und Definitionen der Funktionalanalysis und Numerischen Mathematik, welche für das Verständnis der Thematik unerläßlich sind. Anschließend präsentieren wir die ermittelten numerischen Ergebnisse. Insbesondere zeigen wir die Auswirkungen variierender Sobolevnormen auf die Effizienz der numerischen Verfahren und die Qualität der sich ergebenen Lösungen auf. Diese Ausführungen sollen die in [46] und [47] in der Theorie bewiesenen Behauptungen praktisch bestätigen.

Übersicht

Die Arbeit ist wie folgt gegliedert:

Section 1
In Kapitel 1 wiederholen wir einige Grundlagen aus der Funktionalanalysis, welche für das Verständnis des Themas unabdingbar sind. Dazu gehören die Definitionen von Sobolevräumen und Spuroperatoren. Eine kurze Einführung in die Theorie der elliptischen partiellen Differentialgleichungen schließt das Kapitel ab.

Section 2
In Kapitel 2 wird die Theorie der Multiskalen-Techniken und die Konstruktion der benutzten eindimensionalen biorthogonalen Waveletbasis beschrieben. Darauf aufbauend wird die Tensorproduktkonstruktion für multidimensionale Gebiete dargestellt. Abschließend folgen einige Bemerkungen über Verbesserungen der Vorkonditionierungstechniken und zur Konstruktion von Riesz-Operatoren für beliebige Sobolevräume. Dieser Abschnitt ist essentiell für das Verständnis der numerischen Methoden und Ergebnisse.

Section 3
In Kapitel 3 diskutieren wir Sattelpunktprobleme und ihre Anwendung auf elliptische Randwertprobleme. Wir zeigen, wie schwach formulierte elliptische Randwertprobleme als Sattelpunktprobleme interpretiert werden und wie diese im Waveletrahmen auf natürliche Weise effizient gelöst werden können. Die Konstruktion der Operatoren und der Aufbau des im Rahmen unseres Kontrollproblems benötigten elliptischen Randwertproblems werden ausgeführt. Schließlich geben wir noch die Lösungsalgorithmen der **Konjugierten Gradienten**- und **Uzawa**-Algorithmen für Sattelpunktprobleme an, welche die Lösung mit optimalem Aufwand berechnen.

Section 4
In Kapitel 4 untersuchen wir die Eigenschaften der behandelten Kontrollprobleme im Detail und leiten die verschiedenen äquivalenten Repräsentationen unseres Kontrollproblems durch die Wavelettheorie her. Wir zeigen notwendige und hinreichende Bedingungen in Form von gekoppelten linearen Gleichungssystemen auf, welche die Lösung unseres Kontrollproblems charakterisieren. Zur Darstellung des Kontrollproblems als ℓ_2-Problem benutzen wir hier die vorher behandelte Wavelettheorie. Als letztes präsentieren wir die Lösungsalgorithmen **All-In-One** und **Inexact Gradient**, welche die optimale Kontrolle in asymptotisch optimalem Aufwand ermitteln.

Section 5
In Kapitel 5 präsentieren wir die von uns berechneten numerischen Ergebnisse. Dazu gehören Konditionszahlen, Iterationsstudien, Konvergenzraten und graphische Darstellungen der Lösungen. Dabei soll insbesondere die Effizienz der Waveletmethoden unter Beweis gestellt werden, also die Optimalität der Vorkonditionierungstechniken und die exakte Behandlung der gebrochen rationalen Sobolevnormen in unserem Kontrollproblem. Auf der Modellierungsseite untersuchen wir die Auswirkungen der Parameter im Kontrollfunktional auf die Qualität der berechneten Lösungen. Insbesondere sind die Effekte, welche sich durch die veränderbaren Sobolevindizes im Kontrollfunktional ergeben, von großem Interesse. Auf der numerischen Seite untersuchen wir die Effektivität der numerischen Algorithmen in Form von Iterationsstudien der Lösungsalgorithmen und Konditionszahlen der vorkonditionierten Operatoren.

Section 6
Kapitel 6 enthält die abschließenden Bemerkungen über diese Arbeit und einen Ausblick auf mögliche Verbesserungen und zukünftige Entwicklungen der Thematik.

Appendix A

In diesem Kapitel gebe ich einen Überblick über die von mir geschriebene Software. Dabei kam die Programmiersprache C++ zum Einsatz. Eine Liste der wichtigsten Datenstrukturen und Routinen wird angegeben. Der interessierte Leser findet hier Kontaktinformationen und einen Link auf die Software selbst.

Appendix B

In diesem Abschnitt werden alle wichtigen mathematischen Symbole aus dieser Arbeit und einige wichtige Definitionen, welche nicht im Text stehen, aber benutzt werden aufgelistet.

Introduction

In this thesis, we present wavelet methods for the numerical solution of a control problem constrained by a linear elliptic partial differential equation (PDE) with Dirichlet boundary control. This problem setup is challenging, as not only the solution of an elliptic partial differential equation is sought in the most efficient way possible: we face in addition an optimization problem specified by a target functional.

Recently, **biorthogonal wavelets** have been utilized for problems in numerical analysis, especially for solving partial differential equations. Their solid theoretical background and analytic properties, specifically with respect to Sobolev spaces $H^s(\Omega)$, make them our central ingredient for the numerical treatment of elliptic PDE-constrained control problems.

Optimal Control Problems

Optimization is concerned with problems of the following nature: Find the element u of an admissible set that minimizes a cost functional. The admissible set can be given by an equality or inequality condition, often applied to an operator equation, e.g., a partial differential equation. We set our focus on control problems with linear elliptic PDEs and **Dirichlet boundary control**, which is the practically most important case. This means that the control u appears as a Dirichlet boundary condition in the constraints of the functional to be minimized. The control problem is given as follows:

For some given data y_{Γ_Y} and f, minimize

$$\mathcal{J}(y,u) = \frac{1}{2}\|y - y_{\Gamma_Y}\|^2_{Y,\Gamma_Y} + \frac{\omega}{2}\|u\|^2_{Q,\Gamma} \tag{1}$$

where the **state** y *and the* **control** u *are coupled through the linear elliptic boundary value problem*

$$\begin{aligned} -\nabla\cdot(\mathbf{a}\nabla y) + a_0\, y &= f && \text{in } \Omega, \\ y &= u && \text{on } \Gamma, \\ (\mathbf{a}\nabla y)\cdot\mathbf{n} &= 0 && \text{on } \partial\Omega\setminus\Gamma. \end{aligned} \tag{2}$$

The elliptic boundary value problem (2) is handled by means of a weak formulation. The boundary conditions are enforced weakly by a trace operator B and are bound to the elliptic operator A by the technique of a **Lagrangian multiplier**. This is a favorable approach for the treatment of the essential boundary conditions because it allows for optimal preconditioning. Attaching the boundary conditions in this manner results in a system of equations which is an incarnation of a **saddle point problem**:

$$L\begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ u \end{pmatrix}\ . \tag{3}$$

The saddle point problem L is then attached again to the functional (1) by means of **Lagrangian multipliers**,

$$\hat{\mathcal{J}}(y,p,u,z,\mu) := \mathcal{J}(y,u) + \left\langle \begin{pmatrix} z \\ \mu \end{pmatrix}, L\begin{pmatrix} y \\ p \end{pmatrix} - \begin{pmatrix} f \\ u \end{pmatrix} \right\rangle\ . \tag{4}$$

The necessary optimality conditions for our control problem are thus determined by the first order necessary optimality conditions for the functional (4). For this problem formulation, second order sufficient conditions need not be taken into account since the functional (1) is strongly convex. The resulting equations can be interpreted as a **coupled system of PDEs**.

Efficient Numerical Solution

Several obstacles have to be overcome for the efficient numerical solution of the above mentioned optimality conditions. Let us first specify what we regard as an "efficient" solution strategy. The input data in numerical algorithms is always only given up to **discretization error** accuracy. Even given data, e.g., boundary values, will be discretized upon entering them into a computer system. In this case, the discretization error will be equal to **machine precision**. Generally, we can only expect results of the same precision and accuracy as the input data. The discretization error h_J is usually linked to the level of resolution J in the discretized model and this defines the number of unknowns N_J in our equations. We

seek to compute the solution of our problem in **linear time and complexity** $\mathcal{O}(N_J)$, i.e., the resources and time required grow linearly in the number of unknowns. Since every element of the solution must be computed and thus accessed at least once, one cannot expect any better complexity. A solution method with these properties is said to be of **optimal complexity**.

A first obstacle arises during the modelling of the above control problem. The solution y of the elliptic PDE (2) is generally in a Sobolev space of integral order, e.g., $y \in H^m$ with $m \in \mathbb{N}$. The **trace** $y|_{\partial\Omega}$ is then in $H^{m-1/2}$ which is a Sobolev space of **non-integral** or **fractional** order. In functional (1), the **natural trace norms** are therefore given by the norms of the spaces $Y = Q = H^{m-1/2}$. Sobolev spaces of fractional orders are not as easy to handle as those of integral order. A very popular approach to remedy this problem, especially in finite element theory, is to use a coarser norm, e.g., the L_2-norm, but the problem is then no longer well-defined. As this also deviates from the original problem formulation significantly, we will devise means to handle the natural trace norms. This will be done as accurate as possible and with optimal efficiency, i.e., the natural case will be as easy to handle in the wavelet setting as the L_2-case.

On the numerical side, we face several challenges. Firstly, the size of the anticipated discretized systems prohibits the use of direct solvers, except on very low levels. Instead we use iterative solvers, e.g., the CG-scheme for symmetric positive definite system matrices, and **Uzawa**-type algorithms for the saddle point problem (3). The convergence speed of these iterative solvers increases with the **spectral condition** number $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$ of the operator A. In conventional discretizations, the condition number depends inversely proportional on the discretization error h_J, e.g., $\kappa_2(A_J) \sim h_J^2$. Therefore, we use preconditioning techniques to achieve condition numbers independent of the resolution J. Preconditioners can generally be categorized in **additive** and **multiplicative** type. Examples for additive preconditioners are the **BPX** preconditioner for finite element methods [26,55] and the diagonal wavelet preconditioners. Multigrid methods are multiplicative preconditioners.

Preconditioning will be the essential ingredient for obtaining the solution of our optimal control problem efficiently. For solving the coupled system of PDEs resulting from considering the optimality conditions of (4), two distinct approaches will be given:

(I) A **fully iterative scheme** which was proposed in [46] and which is used for the efficient numerical solution of the resulting coupled linear equations of the state and control on uniformly discretized meshes. This method can be interpreted as an **inexact gradient scheme** with an outer iteration for the control u which depends on a given step size parameter ρ and two inner iterative methods on systems of type (3). In each step of the outer iteration the primal saddle point problem (which is a problem of the state y) and the adjoint saddle point problem are solved up to discretization error accuracy on the current level. Special consideration must be given to the outer iteration parameter ρ since the convergence speed directly depends on its value.

(II) Alternatively, the systems of equations can be reformed into one large (un)symmetric linear system of equations and the conjugate gradient solver can be applied to the corresponding normal equations. This is called the **All-In-One** approach.

The above mentioned claims of optimal preconditioned operators and their application in linear time might appear contradictorily. In order to fulfill all these and address the control problem as accurate as possible, we will utilize wavelet techniques.

Wavelet Discretization

Wavelet techniques offer a number of theoretical and numerical advantages over Finite-Element-Methods (**FEM**). In wavelet discretization, we can address all of the above mentioned modelling and numerical issues in one framework. Firstly, we can derive a natural representation of the functional (1) with proper treatment of the natural trace norms. Additionally, an optimally preconditioned discretization of the elliptic PDE (2), which is applicable in linear time, will be accessible.

We will make use of wavelets in the conceptual framework of **multiresolution analysis** (**MRA**). Here, wavelets form a **Riesz basis** for relevant function spaces such as Sobolev spaces of any (positive and negative) order up to a given bound. Additionally, **biorthogonality**, i.e., having **primal** and **dual** wavelet bases, $\mathbf{\Psi}$, $\widetilde{\mathbf{\Psi}}$, resp., with the property $\langle \mathbf{\Psi}, \widetilde{\mathbf{\Psi}} \rangle = \mathbf{1}$, entails **moment conditions** and **norm**

equivalences independently for the primal and dual spaces. More precisely, current primal wavelets are by construction **piecewise polynomials** with **compact support** which are needed to be given explicitly for the assembly of operators. Dual wavelet bases are chosen to satisfy **Jackson-** and **Bernstein-inequalities** up to a specific order for analytical reasons and are known only implicitly.

The wavelet discretization transforms operators between function spaces, e.g., $A : H^m \to (H^m)'$, into operators on the infinite-dimensional sequence space ℓ_2, e.g., $\mathbf{A} : \ell_2 \to \ell_2$. Using the above mentioned norm equivalences, the resulting linear continuous operators will be **uniformly invertible** ℓ_2-automorphisms with **uniformly bounded** condition numbers. Finite dimensional ℓ_2-operators are derived from their discretized infinite-dimensional counterparts by restricting the wavelet ansatz space to a finite linear subset. Here **stability**, i.e., maintaining an ℓ_2-automorphism at the transition to finite spaces, is ensured by **Galerkin stability** for the elliptic operator $\mathbf{A}$. The discretized saddle point operator $\mathbf{L}$ additionally requires the Ladysenškaya-Babuška-Brezzi (**LBB**)-condition for stability.

Numerical schemes, e.g., Wavelet Methods and FEMs, are always measured by their efficiency and ease of applicability. In our setting, efficiency will be determined by the **Riesz basis** constants and the condition numbers of the resulting matrices $\mathbf{A}_J$ and $\mathbf{L}_J$. We will devise basis transformations specifically to reduce the absolute values of these uniformly bounded condition numbers and, thus, also reduce the numerical costs for each calculation of the solution of (3). With these tools at our disposal, the solution to our control problem is accessible with **linear complexity**.

It should be mentioned that, additionally, wavelets offer a large potential for adaptive refinements. Discretizations on uniform grids quickly reach computational limitations. Also, the solution of a PDE often exhibits isolated singularities caused by the data or the geometry. Adaptive refinement strategies provide a promising potential of achieving a desired accuracy with a minimum of computational complexity required.

Scope of this Thesis

The goal of this document is the elaboration and presentation of the theoretical fundamentals and the numerical results of the above mentioned control problem constrained by a linear elliptic PDE with Dirichlet boundary control. On this account we give an introduction into the matter of such problems to a readership which should be familiar with functional analysis and numerics.

We will then concern ourselves with the construction of wavelets adapted to the interval and optimized with respect to preconditioning. We also employ new techniques to evaluate Sobolev norms of integral and fractional order with linear complexity. Thus, control problems involving fractional Sobolev norms of the control and the state can be given an appropriate representer which is as easy to set up as in the L_2 case.

The control problem will be reformulated in an ℓ_2-setting in terms of a wavelet discretization and the Euler equations of the functional corresponding to (4) will be considered. This discretization yields a representation of the control problem which involves only ℓ_2-norms and -operators. The solutions of the finite dimensional stable analogons of these equations will be computed by the above mentioned **All-In-One** solver or the **Inexact Gradient** scheme. Since the finite dimensional wavelet spaces are nested, these schemes can and will then be employed in a **nested iteration** strategy, i.e., obtaining the solution on a fine level by starting on the coarsest level the discretization permits and prolonging the computed solution to the next finer level once the discretization error accuracy for the solution on this level has been obtained.

Although both techniques are theoretically of optimal complexity in the number of unknowns, parameter choices, as for example the smoothness of the trace space, can have a tremendous impact on the convergence speed. This effect can be described as “the curse of the constants” because the overall cost of the process is still linear although the constants can be very high and the asymptotics may not be verifiable on small levels.
These effects will be investigated in the course of this work. Also part of these studies are iteration histories and plots of controls and states with fixed right hand side and fixed parameters except for the order of the involved trace space. We will also investigate how the parameter ω which weights the importance of the norms in the functional (1) affects the numerical efficiency and the solution itself. Penalizing the control by choosing $\omega > 1$ definitely leads to a different solution than loosening the coupling of the state

$y = y(u)$ and the control u when ω tends to zero. In the limit case $\omega = 0$, the problem statement becomes ill-defined.

These results are to verify the statements of the theoretical estimates put forward in [46] and [47].

Outline

This thesis is structured as follows.

Section 1
In Section 1 basic vocabulary and definitions from functional analysis necessary for a complete understanding of the subject matter are recollected.

Section 2
This section gives an introduction to wavelets and the theory of multiresolution analysis, our main tool in this work. The established wavelet theory is extended by remarks on improving its qualities in applications by enhanced preconditioning and Riesz operators for norm equivalences with precise constants. This part lays the groundwork for a complete theoretical understanding of the numerical schemes.

Section 3
The topic of saddle point problems is discussed in Section 3. First, saddle point problems in general abstract form will be introduced. Secondly, we show how elliptic boundary value problems in weak form can be interpreted as saddle point problems. We also demonstrate the actual construction of the operators on the domain used for our control problem. Lastly, we present the **conjugate gradient** and **Uzawa** algorithms which calculate the solution to these saddle point problems in optimal complexity if the involved operators are preconditioned optimally.

Section 4
This section gives an introduction to control problems and presents different equivalent representations of the control problem at hand. We derive the necessary conditions (which are here also sufficient) describing the optimal control as an unknown in a system of linear equations. The derivation of the ℓ_2-representer of our control problem is shown in terms of of wavelet representation given before. We present the **All-In-One** and **Inexact Gradient** algorithm which both have asymptotically optimal complexity for determining the control and the state.

Section 5
Here we present and discuss numerical results, e.g., iteration histories, convergence rates and solutions obtained. Special emphasis is laid on the efficiency of the wavelet preconditioning schemes in the iterative solvers and on accurate implementations of the fractional Sobolev space norms in our control problem. We will also discuss the intricacies of the modelling of our control problem and the numerical impact of modelling parameter choices. In particular, we investigate the effects of varying smoothness in the Sobolev scale onto the effectiveness of the numerical algorithms and the quality of the produced solution.

Section 6
Finally, this thesis closes with summarizing some insights and final remarks, together with an outlook on possible future continuations and enhancements of this work.

Appendix A
For the implementation of all routines and functions the object oriented programming language C++ was used. A short overview of the used data structures and classes is given together with contact information and information regarding access to the software.

Appendix B
Here we list the most important symbols used in this thesis and state some basic definitions.

1 Fundamentals

We need to recall some fundamental definitions and propositions from functional analysis which are needed in the discussion of elliptic control problems in the later chapters. This concerns especially the Sobolev spaces of fractional order and the trace operators in Section 1.2. The contents of the following sections are generally based on the books [3] and [41]. Other references are stated in place whenever a result is quoted.

1.1 Basic Definitions and Vocabulary

Let X, Y be normed linear spaces over the field $\mathbb{R}$.

Definition 1.1 [Linear Operators and Operator Norms]
We denote all **linear operators** *from X to Y by*

$$L(X;Y) := \{T : X \to Y;\ T \text{ is continuous and linear}\}\ . \tag{1.1.1}$$

For any $T \in L(X;Y)$, the associated **operator norm** *is defined by*

$$\|T\|_{L(X;Y)} := \sup_{x\in X,\ \|x\|_X=1} \|Tx\|_Y, \tag{1.1.2}$$

which is known to be finite for this class of operators (see [3]).

We write $L(X) := L(X;X)$ when X and Y coincide.

Definition 1.2 [Banach Spaces and Equivalent Norms]
A **Banach space** *is a complete vector space B with a norm $\|\cdot\|_B$. A Banach space can have several norms, e.g. $\|\cdot\|_{B_1}$ and $\|\cdot\|_{B_2}$, which are called* **equivalent** *if they induce the same topology. This is equivalent to the existence of positive finite constants c and C such that for all $v \in B$*

$$\|v\|_{B_1} \le c\|v\|_{B_2} \text{ and } \|v\|_{B_2} \le C\|v\|_{B_1}, \tag{1.1.3}$$

written shortly as

$$\|v\|_{B_1} \lesssim \|v\|_{B_2} \text{ and } \|v\|_{B_2} \gtrsim \|v\|_{B_1} \qquad \text{or} \qquad \|v\|_{B_1} \sim \|v\|_{B_2}\ . \tag{1.1.4}$$

Definition 1.3 [Separable Hilbert Space]
A **Hilbert space** *$\mathcal{H}$ is a complete vector space with an inner product $(\cdot,\cdot)_{\mathcal{H}}$ such that the norm is induced by the inner product as $\|\cdot\|_{\mathcal{H}} := \sqrt{(\cdot,\cdot)_{\mathcal{H}}}$. A Hilbert space is called* **separable** *if it contains a countable dense subset, i.e.,*

$$V = \{v_i : i = 1, 2, \ldots\} \subset \mathcal{H}, \qquad \text{such that } \operatorname*{clos}_{\mathcal{H}} V = \mathcal{H}\ . \tag{1.1.5}$$

A Hilbert space is always a Banach space, but the converse does not need to hold. Most spaces relevant for numerical studies are separable since (1.1.5) can equivalently be expressed as

$$\operatorname{dist}(f;V) = 0, \qquad \text{for all } f \in \mathcal{H}, \tag{1.1.6}$$

which, in other words, means that every element of $\mathcal{H}$ can be approximated analytically or numerically with arbitrary precision with elements from the space V. Examples of separable Hilbert spaces are the Lebesgue function spaces $L_p(\Omega)$, $1 \le p < \infty$, and the subspaces $H^s(\Omega)$ of $L_2(\Omega)$. Note that $L_\infty(\Omega)$ is not separable. Any generic Hilbert space considered in this thesis will be separable.

Definition 1.4 [Dual Space]
Let X be a Banach space. The **dual space** *X' of X is the space of all linear continuous functions from X onto the underlying field $\mathbb{R}$. In other words,*

$$X' := L(X;\mathbb{R})\ . \tag{1.1.7}$$

The elements $v' \in X'$ are called **linear functionals**. *The* **dual form** *is defined as $\langle x, x'\rangle_{X\times X'} := x'(x)$.*

1.2 Sobolev Spaces

Most of this section is taken from the books of [50], [1] and [41]. Some details, especially the Fourier analysis notation, are borrowed from the lecture of [57].
In the following, let $\Omega \subset \mathbb{R}^n$ be a bounded domain with piecewise smooth boundary $\partial\Omega$ and Ω being locally on one side. The meaning and degree of smoothness of $\partial\Omega$ will be specified in Section 1.2.2, for now $\partial\Omega$ just shall be considered "sufficiently smooth".
The space $L_2(\Omega)$ is the space of all real-valued square **Lebesgue integrable** functions on Ω. It is equipped with the inner product

$$(u, v)_{L_2(\Omega)} := \int_\Omega u(x)v(x)\, d\mu, \tag{1.2.1}$$

where $\mu = \mu(x)$ is the Lebesgue measure. Functions $u, v \in L_2(\Omega)$ are considered **equal** if $u(x) = v(x)$ holds almost everywhere, i.e., for all $x \in \Omega \setminus A$ and $\mu(A) = 0$. $L_2(\Omega)$ is a **Hilbert space**.

Remark 1.5 *In the following, $\alpha := (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$ is a* **multi-index***. Its definition, along with that of the classical smoothness spaces C^k, $C_0^\infty(\Omega)$,* **Hölder** *spaces $C^{k,\alpha}$ and* **Lipschitz** *spaces $C^{k,1}$ can be found in Appendix B.*

Definition 1.6 [Weak Derivative]
We say $u \in L_2(\Omega)$ has the **weak derivative** *$v = \partial^\alpha u$, if $v \in L_2(\Omega)$ and*

$$(\phi, \partial^\alpha v)_{L_2} = (-1)^{|\alpha|}(\partial^\alpha \phi, u)_{L_2}, \qquad \text{for all } \phi \in C_0^\infty(\Omega)\ . \tag{1.2.2}$$

Remark 1.7 *For convenience purposes, we will omit the domain from the scalar product like we did in (1.2.2), if this does not create confusion.*

If such a v exists, it is unique (in the L_2-sense). In case $u \in C^m(\Omega)$, the weak derivative corresponds to the classical strong derivative and (1.2.2) follows as an application of **Green's formula**.
We now introduce Sobolev spaces as subspaces of L_2, in which elements possess weak derivatives of specific orders.

Definition 1.8 [Sobolev Space on Ω]
For $m \in \mathbb{N}$ we denote by $H^m(\Omega)$ the **Hilbert space** *of all functions $u \in L_2(\Omega)$ for which the weak derivatives $\partial^\alpha u$ for all $|\alpha| \le m$ exist. The inner product of this space is given as*

$$(u, v)_{H^m} := \sum_{|\alpha| \le m} (\partial^\alpha u, \partial^\alpha v)_{L_2} \tag{1.2.3}$$

which is associated to the norm

$$\|u\|_{H^m} := \sqrt{(u, u)_{H^m}} = \sqrt{\sum_{|\alpha| \le m} \|\partial^\alpha u\|_{L_2}^2}\ . \tag{1.2.4}$$

A seminorm is given by

$$|u|_{H^m} := \sqrt{\sum_{|\alpha| = m} \|\partial^\alpha u\|_{L_2}^2}\ . \tag{1.2.5}$$

The Sobolev spaces are obviously nested, i.e., $H^{m+1} \subset H^m$, with the usual definition $H^0 := L_2$. A well known fact is the following

Corollary 1.9 *$C^\infty(\Omega) \cap H^m(\Omega)$ is dense in $H^m(\Omega)$ for $m \in \mathbb{N}_0$.*

The series of spaces $H^m, m \in \mathbb{N}_0$, can be extended to a scale of spaces with continuous smoothness indices, which will be of great importance later. These subspaces of H^m are Sobolev spaces of non-integral order $s \notin \mathbb{N}$ and cannot be characterized by weak derivatives alone as above. Instead, we use the following definition:

Definition 1.10 [Fractional Sobolev Spaces on Ω]
For $s = m + \sigma, m \in \mathbb{N}_0, 0 < \sigma < 1$, we introduce an inner product as

$$(u,v)_{H^s} := (u,v)_{H^m} + \sum_{|\alpha|\le m} \left(\int_\Omega \int_\Omega \frac{|\partial^\alpha u(x) - \partial^\alpha u(y)||\partial^\alpha v(x) - \partial^\alpha v(y)|}{|x-y|^{n+2\sigma}} \, d\mu(x)\, d\mu(y) \right) . \tag{1.2.6}$$

The space $H^s(\Omega)$ is the closure of all functions in $H^m(\Omega)$ for which the norm

$$\|u\|_{H^s} := \sqrt{(u,u)_{H^s}} \tag{1.2.7}$$

is finite. It is thus a Hilbert space.

Remark 1.11 *The Definitions 1.8 and 1.10 also hold in case $\Omega = \mathbb{R}^n$.*

The Sobolev spaces are nested in the following fashion

$$H^{s_1} \subset H^{s_2} \subset L_2, \qquad s_1 > s_2 > 0, \tag{1.2.8}$$

for any domain $\Omega \subseteq \mathbb{R}^n$.

An Alternative Characterization

An alternative approach to define fractional Sobolev spaces is given by means of Fourier Analysis. We can define for $f \in L_2(\mathbb{R}^n)$ the **Fourier transform** $\mathcal{F}(f) \in L_2(\mathbb{R}^n)$ as the limit in the L_2 sense of

$$\int_{|\xi|\le M} \exp(\pm 2\pi i x \cdot \xi) f(\xi) d\xi, \qquad M \to \infty . \tag{1.2.9}$$

The Fourier transform is an **isomorphism** between $L_2(\mathbb{R}^n)$ and itself, with

$$\|\mathcal{F}(f)\|_{L_2} = \|f\|_{L_2},$$

and the identity

$$\mathcal{F}(\partial^\alpha f) = (2\pi i)^{|\alpha|} \xi^\alpha \mathcal{F}(f), \qquad \text{for all } f \in L_2(\mathbb{R}^n)$$

holds. From the above-mentioned remarks we obtain an equivalent characterization of H^m as:

$$H^m(\mathbb{R}^n) = \{ v \,|\, \xi^\alpha \mathcal{F}(v) \in L_2(\mathbb{R}^n) \text{ for all } |\alpha| \le m \} . \tag{1.2.10}$$

Obviously, it does not matter in (1.2.10) whether m is an integer or whether it is positive. It may be easily verified that the characterizations $\xi^\alpha \mathcal{F}(v) \in L_2(\mathbb{R}^n)$ and $(1+|\xi|^2)^{s/2}\mathcal{F}(v) \in L_2(\mathbb{R}^n)$ for $|\alpha| \le s$ are equivalent. The latter is predominantly used in the following alternative version of Definition 1.10 for $\Omega = \mathbb{R}^n$.

Definition 1.12 [Fractional Sobolev Space on $\mathbb{R}^n$]
For $s \in \mathbb{R}$ we define the **Sobolev space of order s**, *$H^s(\mathbb{R}^n)$, as*

$$H^s(\mathbb{R}^n) := \{ v \,|\, (1+|\xi|^2)^{s/2} \mathcal{F}(v) \in L_2(\mathbb{R}^n) \}, \tag{1.2.11}$$

which is a **Hilbert space** *when endowed with the inner product:*

$$(u,v)_{H^s} := \Big((1+|\xi|^2)^{s/2}\mathcal{F}(u), (1+|\xi|^2)^{s/2}\mathcal{F}(v) \Big)_{L_2} . \tag{1.2.12}$$

Remark 1.13 *In case $s \in \mathbb{N}$, the inner products (1.2.3) and (1.2.12) induce equivalent norms, which, however, are not identical.*

For $u \in H^s(\mathbb{R}^n)$, we can define the **restriction** operator onto $\Omega \subseteq \mathbb{R}^n$,

$$u \to u|_\Omega =: \text{ restriction of } u \text{ to } \Omega, \tag{1.2.13}$$

which is continuous and linear. In case $u \in C^k$ holds, the restriction can be defined pointwise.
The space $H^s(\Omega)$ from Definition 1.8 can now be equivalently expressed using the above Definition 1.12 and the following theorem from [50].

Theorem 1.14 *$H^s(\Omega)$ coincides with the space of* **restrictions** *to Ω of the elements of $H^s(\mathbb{R}^n)$.*

Therefore, for any function $u \in H^s(\Omega)$, an element $\tilde{u} \in H^s(\mathbb{R}^n)$ can be specified which defines u by means of local coordinates on the domain Ω. The approach via Fourier transform is in particular applicable for Sobolev spaces to be defined on periodic domains.

1.2.1 Subspaces $H_0^s \subset H^s$

The spaces $H_0^s(\Omega)$ are normally loosely referred to as elements of the spaces $H^s(\Omega)$ with compact support in Ω. The definition of the spaces $H_0^s(\Omega)$ is an extension of Corollary 1.9.

Definition 1.15 [Sobolev Spaces $H_0^s(\Omega)$]
$H_0^s(\Omega)$ is defined as the closure of $\mathcal{D}(\Omega) := C_0^\infty(\Omega)$ with respect to the norm of $H^s(\Omega)$, i.e.,

$$H_0^s(\Omega) := \{\phi \,|\, \exists\, \{\phi_n\} \in \mathcal{D}(\Omega) \text{ and } \phi_n \to \phi \text{ is a Cauchy sequence in } \|\cdot\|_{H^s(\Omega)}\} \ . \tag{1.2.14}$$

Hence the spaces $H_0^s(\mathbb{R}^n)$ and $H^s(\mathbb{R}^n)$ are equal. In general, the spaces $H_0^s(\Omega)$ are closed subspaces of $H^s(\Omega)$. Specifically, we have

$$H_0^s(\Omega) = H^s(\Omega), \qquad 0 \le s \le \frac{1}{2}, \tag{1.2.15}$$

which holds because $\mathcal{D}(\Omega)$ is also dense in $H^s(\Omega)$ for $s \le \frac{1}{2}$, cf. Corollary 1.9. In the other cases we have

$$H_0^s(\Omega) \subsetneq H^s(\Omega), \qquad s > \frac{1}{2}, \tag{1.2.16}$$

which means that $H_0^s(\Omega)$ is strictly contained in $H^s(\Omega)$. It is shown in [41] and [39] that one can also characterize the spaces of (1.2.14) as the following family of functions:

$$H_0^s(\Omega) = \{u \,|\, u \in H^s(\Omega), \partial^\alpha u = 0 \text{ on } \partial\Omega, |\alpha| \le s - \frac{1}{2}\} \ . \tag{1.2.17}$$

These spaces also have an important property regarding their dual spaces which will be seen in Section 1.2.3. We now have the following relations between the Sobolev spaces of integral orders:

$$\begin{array}{ccccccccc} L_2(\Omega) & = & H^0(\Omega) & \supset & H^1(\Omega) & \supset & H^2(\Omega) & \supset \ldots \\ & & \shortparallel & & \cup & & \cup & & \\ & & H_0^0(\Omega) & \supset & H_0^1(\Omega) & \supset & H_0^2(\Omega) & \supset \ldots \end{array} \tag{1.2.18}$$

All inclusions in the above diagram are dense and the embeddings continuous.

1.2.2 Trace Spaces $H^s(\Gamma)$

Trace spaces and trace operators appear naturally in the treatment of the boundary value problem considered in Section 3. To this end, we need to recall a definition of trace spaces and the extension of classical trace operators onto the space $H^s(\Omega)$.
The **constraint** or **trace** $u|_{\partial\Omega}$ of a function $u \in H^s(\Omega)$ cannot simply be defined pointwise because there is no guarantee that functions in L_2 and H^s can be evaluated at specific points $x \in \Omega$. It also makes no sense to define the trace as the continuous limit when approaching the boundary, because firstly, elements of $H^1(\Omega)$ are generally not continuous, and secondly, $\partial\Omega$ is a manifold in $\mathbb{R}^{n-1}$ and, thus, its measure in $\mathbb{R}^n$ is zero. This means we could have $u = v$ a.e. for $u, v \in L_2(\Omega)$ but $u(x) \neq v(x)$ for all $x \in \partial\Omega$.
The trace of functions in Sobolev spaces is defined through a trace operator and is given in local coordinates on an open cover of the boundary $\partial\Omega$. This definition depends also on regularity conditions of the boundary $\partial\Omega$, which we will now formalize.
Let $\Omega \subset \mathbb{R}^n$ be a domain with Lipschitz boundary $\partial\Omega \in C^{k,1}$ to which Ω lies locally on one side. Also, a fixed section $\Gamma \subseteq \partial\Omega$ should have a positive surface measure. The following local coordinate system is thus well defined :
For any $x \in \partial\Omega$, we can specify a neighborhood $V \subset \mathbb{R}^n$ with new orthogonal coordinates $z = (z', z_n)$ where $z' = (z_1, \ldots, z_{n-1})$. Without imposing restrictions, V can be characterized as a cube in these coordinates, i.e.,

$$V = \{(z_1, \ldots, z_n) \,|\, |z_j| \le 1, 1 \le j \le n\},$$

and the first $n-1$ coordinates z' of z span the space

$$V' := \{(z_1, \ldots, z_{n-1}) \,|\, |z_i| \le 1, 1 \le j \le n-1\} \ .$$

Let $\Theta = \{\Theta_j \,|\, j = 1, \ldots, r\}$ be a family of open bounded sets in $\mathbb{R}^n$, covering $\partial\Omega$, such that, for each j, there exists $\varphi_j \in C^{k,1}(V', \Theta_j)$ with positive Jacobian $J(\varphi_j), 1 \le j \le r$ and φ_j is a bijection. Furthermore, we can arrange to have

$$\begin{aligned} &\varphi_j \in C^{k,1}(V_+, \Theta_j \cap \Omega), && V_+ := \{(z', z_n) \in V \,|\, z_n < \varphi_j(z')\}, \\ &\varphi_j \in C^{k,1}(V_0, \Theta_j \cap \partial\Omega), && V_0 := \{(z', z_n) \in V \,|\, z_n = \varphi_j(z')\}, \end{aligned}$$

because of the preliminary requirements to Ω above. In other words, Ω lies locally below the graph of each φ_j and the graph of φ_j is the boundary of Ω in the patch Θ_j. For each j, the pair (φ_j, Θ_j) is called a **coordinate patch** for the boundary part $\partial\Omega \cap \Theta_j$.

Definition 1.16 [Sobolev Spaces $H^s(\partial\Omega)$]
A distribution u on $\partial\Omega$ is in $H^s(\partial\Omega)$ for any real $|s| \le k+1$, if and only if

$$u \circ \Phi_j \in H^s(V' \cap \Phi_j^{-1}(\Theta_j \cap \partial\Omega)) \ . \tag{1.2.19}$$

This is a Banach space when equipped with the norm

$$\|u\|^2_{H^s(\partial\Omega)} := \sum_{j=1}^{r} \|y_j \circ \Phi_j\|^2_{H^s(V' \cap \Phi_j^{-1}(\Theta_j \cap \partial\Omega))} \ . \tag{1.2.20}$$

Remark 1.17 *It was proved in [39] that the above definition is independent of the choice of the system of local maps $\{\varphi_j, \Theta_j\}$.*

The trace space $H^s(\Gamma) \subset H^s(\partial\Omega)$ can be defined analogously by only considering an open cover Θ of Γ, which does not intersect with $\partial\Omega \setminus \Gamma$ (except for parts with zero surface measure), then applying the rest of the definition unchanged.

Remark 1.18 *One can exchange $\partial\Omega \in C^{k,1}$ in the above paragraph by $\partial\Omega \in C^0$ or C^k, the details remain valid if the maps $\{\varphi_j\}$ are adapted appropriately.*

Trace Operators

Trace operators which restrict functions $u \in H^s(\Omega)$ to the boundary can now be constructed as extensions of the classical trace operators of continuous functions,

$$u(x_1, \ldots, x_n)|_{x_n=0} := u(x_1, \ldots, x_{n-1}, 0) \ . \tag{1.2.21}$$

We summarize here the results of this topic; for details see the books of [41], [50] and [39].
We define for any function $u \in C^{k,1}(\bar{\Omega})$ its traces of normal derivatives by

$$\gamma_j(u) := \left.\frac{\partial^j u}{\partial \nu^j}\right|_\Gamma, \qquad 0 \le j \le k, \tag{1.2.22}$$

where $\nu = \nu(x)$ is the outward normal on the boundary of Ω which exists a.e. . We will be referring to γ_0 when talking about *the* trace operator.

Theorem 1.19 *Assume that $s - 1/2 = m + \sigma$, $0 < \sigma < 1, m \in \mathbb{N}_0$ and $s \le k+1$. Then the mapping*

$$u \mapsto \{\gamma_0 u, \gamma_1 u, \ldots, \gamma_m u\}, \tag{1.2.23}$$

which is defined for $u \in C^{k,1}(\bar{\Omega})$, has a unique continuous extension as an operator from

$$H^s(\Omega) \text{ onto } \prod_{j=0}^{m} H^{s-j-1/2}(\Gamma) \ . \tag{1.2.24}$$

We will make no distinction between the classical trace operators and the extensions to Sobolev spaces. In later chapters, we frequently need a classical **Trace Theorem** which holds for domains Ω with Lipschitz continuous boundary $\partial\Omega \in C^{0,1}$.

Theorem 1.20 *For any $u \in H^s(\Omega), 1/2 < s < 3/2$, one can estimate*

$$\|\gamma_0 u\|_{H^{s-1/2}(\Gamma)} \leq c_{T,\Omega} \|u\|_{H^s(\Omega)} \ . \tag{1.2.25}$$

Conversely, for every $h \in H^{s-1/2}(\Gamma)$, there exists some $u \in H^s(\Omega)$ such that $\gamma_0 u = h$ and

$$\|u\|_{H^s(\Omega)} \leq C_{T,\Omega} \|h\|_{H^{s-1/2}(\Gamma)}. \tag{1.2.26}$$

As before, $c_{T,\Omega}$ and $C_{T,\Omega}$ denote positive finite constants, but, as indicated by their subscript, their value usually depends on properties of the domain Ω.

The range of s extends accordingly if $\partial\Omega$ is more regular.

Extension Operators

We can also give estimates which can be seen as a converse counterpart to the above estimate (1.2.25). These are **Whitney-extension** results which state that any function $u \in H^s(\Omega)$ can be extended to a function $\tilde{u} \in H^s(\mathbb{R}^n)$ such that $\tilde{u}|_\Omega = u$ and

$$\|\tilde{u}\|_{H^s(\mathbb{R}^n)} \leq C_{E,\Omega} \|u\|_{H^s(\Omega)}, \qquad s > 0 \ . \tag{1.2.27}$$

This is also true for traces of functions: for any $h \in H^{s-1/2}(\Gamma)$, there exists an extension $\tilde{h} \in H^{s-1/2}(\partial\Omega)$ such that

$$\|\tilde{h}\|_{H^{s-1/2}(\partial\Omega)} \leq C_{E,\partial\Omega} \|h\|_{H^{s-1/2}(\Gamma)}, \qquad s > 0 \ . \tag{1.2.28}$$

Again, $C_{E,\Omega}$ and $C_{E,\partial\Omega}$ denote (domain dependent) positive finite constants.

1.2.3 Dual of Sobolev Spaces

Recall that the dual space of $H^s(\Omega)$ will generally be denoted by $(H^s(\Omega))'$. The dual space of L_2 is related to L_2 again by the **Riesz Representation Theorem**, i.e., $(L_2)' = L_2$, and the **dual form** is given as

$$\langle u, v \rangle_{L_2 \times (L_2)'} := \int u(x)\, v(x)\, d\mu, \qquad u, v \in L_2 \ . \tag{1.2.29}$$

Remark 1.21 *In the following, we will omit the space specifiers in the dual form and write only $\langle \cdot, \cdot \rangle$ if the exact dual form can be ascertained unambiguously from the arguments.*

Thus, we have trivially

$$(H^0(\Omega))' = (H_0^0(\Omega))' = (L_2(\Omega))' = L_2(\Omega) \qquad \text{for domains } \Omega \subseteq \mathbb{R}^n \ . \tag{1.2.30}$$

In case $\Omega = \mathbb{R}^n$ we can use Definition 1.12 with arbitrary negative indices to define Sobolev spaces of negative order on all $\mathbb{R}^n$. These spaces $H^{-s}(\mathbb{R}^n)$ are now the dual spaces of $H^s(\mathbb{R}^n)$ as the following result from [50] shows:

Theorem 1.22 *For all $s > 0$ one has*

$$(H^s(\mathbb{R}^n))' = H^{-s}(\mathbb{R}^n) \ . \tag{1.2.31}$$

This is not true when the domain under consideration is bounded. However, we can identify some of these dual spaces with Sobolev spaces of negative order which we define as follows:

Definition 1.23 [Sobolev Spaces $H^{-s}(\Omega)$]
For $\Omega \subset \mathbb{R}^n$ and $s \in \mathbb{R}_+$ we define a norm for $u \in L_2(\Omega)$ by

$$\|u\|_{H^{-s}(\Omega)} := \sup_{v \in H_0^s(\Omega)} \frac{\langle u, v \rangle_{(L_2)' \times L_2}}{\|v\|_{H_0^s(\Omega)}}, \qquad s > 0 \ . \tag{1.2.32}$$

The closure of $L_2(\Omega)$ with respect to this norm is termed $H^{-s}(\Omega) = (H_0^s(\Omega))'$.

The resulting spaces are obviously bigger than $L_2(\Omega)$ and also nested, and we get the following line of inclusions:

$$\ldots \supset H^{-2}(\Omega) \supset H^{-1}(\Omega) \supset L_2(\Omega) \supset H_0^1(\Omega) \supset H_0^2(\Omega) \supset \ldots \ . \tag{1.2.33}$$

1.2.4 Regularity Properties

The following theorem provides information about the relation of the Sobolev spaces $H^{m+1}(\Omega)$ and $H^m(\Omega)$ for domains Ω with Lipschitz boundary $\partial\Omega \in C^{0,1}$:

Theorem 1.24 *If $\Omega \subset \mathbb{R}^n$ has a Lipschitz boundary, then the embedding $H^{m+1}(\Omega) \hookrightarrow H^m(\Omega)$, $m \in \mathbb{N}_0$, is compact.*

An even more general result for the spaces $H_0^s(\Omega)$ is given by

Theorem 1.25 *The embedding $H_0^s(\Omega) \hookrightarrow H_0^t(\Omega)$, for all $s, t \in \mathbb{R}$ with $s > t$, is continuous and compact.*

Of special interest is obviously the relation of the Sobolev spaces H^s to the classical function spaces C^k. The **Embedding Theorem** by Sobolev establishes this connection.

Theorem 1.26 *If $\Omega \subset \mathbb{R}^n$ has a Lipschitz boundary, then the embedding $H^s(\Omega) \hookrightarrow C^k(\bar{\Omega})$ is continuous for $k \in \mathbb{N}_0$ and $s > k + n/2$.*

1.3 Elliptic Partial Differential Equations

At this point, we need to introduce some basic vocabulary from stationary partial differential equations. The weak formulation of an elliptic PDE will be at the heart of our class of control problems considered in Section 4.
In the following $\Omega \subset \mathbb{R}^n$ will always be a bounded domain with a Lipschitzian boundary $\partial\Omega$.

Definition 1.27 [Partial Differential Equation (PDE)]
Let $a_{\alpha,\beta} \in L_\infty(\Omega)$ be bounded coefficient functions satisfying $a_{\alpha,\beta} = a_{\beta,\alpha}$ for all multi-indices α, β with $|\alpha|, |\beta| \leq m$. A **partial differential equation** *of order 2m,*

$$\mathcal{L}y = f \qquad \text{in } \Omega, \tag{1.3.1}$$

is determined by a linear differential operator of order 2m, i.e.,

$$\mathcal{L} := \sum_{|\alpha|,|\beta|\leq m} (-1)^{|\beta|} \partial^\beta (a_{\alpha,\beta}(x)\partial^\alpha) \ . \tag{1.3.2}$$

We associate to (1.3.2) the polynomial in $\xi = (\xi_1, \ldots, \xi_n)$ given by

$$P(\xi, x) := \sum_{|\alpha|=|\beta|=m} a_{\alpha,\beta}(x)\xi^{\alpha+\beta}, \qquad \xi^\alpha = \prod_{i=1}^n \xi_i^{\alpha_i} \ . \tag{1.3.3}$$

Definition 1.28 [Elliptic Operator]
The operator $\mathcal{L}$ is said to be **elliptic** *if (1.3.3) satisfies*

$$P(\xi, x) \gtrsim \|\xi\|_2^{2m}, \qquad \text{for all } \xi \in \mathbb{R}^n,\ x \in \Omega \ . \tag{1.3.4}$$

Example 1.29 *The* **Laplacian operator**

$$\Delta := \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2} \tag{1.3.5}$$

of order 2 is elliptic. The operator

$$\mathcal{L} = -\Delta + a_0 \, I, \qquad a_0 \in \mathbb{R}_+, \tag{1.3.6}$$

also satisfies these properties.

It is easily seen that the equation $\mathcal{L}y = f$ does not need to have a unique solution. To ensure uniqueness, we have to impose further constraints on the solution space. This is typically done by requiring the solution to attain special boundary values. However, the existence or uniqueness of such boundary values cannot be determined unless we specify which kind of smoothness we require of our solution.

Definition 1.30 [Classical Solution of an Elliptic Boundary Value Problem]
A function $y \in C^{2m}(\Omega) \cap C(\bar{\Omega})$ which solves the **elliptic boundary value problem**

$$\begin{array}{rcll} \mathcal{L}y & = & f & \text{in } \Omega, \\ \dfrac{\partial^i y}{\partial \nu^i} & = & u_i & \text{on } \partial\Omega, \qquad i = 0, \ldots, m-1, \end{array} \tag{1.3.7}$$

pointwise for given data f and u_i is called a **classical solution**.

The above boundary conditions are generally classified into two types. Especially for the important case of $m = 1$, i.e., $\mathcal{L}$ is an operator of order 2, for example the **Laplace operator** (1.3.5), one defines

Definition 1.31 *Constraints of the form*

$$y = u \qquad \text{on } \partial\Omega \tag{1.3.8}$$

are called **Dirichlet boundary conditions**.

and

Definition 1.32 Neumann boundary conditions *are of type*

$$\frac{\partial y}{\partial \nu} = u \qquad \text{on } \partial\Omega \ . \tag{1.3.9}$$

If $u = 0$, the boundary conditions are called **homogeneous**, otherwise **inhomogeneous**.

Example 1.33 *The PDE*

$$\begin{aligned} -\Delta y + a_0 y &= f, && \text{in } \Omega, \\ y &= 0, && \text{on } \partial\Omega, \end{aligned}$$

is called **Helmholtz problem**. *In case* $a_0 = 0$ *it is called* **Poisson problem**.

The existence of a classical solution cannot be proved for arbitrary right hand sides $f \in C^0(\Omega)$ and $u_i \in C^{2i}(\partial\Omega)$. Therefore, we must extend the solution space of (1.3.7) to spaces which permit solutions and ensure uniqueness.

1.3.1 Variational Problems

In this section, we recall some facts about bilinear forms and operators in the abstract setting of a general Hilbert space $\mathcal{H}$ and its dual $\mathcal{H}'$. In the next section, we recall how a solution approach of the elliptic partial differential equation (1.3.1) can be embedded into this setting, thus benefitting from the results collected here.

Definition 1.34 [Continuous and Elliptic Bilinear Forms]
A symmetric **bilinear form** $a : \mathcal{H} \times \mathcal{H} \to I\!R$ *is called* **continuous**, *if and only if*

$$a(y, v) \lesssim \|y\|_{\mathcal{H}} \|v\|_{\mathcal{H}} \qquad \text{for all } y, v \in \mathcal{H} \ . \tag{1.3.10}$$

A continuous bilinear form is called $\mathcal{H}-$**elliptic** *(or* **coercive***), if and only if*

$$a(y, y) \gtrsim \|y\|^2_{\mathcal{H}} \qquad \text{for all } y \in \mathcal{H} \ . \tag{1.3.11}$$

Obviously, such a bilinear form is equivalent to the norm of the Hilbert space, i.e.,

$$\sqrt{a(y, y)} \sim \|y\|_{\mathcal{H}} \ . \tag{1.3.12}$$

Let the operator $A : \mathcal{H} \to \mathcal{H}'$ be defined by

$$\langle y, Av \rangle := a(y, v) \ . \tag{1.3.13}$$

By the **Theorem of Lax-Milgram** (see [3]) it is known that A is an isomorphism inducing the norm equivalence

$$\|Ay\|_{\mathcal{H}'} \sim \|y\|_{\mathcal{H}}, \qquad y \in \mathcal{H} \ . \tag{1.3.14}$$

A **variational problem** can now be phrased like this: Given $f \in \mathcal{H}'$, find $y \in \mathcal{H}$ such that

$$a(y, v) = \langle f, v \rangle, \qquad \text{for all } v \in \mathcal{H}, \tag{1.3.15}$$

or, equivalently, in operator notation

$$Ay = f \ . \tag{1.3.16}$$

The Theorem of Lax-Milgram now ascertains a unique solution $y = A^{-1} f$ to problem (1.3.15) which depends continuously on the right hand side f.

1.3.2 Weak Formulation of Second Order Dirichlet Problems

Let us return to the elliptic partial differential equation (1.3.7) with inhomogeneous Dirichlet boundary conditions. The type of problem we will encounter in Section 3 is given as

$$\begin{array}{rcll} \mathcal{L}y & = & f & \text{in } \Omega, \\ y & = & u & \text{on } \Gamma \subset \partial\Omega, \\ \dfrac{\partial^i y}{\partial \nu^i} & = & 0 & \text{on } \partial\Omega \setminus \Gamma, \qquad i = 1, \ldots, m-1 \ . \end{array} \tag{1.3.17}$$

We now consider the special case $\mathcal{H} = H^m(\Omega)$ with given data $f \in (H^m(\Omega))'$ and $u \in H^{m-1/2}(\Gamma)$. If a classical solution $y \in C^{2m}(\Omega) \cap C(\bar{\Omega})$ exists, multiplication of (1.3.17) with a test function $\phi \in C^\infty(\Omega)$ yields

$$\langle \mathcal{L}y, \phi \rangle = \langle f, \phi \rangle \qquad \text{for all } \phi \in C^\infty(\Omega) \ . \tag{1.3.18}$$

The left hand side is used to define the symmetric bilinear form $a(\cdot,\cdot)$ of Section 1.3.1 after using Green's formula,

$$\begin{aligned} a(y,\phi) &:= \langle \mathcal{L}y, \phi \rangle \\ &= \sum_{|\alpha|,|\beta| \le m} (-1)^{|\alpha|} \int_\Omega \phi(x)\, \partial^\alpha (a_{\alpha,\beta}(x)\, \partial^\beta y(x))\, d\mu(x) \\ &= \sum_{|\alpha|,|\beta| \le m} \int_\Omega a_{\alpha,\beta}(x)\, (\partial^\alpha y)(x)\, (\partial^\beta \phi)(x)\, d\mu(x) \ . \end{aligned} \tag{1.3.19}$$

After applying partial integration $m-$times, the boundary integral terms over $\partial\Omega \setminus \Gamma$ vanish here because of the homogeneous **Neumann boundary conditions** in (1.3.17). Since these boundary conditions are therefore naturally built into the weak formulation, they are called **natural boundary conditions**. The **Dirichlet boundary conditions** must be handled differently and are called **essential boundary conditions**. The integral terms over Γ vanish by restricting our test function space to functions with $\phi|_\Gamma = 0$.
The explicit treatment of the inhomogeneous Dirichlet boundary conditions will be the objective of Section 3. In short, these will be enforced using the **trace operator** γ_0 of Section 1.2.2 and **Lagrangian multipliers**.
The form $a(\cdot,\cdot)$ defined in (1.3.19) is obviously bilinear and symmetric. We can extend the bilinear form $a(\cdot,\cdot)$ onto $H^m(\Omega) \times H^m(\Omega)$ by Corollary 1.9.
The right hand side of (1.3.18) defines a linear functional on the space $H^m(\Omega)$:

$$\langle f, v \rangle = \int_\Omega f(x) v(x)\, d\mu(x) \ . \tag{1.3.20}$$

We will for an instant focus the discussion on the invertibility of the bilinear form (1.3.19) alone, for the special case of homogeneous boundary conditions $u = 0$. In view of Section 1.3.1, the task is then to find the element $y \in H^m(\Omega)$ with $y|_\Gamma = 0$ such that

$$a(y, v) = \langle f, v \rangle \qquad \text{for all } v \in H^m \ . \tag{1.3.21}$$

Definition 1.35 [Weak Solution]
Any function $y \in H^m(\Omega)$ for which (1.3.21) holds will be called **weak solution** *since it is not necessarily in the space $C^{2m}(\Omega) \cap C^0(\bar{\Omega})$.*

Note that the ellipticity of the operator $\mathcal{L}$ does not guarantee the ellipticity of the bilinear form $a(\cdot,\cdot)$. This can only be ensured by requiring additional conditions of the domain Ω and the coefficients $a_{\alpha,\beta}$. One sufficient criterion for ellipticity (1.3.11) in case $m = 1$ is a bounded domain Ω and

$$a_{\alpha,\beta} = 0, \qquad \text{if } |\alpha| + |\beta| \le 1 \ . \tag{1.3.22}$$

In case $m > 1$, additional prerequisites to the coefficient functions $a_{\alpha,\beta}$ are required, see [41].

Remark 1.36 *The Laplacian operator (1.3.5) is of the form (1.3.22) and the induced bilinear form is thus elliptic. This also holds for the operator (1.3.6).*

The other prerequisite of the bilinear form $a(\cdot,\cdot)$ required for the **Theorem of Lax-Milgram** is continuity, which follows directly from the following estimate taken from [41]:

$$|a(y,v)| \leq \sum_{\alpha,\beta} \|a_{\alpha,\beta}\|_{L_\infty(\Omega)} \|y\|_{H^m(\Omega)} \|v\|_{H^m(\Omega)} \ . \tag{1.3.23}$$

Thus, the homogeneous problem (1.3.17) has a unique solution in $H^m(\Omega)$.

Example 1.37 *The bilinear form for the Helmholtz Problem (Example 1.33) can be written as*

$$a(y,v) = (\nabla y, \nabla v)_{L_2} + a_0 \, (y,v)_{L_2} \ . \tag{1.3.24}$$

1.3.3 Galerkin Method

With the elliptic PDE (1.3.17) now stated as a variational problem in weak formulation, we can go on to solve (1.3.21) numerically. Thus we discretize (1.3.21) using elements from a finite dimensional closed subspace $S_j \subset \mathcal{H}$ of level j as trial functions. Let Φ_j be a basis for this space $S_j := S(\Phi_j) := \operatorname{span}\{\phi_{j,k} \,|\, k \in \Delta_j\}$ for some ordered finite index set Δ_j. The task is now to find the unique element $y_j \in S_j$ with

$$a(y_j, v_j) = \langle f, v_j \rangle \qquad \text{for all } v_j \in S_j \ . \tag{1.3.25}$$

The coefficients in the expansion $y_j = \sum_{k\in\Delta_j} y_{j,k}\phi_{j,k}$ are compiled in the vector $\mathbf{y}_j := \big((y_{j,k})_{k\in\Delta_j}\big)^T$. Choosing $v_j = \phi_{j,k}$ subsequentially for all $k \in \Delta_j$, we obtain the linear system of equations

$$\sum_{k\in\Delta_j} y_{j,k} \, a(\phi_{j,i}, \phi_{j,k}) = \langle f, \phi_{j,i} \rangle, \qquad i \in \Delta_j \ . \tag{1.3.26}$$

Abbreviating $\mathbf{A}_j := (a(\phi_{j,i}, \phi_{j,k}))_{i,k\in\Delta_j}$ and $\mathbf{f}_j := \langle f, \phi_{j,i} \rangle_{i\in\Delta_j}$ this reads shortly as

$$\mathbf{A}_j \, \mathbf{y}_j = \mathbf{f}_j \ . \tag{1.3.27}$$

If follows from the symmetry and ellipticity of $a(\cdot,\cdot)$ that $\mathbf{A}_j$ is symmetric positive definite, i.e., $x^T \, \mathbf{A}_j \, x > 0$ for all $0 \neq x \in \mathbb{R}^{\#\Delta_j}$. In particular, this means that $\mathbf{A}_j$ is non-singular and a unique solution $\mathbf{y}_j = \mathbf{A}_j^{-1} \, \mathbf{f}_j$ to (1.3.27) exists. Note that the vector $\mathbf{y}_j$ determines the unique element $y_j = \sum_{k\in\Delta_j} y_{j,k} \, \phi_{j,k} \in S_j$ which is the solution to (1.3.25).

Definition 1.38 [Discretization Error]
We call the level dependent constant

$$h_j := \operatorname{dist}(y, S_j) := \inf_{v_j \in S_j} \|y - v_j\|_{\mathcal{H}}, \qquad y \in \mathcal{H}, \tag{1.3.28}$$

the **discretization error** *of level j (with respect to S_j).*

Of special importance is the following result:

Lemma 1.39 [Céa – Lemma]
Let y resp. y_j denote the solution of the variational problem (1.3.21) resp. (1.3.25) in $\mathcal{H}$ resp. $S_j \subset \mathcal{H}$. Then it follows that

$$\|y - y_j\|_{\mathcal{H}} \lesssim \inf_{v_j \in S_j} \|y - v_j\|_{\mathcal{H}} \ . \tag{1.3.29}$$

In other words, the Galerkin solution y_j is (up to a constant) of the same error to the weak solution y as the best approximation of the trial space S_j. An immediate consequence is the following

Corollary 1.40 *Let the subspaces S_j be nested and their union dense in $\mathcal{H}$, i.e.,*

$$S_j \subset S_{j+1}, \qquad \operatorname*{clos}_{\mathcal{H}} \bigcup_j S_j = \mathcal{H}, \tag{1.3.30}$$

then the Galerkin scheme converges, i.e.,

$$\lim_{j\to\infty} \|y - y_j\|_{\mathcal{H}} = 0 \ . \tag{1.3.31}$$

Typical choices for the trial spaces S_j are **finite element spaces**.
After discretization, the central issue is to solve (1.3.27) numerically. The amount of computational work necessary to obtain $\mathbf{y}_j$ for given $\mathbf{f}_j$ and $\mathbf{A}_j$ increases with the number of unknowns $\#\Delta_j$.
In general, discretizations are arranged such that the size of the system $\{\phi_j\}$ grows exponentially in j. Thus, the matrix $\mathbf{A}_j$ can be extremely large which prohibits the use of direct solvers for (1.3.27). Also, $\mathbf{A}_j$ is ill-conditioned for any single-scale basis Φ_j, i.e., $\kappa_2(\mathbf{A}_j) \gg 1$, which induces increasingly slow convergence and prohibitively high iteration numbers for iterative solvers.
We seek a strategy which is of optimal complexity such that $\mathbf{y}_j$ is obtained with $\mathcal{O}(\#\Delta_j)$ operations and thus in linear time. To achieve this, we aim at constructing preconditioners $\mathbf{C}_j$ for $\mathbf{A}_j$ which have the property that

$$\kappa_2(\mathbf{C}_j^{-1/2}\mathbf{A}_j\mathbf{C}_j^{-1/2}) \ll \kappa_2(\mathbf{A}_j)$$

holds. Ideally, we will later see that preconditioning can yield

$$\kappa_2(\mathbf{C}_j^{-1/2}\mathbf{A}_j\mathbf{C}_j^{-1/2}) \sim 1$$

independent of j.
The next section will therefore be concerned with ingredients from multiscale analysis which allow for fast iterative solving of the equation (1.3.27) by providing a version of $\mathbf{A}_j$ which is **uniformly well-conditioned**. The solution vector $\mathbf{y}_j$ will be proved to be computable within preset discretization error accuracy in linear complexity in the total number of unknowns.
Although later control problems with Dirichlet boundary conditions and saddle point problems will be considered, we demonstrate these techniques for convenience for the simple system (1.3.27) first.

2 Wavelets and Multiresolution Analysis

This section provides an introduction into the theory of wavelets and their construction by means of multiresolution analysis. Although the theoretical groundwork for wavelets was laid as early as the beginning of the 19$^{\text{th}}$-century, it was not until 1909 that the Hungarian mathematician Alfred Haar proposed a piecewise constant, refinable, L_2-orthogonal and compactly supported function suitable for level dependent signal processing. This function is today known as the first wavelet (see [63]). However, the Haar-Wavelet is non-differentiable and thus not suitable for elliptic PDEs, since a weak solution is at least in H^1 and this means, by the Embedding Theorem 1.26, it has to be continuous.
In the 1980's, wavelets constructed by I. Daubechies featured compact support and L_2-orthogonality, yet they were only given implicitly by their coefficients. The later introduced concept of biorthogonality has proved of value since, as the important reconstruction and smoothness properties can now be sought independently.
Today, B-spline-based wavelets with arbitrary smoothness and compact support can be constructed, as first outlined in [19]. These wavelets are the result of a strict theoretical framework and a complex construction process which we will describe in the following sections and which exhibits a beautiful and elegant piece of modern functional analysis.

2.1 Multiscale Decompositions of Function Spaces

The content of this section is based on [25], [46] and [57].

2.1.1 Basics

Let Δ be a (possibly infinite) index set and $\#\Delta$ its cardinality. Then $\ell_2(\Delta)$ is the **Banach space** of elements $\mathbf{v} \in \ell_2(\Delta)$ for which the norm

$$\|\mathbf{v}\|_{\ell_2(\Delta)} := \left(\sum_{k \in \Delta} |v_k|^2 \right)^{1/2} \tag{2.1.1}$$

is finite. The elements $\mathbf{v} \in \ell_2(\Delta)$ are always regarded as **column vectors** of possibly infinite length. Likewise, we define a (countable) collection of functions Φ in a **Hilbert space** $\mathcal{H}$ as a column vector, whose elements are sorted accordingly and in a fixed order. This enables us to introduce the following shorthand notation for an expansion of Φ with a coefficient vector $\mathbf{c}$,

$$\mathbf{c}^T \Phi := \sum_{\phi \in \Phi} c_\phi \phi \ . \tag{2.1.2}$$

Recall from Section 1 the dual form $\langle v, \tilde{v} \rangle := \langle v, \tilde{v} \rangle_{\mathcal{H} \times \mathcal{H}'} := \tilde{v}(v)$. Consequently, for any $\tilde{v} \in \mathcal{H}'$, the quantity $\langle \Phi, \tilde{v} \rangle$ is interpreted as a **column vector** and $\langle \tilde{v}, \Phi \rangle$ as a **row vector** of expansion coefficients $\langle \phi, \tilde{v} \rangle$, $\langle \tilde{v}, \phi \rangle$, $\phi \in \Phi$, respectively. Furthermore, for any two collections $\Phi \subset \mathcal{H}$, $\Theta \subset \mathcal{H}'$ of functions, we frequently work with (possibly infinite) matrices of the form

$$\langle \Phi, \Theta \rangle := (\langle \phi, \theta \rangle)_{\phi \in \Phi, \theta \in \Theta} \ . \tag{2.1.3}$$

For any finite subset $\Phi \subset \mathcal{H}$ the linear span of Φ is abbreviated as

$$S(\Phi) := \operatorname{span} \{\Phi\} \ . \tag{2.1.4}$$

In order to make a function $w \in \mathcal{H}$ numerically accessible, its expansion coefficients $\mathbf{w}$ in a basis Φ of $\mathcal{H}$ should be **unique** and **stable**.

Definition 2.1 [Riesz basis of $\mathcal{H}$]
A family $\Phi = \{\phi_j\}$ of elements of a separable Hilbert space $\mathcal{H}$ is called **Riesz basis**, *if and only if the functions in $\{\phi_j\}$ are linearly independent and for every $\mathbf{c} \in \ell_2$ one has*

$$\|\mathbf{c}\|_{\ell_2} \sim \|\mathbf{c}^T \Phi\|_{\mathcal{H}}, \tag{$\mathcal{S}$)(2.1.5}$$

which is called **Riesz stability** *or just* **stability**.

We will later derive conditions under which multiscale wavelet bases are automatically Riesz bases.

2.1.2 Multiresolution Analysis of $\mathcal{H}$

Recalling Definition 1.3, the inner product of $\mathcal{H}$ is termed $(\cdot,\cdot) := (\cdot,\cdot)_{\mathcal{H}}$, associated with the norm $\|\cdot\|_{\mathcal{H}}$. The elements of $\mathcal{H}$ shall be functions living on a bounded domain $\Omega \subset \mathbb{R}^n$ with values in $\mathbb{R}$. We first consider the univariate case $n = 1$. In case $\Omega \subset \mathbb{R}^n, n > 1$, a tensor product approach is often sufficient for simple domains. We will discuss tensor products and their application in multiresolution analysis frameworks in Section 2.4.

Definition 2.2 [**Multiresolution Analysis (MRA) of $\mathcal{H}$**]
For a fixed parameter $j_0 \in \mathbb{N}_0$, a **multiresolution analysis $\mathcal{S}$ of $\mathcal{H}$** *consists of closed subspaces S_j of $\mathcal{H}$, called multiresolution spaces, which are nested such that their union is dense in $\mathcal{H}$,*

$$S_{j_0} \subset \ldots \subset S_j \subset S_{j+1} \subset \ldots \subset \mathcal{H}, \qquad \operatorname*{clos}_{\mathcal{H}}\Big(\bigcup_{j\geq j_0} S_j\Big) = \mathcal{H}\ . \qquad (\mathcal{R})(2.1.6)$$

Specifically, the multiresolution spaces S_j will be of type

$$S_j = S(\Phi_j), \qquad \Phi_j = \{\phi_{j,k} \,|\, k \in \Delta_j\}, \qquad (2.1.7)$$

each defined by a finite dimensional basis Φ_j with Δ_j being a level dependent finite index set. The bases $(\Phi_j)_{j\geq j_0}$ will be assumed to be **uniformly stable** *in the sense of Definition 2.1, i.e., property $(\mathcal{S})(2.1.5)$ holds uniformly for every $j \geq j_0$.*

The index j always denotes the **level of resolution** or **refinement level** with j_0 being the **coarsest level**. We shall always deal with functions $\phi_{j,k}$ which have the **locality** property, i.e., they are compactly supported with

$$\operatorname{diam}(\operatorname{supp}\phi_{j,k}) \sim 2^{-j}\ . \qquad (\mathcal{L})(2.1.8)$$

For this reason the collection Φ_j is termed **single-scale basis**, since all its members live on the same scale j. It follows from $(\mathcal{S})(2.1.5)$ with $\mathbf{c} = e_k$, that the $\phi_{j,k}$ must be scaled such that

$$\|\phi_{j,k}\|_{\mathcal{H}} \sim 1$$

holds. Here, k is called the **positional index** describing the location of the element ϕ_k in the space V. Considering for a moment $\Omega = \mathbb{R}$, the basis functions for Φ_j can be given by translation and dilation of a single function ϕ called the **generator**, i.e.,

$$\phi_{j,k}(x) := (2^j)^{1/2}\phi(2^j x - k), \qquad k \in \mathbb{Z},\ j \geq j_0\ . \qquad (2.1.9)$$

In the view of the locality condition $(\mathcal{L})(2.1.8)$, (2.1.9) means that $\operatorname{diam}\operatorname{supp}\phi_{j+1,k} \sim \frac{1}{2}\operatorname{diam}\operatorname{supp}\phi_{j,k}$ and therefore Φ_j can model more detail information with increasing level, which led to the designation **multiresolution analysis**. Here ϕ is called the **generator of the MRA** $(\mathcal{R})(2.1.6)$.
It is easy to verify that the $\{\phi_{j,k}\}_{j\geq j_0}$ form a Riesz basis for the space S_j with the same constants as in the case $j = j_0$. Since the MRA spaces are nested, there exists a special sequence of coefficients $\{m_k\}_{k\in\mathbb{Z}} \in \ell_2(\mathbb{Z})$ such that for every $x \in \Omega$

$$\phi(x) = \sum_{k\in\mathbb{Z}^m} m_k\phi(2x - k)\ .$$

Remark 2.3 *For the* **cardinal B-splines**, *the expansion coefficients can be found in [33], see also Appendix B.*

It follows that such a **refinement relation** can also be expressed for any of the functions $\phi_{j,k},\ j \geq j_0$, leading to the existence of matrices $\mathbf{M}_{j,0} = (m^{j,0}_{r,k})_{r\in\Delta_{j+1},k\in\Delta_j}$ such that the **two-scale relation**

$$\phi_{j,k} = \sum_{r\in\Delta_{j+1}} m^{j,0}_{r,k}\phi_{j+1,r}, \qquad k \in \Delta_j, \qquad (2.1.10)$$

is satisfied. The sequence $\mathbf{m}^j_k := (m^{j,0}_{r,k})_{r\in\Delta_{j+1}} \in \ell_2(\Delta_{j+1})$ is called **mask** and each element a **mask coefficient**. Since every function $\phi_{j,k}$ has compact support and only a finite number of functions $\phi_{j+1,k}$

have support intersecting with the support of $\{\phi_{j,k}\}$, non-zero mask coefficients only appear for these functions on level $j+1$. This means $\mathbf{m}_k^j$ has a uniformly, i.e., level independent, bounded number of non-zero entries. This will be crucial in the application of the **fast wavelet transform** in Section 2.1.3. In the sequel, it will be convenient to write (2.1.10) as a matrix-vector equation

$$\Phi_j = \mathbf{M}_{j,0}^T \Phi_{j+1} \ . \tag{2.1.11}$$

Thus, the $\mathbf{m}_k^j$ constitute the columns of the matrix $\mathbf{M}_{j,0} \in I\!R^{(\#\Delta_{j+1})\times(\#\Delta_j)}$. Any family of functions satisfying an equation of this form will be called **refinable**. It is known also for $\Omega \subset I\!R$ that nestedness $(\mathcal{R})$(2.1.6) and stability $(\mathcal{S})$(2.1.5) alone imply the existence of such matrices (see [25]). Obviously, $\mathbf{M}_{j,0}$ is a linear operator from the space $\ell_2(\Delta_j)$ into the space $\ell_2(\Delta_{j+1})$, i.e., recalling Definition 1.1,

$$\mathbf{M}_{j,0} \in L(\ell_2(\Delta_j); \ell_2(\Delta_{j+1})) \ .$$

This matrix is also **uniformly sparse** which means that the number of entries in each row or column are uniformly bounded. Because of the two-scale relation (2.1.10) every $c = \mathbf{c}_j^T \Phi_j$ has a representation $c = \mathbf{c}_{j+1}^T \Phi_{j+1}$, which, in recognition of the norm equivalence $(\mathcal{S})$(2.1.5) applied to both Φ_{j+1} and Φ_j, yields

$$\|\mathbf{c}_j\|_{\ell_2} \sim \|\mathbf{c}_j^T \Phi_j\|_{\mathcal{H}} = \|\mathbf{c}_j^T \mathbf{M}_{j,0}^T \Phi_{j+1}\|_{\mathcal{H}} = \|(\mathbf{M}_{j,0}\mathbf{c}_j)^T \Phi_{j+1}\|_{\mathcal{H}} \sim \|\mathbf{M}_{j,0}\mathbf{c}_j\|_{\ell_2},$$

and consequently, with the definition of operator norm (1.1.2), it follows that

$$\|\mathbf{M}_{j,0}\| = \mathcal{O}(1), \qquad j \geq j_0 \ .$$

Because the spaces Φ_j are nested and their infinite union $\mathcal{S}$ is dense in $\mathcal{H}$, a basis for $\mathcal{H}$ can be assembled from the functions which span the complement of two successive spaces Φ_j and Φ_{j+1}, i.e.,

$$S(\Phi_{j+1}) = S(\Phi_j) \oplus S(\Psi_j), \tag{2.1.12}$$

if we define

$$\Psi_j := \{\psi_{j,k} \,|\, k \in \nabla_j\} \subset S(\Phi_{j+1}), \qquad \nabla_j := \Delta_{j+1} \setminus \Delta_j \ . \tag{2.1.13}$$

The complement spaces $\mathrm{W}_j := S(\Psi_j), j \geq j_0$, are called **detail spaces**.

Definition 2.4 [Wavelets]
The basis functions $\psi_{j,k}$, $j \geq j_0$, $k \in \mathbb{Z}^n$, of the detail spaces $\Psi_j, j \geq j_0$, are denoted as **wavelet functions** *or shortly* **wavelets**.

There is more than one way to choose a basis for the space W_j. One option would be to use the orthogonal complement. Of special interests for the case $\Omega = I\!R$ are those bases of wavelet spaces which can be constructed from a **mother wavelet** ψ by scaling and dilation in the sense of (2.1.9),

$$\psi_{j,k}(x) := (2^j)^{1/2} \psi(2^j x - k), \qquad k \in \mathbb{Z}, j \geq j_0 \ . \tag{2.1.14}$$

Thus, if the mother wavelet is compactly supported, the wavelets also satisfy

$$\operatorname{diam} \operatorname{supp} \psi_{j,k} \sim 2^{-j} \ . \tag{2.1.15}$$

Wavelets of this kind entail a similar band-like structure in $\mathbf{M}_{j,1}$ as seen in $\mathbf{M}_{j,0}$. For this reason, we shall restrict all following discussions to the case of compactly supported generators and compactly supported wavelets. Since every $\psi_{j,k} \in \Psi_j$ is also in the space Φ_{j+1} it has a unique representation

$$\psi_{j,k} = \sum_{r \in \Delta_{j+1}} m_{r,k}^{j,1} \phi_{j+1,r}, \qquad k \in \nabla_j, \tag{2.1.16}$$

which can again be expressed as a matrix-vector equation of the form

$$\Psi_j = \mathbf{M}_{j,1}^T \Phi_{j+1} \tag{2.1.17}$$

with a matrix $\mathbf{M}_{j,1} \in I\!R^{(\#\Delta_{j+1})\times(\#\nabla_j)}$. Furthermore, equation (2.1.12) is equivalent to the fact that the linear operator composed of $\mathbf{M}_{j,0}$ and $\mathbf{M}_{j,1}$,

$$\mathbf{M}_j := (\mathbf{M}_{j,0}, \mathbf{M}_{j,1}) : \begin{array}{ccc} \ell_2(\Delta_j) \times \ell_2(\nabla_j) & \longrightarrow & \ell_2(\Delta_{j+1}) \\ (\mathbf{c}, \mathbf{d}) & \longmapsto & \mathbf{M}_{j,0}\mathbf{c} + \mathbf{M}_{j,1}\mathbf{d} \end{array} \tag{2.1.18}$$

is an **invertible** mapping from $\ell_2(\Delta_j \cup \nabla_j)$ onto $\ell_2(\Delta_{j+1})$. The refinement relations (2.1.11) and (2.1.17) combined lead to

$$\begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{j,0}^T \\ \mathbf{M}_{j,1}^T \end{pmatrix} \Phi_{j+1} =: \mathbf{M}_j^T \Phi_{j+1}, \tag{2.1.19}$$

called **decomposition identity**. This means $\mathbf{M}_j$ performs a change of bases in the space Φ_{j+1}. Of course, we want $\mathbf{M}_j$ to have positive traits which can be exploited for numerical purposes, such as sparseness and invertibility.

Definition 2.5 [Stable Decomposition]
If the union $\{\Phi_j \cup \Psi_j\}$ is uniformly stable in the sense of ($\mathcal{S}$)(2.1.5), i.e.,

$$\|\mathbf{c}\|_{\ell_2(\Delta_{j+1})} \sim \|(\Phi_j^T, \Psi_j^T)\mathbf{c}\|_{\mathcal{H}},$$

then $\{\Phi_j, \Psi_j\}$ is called a **stable decomposition** *of Φ_{j+1}.*

Note that $\mathbf{M}_j$ as a basis transformation must be invertible. We denote its inverse by $\mathbf{G}_j$, which we conveniently write in block structure as

$$\mathbf{M}_j^{-1} =: \mathbf{G}_j = \begin{pmatrix} \mathbf{G}_{j,0} \\ \mathbf{G}_{j,1} \end{pmatrix}, \tag{2.1.20}$$

with $\mathbf{G}_{j,0} \in I\!R^{(\#\Delta_j)\times(\#\Delta_{j+1})}$ and $\mathbf{G}_{j,1} \in I\!R^{(\#\nabla_j)\times(\#\Delta_{j+1})}$. It is known, see [24] for example, that $\{\Phi_j \cup \Psi_j\}$ is uniformly stable if and only if

$$\|\mathbf{M}_j\|, \|\mathbf{G}_j\| = \mathcal{O}(1), \qquad j \to \infty\ . \tag{2.1.21}$$

This condition can be met by any matrix and its inverse with entries whose absolute values are uniformly bounded, e.g., constant, and which are uniformly sparse, i.e., the number of entries in each row and column is independent of j. However, the inverses of sparse matrices are usually densely populated which has made actual construction burdensome in the past. It also draws special attention to the above mentioned choice of the basis Ψ_j, which determines $\mathbf{M}_{j,1}$ through the refinement relation (2.1.16).

Definition 2.6 [Stable Completion]
Any matrix $\mathbf{M}_{j,1}$ which completes $\mathbf{M}_{j,0}$ to a square $(\#\Delta_{j+1})\times(\#\Delta_{j+1})$ matrix, such that $\mathbf{M}_j$ is invertible and (2.1.21) is satisfied, is called **stable completion**.

In other words, the search for a basis Ψ_j of space W_j, consisting of compactly supported wavelets, can be exchanged for the algebraic search of refinement matrices, which are uniformly sparse with uniformly sparse inverses, too. There is a special type of sparse matrices $\mathbf{M}_j$, whose inverses are automatically sparse, namely, orthogonal matrices.

Definition 2.7 [Orthogonal Wavelets]
The wavelets are called **orthogonal** *if*

$$\langle \Psi_j, \Psi_j \rangle = \mathbf{I}, \tag{2.1.22}$$

which is true, if and only if the special situation occurs that $\mathbf{M}_j$ is **orthogonal**, *that is,*

$$\mathbf{G}_j = \mathbf{M}_j^{-1} = \mathbf{M}_j^T\ . \tag{2.1.23}$$

Remark 2.8 *Orthogonality will later be extended by the principle of biorthogonality in Section 2.1.4 .*

Considering $\mathbf{G}_j$ again, it is clear that applying $\mathbf{G}_j^T$ on both sides of (2.1.19) results in the so called **reconstruction identity**,

$$\Phi_{j+1} = \mathbf{G}_j^T \begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \mathbf{G}_{j,0}^T \Phi_j + \mathbf{G}_{j,1}^T \Psi_j, \tag{2.1.24}$$

which enables us now to freely **change representations** of functions between the **single-scale basis** Φ_{j+1} and the **multiscale basis** $\{\Phi_j \cup \Psi_j\}$.

Remark 2.9 *In case $\Omega \subsetneq I\!R$, definitions (2.1.9) and (2.1.14) can be applied only for a limited range of the shifting parameter k. At the boundary $\partial\Omega$ of Ω it might not be applicable at all. Constructions of boundary adapted generators (for various boundary conditions) exist that the assertions of the previous section still hold true in this case.*

2.1.3 Multiscale Transformation

Repeating (2.1.12), starting with a fixed finest level of resolution J up to the coarsest level j_0, yields a multiscale decomposition for the single-scale space $S_J = S(\Phi_J)$,

$$S(\Phi_J) = S(\Phi_{j_0}) \oplus \bigoplus_{j=j_0}^{J-1} S(\Psi_j) \ . \tag{2.1.25}$$

Thus, every $v \in S(\Phi_J)$ with its **single-scale representation**

$$v = \mathbf{c}_J^T \Phi_J = \sum_{k \in \Delta_J} c_{J,k} \phi_{J,k} \tag{2.1.26}$$

can be written in **multiscale form**

$$v = \mathbf{d}_{(J)}^T \Psi_{(J)} := \mathbf{c}_{j_0}^T \Phi_{j_0} + \mathbf{d}_{j_0}^T \Psi_{j_0} + \ldots + \mathbf{d}_{J-1}^T \Psi_{J-1} \tag{2.1.27}$$

with respect to the wavelet basis

$$\Psi_{(J)} := \Phi_{j_0} \cup \bigcup_{j=j_0}^{J-1} \Psi_j = \bigcup_{j=j_0-1}^{J-1} \Psi_j, \qquad \Psi_{j_0-1} := \Phi_{j_0} \ . \tag{2.1.28}$$

We will use the abbreviation

$$\mathbf{d}^T \equiv \mathbf{d}_{(J)}^T := (\mathbf{c}_{j_0}^T, \mathbf{d}_{j_0}^T, \ldots, \mathbf{d}_{J-1}^T) \tag{2.1.29}$$

for the multiscale vector and

$$\mathbf{c}^T \equiv \mathbf{c}_{(J)}^T := \mathbf{c}_J^T \tag{2.1.30}$$

for the single-scale coefficients, omitting the index J, if it does not create confusion. The transformation responsible for computing the single-scale coefficients from the multiscale wavelet coefficients is commonly referred to as the **wavelet transform** or **reconstruction algorithm**

$$\mathbf{T}_J : \ell_2(\Delta_J) \longrightarrow \ell_2(\Delta_J), \qquad \mathbf{d}_{(J)} \mapsto \mathbf{c}_{(J)}, \tag{2.1.31}$$

which, in recognition of the decomposition identity (2.1.19), will involve the application of $\mathbf{M}_j$. In fact, (2.1.18) states

$$\mathbf{c}_j^T \Phi_j + \mathbf{d}_j^T \Psi_j = (\mathbf{M}_{j,0}\mathbf{c}_j + \mathbf{M}_{j,1}\mathbf{d}_j)^T \Phi_{j+1} =: (\mathbf{c}_{j+1})^T \Phi_{j+1},$$

which, if iterated starting from level j_0 to level J, can be visualized as a **pyramid scheme**:

$$\begin{array}{ccccccccccc}
 & & \mathbf{M}_{j_0,0} & & \mathbf{M}_{j_0+1,0} & & & & & \mathbf{M}_{J-1,0} & \\
\mathbf{c}_{j_0} & \longrightarrow & & \mathbf{c}_{j_0+1} & \longrightarrow & \mathbf{c}_{j_0+2} & \longrightarrow & \cdots & \mathbf{c}_{J-1} & \longrightarrow & \mathbf{c}_J \\
 & & \mathbf{M}_{j_0,1} & & \mathbf{M}_{j_0+1,1} & & & & & \mathbf{M}_{J-1,1} & \\
 & \nearrow & & \nearrow & & \nearrow & & & \nearrow & & \\
\mathbf{d}_{j_0} & & & \mathbf{d}_{j_0+1} & & \mathbf{d}_{j_0+2} & & \cdots & \mathbf{d}_{J-1} & &
\end{array} \tag{2.1.32}$$

By this scheme, the operator $\mathbf{T}_J$ can be written as a product of levelwise operators

$$\mathbf{T}_J = \mathbf{T}_{J,J-1} \cdots \mathbf{T}_{J,j_0}, \tag{2.1.33}$$

where each factor has the form

$$\mathbf{T}_{J,j} := \begin{pmatrix} \mathbf{M}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(\#\Delta_J - \#\Delta_{j+1})} \end{pmatrix} \in I\!R^{(\#\Delta_J)\times(\#\Delta_J)} \ . \tag{2.1.34}$$

Conversely, the **inverse wavelet transform**, also known as **decomposition algorithm**,

$$\mathbf{T}_J^{-1} : \ell_2(\Delta_J) \longrightarrow \ell_2(\Delta_J), \qquad \mathbf{c}_{(J)} \mapsto \mathbf{d}_{(J)}, \tag{2.1.35}$$

can be written in a similar product structure by applying the inverses of the matrices $\mathbf{T}_{J,j}$ in reverse order. The inverses of $\mathbf{T}_{J,j}$ can be constructed as

$$\mathbf{T}_{J,j}^{-1} := \begin{pmatrix} \mathbf{G}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(\#\Delta_J - \#\Delta_{j+1})} \end{pmatrix} \in I\!R^{(\#\Delta_J)\times(\#\Delta_J)}, \tag{2.1.36}$$

and the inverse wavelet transform now takes on the form

$$\mathbf{T}_J^{-1} = \mathbf{T}_{J,j_0}^{-1} \cdots \mathbf{T}_{J,J-1}^{-1} \ . \tag{2.1.37}$$

The corresponding pyramid scheme is

$$\begin{array}{ccccccccccc} & \mathbf{G}_{J-1,0} & & \mathbf{G}_{J-2,0} & & & & & \mathbf{G}_{j_0,0} & & \\ \mathbf{c}_J & \longrightarrow & \mathbf{c}_{J-1} & \longrightarrow & \mathbf{c}_{J-2} & \longrightarrow & \cdots & \mathbf{c}_{j_0+1} & \longrightarrow & \mathbf{c}_{j_0} \\ & \mathbf{G}_{J-1,1} & & \mathbf{G}_{J-2,1} & & & & & \mathbf{G}_{j_0,1} & & \\ & \searrow & & \searrow & & \searrow & & & \searrow & & \\ & & \mathbf{d}_{J-1} & & \mathbf{d}_{J-2} & & \cdots & & & \mathbf{d}_{j_0} & \end{array} \tag{2.1.38}$$

Remark 2.10 *Since $\mathbf{M}_j$ and $\mathbf{G}_j$ have only a uniformly bounded number of non-zero entries in each row and column, each can be applied with a number of arithmetic operations that is of order $\mathcal{O}(\#\Delta_{j+1})$. This obviously also holds for the operators $\mathbf{T}_{J,j}$,$\mathbf{T}_{J,j}^{-1}$. Therefore, the application of operators $\mathbf{T}_J$,$\mathbf{T}_J^{-1}$ will always be computed by successively applying each operator $\mathbf{T}_{J,j}$,$\mathbf{T}_{J,j}^{-1}$. We strongly emphasize that the matrices given by $\mathbf{T}_J$,$\mathbf{T}_J^{-1}$* **are never explicitly computed and stored in computer memory**. *Such an action results in a complexity of $\mathcal{O}(\#\Delta_{j+1}\log(\#\Delta_{j+1}))$ and thus an unnecessary computational overhead.*

Let

$$N_j := \#\Delta_j$$

be the length of the coefficient vector $\mathbf{c}_j$ on level j. This leads to the following

Proposition 2.11 *The cost of applying $\mathbf{T}_J$ or $\mathbf{T}_J^{-1}$ using the pyramid scheme is of optimal linear complexity, that is, of order $\mathcal{O}(N_J) = \mathcal{O}(\dim S(\Phi_J))$. This justifies the expression* **Fast Wavelet Transform**.

Proof: Since the support of the basis functions $\phi_{j,k}$ halves from level to level, their number $\#\Delta_j$ must double. It follows that $\#\Delta_j$ grows exponentially in j and so does $\#\nabla_j$. Assuming $\#\Delta_{j+1} \approx 2\,\#\Delta_j$, it follows that the complexity of applying $\mathbf{M}_j$ or $\mathbf{G}_j$ also doubles for each level. But starting from the highest level J this means that the cost of calculating $\mathbf{M}_{J-1}\mathbf{c}_{J-1}$ is halve of what it is for $\mathbf{M}_J\mathbf{c}_J$. Then $N_{j-1} \approx \frac{N_j}{2}$ and the cost of applying $\mathbf{T}_J$ in the sense of (2.1.33) is roughly

$$\mathcal{O}(N_J + N_{J-1} + \ldots + N_{j_0+1}) = \mathcal{O}(N_J(1 + 2^{-1} + 2^{-2} + \ldots + 2^{-(J-j_0+1)})) = \mathcal{O}(2\,N_J) = \mathcal{O}(N_J) \ .$$

■

Remark 2.12 *In contrast, the discrete Fast Fourier Transform needs an overall amount of $\mathcal{O}(N_J\log(N_J))$ arithmetic operations, see [38].*

The fast wavelet transform will play a major part in the representation and fast assembly of operaters between Hilbert spaces with wavelet bases, see Section 2.2.3. It will be essential for preconditioning of the systems of linear equations in Section 4.
By ($\mathcal{R}$)(2.1.6) and (2.1.25), a basis for the whole space $\mathcal{H}$ can be given by letting $J \to \infty$ in (2.1.28),

$$\Psi^{I\!I} := \bigcup_{j=j_0-1}^{\infty} \Psi_j = \{\psi_{j,k} \,|\, (j,k) \in I\!I\}, \qquad I\!I := \bigcup_{j=j_0-1}^{\infty} \{\{j\} \times \nabla_j\}, \tag{2.1.39}$$

recalling $\Psi_{j_0-1} := \Phi_{j_0}$ and $\nabla_{j_0-1} := \Delta_{j_0}$. For any elements $\lambda := (j,k) \in I\!I$ we define $|\lambda| := j$. The interrelation between $\mathbf{T}_J$ and $\Psi^{I\!I}$ is displayed in the next theorem, taken from [25].

Theorem 2.13 *The multiscale Transformations $\mathbf{T}_J$,$\mathbf{T}_J^{-1}$ are well-conditioned,*

$$\|\mathbf{T}_J\|, \|\mathbf{T}_J^{-1}\| = \mathcal{O}(1), \qquad J \geq j_0, \tag{2.1.40}$$

if and only if the collection $\Psi^{I\!I}$ defined by (2.1.39) is a Riesz-Basis for $\mathcal{H}$, i.e.,

$$\|\mathbf{d}\|_{\ell_2(I\!I)} \sim \|\mathbf{d}^T\Psi^{I\!I}\|_{\mathcal{H}}, \qquad \text{for all } \mathbf{d} \in \ell_2(I\!I) \ . \tag{2.1.41}$$

This can be concluded from (2.1.21), see [24].

2.1.4 Dual Multiresolution Analysis of $\mathcal{H}'$

Let Φ_j as before be a Riesz-Basis of a Hilbert space $\mathcal{H}$ decomposed into an MRA as in $(\mathcal{R})$(2.1.6). By the Riesz representation theorem (see [3]), there exists a **dual basis** $\widetilde{\Phi}_j \subset \mathcal{H}'$ in the dual Hilbert space of $\mathcal{H}$. Of course, this basis $\widetilde{\Phi}_j$ is of the same cardinality as Φ_j and is also a **Riesz basis** of $\mathcal{H}'$. Moreover, it is part of a second **multiresolution analysis** $\widetilde{\mathcal{S}}$ of $\mathcal{H}'$, and it holds

$$\langle \Phi_j, \widetilde{\Phi}_j \rangle = \mathbf{I} \ . \tag{2.1.42}$$

We define the spaces

$$\widetilde{S}_j := S(\widetilde{\Phi}_j), \qquad \widetilde{\Phi}_j := \{\widetilde{\phi}_{j,k} \,|\, k \in \Delta_j\}, \tag{2.1.43}$$

where $\widetilde{\Phi}_j$ are designated **dual generator bases**, or just **dual generators**. In this setting, we refer to Φ_j of (2.1.7) as **primal generator bases** or **primal generators**. Furthermore, $P_j : \mathcal{H} \to S_j$ and $\widetilde{P}_j : \mathcal{H}' \to \widetilde{S}_j$ are projectors onto the spaces S_j and $\widetilde{S}_j$ defined by

$$P_j v := \langle v, \widetilde{\Phi}_j \rangle \Phi_j, \qquad v \in \mathcal{H}, \tag{2.1.44}$$

$$\widetilde{P}_j v := \langle v, \Phi_j \rangle \widetilde{\Phi}_j, \qquad v \in \mathcal{H}' \ . \tag{2.1.45}$$

These operators have the **projector property**

$$P_r P_j = P_r, \qquad \widetilde{P}_r \widetilde{P}_j = \widetilde{P}_r, \qquad r \le j, \tag{2.1.46}$$

which entails that $P_j - P_{j-1}$ and $\widetilde{P}_j - \widetilde{P}_{j-1}$ are also projectors. We can now define the primal and dual **detail spaces** employing these projectors as

$$\begin{array}{ll} \mathrm{W}_j := \mathrm{Im}(P_{j+1} - P_j), & \\ \widetilde{\mathrm{W}}_j := \mathrm{Im}(\widetilde{P}_{j+1} - \widetilde{P}_j), & j \ge j_0 \ . \end{array} \tag{2.1.47}$$

Setting $\widetilde{P}_{j_0-1} = P_{j_0-1} := 0$, we can write

$$S_{j_0} = \mathrm{W}_{j_0-1} = \mathrm{Im}(P_{j_0} - P_{j_0-1}), \qquad \widetilde{S}_{j_0} = \widetilde{\mathrm{W}}_{j_0-1} = \mathrm{Im}(\widetilde{P}_{j_0} - \widetilde{P}_{j_0-1}) \ .$$

The detail spaces W_j can also be expressed by

$$\mathrm{W}_j = S(\Psi_j) = S_{j+1} \cap (\widetilde{S}_j)^\perp$$

and concordantly the dual detail spaces as

$$\widetilde{\mathrm{W}}_j = S(\widetilde{\Psi}_j) = \widetilde{S}_{j+1} \cap (S_j)^\perp \ .$$

Nestedness and stability again imply that $\widetilde{\Phi}_j$ is refinable with some matrix $\widetilde{\mathbf{M}}_{j,0}$ similar to (2.1.11),

$$\widetilde{\Phi}_j = \widetilde{\mathbf{M}}_{j,0}^T \widetilde{\Phi}_{j+1} \ . \tag{2.1.48}$$

The main task is now to not only construct wavelet bases $\{\widetilde{\Psi}_j\}_{j \ge j_0}$ such that

$$\widetilde{S}_J = S(\widetilde{\Phi}_{j_0}) \cup \bigcup_{j=j_0}^{J} S(\widetilde{\Psi}_j)$$

is an MRA in $\mathcal{H}'$ analogously to (2.1.7), but also to ensure that the following biorthogonality conditions

$$\begin{array}{lll} S(\Phi_j) \perp S(\widetilde{\Psi}_j), & S(\widetilde{\Phi}_j) \perp S(\Psi_j) & j \ge j_0, \\ S(\Psi_j) \perp S(\widetilde{\Psi}_r), & & j \ne r, \end{array} \tag{2.1.49}$$

are satisfied. The connection between the concept of stable completions, the dual generators and wavelets is made by the following theorem taken from [14], see, e.g. [46].

Theorem 2.14 *Suppose that the biorthogonal collections* $\{\phi_j\}_{j=j_0}^{\infty}, \{\widetilde{\phi}_j\}_{j=j_0}^{\infty}$ *are both uniformly stable and refinable with refinement matrices* $\mathbf{M}_{j,0}, \widetilde{\mathbf{M}}_{j,0}$*, e.g.,*

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

and that they satisfy the duality condition (2.1.42). Assume that $\check{\mathbf{M}}_{j,1}$ *is any stable completion of* $\mathbf{M}_{j,0}$ *such that*

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

satisfies (2.1.21). Then

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

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

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

Moreover, the family of functions

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

form biorthogonal systems

$$\langle \Psi_j, \widetilde{\Psi}_j \rangle = \mathbf{I}, \qquad \langle \Psi_j, \widetilde{\Phi}_j \rangle = \langle \Phi_j, \widetilde{\Psi}_j \rangle = \mathbf{0}, \tag{2.1.50}$$

such that

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

Especially (2.1.42) combined with (2.1.50) implies that the wavelet spaces

$$\Psi^{I\!I} = \bigcup_{j=j_0-1}^{\infty} \Psi_j, \qquad \widetilde{\Psi}^{I\!I} := \bigcup_{j=j_0-1}^{\infty} \widetilde{\Psi}_j := \widetilde{\Phi}_{j_0} \cup \bigcup_{j=j_0}^{\infty} \widetilde{\Psi}_j, \tag{2.1.51}$$

are biorthogonal,

$$\langle \Psi^{I\!I}, \widetilde{\Psi}^{I\!I} \rangle = \mathbf{I} . \tag{$\mathcal{B}$)(2.1.52}$$

Definition 2.15 [Biorthogonal Wavelets]
Two such Riesz-Bases $\Psi^{I\!I}$ *of a MRA* $\mathcal{S} \subset \mathcal{H}$ *and* $\widetilde{\Psi}^{I\!I}$ *of MRA* $\widetilde{\mathcal{S}} \subset \mathcal{H}'$ *with property ($\mathcal{B}$)(2.1.52) are called* **biorthogonal wavelets**. $\Psi^{I\!I}$ *are called* **primal** *wavelets and* $\widetilde{\Psi}^{I\!I}$ **dual** *wavelets.*

With these bases, every $v \in \mathcal{H}$ has a unique expansion

$$v = \sum_{j=j_0-1}^{\infty} \langle v, \widetilde{\Psi}_j \rangle \Psi_j =: \sum_{j=j_0-1}^{\infty} \mathbf{v}_j^T \Psi_j =: \mathbf{v}^T \Psi^{I\!I} \tag{2.1.53}$$

and every $w \in \mathcal{H}'$ has a corresponding unique expansion

$$w = \sum_{j=j_0-1}^{\infty} \langle w, \Psi_j \rangle \widetilde{\Psi}_j =: \sum_{j=j_0-1}^{\infty} \widetilde{\mathbf{w}}_j^T \widetilde{\Psi}_j =: \widetilde{\mathbf{w}}^T \widetilde{\Psi}^{I\!I}, \tag{2.1.54}$$

and these expansions satisfy

$$\|v\|_{\mathcal{H}} \sim \|\langle v, \widetilde{\Psi}^{I\!I} \rangle^T\|_{\ell_2(I\!I)}, \qquad \|w\|_{\mathcal{H}'} \sim \|\langle w, \widetilde{\Psi}^{I\!I} \rangle^T\|_{\ell_2(I\!I)} . \tag{2.1.55}$$

In case $\mathcal{H} = \mathcal{H}'$ it follows that every $v \in \mathcal{H}$ has two unique expansions

$$v = \sum_{j=j_0-1}^{\infty} \langle v, \Psi_j \rangle \widetilde{\Psi}_j = \sum_{j=j_0-1}^{\infty} \langle v, \widetilde{\Psi}_j \rangle \Psi_j . \tag{2.1.56}$$

Accordingly, (2.1.53) is called **primal expansion** and (2.1.54) **dual expansion**. In this case, also the following norm equivalences hold for every $v \in \mathcal{H}$:

$$\|v\|_{\mathcal{H}} \sim \|\langle v, \widetilde{\Psi}^{\mathit{I\!I}} \rangle^T\|_{\ell_2(\mathit{I\!I})} \sim \|\langle v, \Psi^{\mathit{I\!I}} \rangle^T\|_{\ell_2(\mathit{I\!I})} \ . \tag{2.1.57}$$

Two useful expansions can be derived in this case of $\mathcal{H} = \mathcal{H}'$ by applying (2.1.53) for every basis function of $\widetilde{\Psi}^{\mathit{I\!I}}$ and (2.1.54) for $\Psi^{\mathit{I\!I}}$,

$$\Psi^{\mathit{I\!I}} = \langle \Psi^{\mathit{I\!I}}, \Psi^{\mathit{I\!I}} \rangle \widetilde{\Psi}^{\mathit{I\!I}} = \left(\Psi^{\mathit{I\!I}}, \Psi^{\mathit{I\!I}}\right) \widetilde{\Psi}^{\mathit{I\!I}} =: \mathbf{M}_{\mathcal{H}} \widetilde{\Psi}^{\mathit{I\!I}}, \tag{2.1.58}$$

$$\widetilde{\Psi}^{\mathit{I\!I}} = \langle \widetilde{\Psi}^{\mathit{I\!I}}, \widetilde{\Psi}^{\mathit{I\!I}} \rangle \Psi^{\mathit{I\!I}} = \left(\widetilde{\Psi}^{\mathit{I\!I}}, \widetilde{\Psi}^{\mathit{I\!I}}\right) \widetilde{\Psi}^{\mathit{I\!I}} =: \widetilde{\mathbf{M}}_{\mathcal{H}'} \Psi^{\mathit{I\!I}}, \tag{2.1.59}$$

which, when combined, bring forward an identity for the **Gramian matrices** $\mathbf{M}_{\mathcal{H}}, \widetilde{\mathbf{M}}_{\mathcal{H}'}$,

$$\mathbf{M}_{\mathcal{H}} \widetilde{\mathbf{M}}_{\mathcal{H}'} = \left(\Psi^{\mathit{I\!I}}, \Psi^{\mathit{I\!I}}\right) \left(\widetilde{\Psi}^{\mathit{I\!I}}, \widetilde{\Psi}^{\mathit{I\!I}}\right) = \mathbf{I} \ . \tag{2.1.60}$$

The **dual wavelet transformation** is designated as

$$\widetilde{\mathbf{T}}_J : \widetilde{\mathbf{d}}_{(J)} \longmapsto \widetilde{\mathbf{c}}_{(J)} \tag{2.1.61}$$

and from the biorthogonality equations (2.1.42) and ($\mathcal{B}$)(2.1.52), we can deduct

$$\mathbf{I} = \langle \widetilde{\Psi}_J^{\mathit{I\!I}}, \Psi_J^{\mathit{I\!I}} \rangle = \langle \widetilde{\mathbf{T}}_J^T \widetilde{\Phi}_J, \mathbf{T}_J^T \Phi_J \rangle = \widetilde{\mathbf{T}} \langle \widetilde{\Phi}_J, \Phi_J \rangle \mathbf{T}_J^T = \widetilde{\mathbf{T}}_J \mathbf{T}_J^T$$

or $\widetilde{\mathbf{T}}_J = \mathbf{T}_J^{-T}$ and, consequently, $\widetilde{\mathbf{T}}_J^{-1} = \mathbf{T}_J^T$, see Figure 2.1. It follows that $\widetilde{\mathbf{T}}_J$ has the same properties as $\mathbf{T}_J$, e.g. uniform sparseness and uniformly bounded condition numbers.
In Section 2.2.3 we will show how to apply $\mathbf{T}_J$ for preconditioning of linear elliptic operators. To this end, the assembly of operators is first done in terms of the single-scale functions Φ_j, and the fast wavelet transform is used to attain the wavelet representation. Hence, assembly is simple and computation fast. It should be pointed out that the dual basis functions $\widetilde{\Phi}_j$ and wavelets $\widetilde{\Psi}_j$ are not needed explicitly in this thesis. All which must be known is the dual wavelet transform, which is given by Figure 2.1.

	primal	dual
reconstruction	$\mathbf{T}_J : \mathbf{d} \longmapsto \mathbf{c}$ $\mathbf{T}_J = \widetilde{\mathbf{T}}_J^{-T}$	$\widetilde{\mathbf{T}}_J : \widetilde{\mathbf{d}} \longmapsto \widetilde{\mathbf{c}}$ $\widetilde{\mathbf{T}}_J = \mathbf{T}_J^{-T}$
decomposition	$\mathbf{T}_J^{-1} : \mathbf{c} \longmapsto \mathbf{d}$ $\mathbf{T}_J^{-1} = \widetilde{\mathbf{T}}_J^T$	$\widetilde{\mathbf{T}}_J^{-1} : \widetilde{\mathbf{c}} \longmapsto \widetilde{\mathbf{d}}$ $\widetilde{\mathbf{T}}_J^{-1} = \mathbf{T}_J^T$

Figure 2.1: Correlation of primal and dual multiscale transformations

2.2 Multiresolutions of L_2 and H^s

In the view of the application of the MRA framework to PDEs, we now need to consider elements of the Hilbert space $\mathcal{H}$ to be functions $f : \Omega \to I\!R, \Omega \subset I\!R$, lying in the function space L_2 or a subspace $H^s \subset L_2$. Let $\mathcal{S}$ be a multiresolution sequence of $\mathcal{H}$ as in Section 2.1.2, possibly incorporating boundary conditions of Ω.
Taking $\mathcal{H} = L_2(\Omega)$ conforms to the case $\mathcal{H} = \mathcal{H}'$ with the dual pairing

$$\langle f, g\rangle_{L_2(\Omega)\times L_2(\Omega)} := \int_\Omega f(x)g(x)d\mu, \qquad \text{for all } f, g \in L_2(\Omega) \ .$$

In case $\mathcal{H} = H^s, s > 0$, recall that $\mathcal{H}' = (H^s)', s > 0$, is a significantly larger space than H^s and it holds

$$H^s(\Omega) \subset L_2(\Omega) \subset (H^s(\Omega))', \qquad s > 0, \tag{2.2.1}$$

where the embedding is continuous and dense. This identity is an example of a **Gelfand triple**.

2.2.1 Approximation and Regularity Properties

Approximation properties refer to the ability to reproduce certain classes of functions with linear combinations of Φ_j. Of special interest are the spaces of polynomials

$$\Pi_r := \left\{ \sum a_i x^i : 0 \le i \le r-1 \right\} \ . \tag{2.2.2}$$

It will be important in the sequel that there are constants $d, \tilde{d} \in I\!N$ such that the space Π_d is contained in $S(\Phi_{j_0})$ and accordingly $\Pi_{\tilde{d}} \subset S(\widetilde{\Phi}_{j_0})$, and the following identities hold:

$$x^r = \sum_l \langle (\cdot)^r, \widetilde{\phi}_{j_0}(\cdot - k)\rangle \phi_{j_0}(x-k), \qquad r = 0, \dots, d-1, \tag{$\mathcal{P}$}(2.2.3)$$

$$x^r = \sum_l \langle (\cdot)^r, \phi_{j_0}(\cdot - k)\rangle \widetilde{\phi}_{j_0}(x-k), \qquad r = 0, \dots, \tilde{d}-1 \ . \tag{$\widetilde{\mathcal{P}}$}(2.2.4)$$

Since the spaces $S(\Phi_j), S(\widetilde{\Phi}_j)$ are nested, this also holds true for $S(\Phi_j), S(\widetilde{\Phi}_j), j \ge j_0$. By the biorthogonality conditions (2.1.50), this yields the so called **moment conditions**,

$$\int_\Omega x^r \psi_{j,k}(x)d\mu = 0, \qquad r = 0, \dots, \tilde{d}-1, \tag{$\mathcal{V}$}(2.2.5)$$

$$\int_\Omega x^r \widetilde{\psi}_{j,k}(x)d\mu = 0, \qquad r = 0, \dots, d-1, \tag{$\widetilde{\mathcal{V}}$}(2.2.6)$$

which means that the wavelets $\psi_{j,k}, \widetilde{\psi}_{j,k}$ are orthogonal to all polynomials up to order $\tilde{d}, d$, respectively. The wavelets $\psi_{j,k}, \widetilde{\psi}_{j,k}$ are said to have $\tilde{d}$-, d-th order **vanishing moments**.
Now we turn to **regularity** properties, commonly referred to as **smoothness**. We quote from [59] that every generator $\phi \in L_2$ of an MRA $\mathcal{S}$ is also contained in H^s for a certain range $[0, s), s > 0$. We define regularity properties which will play an integral part in the norm equivalence proposition in Section 2.2.2:

Definition 2.16 *The* **regularity** *of the MRAs $\mathcal{S}$ and $\widetilde{\mathcal{S}}$ is characterized by*

$$\gamma := \sup\{s \,|\, S(\Phi_j) \subset H^s, j \ge j_0\}, \qquad \tilde{\gamma} := \sup\{s \,|\, S(\widetilde{\Phi}_j) \subset H^s, j \ge j_0\} \ . \tag{2.2.7}$$

It is necessary to find the optimal balance between the three properties **regularity** $\gamma, \tilde{\gamma}$ (2.2.7), **polynomial exactness** d $(\mathcal{P})$(2.2.3) and **vanishing moments** $\tilde{d}$ $(\mathcal{V})$(2.2.5) of the trial spaces $S(\Phi_j)$ for any problem at hand.
The choice of $d, \tilde{d}$ is not entirely free. Existence of a compactly supported dual scaling function $\widetilde{\Phi}_j$ was proved in [22] for

$$\tilde{d} > d, \qquad d + \tilde{d} = \text{even} \ . \tag{2.2.8}$$

In case of biorthogonal **spline wavelets** it is known that the support of the generators is linked to the **polynomial exactness** (see [19])

$$\operatorname{supp}\phi_{j,k} = \mathcal{O}(d), \qquad \operatorname{supp}\widetilde{\phi}_{j,k} = \mathcal{O}(\tilde{d}) \ . \tag{2.2.9}$$

This in turn determines the length of the **mask** (2.1.10) and, thus, the constants involving the cost of applying the **fast wavelet transform** (2.1.31).
We will see in the following sections that the wavelet discretization of a differential operator $\mathcal{L} : H^{+t} \to (H^{+t})'$ of order 2t requires $\gamma, \tilde{\gamma} > |t|$.

2.2.2 Norm Equivalences for Sobolev Spaces $H^s \subset L_2$

The inner product $(\cdot,\cdot)_{H^s}$ and norm $\|\cdot\|_{H^s}$ of Sobolev spaces $H^s, s \in \mathbb{R}$, cannot be expressed analytically for arbitrary values of s like for L_2. Therefore, we must resort to norm equivalences which we will introduce and analyze now.
Up to now, Riesz stability is given uniformly for all spaces S_j, $j \geq j_0$, see Definition 2.2. In general, this does not immediately imply stability with respect to several levels, as is needed for infinite sums of elements of these spaces. Results of this kind combined with Theorem 2.13 ensure the Riesz-Basis properties (2.1.41) and (2.1.55) of $\Psi^{\mathbb{I}}$. To this end, regularity and approximation properties are required of $\mathcal{S}$ and $\widetilde{\mathcal{S}}$, which are formalized in the following theorem from [26], see also [32] and [45].

Theorem 2.17 *Let $\mathcal{S} := \{S_j\}_{j\geq j_0}$ and $\widetilde{\mathcal{S}} := \{\widetilde{S}_j\}_{j\geq j_0}$ be multiresolution sequences with bases Φ_j,$\widetilde{\Phi}_j$ satisfying properties Stability ($\mathcal{S}$)(2.1.5), Refinability ($\mathcal{R}$)(2.1.6), Locality ($\mathcal{L}$)(2.1.8) and Biorthogonality ($\mathcal{B}$)(2.1.52) and let P_j,$\widetilde{P}_j$ be defined by (2.1.44), (2.1.45). If $\mathcal{S}$,$\widetilde{\mathcal{S}}$ both satisfy the* **Jackson inequality**

$$\inf_{v_j \in \overline{S}_j} \|v - v_j\|_{L_2(\Omega)} \lesssim 2^{-sj}\|v\|_{H^s(\Omega)}, \qquad v \in H^s(\Omega),\ 0 \leq s \leq \overline{d}, \tag{2.2.10}$$

and the **Bernstein inequality**

$$\|v_j\|_{H^s(\Omega)} \lesssim 2^{sj}\|v_j\|_{L_2(\Omega)}, \qquad v_j \in \overline{S}_j,\ 0 \leq s < \overline{\gamma}, \tag{2.2.11}$$

for the spaces $\overline{S}_j = S_j, \widetilde{S}_j$ with order $\overline{d} = d$, $\tilde{d}$ and $\overline{\gamma} = \gamma, \tilde{\gamma}$, respectively, then for

$$0 < \sigma := \min\{d, \gamma\}, \qquad 0 < \tilde{\sigma} := \min\{\tilde{d}, \tilde{\gamma}\}$$

one obtains the norm equivalences

$$\begin{aligned}
\|v\|^2_{H^s(\Omega)} &\sim \sum_{j=j_0}^{\infty} \|(P_j - P_{j-1})v\|^2_{H^s(\Omega)} \\
&\sim \sum_{j=j_0}^{\infty} 2^{2sj}\|(P_j - P_{j-1})v\|^2_{L_2(\Omega)} \\
&\sim \sum_{j=j_0-1}^{\infty} 2^{2sj}\|\langle v, \widetilde{\Psi}_j\rangle^T\|^2_{\ell_2(\nabla_j)}, \qquad s \in (-\tilde{\sigma}, \sigma)\ .
\end{aligned}$$

Note that here is set $H^s = (H^{-s})'$ for $s < 0$.

Proof: Since this is a central result we give a short sketch of the proof. The interested reader should turn to [45] for complete details.
From the Bernstein inequality (2.2.11) follows with a **discrete Hardy inequality** (see, for example, [45]),

$$\|v\|_{H^s} \lesssim \left(\sum_{j=j_0}^{m} 2^{2sj}\|(P_j - P_{j-1})v\|^2_{L_2}\right)^{1/2} .$$

The lower estimate can only be shown to be of the form

$$\|v\|_{H^s} \gtrsim \frac{1}{\nu_m}\left(\sum_{j=j_0}^{m} 2^{2sj}\|(P_j - P_{j-1})v\|^2_{L_2}\right)^{1/2} ,$$

with a constant $\nu_m = \nu_m(S_j, P_j)$ depending on properties of the space S_j and the projector P_j. Obviously, we aim at ensuring $\nu_m = \mathcal{O}(1)$ for $m \to \infty$. It is proved in [23] that under the assumptions of this theorem from the **projector property** (2.1.46) automatically

$$\|P_j\| = \mathcal{O}(1), \qquad \text{for all } j \geq j_0,$$

follows. The Jackson Inequality (2.2.10) can be used to show an upper limit for ν_m,

$$\nu_m \lesssim \max\{ \sup_{0 \neq g \in S_j} (\frac{\|P_j g\|_{L_2}}{\|g\|_{L_2}} + 1) \,|\, j = 0, \ldots, m\},$$

which can be bounded from above by the norm of the projector P_j, which is bounded by the above remark. Thus, the existence of a lower bound for $1/\nu_m$ for $m \to \infty$ follows.
The proof of the equivalence $\sum \|(P_j - P_{j-1})v\|^2_{H^s(\Omega)} \sim \sum 2^{2sj} \|(P_j - P_{j-1})v\|^2_{L_2(\Omega)}$ can be found in [24]. Lastly, the equivalence to the wavelet expansion $\sum 2^{2sj} \|\langle v, \widetilde{\Psi}_j \rangle^T\|^2_{\ell_2(\nabla_j)}$ follows directly from the assertions of Theorem 2.14. ∎

Remark 2.18 *In particular for $s = 0$, we regain the* **Riesz basis** *property (2.1.57) for $\mathcal{H} = L_2$.*

Remark 2.19 *By interchanging the roles of $\mathcal{S}$ and $\widetilde{\mathcal{S}}$, we obtain*

$$\|v\|^2_{H^s(\Omega)} \sim \sum_{j=j_0}^{\infty} 2^{2sj} \|(\widetilde{P}_j - \widetilde{P}_{j-1})v\|^2_{L_2(\Omega)}, \qquad s \in (-\sigma, \tilde{\sigma}) \ .$$

Remark 2.20 *Usually $\gamma < d$ and $\tilde{\gamma} < \tilde{d}$ holds, in which case $\sigma = \gamma$ and $\tilde{\sigma} = \tilde{\gamma}$ follows.*

As stated in the above proof, the projectors $\overline{P} = P_j, \widetilde{P}_j$ are uniformly bounded in $H^s(\Omega)$, i.e.,

$$\|\overline{P}v\|_{H^s(\Omega)} \lesssim \|v\|_{H^s(\Omega)}, \qquad \text{for all } v \in H^s(\Omega),$$

for s up to $\sigma, \tilde{\sigma}$, respectively. Theorem 2.17 also implies that the unique wavelet expansions (2.1.53), (2.1.54) in bases $\Psi^{I\!I}, \widetilde{\Psi}^{I\!I}$ for every $v \in H^{+s}$, $\widetilde{v} \in (H^{+s})'$ satisfy the following **norm equivalences**

$$\|v\|_{H^{+s}} \sim \|\mathbf{D}^{+s}\mathbf{v}\|_{\ell_2(I\!I)}, \qquad v = \mathbf{v}^T\Psi^{I\!I} = \langle v, \widetilde{\Psi}^{I\!I}\rangle \Psi^{I\!I}, \tag{2.2.12}$$

$$\|\widetilde{v}\|_{H^{-s}} \sim \|\mathbf{D}^{-s}\widetilde{\mathbf{v}}\|_{\ell_2(I\!I)}, \qquad \widetilde{v} = \widetilde{\mathbf{v}}^T\widetilde{\Psi}^{I\!I} = \langle \widetilde{v}, \Psi^{I\!I}\rangle \widetilde{\Psi}^{I\!I}, \tag{2.2.13}$$

with the diagonal matrices $\mathbf{D}^{\pm s} = \mathbf{D}_1^{\pm s}$ defined by

$$\left(\mathbf{D}_1^{\pm s}\right)_{\lambda,\lambda'} := 2^{\pm|\lambda|s}\delta_{(\lambda,\lambda')} \ . \tag{2.2.14}$$

Here we set for any indexes $\lambda = (j,k)$, $\lambda' = (j',k')$ with $|\lambda| = j$

$$\lambda = \lambda' :\Longleftrightarrow j = j' \wedge k = k' \ .$$

Remark 2.21 *Other diagonal matrices exist for which the norm equivalences hold. In the following sections, the diagonal matrices $\mathbf{D}^{+s}, \mathbf{D}^{-s}$ should be understood as a placeholder for* **any** *diagonal matrix ensuring (2.2.12) and (2.2.13). The choice of the entries of the diagonal matrix is not important for theoretical considerations, but we will see the impact of their precise form in numerical applications in Section 5.*

Corollary 2.22 *For any diagonal matrix $\mathbf{D}^{+s}, \mathbf{D}^{-s}$ satisfying (2.2.12),(2.2.13), the wavelet bases $\Psi^s := \mathbf{D}^{-s}\Psi^{I\!I}$, $\widetilde{\Psi}^s := \mathbf{D}^{+s}\widetilde{\Psi}^{I\!I}$ constitute* **Riesz bases** *for H^{+s}, $(H^{+s})'$, respectively.*

The diagonal scaling can be seen as a smoothing of the wavelet basis for positive Sobolev indices and a roughening for negative indices. In the context of this thesis, an important consequence of norm equivalences is their ability to prove that operators in properly scaled wavelet discretizations are asymptotically optimally preconditioned.

2.2.3 Operator Representation and Preconditioning

To this end, we will describe how the wavelet representation of an operator as a successive application of linear operators is constructed. For later purposes, it will be convenient to derive this in a very general setting. Let Ω_A, Ω_B be two open, bounded domains in $\mathbb{R}^n$, not necessarily distinct, and let A be a linear operator

$$A : H^s(\Omega_A) \to (H^t(\Omega_B))', \qquad A : v_A \mapsto w_B \tag{2.2.15}$$

from the Sobolev space $H^s(\Omega_A), s \geq 0$, into the dual of the Sobolev space $H^t(\Omega_B), t \geq 0$. Let there be biorthogonal wavelet bases in both spaces at our disposal, both satisfying the **Riesz basis** criterion with diagonal matrices,

$$\begin{aligned} \Psi_A^s &:= \mathbf{D}_A^{-s}\Psi_A^{\mathbb{I}} && \text{(primal) basis of } H^s(\Omega_A), \\ \widetilde{\Psi}_A^s &:= \mathbf{D}_A^{+s}\widetilde{\Psi}_A^{\mathbb{I}} && \text{(dual) basis of } (H^s(\Omega_A))', \end{aligned}$$

and accordingly for $H^t(\Omega_B)$ and $(H^t(\Omega_B))'$. Henceforth, all Sobolev norms are only taken for values which the smoothness of primal and dual bases permit by Theorem 2.17. Now we can express v_A and w_B in terms of these wavelet bases as

$$v_A = \mathbf{v}^T\Psi_A^s := \langle v, \widetilde{\Psi}_A^s\rangle\Psi_A^s, \qquad w_B = \mathbf{w}^T\widetilde{\Psi}_B^t := \langle w, \Psi_B^t\rangle\widetilde{\Psi}_B^t,$$

and therefore it follows

$$\mathbf{w} = \langle \Psi_B^t, w\rangle = \langle \Psi_B^t, Av\rangle = \langle \Psi_B^t, A\Psi_A^s\rangle\mathbf{v},$$

which can be seen as a discretized infinite-dimensional operator equation

$$\mathbf{A}\mathbf{v} = \mathbf{w}, \tag{2.2.16}$$

upon setting

$$\mathbf{A} := \langle \Psi_B^t, A\Psi_A^s\rangle \ . \tag{2.2.17}$$

This is called the **standard representation** of the operator A in wavelet coordinates. Equation (2.2.17) can be reformulated as

$$\mathbf{A} = \langle \Psi_B^t, A\Psi_A^s\rangle = \mathbf{D}_B^{-t}\langle \Psi_B^{\mathbb{I}}, A\Psi_A^{\mathbb{I}}\rangle\mathbf{D}_A^{-s} = \mathbf{D}_B^{-t}\mathbf{T}_B^T\langle \Phi_B, A\Phi_A\rangle\mathbf{T}_A\mathbf{D}_A^{-s} \tag{2.2.18}$$

using the **wavelet transform** to express the multiscale bases with respect to the corresponding single-scale bases. It follows that the **adjoint operator** $A' : H^t(\Omega_B) \to (H^s(\Omega_A))'$ defined by

$$\langle A'w, v\rangle_{(H^s)'\times H^s} := \langle w, Av\rangle_{H^t\times(H^t)'}, \qquad \text{for all } v \in H^s(\Omega_A),\ w \in H^t(\Omega_B), \tag{2.2.19}$$

then has the representation

$$\mathbf{A}' = \langle \Psi_A^s, A'\Psi_B^t\rangle = \langle A'\Psi_B^t, \Psi_A^s\rangle^T = \langle \Psi_B^t, A\Psi_A^s\rangle^T = \mathbf{A}^T \ . \tag{2.2.20}$$

Remark 2.23 *In case $A : H^s(\Omega_A) \to H^t(\Omega_B)$ with $s, t \geq 0$, the construction process works accordingly. The role of the primal and dual wavelet bases of $H^t(\Omega_B)$ should be arranged such that the primal side is associated with the positive Sobolev scale index and hence with a smoothing of the wavelet basis.*

Preconditioning

Assuming $A : H^s(\Omega_A) \to (H^t(\Omega_B))'$ is an **isomorphism**, e.g.,

$$\|Av_A\|_{(H^t)'} \sim \|v_A\|_{H^s}, \qquad \text{for all } v_A \in H^s, \tag{2.2.21}$$

we can prove the following

Theorem 2.24 *The mapping $\mathbf{A} : \ell_2(\mathbb{I}) \to \ell_2(\mathbb{I})$ is an* **isomorphism** *on $\ell_2(\mathbb{I})$,*

$$\|\mathbf{A}\mathbf{v}\|_{\ell_2(\mathbb{I})} \sim \|\mathbf{v}\|_{\ell_2(\mathbb{I})} \sim \|\mathbf{A}^{-1}\mathbf{v}\|_{\ell_2(\mathbb{I})}, \qquad \textit{for all } \mathbf{v} \in \ell_2(\mathbb{I}), \tag{2.2.22}$$

with bounded spectral condition number

$$\kappa_2(\mathbf{A}) := \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 = \mathcal{O}(1) \ . \tag{2.2.23}$$

Proof: Expanding $v_A = \mathbf{v}_A^T \Psi_A^s$ in the properly scaled wavelet basis, we have by the norm equivalences (2.2.12), (2.2.13) and by assumption (2.2.21),

$$\begin{aligned} \|\mathbf{v}_A\|_{\ell_2} &\sim \|\mathbf{v}_A^T \Psi_A^s\|_{H^s} \\ &\sim \|\mathbf{v}_A^T A \Psi_A^s\|_{(H^t)'} \\ &= \|\mathbf{v}_A^T \langle A\Psi_A^s, \Psi_B^t\rangle \widetilde{\Psi}_B^t\|_{(H^t)'} \\ &\sim \|\langle \Psi_B^t, A\Psi_A^s\rangle \mathbf{v}_A\|_{\ell_2} \\ &= \|\mathbf{A}\mathbf{v}_A\|_{\ell_2} \ . \end{aligned}$$

Since a completely analogous equivalence relation holds true for the inverse $\mathbf{A}^{-1}$, the second equivalence in (2.2.22) follows. From this we can directly infer (2.2.23). ∎

Thus, the multiplication by the diagonal matrices $\mathbf{D}_B^{-t}$, $\mathbf{D}_A^{-s}$ has the capability of undoing the effect of A in the sense of the Sobolev scale. Hence, the diagonal matrices can be seen as **preconditioning** of the discretized linear system.

2.2.4 Finite Discretizations

Let $\Psi^{\mathcal{I}}, \widetilde{\Psi}^{\mathcal{I}}$ be two biorthogonal wavelet bases as in Section 2.1. We will denote all scaled wavelet bases up to a certain level of resolution J by

$$\Psi_J^s := \mathbf{D}_J^{-s} \Psi_J^{\mathcal{I}} = \{(\mathbf{D}^{-s})_{\lambda,\lambda}\, \psi_\lambda \,|\, \lambda \in \mathcal{I}_J\} \subset H^s, \qquad \mathcal{I}_J := \{\lambda \in \mathcal{I} \,|\, |\lambda| \le J\}, \tag{2.2.24}$$

where the finite diagonal scaling operator $\mathbf{D}_J^{-s}$ is constructed from $\mathbf{D}^s$ by deleting all rows and columns of indexes not in $\mathcal{I}_J$,

$$\mathbf{D}_J^{-s} := \left(\left(\mathbf{D}^{-s}\right)_{\lambda,\lambda'} \right)_{\lambda \in \mathcal{I}_J, \lambda' \in \mathcal{I}_J} \in I\!R^{(\#\mathcal{I}_J)\times(\#\mathcal{I}_J)} \ . \tag{2.2.25}$$

The unscaled finite wavelet basis of levels up to J is now defined as

$$\Psi_J^{\mathcal{I}} := \{\psi_{j,k} \in \Psi^{\mathcal{I}} \,|\, (j,k) \in \mathcal{I}_J\} \ . \tag{2.2.26}$$

Note that the wavelet space $\Psi_J^{\mathcal{I}}$ need not be the same as $\Psi_{(J)}$ in (2.1.28), for example in a tensor product setting of Section 2.4. Since definition (2.2.26) extends (2.1.28), it will be used primarily.

Remark 2.25 *This technique creates spaces which are* **linear***, which makes them easy to handle in applications. Conversely,* **non-linear** *wavelet spaces can be constructed by best* **N-Term approximations***. The resulting* **adaptive techniques** *give rise to higher approximation orders for non-smooth functions when compared to linear methods requiring the same storage amount in a computer system. We focus on the* **linear** *case on uniform grids in this work, since this is the optimal case for smooth given data and solutions. When dealing with singularities in the data or solution, enhancements can be achieved using adaptive methods. An introduction to non-linear approximations can be found in [35].*

We can directly conclude from (2.2.10) the value of the **discretization error** (1.3.28) with respect to Ψ_J^s as $h_j = 2^{-sj}$.

Best Approximations

An important question in the current setting is to find the **best approximation** $y^* := y_J^*$ of an element $y \in H^s$ in the subspace $S(\Psi_J^{\mathcal{I}}) \subseteq S(\Psi^{\mathcal{I}})$, or to give error bounds for $\|y - y^*\|_{H^s}$ for any valid value of s. The best approximation y^* of y is defined as the element for which (1.3.28) is minimized, i.e.,

$$\|y - y^*\|_{H^s} = \inf_{v \in S(\Psi_J^{\mathcal{I}})} \|y - v\|_{H^s} \ . \tag{2.2.27}$$

The natural candidate for y^* is obviously the orthogonal projection onto the space $\Psi_J^{\mathcal{I}}$ by means of the projectors P_J of (2.1.44). We can use these projectors to prove error bounds for (2.2.27) given in the following

Theorem 2.26 *For $y \in H^s, 0 < s \le d$, and $r \le s$ one has*

$$\inf_{v \in S(\Psi_J^{I\!I})} \|y - v\|_{H^r} \lesssim 2^{-(s-r)J} \|y\|_{H^s} \ .$$

Proof: We expand y in a telescopic sum

$$y = \sum_{l=j_0}^{\infty} (P_\ell - P_{\ell-1})\, y, \tag{2.2.28}$$

recalling $P_{j_0-1} = 0$ and the projector property $P_J P_j = P_J, j \ge J$. By Theorem 2.17, we have

$$\begin{aligned}
\inf_{v \in S(\Psi_J^{I\!I})} \|y - v\|_{H^r}^2 &\le \|y - P_J y\|_{H^r}^2 \\
&= \| \sum_{\ell=J+1}^{\infty} (P_\ell - P_{\ell-1})\, y\|_{H^r}^2 \\
&= \sum_{\ell=J+1}^{\infty} 2^{+2r\ell} \| (P_\ell - P_{\ell-1})\, y\|_{L_2}^2 \\
&= \sum_{\ell=J+1}^{\infty} 2^{-2(s-r)\ell} 2^{+2s\ell} \| (P_\ell - P_{\ell-1})\, y\|_{L_2}^2 \\
&\lesssim 2^{-2(s-r)J} \|y\|_{H^s}^2,
\end{aligned}$$

where we used the geometric series formula in the last step. The norm equivalence is still valid because only a finite number of coefficients were omitted in the sum, although the constants involved are probably higher. Assertion (2.2.28) now follows after taking the square root on both sides. ∎

The above result reads for $r = 0$:

$$\inf_{v \in S(\Psi_J^{I\!I})} \|y - v\|_{L_2} \lesssim 2^{-sJ} \|y\|_{H^s} \ . \tag{2.2.29}$$

This means that the convergence speed of the approximation by wavelet spaces (as $J \to \infty$) is directly linked to the **smoothness** $s > 0$ of the function to be approximated.

2.2.5 Stability of the Discretizations

Since we now have optimally conditioned infinite dimensional wavelet-discretized operators at our disposal, we only need to ensure stability when truncating all multiscale bases Ψ_j above a certain level $j > J$. Then we have **uniformly stable** finite dimensional operators which are easy to set up by means of the **single-scale basis** and the **fast wavelet transform** and yet carry all the advantages induced by the attributes $(\mathcal{S})$(2.1.5), $(\mathcal{R})$(2.1.6), $(\mathcal{L})$(2.1.8) and $(\mathcal{B})$(2.1.52). There are two types of criteria relevant for our control problems which ensure stability of the finite discretized systems, both of which we specify now.

Galerkin Stability

We fix a refinement level J at which we wish to find the solution $y_J \in S_J$ of an elliptic differential operator $A : H^{+t} \to (H^{+t})'$ in a **Galerkin scheme** (see Section 1.3.3),

$$\langle \Psi_J^t, A y_J \rangle = \langle \Psi_J^t, f \rangle, \tag{2.2.30}$$

with respect to the wavelet basis $\Psi_J^s \subset \Psi^{I\!I} \subset \mathcal{H}$ with the finite index set $I\!I_J \subset I\!I$.

Definition 2.27 [Galerkin Stability]
The Galerkin Scheme (2.2.30) is called **(t,-t)-stable**, *or* **Galerkin stable**, *if*

$$\|P_J v\|_{H^t} \sim \|\widetilde{P}_J A P_J v\|_{(H^t)'}, \qquad v \in S_J, \tag{2.2.31}$$

holds uniformly in J, with the projectors of (2.1.44) and (2.1.45).

Expanding $y_J = \mathbf{y}_J^T \Psi_J^t = \mathbf{y}_J^T \mathbf{D}_J^{-t} \Psi_J^{\mathbb{I}}$ and $f = \langle f, \Psi_J^t \rangle \widetilde{\Psi}_J^t = \mathbf{f}_J^T \widetilde{\Psi}_J^t = \mathbf{f}_J^T \mathbf{D}_J^{+t} \widetilde{\Psi}_J^{\mathbb{I}}$ in (2.2.30), we obtain

$$\mathbf{A}_J := \langle \Psi_J^t, A\Psi_J^t \rangle = \mathbf{D}_J^{-t} \langle \Psi_J^{\mathbb{I}}, A\Psi_J^{\mathbb{I}} \rangle \mathbf{D}_J^{-t}, \tag{2.2.32}$$

and analogously to (2.2.16) we can write (2.2.30) as a finite-dimensional discretized operator equation

$$\mathbf{A}_J \mathbf{y}_J = \mathbf{f}_J \ . \tag{2.2.33}$$

Galerkin stability now ensures the stability of our discretized operator.

Proposition 2.28 *If the Galerkin scheme is (t, -t)-stable and it holds*

$$|t| < \gamma, \tilde{\gamma},$$

then the matrices

$$\mathbf{A}_J = \mathbf{D}_J^{-t} \langle \Psi_J^{\mathbb{I}}, A\Psi_J^{\mathbb{I}} \rangle \mathbf{D}_J^{-t} \tag{2.2.34}$$

have uniformly bounded condition numbers.

Proof: The result follows from applying the norm equivalences (2.2.12),(2.2.13) combined with (2.2.30) and is altogether very similar to the proof of Theorem 2.24:
For $P_J v = \mathbf{v}_J^T \Psi_J^{\mathbb{I}} \in H^{+t}$, equivalence (2.2.31) can be written as

$$\|(\mathbf{A}_J \mathbf{v}_J)^T \widetilde{\Psi}_J^{\mathbb{I}}\|_{(H^t)'} \sim \|\mathbf{v}_J^T \Psi_J^{\mathbb{I}}\|_{H^t} \ .$$

Together with (2.2.12),(2.2.13) this yields

$$\begin{aligned} \|\mathbf{v}_J\|_{\ell_2(\mathbb{I}_J)} &\sim \|\mathbf{v}_J^T \mathbf{D}_J^{-t} \Psi_J^{\mathbb{I}}\|_{H^t} \\ &\sim \|(\langle \Psi_J^{\mathbb{I}}, A\Psi_J^{\mathbb{I}} \rangle \mathbf{D}_J^{-t} \mathbf{v}_J)^T \widetilde{\Psi}_J^{\mathbb{I}}\|_{(H^t)'} \\ &\sim \|(\mathbf{D}_J^{-t} \langle \Psi_J^{\mathbb{I}}, A\Psi_J^{\mathbb{I}} \rangle \mathbf{D}_J^{-t}) \mathbf{v}_J\|_{\ell_2(\mathbb{I}_J)} \\ &= \|\mathbf{A}_J \mathbf{v}_J\|_{\ell_2(\mathbb{I}_J)}, \end{aligned}$$

and in perfect analogy

$$\|\mathbf{v}_J\|_{\ell_2(\mathbb{I}_J)} \sim \|\mathbf{A}_J^{-1} \mathbf{v}_J\|_{\ell_2(\mathbb{I}_J)} \ .$$

Thus, by definition, we arrive at $\kappa_2(\mathbf{A}_J) = \mathcal{O}(1)$. ∎

In other words, Galerkin stability entails that $\mathbf{A}_J$ is still an **isomorphism** on $\ell_2(\mathbb{I}_J)$ **uniformly in** J.

Remark 2.29 *Galerkin stability is trivially satisfied if the operator A is given by* $\langle y, Av \rangle = a(y, v)$ *as in Definition 1.34 since then* $a(v, v) \sim \|v\|_{H^t}^2$ *holds. The operator* $\mathbf{A}_J$ *is then called* **stiffness matrix**.

The LBB-Condition

Galerkin stability is sufficient for the homogeneous version of the elliptic PDE (1.3.7). Later, additional stability conditions will be required during the discretization of elliptic boundary value problems as **saddle point problems**. These will be given by the Ladysenškaya-Babuška-Brezzi (**LBB**)-Condition. We will introduce and discuss the LBB-Condition in Section 3.

2.2.6 Riesz Stability Properties

As we will see in Section 5, theoretical equivalence relations have substantial impact on the computed solutions in applications. The computational problem arises from the constants inherent in every norm equivalence which we up to now gracefully ignored with our convenient short writing symbols like "$\sim$". To get a quantitative measurement of the condition of a **Riesz basis**, we must establish lower and upper bounds for the equivalence relation $(\mathcal{S})$(2.1.5). To this end, we define the **Riesz bounds** for a **Riesz basis** $\Phi \subset \mathcal{H}$,

$$\begin{aligned} c_\Phi &:= \sup \{c \,|\, c\|\mathbf{v}\|_{\ell_2} \leq \|\mathbf{v}^T \Phi\|_{\mathcal{H}}\}, \\ C_\Phi &:= \inf \{C \,|\, C\|\mathbf{v}\|_{\ell_2} \geq \|\mathbf{v}^T \Phi\|_{\mathcal{H}}\}, \end{aligned}$$

with which we can rewrite $(\mathcal{S})(2.1.5)$ to

$$c_\Phi \|\mathbf{v}\|_{\ell_2} \leq \|\mathbf{v}^T\Phi\|_{\mathcal{H}} \leq C_\Phi \|\mathbf{v}\|_{\ell_2}, \qquad \text{for all } \mathbf{v} \in \ell_2 \ . \tag{2.2.35}$$

If $\lambda_{\min}(\Phi_j, \Phi_j)$ resp. $\lambda_{\max}(\Phi_j, \Phi_j)$ denote the smallest resp. largest eigenvalue of the **Gramian matrix** $(\Phi_j, \Phi_j) := (\Phi_j, \Phi_j)_{\mathcal{H}}$, it is known (see [42]) that

$$c_{\Phi_j} = \sqrt{\lambda_{\min}(\Phi_j, \Phi_j)} \sim 1, \qquad C_{\Phi_j} = \sqrt{\lambda_{\max}(\Phi_j, \Phi_j)} \sim 1 \ . \tag{2.2.36}$$

If the **basis condition**

$$\kappa_{\Phi_j} := \left(\frac{C_{\Phi_j}}{c_{\Phi_j}}\right)^2 \tag{2.2.37}$$

is a large constat, Φ_j is said to be **poorly conditioned**. Unfortunately, the Gramian matrix is not explicitly computable when $\mathcal{H} = H^s$ for arbitrary values of s, and so the quality of the basis cannot be judged accurately. To improve the constants involved in (2.2.35) when evaluating the norm of any element $v \in \mathcal{H}$, we take on a different approach from [12] which builds on the **Riesz Representation Theorem** using **Riesz operators**.

2.2.7 Riesz Operators for H^s

We describe this for the general case $\mathcal{H} = H^{+s}, \mathcal{H}' = (H^{+s})'$ with wavelet bases $\Psi^s, \widetilde{\Psi}^s$.
We are now interested in constructing **Riesz maps** $R_{\mathcal{H}} : \mathcal{H} \to \mathcal{H}'$ defined by

$$\langle R_{\mathcal{H}} v, w\rangle := (v, w)_{\mathcal{H}}, \qquad \text{for all } v, w \in \mathcal{H} \ . \tag{2.2.38}$$

The **Riesz Representation Theorem** establishes for Riesz maps $\langle R_{\mathcal{H}} v, v\rangle_{\mathcal{H}'\times\mathcal{H}} \sim \|R_{\mathcal{H}} v\|_{\mathcal{H}'}\|v\|_{\mathcal{H}}$, from which it follows that Riesz operators are generally spectral equivalent to the identity, i.e.,

$$\|R_{\mathcal{H}}\|_{L(\mathcal{H};\mathcal{H}')} \sim 1 \ . \tag{2.2.39}$$

Observe that for any $v = \mathbf{v}^T\Psi^s \in H^s$ one has

$$\|v\|^2_{H^s} = \left(\mathbf{v}^T\Psi^s, \mathbf{v}^T\Psi^s\right)_{H^s} = \mathbf{v}^T(\Psi^s, \Psi^s)_{H^s}\mathbf{v} = \mathbf{v}^T\mathbf{M}_{H^s}\mathbf{v}, \tag{2.2.40}$$

where $\left(\Psi^{I\!I}, \Psi^{I\!I}\right)_{H^s} = \mathbf{M}_{H^s}$ is the **Gramian matrix** with respect to the H^s-inner product. Since

$$\|\mathbf{M}^{1/2}_{H^s}\mathbf{v}\|^2_{\ell_2} = \mathbf{v}^T\mathbf{M}_{H^s}\mathbf{v} \equiv \|\mathbf{R}^{1/2}_{H^s}\mathbf{v}\|^2_{\ell_2},$$

we conclude that the exact discretization of the Riesz map R_{H^s} would be $\mathbf{R}_{H^s} = \mathbf{M}_{H^s}$.

Remark 2.30 *(i) Recall that the exact* **Gramian matrix** $\mathbf{M}_{H^s}$ *is inaccessible for* $s \notin \mathbb{Z}$.

(ii) Note that $\mathbf{R}_{\mathcal{H}}$ *is symmetric positive definite for any space* $\mathcal{H}$ *and, thus,* $\mathbf{R}^{1/2}_{\mathcal{H}}$ *is always well defined.*

For $s \in \mathbb{Z}$, we can represent $\mathbf{R}_{\mathcal{H}}$ exactly. For example, in the cases $s \in \{0, 1\}$, it follows with the definitions of the L_2-product (1.2.1) and the H^1-product (1.2.3) that we have

$$\mathbf{R}_{L_2} := \mathbf{M}_{L_2} \qquad \text{and} \qquad \mathbf{R}_{H^1} := \mathbf{D}^{-1}(\mathbf{S}_{H^1} + \mathbf{M}_{L_2})\mathbf{D}^{-1}, \tag{2.2.41}$$

where $\mathbf{S}_{H^1}$ is the **Laplace matrix** and $\mathbf{M}_{L_2}$ the **mass matrix**, for further details see [11]. With these matrices the norm equivalences $\|v\|_{L_2} \sim \|\mathbf{v}\|_{\ell_2}$ and $\|w\|_{H^1} \sim \|\mathbf{w}\|_{\ell_2}$ can be obtained with constants equal to 1 and not of order $\mathcal{O}(1)$.

Riesz Operators based on Scaling

Because we will have to deal with fractional Sobolev spaces in Section 4, we use another construction based on diagonal scaling. Since the Sobolev spaces are nested as in (1.2.8), there is an inclusion operator $\iota : H^s \to (H^s)'$. In wavelet coordinates, this change of bases is merely a diagonal scaling when the wavelet base of the target space is the same as that of the initial space. This is not the case here, but the construction can nevertheless be carried out as follows.
By Theorem 2.17 can we interpret $\widehat{\mathbf{D}}^{+s} := \mathbf{D}_1^{+s}$ as a shifting operator in the Sobolev scale, i.e.,

$$\widehat{\mathbf{D}}^{+s} : H^{t+s} \to H^t, \qquad (\widehat{\mathbf{D}}^{+s})_{(j,j')(k,k')} = 2^{+js}\delta_{(j,j')}\delta_{(k,k')}, \tag{2.2.42}$$

so that

$$\widehat{R}_{H^s} := \widehat{\mathbf{D}}^{+2s} : H^s \to (H^s)' \tag{2.2.43}$$

can be used to shift elements of $\mathcal{H}$ into $\mathcal{H}'$. Using the formulation of Section 2.2.3, the standard wavelet representation $\widehat{\mathbf{R}}_{H^s}$ of $\widehat{R}_{H^s}$ is given by

$$\begin{aligned}
\widehat{\mathbf{R}}_{H^s} &= \Big\langle \Psi^s, \widehat{R}_{H^s}\Psi^s \Big\rangle_{H^s\times(H^s)'} \\
&= \mathbf{D}^{-s}\Big\langle \Psi^{I\!I}, \widehat{R}_{H^s}\Psi^{I\!I} \Big\rangle_{H^s\times(H^s)'}\mathbf{D}^{-s} \\
&= \mathbf{D}^{-s}\Big\langle \Psi^{I\!I}, \widehat{\mathbf{D}}^{+s}\widehat{\mathbf{D}}^{+s}\Psi^{I\!I} \Big\rangle_{H^s\times(H^s)'}\mathbf{D}^{-s} \\
&= \mathbf{D}^{-s}\widehat{\mathbf{D}}^{+s}\big(\Psi^{I\!I}, \Psi^{I\!I}\big)_{L_2}\widehat{\mathbf{D}}^{+s}\mathbf{D}^{-s} \ .
\end{aligned} \tag{2.2.44}$$

Remark 2.31 *Note that the Riesz operator scaling $\widehat{\mathbf{D}}^{+s}$ and the diagonal scaling $\mathbf{D}^{-s}$ could cancel each other out by the use of $\mathbf{D}^{-s} = \mathbf{D}_1^{-s}$. However, this does not occur if we choose $\mathbf{D}^{-s}$ differently, e.g., see Remark 3.19.*

Remark 2.32 *The choice $\widehat{\mathbf{D}}^{+s} = \mathbf{D}_1^{+s}$ is not without reason. By changing $\widehat{\mathbf{D}}$, we change the way we weight functions in the norm equivalences. Although any matrix satisfying the norm equivalence could potentially be used as $\widehat{\mathbf{D}}$, the actual choice has an impact on the quality of numerical experiments, see Section 5.2.3.*

The inverse operator $\widehat{R}_{H^s}^{-1} : H^{-s} \to H^{+s}$ is trivially given by $\widehat{\mathbf{D}}^{-2s}$ and can be represented as the inverse of (2.2.44),

$$\begin{aligned}
\widehat{\mathbf{R}}_{H^s}^{-1} &= \mathbf{D}^{+s}\widehat{\mathbf{D}}^{-s}\big(\Psi^{I\!I}, \Psi^{I\!I}\big)_{L_2}^{-1}\widehat{\mathbf{D}}^{-s}\mathbf{D}^{+s} \\
&= \mathbf{D}^{+s}\widehat{\mathbf{D}}^{-s}\Big(\widetilde{\Psi}^{I\!I}, \widetilde{\Psi}^{I\!I}\Big)_{L_2}\widehat{\mathbf{D}}^{-s}\mathbf{D}^{+s},
\end{aligned} \tag{2.2.45}$$

where we used identity (2.1.60) in the last step. The case $s = 0$ thus again becomes

$$\begin{array}{rcccl}
\widehat{\mathbf{R}}_{L_2} & = & \mathbf{M}_{L_2} & = & \big(\Psi^{I\!I}, \Psi^{I\!I}\big)_{L_2}, \\
\widehat{\mathbf{R}}_{L_2}^{-1} & = & \widetilde{\mathbf{M}}_{L_2} & = & \Big(\widetilde{\Psi}^{I\!I}, \widetilde{\Psi}^{I\!I}\Big)_{L_2} .
\end{array} \tag{2.2.46}$$

Numerical tests, see Section 5.2.3, show that using $\widehat{\mathbf{R}}_{H^s}$ gives better results than using no Riesz operator at all. This observation can be justified by the following deliberation: Fix $c_0, C_0 < \infty$ as the **Riesz bounds** of (2.2.35) for $\mathcal{H} = L_2$. Then it follows

$$\kappa_{L_2} := \left(\frac{C_0}{c_0}\right)^2 = \frac{\lambda_{\max}(\mathbf{M}_{L_2})}{\lambda_{\min}(\mathbf{M}_{L_2})} = \kappa_2(\mathbf{M}_{L_2}) \sim 1, \tag{2.2.47}$$

and for $\mathcal{H} = H^s$ with constants $c_s, C_s < \infty$

$$\kappa_{H^s} := \left(\frac{C_s}{c_s}\right)^2 = \frac{\lambda_{\max}(\mathbf{M}_{H^s})}{\lambda_{\min}(\mathbf{M}_{H^s})} = \kappa_2(\mathbf{M}_{H^s}) \sim 1 \ . \tag{2.2.48}$$

These can easily be combined to give error bounds for the H^s-norm with respect to the ℓ_2-norm of $\|\mathbf{M}_{L_2}^{1/2}\mathbf{v}\|_{\ell_2}$, which is not equal to $\|v\|_{L_2}$, since the coefficient vector $\mathbf{v}$ is scaled by $\mathbf{D}^{+s}$. It holds

$$\frac{c_s}{C_0}\|\mathbf{M}_{L_2}^{1/2}\mathbf{v}\|_{\ell_2} \leq \|\mathbf{M}_{H^s}^{1/2}\mathbf{v}\|_{\ell_2} \leq \frac{C_s}{c_0}\|\mathbf{M}_{L_2}^{1/2}\mathbf{v}\|_{\ell_2} \ . \tag{2.2.49}$$

These estimates are not sharp, e.g., for $s = 0$ we should have equality but only obtain equivalence up to the value of κ_{L_2}, and we witness better results in practice than can be predicted here.

Lemma 2.33 *We have for every $v = \mathbf{v}^T\Psi^s = \langle v, \widetilde{\Psi}^s\rangle\Psi^s \in H^s$ the following chain of equivalences:*

$$\|v\|_{H^s} = \|\mathbf{R}_{H^s}^{1/2}\mathbf{v}\|_{\ell_2} \sim \|\widehat{\mathbf{R}}_{H^s}^{1/2}\mathbf{v}\|_{\ell_2} \sim \|\mathbf{R}_{L_2}^{1/2}\mathbf{v}\|_{\ell_2} \sim \|\mathbf{v}\|_{\ell_2}, \qquad s \in (-\tilde{\sigma}, \sigma) \ . \tag{2.2.50}$$

In other words, every one of the operators $\mathbf{R}_{H^s}, \widehat{\mathbf{R}}_{H^s}, \mathbf{R}_{L_2}$ and $\mathbf{I}$ can be used as a Riesz operator for H^s.

Riesz Operators based on Interpolation

The construction of the Riesz operator outlined here was introduced in [11]. We only quote the results and refer to that work for details.
Since the exact Riesz operator for L_2 and H^1 are known, these can be used to construct new Riesz operators by interpolating linearly between them.

Theorem 2.34 *For $s \in [0,1]$, the norm defined by*

$$v = \mathbf{v}^T\Psi \in H^s, \qquad \|v\|_s^2 := (1-s)\,\mathbf{v}^T\mathbf{D}^{+s}\mathbf{R}_{L_2}\mathbf{D}^{+s}\mathbf{v} + s\,\mathbf{v}^T\mathbf{D}^{+s}\mathbf{R}_{H^1}\mathbf{D}^{+s}\mathbf{v}, \tag{2.2.51}$$

or alternatively written in the scaled wavelet basis as

$$v = \mathbf{v}^T\Psi^s \in H^s, \qquad \|v\|_s^2 = \mathbf{v}^T\left((1-s)\mathbf{M}_{L_2} + s\mathbf{D}^{-1}(\mathbf{S}_{H^1} + \mathbf{M}_{L_2})\mathbf{D}^{-1}\right)\mathbf{v}, \tag{2.2.52}$$

is equal to the standard Sobolev norms for $s \in \{0,1\}$ and equivalent for $s \in (0,1)$. It can be computed in linear time.

In the following, we denote this Riesz operator as

$$\widetilde{\mathbf{R}}_{H^s} := (1-s)\,\mathbf{R}_{L_2} + s\,\mathbf{R}_{H^1}, \qquad 0 \leq s \leq 1 \ . \tag{2.2.53}$$

Just as the summands, the Riesz operator $\widetilde{\mathbf{R}}_{H^s}$ is spectrally equivalent to the identity matrix and thus uniformly well-conditioned.

Remark 2.35 *This construction can be extended for all $s \in \mathbb{R}$ with exact interpolation for all integer $s \in \mathbb{Z}$, see [11].*

These Riesz operators will be used in Section 4 to improve the constants in the above norm equivalences. This will somewhat improve the discrepancy between the original analytical problem formulation and the discretized wavelet formulation.

2.3 Wavelets on the Interval

In the preceding sections we described general properties of wavelet spaces. This theoretical work will now be applied to a practical construction of B-spline based (primal) wavelets. The construction process presented here was first proposed in [19]. Actual implementation results can be found in [30].
The objective is the construction of wavelets adapted to the interval $\mathrm{I} = (0,1)$ endowed with the following properties:

($\mathcal{S}$)(2.1.5) The wavelets form a Riesz-Basis for $L_2(\mathrm{I})$, and norm equivalences with respect to Sobolev spaces $H^s(\mathrm{I})$ for a certain range of the smoothness parameter s hold.

($\mathcal{R}$)(2.1.6) The wavelets are refinable with masks of uniformly bounded length.

($\mathcal{L}$)(2.1.8) All generators and wavelets on the primal and dual side have compact support.

($\mathcal{B}$)(2.1.52) The primal and dual wavelets form a biorthogonal pair.

($\mathcal{P}$)(2.2.3) The primal MRA consists of spline spaces of order up to $d-1$, and thus has polynomial exactness of order d.

($\widetilde{\mathcal{P}}$)(2.2.4) The dual MRA has polynomial exactness of order $\tilde{d}-1$.

($\mathcal{V}$)(2.2.5) As a consequence of ($\widetilde{\mathcal{P}}$), the wavelets have $\tilde{d}$ vanishing moments.

The constants d and $\tilde{d}$ are preassigned such that they satisfy (2.2.8) before the construction process.

2.3.1 The Construction

We outline the construction process for B-splines with general order such that the structure of the construction becomes more apparent. There are several parameters which can be chosen arbitrarily during the construction process. Later we will investigate the impact of these parameters, in particular, in Lemma 2.41, in applications. Full details about the construction process can be found in [30].
Later we will employ a modified construction with orders $d = 2, \tilde{d} = 4$ in Section 5 which is sufficient for the control problems treated in this thesis.

B-Spline Wavelets

We begin the construction by setting up a **Riesz basis** for $L_2(\mathbb{R})$, which is then restricted to the interval $(0,1)$. Let $\phi^d(x)$ be the **cardinal B-spline** of order $d \in \mathbb{N}$ (see Appendix B for the definition). These B-splines are symmetric and centered around $\mu(d) := d \mod 2$. We will use these B-splines as **primal generators**, since they are easy to set up, have finite support,

$$\operatorname{supp}\phi^d = [\ell_1, \ell_2], \qquad \ell_1 := -\lfloor \frac{d}{2} \rfloor,\ \ell_2 := \lceil \frac{d}{2} \rceil, \tag{2.3.1}$$

and are known to be **refinable** with mask $\mathbf{a}^d = \{a_k^d\} \in \ell_2$,

$$a_k^d := 2^{1-d} \begin{pmatrix} d \\ k + \lfloor \frac{d}{2} \rfloor \end{pmatrix}, \qquad k = \ell_1, \ldots, \ell_2\ . \tag{2.3.2}$$

These generators also offer the advantage of being scaled correctly in the sense of the Riesz basis property in Definition 2.1. The question whether a refinable **dual basis** exists for any $d \in \mathbb{N}$ has been proved in [19] and we cite this result from [30].

Theorem 2.36 *For each d and any $\tilde{d} \geq d$ with $d + \tilde{d}$ even, there exists a function $\phi^{d,\tilde{d}} \in L_2(\mathbb{R})$ such that*

(i) $\phi^{d,\tilde{d}}$ has compact support, e.g.

$$\operatorname{supp}\phi^{d,\tilde{d}} = [\tilde{\ell}_1, \tilde{\ell}_2], \qquad \tilde{\ell}_1 := \ell_1 - \tilde{d} + 1,\ \tilde{\ell}_2 := \ell_2 + \tilde{d} - 1$$

(ii) $\phi^{d,\tilde{d}}$ *is also centered around* $\mu(d)$.

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

(iv) $\phi^{d,\tilde{d}}$ *is exact of order* $\tilde{d}$.

(v) ϕ^d *and* $\phi^{d,\tilde{d}}$ *form a* **dual pair**, *i.e.,*

$$\Big(\phi^d, \phi^{d,\tilde{d}}(\cdot - k)\Big)_{L_2(\mathbb{R})} = \delta_{(0,k)}, \qquad k \in \mathbb{Z} \ .$$

(vi) The regularity of $\phi^{d,\tilde{d}}$ *increases proportionally with* $\tilde{d}$.

Thus, for d,$\tilde{d}$ fixed, we write $\phi := \phi^d$, $\widetilde{\phi} := \phi^{d,\tilde{d}}$ and define the generator bases $\Phi_j = \{\phi_{j,k}\}$, $\widetilde{\Phi}_j = \{\widetilde{\phi}_{j,k}\}$ according to (2.1.9). We also deduce that $\mathcal{S} := \{S(\Phi_j), j \geq j_0\}$ is a **multiresolution analysis** of L_2.
The primal mother wavelet can now be constructed from the primal basis functions and the dual mask $\tilde{\mathbf{a}}$ as (see [19])

$$\psi(x) := \sum_{k\in\mathbb{Z}} b_k \phi(2x-k), \qquad b_k := (-1)^k \tilde{a}_{1-k}^d \ . \tag{2.3.3}$$

The primal wavelets are then derived as depicted in (2.1.14). Thus, the elements of the refinement relation (2.1.16) are given by the b_k above. The dual wavelets are defined accordingly.

Corollary 2.37 *The bases* $\{\Phi_{j_0} \cup \bigcup_{j\geq j_0} \Psi_j\}$, $\{\widetilde{\Phi}_{j_0} \cup \bigcup_{j\geq j_0} \widetilde{\Psi}_j\}$ *are indeed* **biorthogonal** *wavelet bases in the multiresolution framework of Section 2.1.*

Adaptation to the Interval

With proper generators and wavelets at our disposal, we can not simply restrict the collections $\Phi_j, \widetilde{\Phi}_j$ to the interval $(0,1)$ in the hope of constructing adapted wavelets, as this would violate **biorthogonality**. Ruling out any $\phi_{j,k}, \widetilde{\phi}_{j,k}$ whose support is not fully contained in $(0,1)$ would lead to primal and dual bases of different cardinality and thus also break biorthogonality. In addition, the approximation property (2.2.10) would not longer hold near the ends of the interval. Last but not least, we also need to take boundary conditions into account.
The actual adaptation is done in three steps. First, we take out every function whose support is not fully contained within $(0,1)$. Then we insert new basis functions at the boundaries to compensate for the reduction. These shall incorporate the boundary conditions and have properties $(\mathcal{V})$ and $(\widetilde{\mathcal{V}})$, thus preserving the **vanishing moments** and **polynomial reproduction** orders d, $\tilde{d}$ for the new basis. Lastly, the new basis functions are again biorthogonalized by a local basis transformation regarding only functions near the boundary.
Fixing a parameter $\ell \geq -\ell_1$, let Φ_j^0 be the family of basis functions inside $\mathrm{I} = (0,1)$ with distance $2^{-j}|\ell - (-\ell_1)|$ from the edges, i.e.,

$$\Phi_j^0 := \{\phi_{j,k} \in \Phi_j \mid \operatorname{supp}\phi_{j,k} \subseteq 2^{-j}[|\ell+\ell_1|, 1-2^{-j}|\ell+\ell_1|] \subseteq (0,1)\} \ . \tag{2.3.4}$$

We infer that the cardinality of this set is $2^j - 2\,\ell - \mu(d)$ and associate it with the index set

$$\Delta_j^0 := \{\ell, \ldots, 2^j - \ell - \mu(d)\} \ . \tag{2.3.5}$$

The new set of functions at the left boundary with its index set is designated as

$$\Phi_j^L := \{\phi_{j,k}^L \mid k \in \Delta_j^L\}, \qquad \Delta_j^L := \{\ell - d, \ldots, \ell - 1\}, \tag{2.3.6}$$

where each element $\phi_{j,k}^L$ is given by (see [30]),

$$\phi_{j,\ell-d+r}^L := \sum_{m=-\ell_2+1}^{\ell-1} \Big((\cdot)^r, \widetilde{\phi}(\cdot - m)\Big)_{L_2(\mathbb{R})} \phi_{j,m}\Big|_{(0,1)}, \qquad r = 0, \ldots, d-1, \tag{2.3.7}$$

thus incorporating the space Π_d into $\operatorname{span}\{\phi_{j,k}^L\}$ while preserving the regularity order γ in the new function set $\phi_{j,k}^L$.

Remark 2.38 *Homogeneous boundary conditions on either side can be achieved by demanding only* $r = 1, \dots, d-1$*, i.e., deleting the monomials of order 0 from the appropriate index sets. Equal cardinality on the primal and dual side can always be achieved by a suitable modification of* ℓ *and* $\tilde{\ell}$*.*

The functions (2.3.7) are known to be refinable with a finite mask, see [30], a fact which we save to prove in full detail here. By permutation and renumbering, we can employ these refinement coefficients and functions at the right hand border of $(0,1)$,

$$\Phi_j^R := \{\phi_{j,k}^R \,|\, k \in \Delta_j^R\}, \qquad \Delta_j^R := \{2^j - \ell - \mu(d) + 1, \dots, 2^j - \ell - \mu(d) + d\}, \tag{2.3.8}$$

with

$$\phi_{j,2^j-\ell-\mu(d)-d-r}^R(1-x) := \phi_{j,\ell-d+r}^L(x), \qquad r = 0, \dots, d-1, \tag{2.3.9}$$

thus achieving symmetry. The next result is also taken from [30].

Proposition 2.39 *By defining the family of functions*

$$\Phi_j' := \Phi_j^L \cup \Phi_j^0 \cup \Phi_j^R, \qquad \Delta_j := \Delta_j^L \cup \Delta_j^0 \cup \Delta_j^R, \tag{2.3.10}$$

the spaces $S_j := S(\Phi_j'), j \geq j_0$*, are nested and exact of order* d*, i.e.,*

$$S_j \subset S_{j+1}, \qquad \Pi_d((0,1)) \subset S_j, \qquad j \geq j_0\ . \tag{2.3.11}$$

Moreover, the same regularity order γ *is maintained as from* $\Omega = I\!R$*.*

This construction can be carried to the dual side accordingly with parameter $\tilde{\ell} \geq -\tilde{\ell}_1$, and Proposition 2.39 holds true for $\widetilde{\Phi}_j'$ and $\widetilde{\Delta}_j'$ defined analogously. The number of boundary functions to add on the primal side is always d and $\tilde{d}$ and the dual side, although $\tilde{d} \geq d$ and $\Pi_{\tilde{d}} \subset \widetilde{\Phi}_j'$ needs to be achieved on the dual side. Hence, the cardinality of both index sets $\Delta_j, \widetilde{\Delta}_j$ can be made equal by choosing

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

for the dual side. This is necessary since equal cardinality is a prerequisite of biorthogonality.
To reestablish biorthogonality, a local basis transformation is applied on the left and right boundaries. To this end, we enlarge the primal index set $\Delta_j^X, X \in \{L, R\}$, by the next $\tilde{d} - d \geq 0$ adjacent functions of Φ_j^0 and get

$$\#\Delta_j^X = \#\widetilde{\Delta}_j^X, \qquad X \in \{L, R\}\ . \tag{2.3.12}$$

We can now define the square matrices

$$\Gamma_j^L := \left(\left(\phi_{j,k}^L, \widetilde{\phi}_{j,k'}^L \right)_{(0,1)} \right)_{k,k' \in \Delta_j^L}, \qquad \Gamma_j^R := \left(\left(\phi_{j,k}^R, \widetilde{\phi}_{j,k'}^R \right)_{(0,1)} \right)_{k,k' \in \Delta_j^R}\ . \tag{2.3.13}$$

The next theorem was proved in [30].

Theorem 2.40 *The matrices* Γ_j^L, Γ_j^R *are nonsingular, symmetric and independent of j, i.e.,*

$$\Gamma := \Gamma^L = \Gamma_j^L, \qquad \Gamma^{\updownarrow} = \Gamma^R = \Gamma_j^R,$$

where $(C^{\updownarrow})_{i,j} := (C)_{n-i,n-j}$ *for* $n \times n$*-matrices.*

By defining the bases

$$\Phi_j := \Phi_j', \qquad \widetilde{\Phi}_j := \Gamma_j^{-T} \widetilde{\Phi}_j' := \begin{pmatrix} \Gamma^L & & \\ & \mathbf{I}_{\#\widetilde{\Delta}_j^0} & \\ & & \Gamma^R \end{pmatrix}^{-T} \widetilde{\Phi}_j' \tag{2.3.14}$$

we achieve $(\mathcal{B})$ biorthogonal, $(\mathcal{S})$ uniformly stable and $(\mathcal{L})$ local single-scale bases with the desired $(\mathcal{V})$, $(\widetilde{\mathcal{V}})$ vanishing moment conditions. Moreover, these bases are $(\mathcal{R})$ refinable with matrices

$$\mathbf{M}_{j,0} := \mathbf{M}_{j,0}', \qquad \widetilde{\mathbf{M}}_{j,0} := \Gamma_{j+1}\, \widetilde{\mathbf{M}}_{j,0}'\, \Gamma_j^{-1}, \tag{2.3.15}$$

where $\mathbf{M}_{j,0}'$, $\widetilde{\mathbf{M}}_{j,0}'$ are the refinement matrices of the bases Φ_j' and $\widetilde{\Phi}_j'$.

Stable Completions

All which remains to be done is to construct a **stable completion** suitable for our needs. We will first outline the construction of an **initial** stable completion, and then apply Theorem 2.14, thereby finishing the construction process.
We assume d to be even, because this is the case that we will be using in Section 2.3.2. For odd d a similar reasoning can be used with some obvious modifications. We can thus infer $\mu(d) \equiv 0$ from the previous section.
To begin, we first establish the block matrix view of the refinement matrix $\mathbf{M}_{j,0}$,

$$\mathbf{M}_{j,0} = \begin{array}{c} \mathbf{M}_L \\ \\ \mathbf{Z}_j \\ \\ \mathbf{M}_R \end{array} \tag{2.3.16}$$

where the inner block $\mathbf{Z}_j$ should contain the repeated columns of the mask vector $\mathbf{a}$ of (2.3.2). The border blocks $\mathbf{M}_L$ and $\mathbf{M}_R$ are independent of j and contain the refinement relations for the new boundary functions Φ_j^L and Φ_j^R. The boundary symmetry (2.3.9) inherited in the construction process assures $\mathbf{M}_L = \mathbf{M}_R^{\updownarrow}$. We have $\mathbf{Z}_j \in I\!R^{p(j)\times q(j)}$ with

$$\begin{aligned} p(j) &:= 2^j - 2\,\ell + 1, \\ q(j) &:= 2^{j+1} - 4\,\ell + d + 1\ . \end{aligned}$$

We can perform Gaussian elimination with some matrices $\mathbf{H}_j^{(i)} \in I\!R^{q(j)\times q(j)}$ on $\mathbf{Z}_j^{(0)} := \mathbf{Z}$, which is possible without pivoting because of the B-spline construction. After d steps, we have

$$(\mathbf{Z}_j^{(d)})_{m,k} = b\,\delta_{(m-2k,0)},$$

in other words, the matrix contains only one constant non-zero entry per column. Introducing the two new matrices

$$\mathbf{U}_j := b^{-2}\left(\mathbf{Z}_j^{(d)}\right)^T, \qquad (\mathbf{F}_j)_{m,k} := \delta_{(m-2k,-1)},$$

and we can conclude

$$\mathbf{U}_j\mathbf{Z}_j^{(d)} = \mathbf{I}_{p(j)}, \qquad \mathbf{U}_j\mathbf{F}_j = 0\ .$$

These matrices $\mathbf{Z}_j^{(i)}, \mathbf{U}_j^T, \mathbf{F}_j$ are then embedded into matrices

$$\mathbf{Z}_j^{(i)} \to \hat{\mathbf{Z}}_j^{(i)}, \qquad \mathbf{U}_j^T \to \hat{\mathbf{U}}_j^T, \qquad \mathbf{F}_j \to \hat{\mathbf{F}}_j, \tag{2.3.17}$$

according to the following scheme:

$$\left.\begin{array}{c} \hat{\mathbf{Z}}_j^{(i)} \\ \hat{\mathbf{U}}_j^T \end{array}\right\} := \begin{array}{c} \mathbf{I}_d \\ \\ \left\{\begin{array}{c} \mathbf{Z}_j^{(i)} \\ \mathbf{U}_j^T \end{array}\right. \\ \\ \mathbf{I}_d \end{array} \qquad \hat{\mathbf{F}}_j := \begin{array}{c} \mathbf{I}_{\ell+\mu(d)-1} \\ \\ \mathbf{F}_j \\ \\ \mathbf{I}_\ell \end{array} \tag{2.3.18}$$

The matrices $\hat{\mathbf{Z}}_j^{(i)}$ and $\hat{\mathbf{U}}_j^T$ have the dimensions $\#\Delta_{j+1} \times \#\Delta_j$, $\hat{\mathbf{F}}_j$ is of size $\#\Delta_{j+1} \times \#\nabla_j$. We embed each $\mathbf{H}_j^{(i)}$ into a square $\#\Delta_{j+1} \times \#\Delta_{j+1}$-matrix $\hat{\mathbf{H}}_j^{(i)}$ in the same manner (see (2.3.21) below) and can thus write

$$\hat{\mathbf{Z}}_j^{(0)} = \hat{\mathbf{H}}_j^{-1} \hat{\mathbf{Z}}_j^{(d)}, \qquad \hat{\mathbf{H}}_j := \hat{\mathbf{H}}_j^{(d)} \cdots \hat{\mathbf{H}}_j^{(1)}, \tag{2.3.19}$$

With these definitions, the following is an extension of a result from [30] proposed in [11]:

Lemma 2.41 *For any $r \neq 0$, it holds that*

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

Lastly, there exists a square matrix $\hat{\mathbf{P}}_j \in I\!R^{(\#\Delta_{j+1})\times(\#\Delta_{j+1})}$ such that

$$\mathbf{M}_{j,0} = \hat{\mathbf{P}}_j \hat{\mathbf{Z}}_j^{(0)}, \tag{2.3.20}$$

and the layout of $\hat{\mathbf{P}}_j$ can be seen below:

$$\hat{\mathbf{H}}_j^{(i)} := \begin{array}{ccc} \mathbf{I}_{\ell+\ell_2} & & \\ & \mathbf{H}_j & \\ & & \mathbf{I}_{\ell+\ell_2} \end{array} \qquad \hat{\mathbf{P}}_j := \begin{array}{ccc} & & \mathbf{M}_L \\ & \mathbf{I}_{\#\Delta_{j+1}-2d} & \\ \mathbf{M}_R & & \end{array} \tag{2.3.21}$$

As the final result, the **initial stable completion** is given by

$$\check{\mathbf{M}}_j = (\mathbf{M}_{j,0}, \check{\mathbf{M}}_{j,1}) = \hat{\mathbf{P}}_j \hat{\mathbf{H}}_j^{-1} (\mathbf{Z}_j^{(d)}, r\hat{\mathbf{F}}_j), \tag{2.3.22}$$

with its inverse

$$\check{\mathbf{G}}_j = \begin{pmatrix} \check{\mathbf{G}}_{j,0} \\ \check{\mathbf{G}}_{j,1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{U}}_j \\ r^{-1}\hat{\mathbf{F}}_j^T \end{pmatrix} \hat{\mathbf{H}}_j \hat{\mathbf{P}}_j^{-1} \ . \tag{2.3.23}$$

We can now apply Theorem 2.14 with the matrices $\mathbf{M}_{j,0}$, $\widetilde{\mathbf{M}}_{j,0}$ of (2.3.15) and $\check{\mathbf{M}}_{j,1}$ of (2.3.22) and thus the construction of biorthogonal wavelet bases $\Psi, \widetilde{\Psi}$ on the interval $\mathrm{I} = (0,1)$ is complete.

2.3.2 The Case $d = 2$, $\tilde{d} = 4$, $j_0 = 3$

We will not specialize the entire construction process of Section 2.3.1. Rather we show the resulting bases and wavelets and their refinement matrices. These results were first established in [30] and further investigated in [11]. The original refinement matrices can also be downloaded from the web at [29]. A construction using only rational numbers which avoids further rounding errors has been carried out in [11].
By choosing $d = 2$, a primal function consists of piecewise constant and linear polynomials. The minimum level in this case is $j_0 = 3$, and we have

$$\#\Delta_j = 2^j + 1, \qquad \#\nabla_j = 2^j \ . \tag{2.3.24}$$

Let the **characteristic function** and the **hat function** be given as

$$\chi_{j,k}(x) := \begin{cases} 1 & 2^j x \in [k, k+1] \\ 0 & \text{otherwise} \end{cases}, \qquad \varphi_{j,k}(x) := \begin{cases} 2^j x - (k-1) & 2^j x \in [k-1, k) \\ -2^j x + (k+1) & 2^j x \in [k, k+1] \\ 0 & \text{otherwise} \end{cases},$$

and the primal **generators** on level $j \geq j_0$ are given by

$$\phi_{j,k} = 2^{j/2} \begin{cases} \chi_{j,k} + \varphi^-_{j,k+1} & k = 0 \\ \varphi_{j,k} & k = 1, \ldots, 2^j - 1 \\ \varphi^+_{j,k-2} + \chi_{j,k-1} & k = 2^j \end{cases} , \tag{2.3.25}$$

where $\varphi^+_{j,k}$ is the first part of the hat function (with positive slope) and $\varphi^-_{j,k}$ the second part (with negative slope). The **hat function** is known to be refinable with mask

$$\{\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}\}, \tag{2.3.26}$$

and it is easy to verify that the boundary generators are refinable with mask

$$\{\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}}\} . \tag{2.3.27}$$

The general structure of the refinement matrix $\mathbf{M}_{j_0,0}$ can be seen in Figure 2.2. Higher level matrices $\mathbf{M}_{j,0}, j > j_0$, are formed by inserting empty rows and columns in the middle of $\mathbf{M}_{j_0,0}$ and repeating the mask (2.3.26) in every column, offsetting by two on every turn. The sparsity of $\mathbf{M}_{j,0}$ is thus obvious: we have at most three non-zero values per column.
The wavelets are specified by the refinement matrix $\mathbf{M}_{j,1}$ which still has an open degree of freedom in its construction: the parameter r from Lemma 2.41 can be chosen arbitrarily. Although this parameter does not affect plots of the wavelets, it does affect computations by having an impact on the absolute value of the condition numbers of the finite discretized systems. There are two values of r that have proved to be of value in applications and we will denote them by

$$r_{DKU} := 1 , \qquad r_B := \sqrt{2} . \tag{2.3.28}$$

Setting $r = r_{DKU}$ leads to the refinement matrices of [29] and setting $r = r_B$ is the construction from [11]. The wavelet refinement matrix $\mathbf{M}_{j_0,1}$ can be seen in Figure 2.3. The matrices $\mathbf{M}_{j,1}$ are also constructed in a repetitive manner by repeating and offsetting any of the third to sixth row. The bandwidth is now $9 \approx 2\tilde{d}$, which is a direct consequence of (2.3.3) combined with (2.36(i),(iii)).
The dual wavelets and generators can only be plotted approximately, since they are not known explicitly. Recall that only their existence is assured by Theorem 2.36 which suffices for our purpose. However, by (2.1.58) and (2.1.60), we can project the space $S(\widetilde{\Psi}_j)$ onto the space $S(\Phi_{j+1})$, thus approximating the dual wavelets with piecewise linear functions. Obviously this method is inexact, but choosing $j \gg j_0$ and then plotting only dual functions on level $j = j_0$ will lead to a sufficiently accurate visualization. These pictures can be found in Figure 2.4 to Figure 2.6.
It is well known (see [32]), that piecewise linear and globally continuous wavelets are contained in the Sobolev space H^s up to $s < 3/2$, or, by Definition 2.16 we have $\gamma = 3/2$.
It was pointed out in [19] that in our case of $d = 2$, $\tilde{d} = 4$, the dual generators decay in the following fashion:

$$|\widetilde{\phi}(x)| \leq C(1 + |x|)^{-\alpha}, \qquad \alpha > 1.2777 .$$

With Definition 1.10 this directly translates to $\tilde{\gamma} > 1.2777$. Note that the case $d = 2, \tilde{d} = 2$ would only yield $\tilde{\gamma} > 0.6584$ (see [19]), which would not suffice for the applications considered in this thesis. Hence, our wavelets satisfy the norm equivalences of Theorem 2.17 for the range from $(H^1(\mathrm{I}))'$ to $H^1(\mathrm{I})$. Thus, we are in a position to apply the standard wavelet representation of Section 2.2.3 to any elliptic differential operator $A : H^1(\mathrm{I}) \to (H^1(\mathrm{I}))'$ of order 2.

Remark 2.42 *The above outlined construction is one possible method for the construction of boundary adapted spline based wavelets. Another approach is given by splines with coinciding nodes at the boundary. Such a construction was first proposed in [16] and further optimized by Miriam Primbs. In the experiments conducted later, we will also use the mask coefficients from [56] for comparison.*

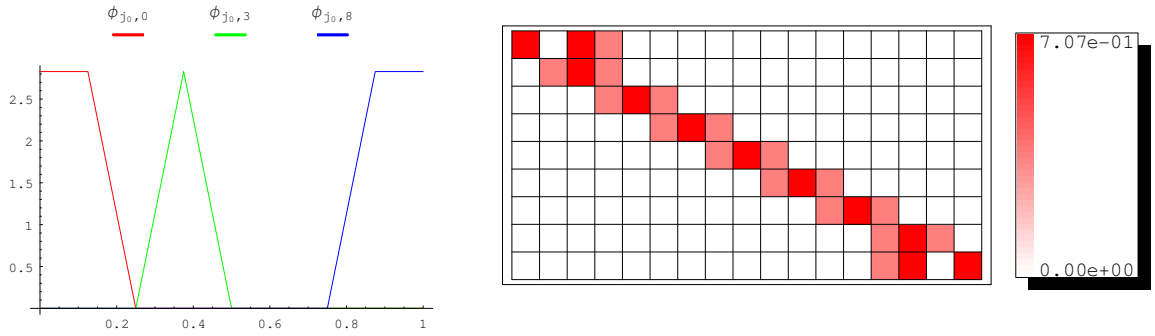


Figure 2.2: On the left, we show the 1st, 4th and 9th primal generators on level $j_0 = 3$ and on the right, a density plot of the refinement matrix $\mathbf{M}_{j_0,0}^T$. Brighter boxes indicate entries of smaller absolute value, white boxes represent 0. The first generator $\phi_{j_0,0}$ is drawn red, $\phi_{j_0,3}$ green and $\phi_{j_0,8}$ blue. The entries of $\mathbf{M}_{j_0,0}^T$ belonging to these three generators are the 1st, 4th and 9th row, respectively.

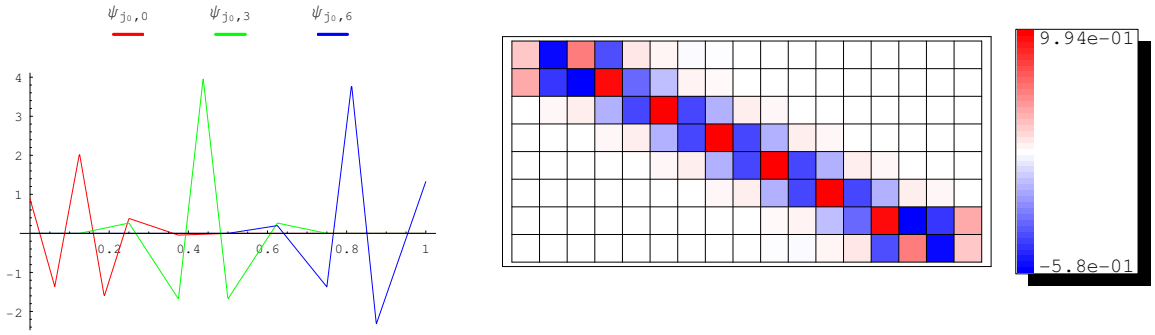


Figure 2.3: On the left, we show the 1st, 4th and 7th primal wavelet on level $j_0 = 3$ for $r_B = \sqrt{2}$ in colors red, green and blue, respectively. Only four wavelets, the 3rd to 6th wavelet, on this level are left untouched by the homogeneous boundary conditions. On the right, the refinement matrix $\mathbf{M}_{j_0,1}^T$ is plotted. The 1st, 4th and 7th row of $\mathbf{M}_{j_0,1}^T$ contains the masks of these wavelets.

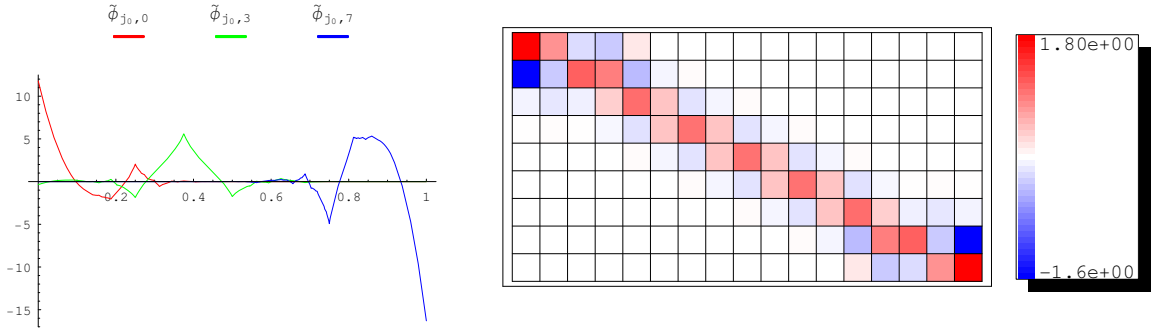


Figure 2.4: Three boundary adapted dual generators, number 1, 4 and 8, on level $j_0 = 3$, projected onto $S(\Phi_{10})$. This means every function is approximated by piecewise linear functions of support $\approx 2^{-10}$. On the right, we show the matrix $\mathbf{G}_{j_0,0}$. The refinement coefficients for these generators are given in the first, fourth and eighth row, respectively.

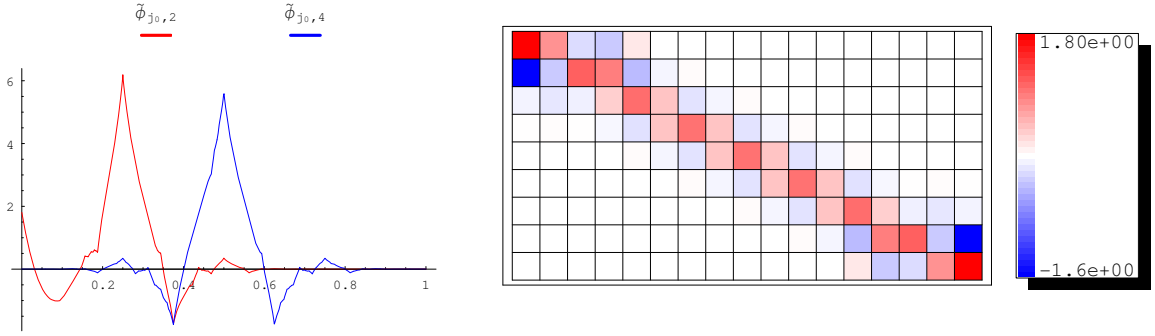


Figure 2.5: The last boundary adapted dual generator, number 3, and the only generator unchanged by the boundary conditions, approximated with $S(\Phi_{10})$. It is this generator's mask that is used in the repetitive structure of the matrices $\mathbf{G}_{j,0}$. The corresponding rows in the matrix $\mathbf{G}_{j_0,0}$ on the right are rows number 3 and 5.

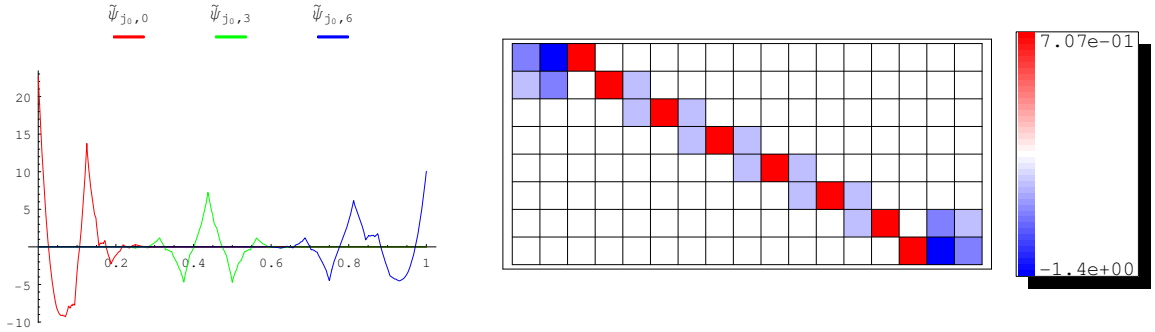


Figure 2.6: Here we show the dual wavelets on level $j_0 = 3$ projected onto $S(\Phi_{10})$. Since $d = 2$, we see a small mask length and small boundary blocks in $\mathbf{G}_{j,1}$ on the right. The wavelets incorporate the scaling of $r = r_B$.

2.3.3 Basis Transformations

Numerical studies show that condition numbers of operators obtained using wavelet discretizations are indeed uniformly bounded, if preconditioned correctly. The involved constants can nevertheless be quite high, and condition numbers of magnitude $10^2 - 10^3$ are seen often. We now show some approaches to improve the wavelet bases to achieve lower absolute values of condition numbers and thus faster program executions in applications.

Reducing Boundary Effects

The single scale basis (2.3.25) generally exhibits worse stability constants than the basis without the boundary adapted generator. This means that the absolute values of the condition numbers of differential operators in wavelet discretization of Section 2.3.2 are usually higher than those using free boundaries on the interval. A common approach to remedy the situation is the application of a basis transform, i.e.,

$$\Phi'_j := \mathbf{C}_j \Phi_j, \tag{2.3.29}$$

which acts local in the sense that $\mathbf{C}_j$ only affects the boundary blocks, i.e.,

$$\mathbf{C}_j := \begin{pmatrix} C & & \\ & \mathbf{I}_{\#\Delta_j - 2m} & \\ & & C^{\updownarrow} \end{pmatrix} \in I\!R^{(\#\Delta_j)\times(\#\Delta_j)}, \tag{2.3.30}$$

and the matrix $C \in I\!R^{m\times m}$ is independent of j. Recall $(C^{\updownarrow})_{m-i,m-j} := (C)_{i,j}$. Obviously, the modified **single-scale basis** implies new refinement matrices $\mathbf{M}'_{j,0}, \mathbf{M}'_{j,1}$:

$$\mathbf{M}'_{j,0} = \mathbf{C}_{j+1}^{-T} \mathbf{M}_{j,0} \mathbf{C}_j^T, \tag{2.3.31}$$

$$\mathbf{M}'_{j,1} = \mathbf{C}_{j+1}^{-T} \mathbf{M}_{j,1}\ . \tag{2.3.32}$$

The altered multiscale transformation $\mathbf{T}'_J$ can easily be shown to be of the form

$$\mathbf{T}'_J = \mathbf{C}_J^{-T} \mathbf{T}_J \begin{pmatrix} \mathbf{C}_{j_0}^T & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix}\ . \tag{2.3.33}$$

Let $\mathcal{L} : \mathcal{H} \to \mathcal{H}'$ be any operator with bases Φ_J and $\Psi_{(J)}$ in $\mathcal{H}$, e.g., $\mathcal{L}$ is the differential operator from Section 1.3.2. The discretized operator $\mathbf{L}$ in standard form (2.2.17) with respect to the wavelet basis $\Psi'_{(J)} = {\mathbf{T}'_J}^T \Phi'_J$ has the representation

$$\mathbf{L}_{\Psi'_{(J)}} = \begin{pmatrix} \mathbf{C}_{j_0} & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix} \mathbf{L}_{\Psi_{(J)}} \begin{pmatrix} \mathbf{C}_{j_0}^T & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix}, \tag{2.3.34}$$

thus, the new operator is obtained from the standard operator matrix by the application of the transformation $\mathbf{C}_{j_0}$ on the coarsest level.
A suitable choice for the setup of $\mathbf{C}_{j_0}$ can be constructed as follows: We take an upper block

$$\left(\mathbf{L}_{\Psi_{(j_0)}}\right)_{i,j=1,\ldots,m} \in I\!R^{m\times m},$$

with $m \le \lfloor \#\Delta_{j_0}/2 \rfloor$, thereby not changing all of the basis function of the generator basis. We compute the singular value decomposition of this block, i.e.,

$$\left(\mathbf{L}_{\Psi_{(j_0)}}\right)_{i,j=1,\ldots,m} = \mathbf{U}\mathbf{S}\mathbf{U}^T := \mathbf{U} \begin{pmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_m \end{pmatrix} \mathbf{U}^T, \tag{2.3.35}$$

with an orthogonal matrix $\mathbf{U} \in O(m)$, and we set for $q > 0$,

$$C := \sqrt{q} \begin{pmatrix} 1/\sqrt{s_1} & & & \\ & 1/\sqrt{s_2} & & \\ & & \ddots & \\ & & & 1/\sqrt{s_m} \end{pmatrix} \mathbf{U}^T\ . \tag{2.3.36}$$

Using this matrix as outlined above yields

$$\mathbf{L}_{\Psi'_{(j_0)}} = \mathbf{C}_{j_0}\mathbf{L}_{\Psi_{(j_0)}}\mathbf{C}_{j_0}^T = \left(\begin{array}{c|c|c} q\,\mathbf{I}_c & * & \\ \hline * & \mathbf{L}_\square & * \\ \hline & * & q\,\mathbf{I}_c \end{array}\right) . \tag{2.3.37}$$

The middle square block $\mathbf{L}_\square \in I\!R^{(\#\Delta_{j+1}-2m)\times(\#\Delta_{j+1}-2m)}$ of $\mathbf{L}_{\Psi_{(j_0\prime)}}$ consists of entries that remain unchanged with respect to the unmodified wavelet basis $\Psi_{(J)}$. The blocks marked with asterisks contain new non-zero entries. This basis transformation is particularly cheap in terms of complexity, since it is only used on the coarsest level. The impact of these blocks on the condition number can be influenced by the parameter q.

Remark 2.43 *A Cholesky decomposition can also be used instead of the singular value decomposition. In this case the resulting matrix* $\mathbf{L}_{\Psi'_{(j_0)}}$ *has fewer non-zero values, but no further decrease of the condition number is achieved.*

Operator Adaptation to Preconditioning

We now introduce a basis transformation specifically designed for lowering the absolute values of the condition number of the **stiffness matrix**

$$\mathbf{A}_J := (a(\psi_\lambda, \psi_{\lambda'}))_{\lambda,\lambda'\in I\!I_J} , \tag{2.3.38}$$

with the **bilinear form** defined in (1.3.24). The condition number of this positive definite symmetric matrix in wavelet discretization depends on the properties of the wavelets as well as on the generator basis. Of course, the condition $\kappa(\mathbf{A}_J)$ can never be smaller than $\kappa(\mathbf{A}_{j_0})$. Therefore, we seek a generator basis adapted to the operator to minimize the absolute value of its condition number.
We make use of an **orthogonal** transformation matrix $\mathrm{O} \in O(\#\Delta_{j_0})$ to create a new, albeit completely equivalent, generator basis for the coarsest level j_0:

$$\Phi'_{j_0} := \mathrm{O}^T\,\Phi_{j_0}, \tag{2.3.39}$$

while leaving the higher level generator bases unchanged by this transformation. The resulting MRA

$$\mathrm{O} : \Psi^{I\!I} = \{\Phi_{j_0}, \Psi_{j_0}, \Psi_{j_0+1}, \ldots\} \quad \longmapsto \quad \Psi^{I\!I'} := \{\Phi'_{j_0}, \Psi_{j_0}, \Psi_{j_0+1}, \ldots\} . \tag{2.3.40}$$

is also completely equivalent to the original MRA since the orthogonal transformation does not change the stability constants $(\mathcal{S})$(2.1.5). This change obviously requires an adaptation of the **two-scale relation** (2.1.11) for level j_0:

$$\Phi'_{j_0} = (\mathbf{M}'_{j_0,0})^T\Phi_{j_0+1} := (\mathbf{M}_{j_0,0}\mathrm{O})^T\Phi_{j_0+1} . \tag{2.3.41}$$

To still ensure biorthogonality, the dual MRA must also be adapted accordingly. From now on, the complete change of bases will always be accomplished by using $\mathbf{M}'_{j_0,0}$ instead of $\mathbf{M}_{j_0,0}$ in the course of the wavelet transform (2.1.31). This is no antagonism, because operators are assembled in the generator base Φ_J and then transform into the wavelet representation by the FWT. This can be implemented by applying O subsequent to the wavelet transform $\mathbf{T}_J$:

$$\mathbf{T}'_J := \mathbf{T}_J \left(\begin{array}{cc} \mathrm{O} & \\ & \mathbf{I}_{\#(\Delta_J\setminus\Delta_{j_0})} \end{array}\right) . \tag{2.3.42}$$

Theorem 2.13 is still valid as the relations (2.1.40) still hold with the same constants.

Lemma 2.44 *The form of the stiffness matrix* $\mathbf{A}'_J$ *in the wavelet base* $\Psi^{I\!I'}$ *is*

$$\mathbf{A}'_J = \left(\begin{array}{c|c} \mathrm{O}^T\mathbf{A}_{j_0}\mathrm{O} & (\mathbf{a}\mathrm{O})^T \\ \hline \mathbf{a}\mathrm{O} & \mathbf{A}_{J\setminus j_0} \end{array}\right) \tag{2.3.43}$$

The block $\mathbf{A}_{J\setminus j_0} \in I\!R^{\#(\Delta_J\setminus\Delta_{j_0})\times\#(\Delta_J\setminus\Delta_{j_0})}$ *remains unaffected.*

Proof: Assuming the following block structure of the symmetric matrix $\mathbf{A}_J$,

$$\mathbf{A}_J = \left(\begin{array}{c|c} \mathbf{A}_{j_0} & \mathbf{a}^T \\ \hline \mathbf{a} & \mathbf{A}_{J\setminus j_0} \end{array} \right),$$

with blocks $\mathbf{a} \in I\!R^{\#(\Delta_J \setminus \Delta_{j_0}) \times \#\Delta_{j_0}}$, we can directly deduce

$$\begin{aligned} \mathbf{A}'_J &= (\mathbf{T}'_J)^T \mathbf{A}_{\Phi_J} \mathbf{T}'_J \\ &= \begin{pmatrix} \mathrm{O}^T & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix} \mathbf{A}_J \begin{pmatrix} \mathrm{O} & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{O}^T & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix} \left(\begin{array}{c|c} \mathbf{A}_{j_0} & \mathbf{a}^T \\ \hline \mathbf{a} & \mathbf{A}_{J\setminus j_0} \end{array} \right) \begin{pmatrix} \mathrm{O} & \\ & \mathbf{I}_{\#(\Delta_J \setminus \Delta_{j_0})} \end{pmatrix} \\ &= \left(\begin{array}{c|c} \mathrm{O}^T \mathbf{A}_{j_0} \mathrm{O} & (\mathbf{a}\mathrm{O})^T \\ \hline \mathbf{a}\mathrm{O} & \mathbf{A}_{J\setminus j_0} \end{array} \right). \end{aligned}$$

■

The eigenvalues of $\mathbf{A}'_J$ are not different of $\mathbf{A}_J$, because applying an orthogonal matrix does not change these. The trick is to choose the orthogonal matrix $\mathrm{O} \in O(\#I\!I_{j_0})$ such that $D_{j_0} = \mathrm{O}^T \mathbf{A}_{j_0} \mathrm{O}$ is a **diagonal matrix**. This is possible because $\mathbf{A}_{j_0}$ is symmetric positive definite. Fixing this matrix O leads to

$$\mathbf{A}'_J = \left(\begin{array}{c|c} \diagdown & (\mathbf{a}\mathrm{O})^T \\ \hline \mathbf{a}\mathrm{O} & \mathbf{A}_{J\setminus j_0} \end{array} \right). \tag{2.3.44}$$

The setup allows for an improved optimal preconditioner for this operator, since any diagonal matrix can easily be preconditioned by its own inverse. The matrix O does not change the spectral elements of $\mathbf{A}'_J$ corresponding to the resolution levels $j > j_0$, hence it has no negative impact. Using any other preconditioner, for example $\mathbf{D}_1^{\pm s}$ of (2.2.14), will usually not result in better preconditioning of the operator $\mathbf{A}'_J$. We define the following diagonal matrix for preconditioning

$$\left(\mathbf{D}^{\pm s}_{\{O,X\}}\right)_{\lambda,\lambda'} := \delta_{(\lambda,\lambda')} \cdot \begin{cases} \left((D_{j_0}^{-1/2})_{\lambda,\lambda'}\right)^{\pm s} & |\lambda| = |\lambda'| = j_0 \\ \left(\mathbf{D}_X^{\pm s}\right)_{\lambda,\lambda'} & otherwise \end{cases}, \tag{2.3.45}$$

where $\mathbf{D}_X^{\pm s}$ could be any other preconditioner, for example (2.2.14) or (3.2.34).

Remark 2.45 *The chosen orthogonal matrix* O *will be densely populated. In our wavelet construction of Section 2.3.2, the minimum level is* $j_0 = 3$ *and the application of* O *will therefore require 81 floating point operations. The wavelet transform* $\mathbf{T}_{J,j_0}$*, on the other hand, requires 95 floating point multiplications. The overhead induced by* O *can thus lead to slightly higher execution times on level* $J = j_0 + 1$*, but is totally negligible on higher levels. Thus, the application of* $\mathbf{A}'_J$ *is still linear in time with respect to the number of unknowns.*

As we will see later in Section 5, this technique can lower the condition number of $\mathbf{A}_J$ in 1D by several orders of magnitude over all levels.

2.4 Multivariate Wavelets

There are several ways to construct wavelets on manifolds in higher dimensions. Of course, the construction of Section 2.3.1 could be carried out directly. A relatively new approach for complex and irregular domains $\Omega \subset I\!R^n$ is given by the **Finite-Element-Wavelets** established in [58] and [44]. On domains $\Omega \subset I\!R^n$ which can be expressed as Cartesian products of intervals (perhaps after applying domain decomposition strategies, see [13,31], further optimized and implemented on the sphere in [48]), multivariate wavelets can be constructed by tensor products of wavelet bases on these intervals.

2.4.1 Cartesian Products

Given a MRA for the interval $\mathrm{I} = (0,1)$, we can use it to form a MRA for the n-dimensional hypercube $\square := \square^n := (0,1)^n$ preserving the **regularity** $\gamma, \tilde{\gamma}$ and **moment conditions** $d, \tilde{d}$ of Section 2.2.1. Thus, we can construct multivariate wavelets on generally rectangular domains. By a **tensor product** we mean combining **univariate bases** $\{\Phi_{(j,l)}\}_{l=1,\dots,n}$ into one **multivariate basis**

$$\Phi_{(\square,j)} := \Phi_{(\square^n,j)}(x) := \bigotimes_{l=1}^{n} \Phi_{(j,l)}(x_l), \qquad x := (x_1,\dots,x_n), \tag{2.4.1}$$

which forms a single-scale basis of refinement level j on the domain $\square^n$. We will focus on the special case where

$$\Phi_{(j,l_1)} = \Phi_{(j,l_2)}, \qquad \text{for all } l_1, l_2,$$

which is the reason we will write $\Phi_{(\square,j)}$ instead of $\Phi_{(\square,j,l)}$. Each function $\phi_{(\square,j,k)} \in \Phi_{(\square,j)}$, with $k = (k_1,\dots,k_n)$ now being a **multi-index**, has approximately support of $2^{-j}(0,1)^n$. We can associate $\Phi_{(\square,j)}$ with the index set

$$\Delta_j^{\square} := \Delta_j^1 \times \cdots \times \Delta_j^n \ . \tag{2.4.2}$$

Analogously to the above, we can build tensor products of the wavelet basis $\Psi_{(J)}$ of (2.1.28), i.e.,

$$\Psi_{(\square,J)} := \Psi_{(\square,J)}(x) := \bigotimes_{i=1}^{n} \Psi_{(J)}(x_i) \ . \tag{2.4.3}$$

Here, functions on different levels in different spatial dimensions are coupled. This construction is called **anisotropic** tensorization in contrast to the **isotropic** construction outlined in [24] and implemented in, for expample, [11]. The anisotropic approach has the benefit of being easier handling of the generator basis $\Phi_{(\square,j)}$. We focus entirely on the anisotropic case. The wavelet basis functions $\psi_{j,k}(x) \in \Psi_{(\square,J)}$ are indexed as

$$\psi_{j,k}(x) := \psi_{j_1,\dots,j_n;k_1,\dots,k_n}(x) := \prod_{i=1}^{n} \psi_{j_i,k_i}(x_i) \ . \tag{2.4.4}$$

The tensor product wavelet space analogous to (2.1.51) will be referred to as

$$\Psi_{(\square)}^{I\!I} := \bigcup_{j=j_0-1}^{\infty} \Psi_{(\square,J)}, \tag{2.4.5}$$

with the infinite index set $I\!I_{(\square)}$ also given by the external product of the n index sets $I\!I$ of (2.1.39),

$$I\!I_{(\square)} := I\!I^1 \times \cdots \times I\!I^n \ . \tag{2.4.6}$$

We can now define the finite linear subspaces $\Psi_{(\square,J)}^{I\!I} \subset \Psi_{\square}^{I\!I}$ by truncation of the index set $I\!I_{(\square)}$ exactly as in (2.2.26). The definition of the finite index set for tensor product wavelets is now an extension of the previous definition (2.2.24),

$$I\!I_{(\square,J)} := \{\lambda \in I\!I_{(\square)} \,|\, |\lambda|_* \le J\}, \tag{2.4.7}$$

where $|\lambda|_* := \max\{j_1,\dots,j_n\}$ is the maximum norm. The connection to the wavelet basis $\Phi_{(\square,J)}$ is established by the wavelet transform $\mathbf{T}_{(\square,J)}$,

$$\Psi_{(\square,J)} = \mathbf{T}_{(\square,J)}^T \Phi_{(\square,J)} \ . \tag{2.4.8}$$

The tensor product wavelet transform $\mathbf{T}_{(\square,J)}$ can be constructed as the tensor product of the univariate transformations (2.1.31), i.e.,

$$\mathbf{T}_{(\square,J)} := \mathbf{T}_{(\square,J,J-1)} \cdots \mathbf{T}_{(\square,J,j_0)}, \qquad \mathbf{T}_{(\square,J,j)} := \bigotimes_{i=1}^{n} \mathbf{T}_{J,j} \ . \tag{2.4.9}$$

The multiplicative cascading structure obviously retains the properties of the multiscale transform. In the same way can this construction process be applied to the dual wavelets and to the inverse wavelet transform. This construction also preserves the biorthogonality of the primal and dual tensor wavelets.

2.4.2 Multidimensional Operators

One of the advantages of tensor products is the ease of the generalization of the involved operators to higher dimensions. We will now address the multi-dimensional counterparts to the one-dimensional operators, which we will be using later on.

Gramian Matrix

The **Gramian matrix** is also called **mass matrix**, because it is the matrix of the basis' own L_2-norm, i.e.,

$$\mathbf{M}_J := \left(\Psi_{(J)}, \Psi_{(J)}\right)_{L_2} := \left(\int_I \psi_\lambda \, \psi_{\lambda'} d\mu\right)_{\lambda,\lambda' \in \mathbb{I}_J} \ . \tag{2.4.10}$$

We can conclude from the locality property $(\mathcal{L})$(2.1.8) that the Gramian matrix is uniformly sparse and thus contains only $\mathcal{O}(\#\mathbb{I}_J)$ non-zero entries.

Proposition 2.46 *The n-dimensional mass matrix $\mathbf{M}_{(\square,J)}$ can be set up for tensor product wavelets as*

$$\begin{aligned} \mathbf{M}_{(\square,J)} &= \mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J & (2.4.11)\\ &= \mathbf{T}^T_{(\square,J)} \left(\mathbf{M}_{\Phi_J} \otimes \cdots \otimes \mathbf{M}_{\Phi_J}\right) \mathbf{T}_{(\square,J)} \ . & (2.4.12) \end{aligned}$$

Proof: By definition of the tensor product one has

$$\begin{aligned} \mathbf{M}_{(\square,J)} &= \left(\Psi_{(\square,J)}, \Psi_{(\square,J)}\right)_{L_2} \\ &= \left(\int_\square \psi_{\lambda_1} \cdots \psi_{\lambda_n} \psi_{\lambda'_1} \cdots \psi_{\lambda'_n} d\mu\right)_{\lambda,\lambda' \in \mathbb{I}_{(\square,J)}} \\ &= \left(\int_\square \psi_{\lambda_1} \psi_{\lambda'_1} \cdots \psi_{\lambda_n} \psi_{\lambda'_n} d\mu\right)_{\lambda,\lambda' \in \mathbb{I}_{(\square,J)}} \\ &= \left(\int_\mathrm{I} \psi_{\lambda_1} \psi_{\lambda'_1} d\mu \cdots \int_\mathrm{I} \psi_{\lambda_n} \psi_{\lambda'_n} d\mu\right)_{\lambda,\lambda' \in \mathbb{I}_{(\square,J)}} \\ &= \mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J \ . \end{aligned}$$

By (2.4.8), we can expand (2.4.11) as

$$\mathbf{M}_{(\square,J)} = \mathbf{T}^T_{(\square,J)} \left(\mathbf{M}_{\Phi_J} \otimes \cdots \otimes \mathbf{M}_{\Phi_J}\right) \mathbf{T}_{(\square,J)} \ .$$

where $\mathbf{M}_{\Phi_J}$ is of course the mass matrix with respect to the single-scale basis Φ_J. ∎

Hence, $\mathbf{M}_{(\square,J)}$ is also uniformly sparse and a matrix-vector multiplication of $\mathbf{M}_{(\square,J)}$ can be done with $\mathcal{O}(\#\mathbb{I}_{(\square,J)})$ arithmetic operations.

Stiffness Matrix

The **stiffness matrix** (2.3.38) can be decomposed into the **Laplace matrix** and the **mass matrix**

$$\mathbf{A}_J = \mathbf{S}_J + a_0 \, \mathbf{M}_J := \left(\left(\nabla\psi_\lambda, \nabla\psi_{\lambda'}\right)_{L_2}\right)_{\lambda,\lambda' \in \mathbb{I}_J} + a_0 \left(\left(\psi_\lambda, \psi_{\lambda'}\right)_{L_2}\right)_{\lambda,\lambda' \in \mathbb{I}_J} \tag{2.4.13}$$

using the two terms of (1.3.24). Employing the same arguments in Proposition 2.46, we conclude that the stiffness matrix (and the decomposition) is uniformly sparse.

Lemma 2.47 *The n-dimensional stiffness matrix* $\mathbf{A}_{(\square,J)}$ *is given by*

$$\mathbf{A}_{(\square,J)} := (a(\psi_\lambda, \psi_{\lambda'}))_{\lambda,\lambda' \in \mathcal{I}_{(\square,J)}} , \tag{2.4.14}$$

where the tensor product structure can again be exploited to yield the equivalent form

$$\begin{aligned} \mathbf{A}_{(\square,J)} \quad = \quad & \mathbf{S}_J \otimes \mathbf{M}_J \otimes \mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J \\ + \quad & \mathbf{M}_J \otimes \mathbf{S}_J \otimes \mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J \\ & \vdots \\ + \quad & \mathbf{M}_J \otimes \mathbf{M}_J \otimes \mathbf{M}_J \otimes \cdots \otimes \mathbf{S}_J \\ + \quad & a_0 \, \mathbf{M}_J \otimes \mathbf{M}_J \otimes \mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J \ . \end{aligned} \tag{2.4.15}$$

Proof: We focus on the first term in (2.4.13), since the mass term has already been dealt with in Proposition 2.46. The proof of Proposition 2.46 can easily be extended to show the following calculation rule for general functions $f = f_1 \otimes \cdots \otimes f_n$, $g = g_1 \otimes \cdots \otimes g_n$:

$$\begin{aligned} (f,g)_{L_2} \quad &= \quad (f_1 \otimes \cdots \otimes f_n, g_1 \otimes \cdots \otimes g_n)_{L_2} \\ &= \quad (f_1, g_1)_{L_2} \otimes \cdots \otimes (f_n, g_n)_{L_2} \ . \end{aligned}$$

Setting $\psi_\lambda := \psi_{\lambda_1} \otimes \cdots \otimes \psi_{\lambda_n}$ and $\psi_{\lambda'} := \psi_{\lambda'_1} \otimes \cdots \otimes \psi_{\lambda'_n}$, we now have for the first term

$$\begin{aligned} \big((\nabla\psi_\lambda, \nabla\psi_{\lambda'})_{L_2}\big)_{\lambda,\lambda' \in \mathcal{I}_{(\square,J)}} \quad &= \quad \left(\sum_{i=1}^n (\partial_i \psi_\lambda, \partial_i \psi_{\lambda'})_{L_2} \right)_{\lambda,\lambda' \in \mathcal{I}_{(\square,J)}} \\ &= \quad \Bigg(\sum_{i=1}^n (\psi_{\lambda_1} \otimes \cdots \otimes \partial_i \psi_{\lambda_i} \otimes \cdots \otimes \psi_{\lambda_n}, \\ &\qquad\qquad \psi_{\lambda_1} \otimes \cdots \otimes \partial_i \psi_{\lambda'_i} \otimes \cdots \otimes \psi_{\lambda'_n} \big)_{L_2} \Bigg)_{\lambda,\lambda' \in \mathcal{I}_{(\square,J)}} \\ &= \quad \Bigg(\sum_{i=1}^n \big(\psi_{\lambda_1}, \psi_{\lambda'_1}\big)_{L_2} \otimes \cdots \otimes \big(\partial_i \psi_{\lambda_1}, \partial_i \psi_{\lambda'_1}\big)_{L_2} \otimes \cdots \\ &\qquad\qquad \cdots \otimes \big(\psi_{\lambda_n}, \psi_{\lambda'_n}\big)_{L_2} \Bigg)_{\lambda,\lambda' \in \mathcal{I}_{(\square,J)}} \\ &= \quad \sum_{i=1}^n \underbrace{\mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J}_{i-1} \otimes \mathbf{S}_J \otimes \underbrace{\mathbf{M}_J \otimes \cdots \otimes \mathbf{M}_J}_{n-i} \ . \end{aligned}$$

Here we used the product rule in the second step and $\partial_i f_j = 0$ for $i \neq j$. Now follows (2.4.15) with Proposition 2.46. ∎

Proposition 2.48 *Designating by* $\mathbf{S}_{\Phi_J}$ *and* $\mathbf{M}_{\Phi_J}$ *the matrices from (2.4.13), assembled in the single-scale basis* Φ_J*, we can use the fast wavelet transform* $\mathbf{T}_{(\square,J)}$ *and compute* $\mathbf{A}_{(\square,J)}$ *as*

$$\begin{aligned} \mathbf{A}_{(\square,J)} \quad = \quad \mathbf{T}^T_{(\square,J)} (\quad & \mathbf{S}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \cdots \otimes \mathbf{M}_{\Phi_J} \\ + \quad & \mathbf{M}_{\Phi_J} \otimes \mathbf{S}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \cdots \otimes \mathbf{M}_{\Phi_J} \\ & \vdots \\ + \quad & \mathbf{M}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \cdots \otimes \mathbf{S}_{\Phi_J} \\ + \quad & a_0 \, \mathbf{M}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \mathbf{M}_{\Phi_J} \otimes \cdots \otimes \mathbf{M}_{\Phi_J} \ ) \ \mathbf{T}_{(\square,J)} \ . \end{aligned} \tag{2.4.16}$$

Since only uniformly sparse matrices are used in the setup of this matrix, we can again conclude that $\mathbf{A}_{(\square,J)}$ can be applied in $\mathcal{O}(\#\mathcal{I}_{(\square,J)})$ operations.

Diagonal Scaling

As noted before, there are several distinct diagonal scalings which satisfy Corollary 2.22, and every one differs in its tensor product construction. We will now show the construction of the matrix $\mathbf{D}^{\pm s}_{1,J}$,

which is defined through (2.2.14) and (2.2.25).
Definition (2.2.14) for the diagonal scaling operator $\mathbf{D}^s_{1,J}$ needs be modified in the tensor product setting using **multi-index** notation.

Proposition 2.49 *We can express $\mathbf{D}^s_{1,(\square,J)}$ for $s \geq 0$ with the following rule:*

$$\begin{array}{rcl} \mathbf{D}^s_{1,(\square,J)} & = & \frac{1}{n} \Big(\quad \mathbf{D}^s_{1,J} \otimes \mathbf{I}_J \otimes \mathbf{I}_J \otimes \cdots \otimes \mathbf{I}_J \\ & & + \quad \mathbf{I}_J \otimes \mathbf{D}^s_{1,J} \otimes \mathbf{I}_J \otimes \cdots \otimes \mathbf{I}_J \\ & & \vdots \qquad\qquad\quad \ddots \\ & & + \quad \mathbf{I}_J \otimes \mathbf{I}_J \otimes \cdots \otimes \mathbf{I}_J \otimes \mathbf{D}^s_{1,J} \quad \Big) \, . \end{array} \tag{2.4.17}$$

For $s < 0$, we set

$$\mathbf{D}^s_{1,(\square,J)} := \left(\mathbf{D}^{-s}_{1,(\square,J)}\right)^{-1}, \tag{2.4.18}$$

which can be computed efficiently by element-wise inversion of (2.4.17), because $\mathbf{D}^s_{1,(\square,J)}$ is a diagonal matrix.

This concludes our chapter on wavelets. We know how to construct the wavelet form of operators on domains of cartesian products of the interval using our wavelets of Section 2.3.2. We will use this knowledge for discretization of our control problem with Dirichlet boundary control. For this, we need an appropriate weak form of an elliptic PDE which allows for a flexible update of the Dirichlet boundary conditions. As we will see, this can be very elegantly accomplished by formulating the PDE and the boundary conditions as a **saddle point problem**.

3 Elliptic Boundary Value Problems

This chapter serves as an introduction into **saddle point problems** and especially to formulate **elliptic boundary value problems** as saddle point problems.
There is more than one way to weakly formulate the elliptic boundary value problem from Section 3.2, with specific consideration of the essential boundary conditions. One of the first works on the **boundary penalty method** (**BPM**) is by Babuška [5] from 1973. It uses a penalty parameter ϵ to append the Dirichlet boundary conditions weakly; thus, forming a problem with **Robin boundary conditions** (see [61]). This method can simultaneously handle Dirichlet and Neumann boundary conditions. However, the method is not recommended for iterative solvers (see [10]) since the penalty parameter has a strong negative impact on the condition of the problem. Up to now, no optimal preconditioner is known for this situation.
As an alternative, the **Lagrange multiplier method** was introduced by Babuška in 1973, see [4]. Here the space of test functions is chosen not to incorporate any Dirichlet boundary conditions. Instead, these are supposed to be attained attained by the Lagrangian multipliers only where needed. Important for us is that this method allows for optimal preconditioning by wavelets.

3.1 Saddle Point Problems

Let us first formulate saddle point problems in an abstract setting. Consider Hilbert spaces X and Q with their dual spaces X' and Q' together with their respective dual forms $\langle\cdot,\cdot\rangle_{X\times X'}$, $\langle\cdot,\cdot\rangle_{Q\times Q'}$. We define the product Hilbert space $\mathcal{H}$ and its dual $\mathcal{H}'$ as

$$\mathcal{H} := X \times Q', \qquad \mathcal{H}' := X' \times Q \ . \tag{3.1.1}$$

This definition of $\mathcal{H}$ involving Q' instead of Q is more convenient for the specification considered later. The $\mathcal{H}$-inner product will be given by

$$(\cdot,\cdot)_{\mathcal{H}} := (\cdot,\cdot)_X + (\cdot,\cdot)_{Q'}, \tag{3.1.2}$$

inducing the canonical norm $\|\cdot\|_{\mathcal{H}}$ on $\mathcal{H}$ as

$$\left\|\begin{pmatrix} v \\ \tilde{q}\end{pmatrix}\right\|^2_{\mathcal{H}} := (v,v)_X + (\tilde{q},\tilde{q})_{Q'} = \|v\|^2_X + \|\tilde{q}\|^2_{Q'} \qquad \text{for all } v \in X, \tilde{q} \in Q' \ . \tag{3.1.3}$$

3.1.1 Abstract Saddle Point Problems

Suppose $a(\cdot,\cdot) : X \times X \to I\!R$ is a continuous bilinear form with some constant $\alpha_2 > 0$

$$a(v,w) \le \alpha_2 \|v\|_X \|w\|_X, \qquad \text{for all } v,w \in X, \tag{3.1.4}$$

and likewise $b(\cdot,\cdot) : X \times Q' \to I\!R$ with some $\beta_2 > 0$,

$$b(v,\tilde{q}) \le \beta_2 \|v\|_X \|\tilde{q}\|_{Q'}, \qquad \text{for all } v \in X, \tilde{q} \in Q' \ . \tag{3.1.5}$$

We consider the following problem:

Problem 3.1 [Saddle Point Problem]
For given $f \in X'$ and $u \in Q$, find $y \in X$, $p \in Q'$ which solve the extremal problem

$$\inf_{v\in X} \sup_{\tilde{q}\in Q'} \frac{1}{2} a(v,v) - \langle f, v\rangle_{X'\times X} + b(v,\tilde{q}) - \langle u,\tilde{q}\rangle_{Q\times Q'} \ . \tag{3.1.6}$$

We can formulate Problem 3.1 equivalently as the search for the minimum $y \in X$ of

$$J(v) := \frac{1}{2} a(v,v) - \langle f, v\rangle_{X'\times X} \tag{3.1.7}$$

under the constraint of

$$b(v,\tilde{q}) = \langle u,\tilde{q}\rangle_{Q\times Q'}, \qquad \text{for all } \tilde{q} \in Q' \ . \tag{3.1.8}$$

This minimization problem can be solved by appending the constraints by means of a **Lagrangian multiplier** to the functional (3.1.7), by defining

$$K(v,\tilde{q}) := J(v) + b(v,\tilde{q}) - \langle u,\tilde{q}\rangle_{Q\times Q'} \tag{3.1.9}$$

and solving the system of equations resulting from the necessary, and (as we will see later) in our case also sufficient minimization conditions

$$\partial K(v,\tilde{q}) = 0\ . \tag{3.1.10}$$

This procedure is justified by the fact that the minimum of (3.1.7) will also be the minimum of (3.1.9), if $\tilde{q}$ is indeed an element for which (3.1.8) is fulfilled.
Explicitly calculating (3.1.10) leads to the following reformulation of Problem 3.1:

Problem 3.2 [Saddle Point Problem - Optimality Conditions]
Given $(f,u)\in\mathcal{H}'$, find $(y,p)\in\mathcal{H}$ such that

$$\begin{array}{lcll} a(y,v)+b(v,p) & = & \langle f,v\rangle_{X'\times X}, & \text{for all } v\in X,\\ b(y,\tilde{q}) & = & \langle u,\tilde{q}\rangle_{Q\times Q'}, & \text{for all } \tilde{q}\in Q', \end{array} \tag{3.1.11}$$

Any solution $(y,p)\in\mathcal{H}$ of Problem 3.2 satisfies

$$K(y,\tilde{q}) \le K(y,p) \le K(v,p), \qquad \text{for all } v\in X, \tilde{q}\in Q', \tag{3.1.12}$$

which is therefore called **saddle point property**. Recall from Section 1.3.1 that the bilinear forms $a(\cdot,\cdot), b(\cdot,\cdot)$ define linear continuous operators $A: X\to X'$, $B: X\to Q$, along with their adjoints $A': X'\to X$, $B': Q'\to X'$, by

$$\langle v, A'w\rangle_{X\times X'} = \langle Av, w\rangle_{X'\times X} := a(v,w), \tag{3.1.13}$$

$$\langle v, B'\tilde{q}\rangle_{X\times X'} = \langle Bv, \tilde{q}\rangle_{Q\times Q'} := b(v,\tilde{q})\ . \tag{3.1.14}$$

Hence, these operators are defined by their roles as functionals $Av\in X'$ and $B'\tilde{q}\in Q'$ acting on elements of the space X. We can use these operators to rephrase Problem 3.2 as follows:

Problem 3.3 [Saddle Point Problem - Operator Representation]
Given $(f,u)\in\mathcal{H}'$, find $(y,p)\in\mathcal{H}$ such that

$$\begin{array}{lcl} A\,y + B'\,p & = & f,\\ B\,y & = & u, \end{array} \tag{3.1.15}$$

which can be written as one linear system as

$$\begin{pmatrix} A & B'\\ B & 0 \end{pmatrix}\begin{pmatrix} y\\ p\end{pmatrix} = \begin{pmatrix} f\\ u\end{pmatrix}\ . \tag{3.1.16}$$

Next, we discuss under which conditions a solution $(y,p)\in\mathcal{H}$ to Problem 3.3 exists and is unique.
To this end, we define the **kernel** of operator B as

$$\ker B := \{v\in X \,|\, b(v,\tilde{q}) = 0 \ \text{ for all } \tilde{q}\in Q'\}\subset X\ . \tag{3.1.17}$$

We quote the following result from [46] which was first established by F. Brezzi in [9].

Theorem 3.4 *Let the linear operator A be invertible on $\ker B\subseteq X$, i.e., for some constant $\alpha_1>0$*

$$\begin{array}{lcl} \inf\limits_{v\in\ker B}\sup\limits_{w\in\ker B}\dfrac{\langle Av,w\rangle_{X'\times X}}{\|v\|_X\|w\|_X} & \ge & \alpha_1,\\ \inf\limits_{v\in\ker B}\sup\limits_{w\in\ker B}\dfrac{\langle A'v,w\rangle_{X'\times X}}{\|v\|_X\|w\|_X} & \ge & \alpha_1\ . \end{array} \tag{3.1.18}$$

Let further the range of B be closed in Q, e.g. for some constant $\beta_1>0$ the **inf-sup condition**

$$\inf_{\tilde{q}\in Q'}\sup_{v\in X}\frac{\langle Bv,\tilde{q}\rangle_{Q\times Q'}}{\|v\|_X\|\tilde{q}\|_{Q'}} \ge \beta_1 > 0 \tag{3.1.19}$$

holds. Then there exists a unique solution $(y,p) \in \mathcal{H}$ *to Problem 3.3 for given* $(f,u) \in \mathcal{H}'$*. That is,*

$$L := \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} : \mathcal{H} \to \mathcal{H}' \tag{3.1.20}$$

is an isomorphism, and one has the norm equivalence

$$c_L \left\| \begin{pmatrix} v \\ \tilde{q} \end{pmatrix} \right\|_{\mathcal{H}} \le \left\| L \begin{pmatrix} v \\ \tilde{q} \end{pmatrix} \right\|_{\mathcal{H}'} \le C_L \left\| \begin{pmatrix} v \\ \tilde{q} \end{pmatrix} \right\|_{\mathcal{H}} \tag{3.1.21}$$

for any $(v,\tilde{q}) \in \mathcal{H}$*, where the constants* c_L, C_L *are given as*

$$\begin{aligned} c_L &:= \left(\frac{1}{\alpha_1 \beta_1} \left(1 + \frac{\alpha_2}{\alpha_1} \right) + \max \left\{ \frac{2}{\alpha_1^2} \, , \, \frac{1}{\beta_1^2} \left(1 + \frac{\alpha_2}{\alpha_1} \right)^2 + \left(\frac{\alpha_2}{\beta_1^2} \left(1 + \frac{\alpha_2}{\alpha_1} \right) \right)^2 \right\} \right)^{-1/2} \\ C_L &:= \sqrt{2(\alpha_2^2 + \beta_2^2)} \, . \end{aligned}$$

Remark 3.5 *The first preprequisite is trivially fulfilled if* A *is invertible on all* X*. In this case,*

$$\alpha_1 \|v\|_X \le \|Av\|_{X'} \le \alpha_2 \|v\|_X \iff \|Av\|_{X'} \sim \|v\|_X, \qquad \text{for all } v \in X,$$

follows with continuity (3.1.4). Our bilinear form, stemming from the elliptic partial differential equations of Section 1.3.2, is of this type, see Section 3.2.

The **inf-sup condition** (3.1.19) is always satisfied if B is **surjective**, i.e., range $B = Q$. This will be the case in Section 3.2 when we consider B to be the trace operator onto the boundary $\Gamma \subseteq \partial\Omega \subset \mathbb{R}^{n-1}$ of a domain $\Omega \subset \mathbb{R}^n$.

3.1.2 Wavelet Discretization

Since we now have an isomorphism $L : \mathcal{H} \to \mathcal{H}'$, we can use the theory from Section 2.2.3 and represent L in standard wavelet representation, which will be denoted by $\mathbf{L}$.
We choose specifically the spaces $X = H^{+m}(\Omega)$ and $Q = H^{+s}(\Gamma)$ as Sobolev spaces of orders $m, s \ge 0$ on the bounded domains bounded in $\mathbb{R}^n$. In particular, the domain Γ can be a subset of Ω. For example, if $\Omega = \square^n := (0,1)^n$, then $\Gamma \subseteq \partial\Omega$ may be an edge or a face of this cube.
With the background of Section 2, we have assured the existence of biorthogonal wavelet bases Ψ_Ω^m, $\widetilde{\Psi}_\Omega^m$ and Ψ_Γ^s,$\widetilde{\Psi}_\Gamma^s$ for the spaces X, Q and their duals X',Q', such that the norm equivalences (2.2.12), (2.2.13) hold for the required ranges,

$$\|v\|_X \sim \|\mathbf{D}_\Omega^{-m} \mathbf{v}\|_{\ell_2(\mathcal{I}_X)}, \qquad \|\tilde{v}\|_{X'} \sim \|\mathbf{D}_\Omega^{+m} \tilde{\mathbf{v}}\|_{\ell_2(\mathcal{I}_X)}, \tag{3.1.22}$$

and

$$\|q\|_Q \sim \|\mathbf{D}_\Gamma^{-s} \mathbf{v}\|_{\ell_2(\mathcal{I}_Q)}, \qquad \|\tilde{q}\|_{Q'} \sim \|\mathbf{D}_\Gamma^{+s} \tilde{\mathbf{q}}\|_{\ell_2(\mathcal{I}_Q)} \, . \tag{3.1.23}$$

Hence, in accordance with the notation introduced in Corollary 2.22, a wavelet basis for $\mathcal{H} = X \times Q'$ is given by $\Psi_\mathcal{H} := \left(\Psi_\Omega^m, \widetilde{\Psi}_\Gamma^s \right)^T$ with the index set $\mathcal{I}_\mathcal{H} := \mathcal{I}_X \times \mathcal{I}_Q$. Likewise, a basis for the dual space $\mathcal{H}' = X' \times Q$ is $\widetilde{\Psi}_\mathcal{H} := \left(\widetilde{\Psi}_\Omega^m, \Psi_\Gamma^s \right)^T$.
We can expand the right hand side $F := (f,u)^T \in \mathcal{H}'$ in these scaled wavelet bases as

$$F = (f,u)^T = \left(\mathbf{f}^T \widetilde{\Psi}_\Omega^m, \mathbf{u}^T \Psi_\Gamma^s \right)^T =: \mathbf{F}^T \widetilde{\Psi}_\mathcal{H} \, . \tag{3.1.24}$$

The solution vector $Y = (y,p)^T \in \mathcal{H}$ has an analogous expansion

$$Y = (y,p)^T = \left(\mathbf{y}^T \Psi_\Omega^m, \mathbf{p}^T \widetilde{\Psi}_\Gamma^s \right)^T =: \mathbf{Y}^T \Psi_\mathcal{H} \, . \tag{3.1.25}$$

By Section 2.2.3, the discretized infinite-dimensional operator $\mathbf{L}$ from Problem 3.3 is now given by

$$\mathbf{L} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} := \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{u} \end{pmatrix}, \tag{3.1.26}$$

with the operators $\mathbf{A}$,$\mathbf{B}$ in standard wavelet form, see Section 2.2.3,

$$\begin{array}{rcl} \mathbf{A} & := & \langle \Psi_\Omega^m, A\Psi_\Omega^m\rangle \;=\; \mathbf{D}_\Omega^{-m}\langle \Psi_\Omega, A\Psi_\Omega\rangle \mathbf{D}_\Omega^{-m}, \\ \mathbf{B} & := & \langle \widetilde{\Psi}_\Gamma^s, B\Psi_\Omega^m\rangle \;=\; \mathbf{D}_\Gamma^{+s}\langle \widetilde{\Psi}_\Gamma, B\Psi_\Omega\rangle \mathbf{D}_\Omega^{-m}\ . \end{array} \tag{3.1.27}$$

The coefficient vectors $\mathbf{f}, \mathbf{g}$ are calculated by the primal and dual expansions (2.1.53), (2.1.54), i.e.,

$$\mathbf{f} = \mathbf{D}_\Omega^{-m}\langle \Psi_\Omega, f\rangle, \qquad \mathbf{u} = \mathbf{D}_\Gamma^{+s}\langle \widetilde{\Psi}_\Gamma, u\rangle\ . \tag{3.1.28}$$

Since all prerequisites from Section 2.2.3 are fulfilled, we can formulate the following result.

Corollary 3.6 *The linear operator* $\mathbf{L}$ *defined in (3.1.26) is an* ℓ_2*-automorphism, i.e., for every* $(\mathbf{v}, \tilde{\mathbf{q}}) \in \ell_2(I\!I_X \times I\!I_Q)$ *we have*

$$\mathbf{c_L}\left\|\begin{pmatrix}\mathbf{v}\\ \tilde{\mathbf{q}}\end{pmatrix}\right\|_{\ell_2} \le \left\|\mathbf{L}\begin{pmatrix}\mathbf{v}\\ \tilde{\mathbf{q}}\end{pmatrix}\right\|_{\ell_2} \le \mathbf{C_L}\left\|\begin{pmatrix}\mathbf{v}\\ \tilde{\mathbf{q}}\end{pmatrix}\right\|_{\ell_2} \tag{3.1.29}$$

with constants $\mathbf{c_L}$, $\mathbf{C_L}$ *only depending on* c_L,C_L *from (3.1.21) and the constants in the norm equivalences (3.1.22) and (3.1.23).*

3.1.3 Stability - The LBB-Condition

We must now investigate under which conditions the infinite-dimensional operator $\mathbf{L}$ remains an ℓ_2-isomorphism when we progress from the infinite index sets $I\!I_X$ and $I\!I_Q$ to the finite index set $I\!I_{X,\sigma} \subset I\!I_X$ and $I\!I_{Q,\pi} \subset I\!I_Q$. Here $J_X \equiv \sigma$ and $J_Q \equiv \pi$ should be the level of resolutions on the space X and Q, respectively.
We define the finite subset of indices for the product space $\mathcal{H}$ with $J_\mathcal{H} := (\sigma, \pi)$ as

$$I\!I_{\mathcal{H},J_\mathcal{H}} := I\!I_{X,\sigma} \times I\!I_{Q,\pi} \subset I\!I_\mathcal{H}, \tag{3.1.30}$$

which defines a subspace $\Psi_{(\mathcal{H},J_\mathcal{H})}$ of the wavelet space $\Psi_\mathcal{H}$. Defining $I\!I_{X,\sigma}, I\!I_{Q,\pi}$ as in (2.4.7), this subspace is linear. The corresponding spaces are denoted as

$$X_\sigma := S(\Psi_{(\Omega,\sigma)}), \qquad Q_\pi := S(\Psi_{(\Gamma,\pi)}), \tag{3.1.31}$$

and accordingly for the dual side X'_σ, Q'_π. Hence we have the finite dimensional product spaces

$$\mathcal{H}_{J_\mathcal{H}} := \mathcal{H}_{(\sigma,\pi)} := X_\sigma \times Q'_\pi \subset \mathcal{H}, \qquad \mathcal{H}'_{J_\mathcal{H}} := \mathcal{H}'_{(\sigma,\pi)} := X'_\sigma \times Q_\pi \subset \mathcal{H}'\ . \tag{3.1.32}$$

The finite analogons to any set, vector and operator are constructed by deleting any elements, rows or columns referring to indexes not contained in the finite index sets.
The right hand side $F_{J_\mathcal{H}} := (f_\sigma, u_\pi)$ and the solution vector $Y_{J_\mathcal{H}} := (y_\sigma, p_\pi)$ are, as in Section 2.2.5, expanded in terms of the finite wavelet bases as

$$\begin{array}{rcl} F_{J_\mathcal{H}} & = & \mathbf{F}_{J_\mathcal{H}}^T \mathbf{D}_{\mathcal{H},J_\mathcal{H}} \Psi_{(\mathcal{H},J_\mathcal{H})} \;:=\; (\mathbf{f}_\sigma^T \mathbf{D}_{\Omega,\sigma}^{-m} \widetilde{\Psi}_{(\Omega,\sigma)}, \mathbf{u}_\pi^T \mathbf{D}_{\Gamma,\pi}^{+s} \Psi_{(\Gamma,\pi)}) \in \mathcal{H}'_{J_\mathcal{H}}, \\ Y_{J_\mathcal{H}} & = & \mathbf{Y}_{J_\mathcal{H}}^T \mathbf{D}_{\mathcal{H}',J_\mathcal{H}} \widetilde{\Psi}_{(\mathcal{H},J_\mathcal{H})} \;:=\; (\mathbf{y}_\sigma^T \mathbf{D}_{\Omega,\sigma}^{+m} \Psi_{(\Omega,\sigma)}, \mathbf{p}_\pi^T \mathbf{D}_{\Gamma,\pi}^{-s} \widetilde{\Psi}_{(\Gamma,\pi)}) \in \mathcal{H}_{J_\mathcal{H}}\ . \end{array} \tag{3.1.33}$$

The finite analogons of the operators (3.1.27) are

$$\begin{array}{rcl} \mathbf{A}_\sigma & := & \mathbf{D}_{\Omega,\sigma}^{-m}\langle \Psi_{(\Omega,\sigma)}, A\Psi_{(\Omega,\sigma)}\rangle \mathbf{D}_{\Omega,\sigma}^{-m}, \\ \mathbf{B}_{\sigma,\pi} & := & \mathbf{D}_{\Gamma,\pi}^{+s}\langle \widetilde{\Psi}_{(\Gamma,\pi)}, B\Psi_{(\Omega,\sigma)}\rangle \mathbf{D}_{\Omega,\sigma}^{-m}\ . \end{array} \tag{3.1.34}$$

The **discrete finite** saddle point problem can then be formulated in operator form as follows.

Problem 3.7 [Saddle Point Problem - Finite Discretization]
Given $(\mathbf{f}_\sigma, \mathbf{g}_\pi)$*, find* $(\mathbf{y}_\sigma, \mathbf{p}_\pi)$ *such that*

$$\mathbf{L}_{J_\mathcal{H}}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} := \begin{pmatrix}\mathbf{A}_\sigma & \mathbf{B}_{\sigma,\pi}^T\\ \mathbf{B}_{\sigma,\pi} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} = \begin{pmatrix}\mathbf{f}_\sigma\\ \mathbf{u}_\pi\end{pmatrix} \tag{3.1.35}$$

holds.

To guarantee the existence of a solution to this problem, we can again use Theorem 3.4, which specifically applies to finite dimensional spaces. We have to assure that the spaces X_σ, Q_π are chosen such that the ellipticity conditions (3.1.18) for $a(\cdot,\cdot)$ and the inf-sup condition (3.1.19) for $b(\cdot,\cdot)$ are satisfied with respect to these spaces.
The continuity inequalities (3.1.4), (3.1.5) are obviously fulfilled for subspaces $X_\sigma \subset X$ and $Q_\pi \subset Q$ and hold uniformly with the same constants α_2, β_2.
In contrast, the **discrete ellipticity condition**, the analogon to (3.1.18),

$$\begin{aligned} \inf_{\mathbf{v}\in\ker \mathbf{B}_{\sigma,\pi}} \sup_{\mathbf{w}\in\ker \mathbf{B}_{\sigma,\pi}} \frac{\langle \mathbf{A}_\sigma \mathbf{v},\mathbf{w}\rangle_{X'\times X}}{\|\mathbf{v}\|_X\|\mathbf{w}\|_X} &\geq \tilde{\alpha}_1, \\ \inf_{\mathbf{v}\in\ker \mathbf{B}_{\sigma,\pi}} \sup_{\mathbf{w}\in\ker \mathbf{B}_{\sigma,\pi}} \frac{\langle \mathbf{A}_\sigma^T \mathbf{v},\mathbf{w}\rangle_{X'\times X}}{\|\mathbf{v}\|_X\|\mathbf{w}\|_X} &\geq \tilde{\alpha}_1, \end{aligned} \tag{3.1.36}$$

does not immediately carry over, since not automatically

$$\ker \mathbf{B}_{\sigma,\pi} \subset \ker B, \tag{3.1.37}$$

holds for the kernel of $\mathbf{B}_{\sigma,\pi}$,

$$\ker \mathbf{B}_{\sigma,\pi} := \{v_\sigma \in X_\sigma \mid b(v_\sigma,\tilde{q}_\pi) = 0 \text{ for all } \tilde{q}_\pi \in Q'_\pi\} \subset X_\sigma \subset X\ . \tag{3.1.38}$$

The next definition is taken from [21].

Definition 3.8 [FEP]
The discretization $X_\sigma \subset X, Q_\pi \subset Q$ is said to have **Full Equilibrium Property** *if (3.1.37) holds.*

Obviously, this is a very strong property and it cannot be hoped that this is valid automatically for any operator B and its discretizations. Of course, if (3.1.37) is assured to be true, then the discrete ellipticity (3.1.36) follows directly from general ellipticity (3.1.18).
The discrete analogon of the inf-sup condition (3.1.19) for the pair X_σ, Q'_π reads as follows.

Definition 3.9 [LBB-condition]
We say $\mathbf{B}_{\sigma,\pi}$ satisfies the **Ladysenskaya-Babuška-Brezzi-condition** *if a constant $\tilde{\beta}_1 > 0$ exists such that*

$$\inf_{\tilde{q}\in Q'_\pi} \sup_{v\in X_\sigma} \frac{\langle \mathbf{B}_{\sigma,\pi} v,\tilde{q}\rangle_{Q\times Q'}}{\|v\|_X\|\tilde{q}\|_{Q'}} \geq \tilde{\beta}_1 > 0, \tag{3.1.39}$$

holds uniformly in σ,π.

The LBB-condition can be interpreted as a way of ensuring that no element $\tilde{q} \in Q'_\pi$ is orthogonal to any element $\mathbf{B}_{\sigma,\pi} v \in Q$ with respect to $\langle\cdot,\cdot\rangle_{Q\times Q'}$ and the parameter $\tilde{\beta}_1$ expresses the magnitude of that orthogonality property.
There are now several criteria which ensure the validity of the LBB-condition and the discrete ellipticity condition. The wavelet setting of this section provides some necessary and sufficient conditions which rely especially on biorthogonality $(\mathcal{B})$(2.1.52) and the explicit representation of the dual space by the dual MRA of Section 2.1.4. We give a short summary of some results which can be used in the framework for our control problem of Section 5.

Probably the most apparent way to fulfill the LBB- and FEP-conditions is given by a general result from [21].

Corollary 3.10 *If the spaces X_σ and Q_π fulfill $B(X_\sigma) = Q_\pi$ or equivalently $B'(Q'_\pi) = X'_\sigma$, then both the LBB- and the FEP-condition are valid.*

One situation where this applies is when the domain Ω is given as the cartesian product of, for example, an interval $\mathrm{I} = (0,1)$ and $\Gamma = (0,1) \subset \partial\Omega$. We can then choose biorthogonal wavelet spaces $\Psi_\Gamma, \widetilde{\Psi}_\Gamma$ on $(0,1)$ and use the tensor product construction of Section 2.4 with these wavelets on the domain $\Omega = \square^n$. If the operator B is given through the trace operator

$$\gamma_0\, v = v|_\Gamma, \qquad v \in X,$$

then the prerequisites of Corollary 3.10 are satisfied.

Another way to satisfy the LBB-condition, which fits the wavelet setting is **Fortin's criterion** from [36].

Theorem 3.11 *Let the bilinear form $b(\cdot,\cdot) : X \times Q' \to \mathbb{R}$ fulfill the* **inf-sup condition** *(3.1.19). If there is a uniformly bounded linear projector $\Pi_\sigma : X \to X_\sigma$ such that for any $v \in X_\sigma$*

$$b(v - \Pi_\sigma v, \tilde{q}_\pi) = 0 \qquad \text{for all } \tilde{q}_\pi \in Q'_\pi, \tag{3.1.40}$$

then the **LBB**-*condition (3.1.39) holds for $b(\cdot,\cdot)$.*

In the wavelet setting a bounded linear projector appears naturally during the construction process, cf. (2.1.44) and (2.1.45).

A general approach is given in [27]. In case B is not a trace operator from the space X onto the space Q, the equilibrium condition (3.1.37) may not hold. Thus, as a generalization of the above setting, we consider the case where Q_π is explicitly not given as the trace of the space X_σ.
This setting depends only on stable multiresolution spaces $S_j, \widetilde{S}_j$ of MRAs $\mathcal{S}, \widetilde{\mathcal{S}}$ on both Ω and Γ which satisfy **Jackson**- and **Bernstein**-inequalities (2.2.10),(2.2.11) of certain orders. Also, the boundedness of operator B must be given and a Trace Theorem 1.20 must hold. The main result of [27] states that the LBB-condition is satisfied whenever

$$\sigma = \pi + L, \qquad L = L(\Omega, \Gamma, \mathcal{S}, \widetilde{\mathcal{S}}, B) > 0, \tag{3.1.41}$$

which, in other words, means that the discretization level on the domain Ω must be chosen somewhat higher than the mesh size of the space on Γ. This result has the advantage of enforcing no constraints to the spatial dimensions or requiring any interrelation of the domains Ω and Γ.

3.1.4 Numerical Considerations

Now that we can ensure finite stability of the discretized saddle point problems, it remains to solve the system of equations (3.1.35) numerically. The implementation details of the algorithms for solving (3.1.26) can be found in Section 3.3.

Error Estimates

We quote a general result from [10] which gives an error estimate (cf. Lemma 1.39) for the solution of the discretized problem (3.7) with respect to the solution of the continuous problem (3.3).

Proposition 3.12 *Let $(y,p) \in X \times Q'$ be the solution of the continuous saddle point problem 3.3 and $(y_\sigma, p_\pi) \in X_\sigma \times Q'_\pi$ be the solution of the discrete finite problem 3.7. Then the error estimates*

$$\|y - y_\sigma\|_X \leq \left(1 + \frac{\alpha_2}{\tilde{\alpha}_1}\right)\left(1 + \frac{\beta_2}{\tilde{\beta}_1}\right) \inf_{v \in X_\sigma} \|y - v\|_X + \frac{\beta_2}{\tilde{\alpha}_1} \inf_{\tilde{q} \in Q'_\pi} \|p - \tilde{q}\|_{Q'} \tag{3.1.42}$$

$$\|p - p_\pi\|_{Q'} \leq \left(1 + \frac{\beta_2}{\tilde{\beta}_1}\right) \inf_{\tilde{q} \in Q'_\pi} \|p - \tilde{q}\|_{Q'} + \frac{\alpha_2}{\tilde{\beta}_1} \inf_{\mathbf{v} \in X} \|y - v\|_X \tag{3.1.43}$$

hold with the constants from (3.1.18), (3.1.19), (3.1.36) and (3.1.39).

This means the numerical error between y and y_σ can be estimated by the best approximation from the spaces X_σ and Q'_π. The ellipticity and continuity constants are uniformly bounded, i.e., they do not depend on σ nor π. With the assumption $(\mathcal{R})$(2.1.6), we can conclude $y_\sigma \to y$ for $\sigma \to \infty$.
These results can be further refined in case the FEP-condition is valid. We quote from [8] the following

Corollary 3.13 *Let the prerequisites of Proposition 3.12 and additionally the FEP-condition be satisfied. Then one has*

$$\|y - y_\sigma\|_X \leq \frac{\alpha_2}{\tilde{\alpha}_1} \inf_{v \in X_\sigma} \|y - v\|_X \ . \tag{3.1.44}$$

Also, we can derive for uniform refinements as in Section 2.2.4 the estimate

$$\inf_{v \in X_\sigma} \|y - v\|_X \lesssim \|y\|_X \ . \tag{3.1.45}$$

Together with (3.1.44) this yields

$$\|y - y_\sigma\|_X \lesssim \|y\|_X \ . \tag{3.1.46}$$

The right hand side of (3.1.46) can be bounded by **a-priori error estimates** for the unique solution $(y,p) \in \mathcal{H}$ with right hand side $(f,u) \in \mathcal{H}'$, i.e.,

$$\begin{aligned} \|y\|_X &\leq \frac{1}{\alpha_1}\|f\|_{X'} + \frac{1}{\beta_1}\left(1+\frac{\alpha_2}{\alpha_1}\right)\|u\|_Q, & (3.1.47)\\ \|p\|_{Q'} &\leq \frac{1}{\beta_1}\left(1+\frac{\alpha_2}{\alpha_1}\right)\|f\|_{X'} + \frac{\alpha_2}{\beta_1^2}\left(1+\frac{\alpha_2}{\alpha_1}\right)\|u\|_Q, & (3.1.48) \end{aligned}$$

see [10] for instance.

Assembly in Wavelet Coordinates

In accordance with Section 2.2.3, the operator $\mathbf{L}_{J_\mathcal{H}}$ will **not** be assembled during the computations. Recall that we always use (2.2.18). Let the following relations hold for the domain Ω, the wavelets and their single-scale functions:

$$\Psi_{(\Omega,J_\mathcal{H})} = \mathbf{T}^T_{\Omega,J_\mathcal{H}}\Phi_{\Omega,J_\mathcal{H}}, \qquad \widetilde{\Psi}_{(\Omega,J_\mathcal{H})} = \widetilde{\mathbf{T}}^T_{\Omega,J_\mathcal{H}}\widetilde{\Phi}_{\Omega,J_\mathcal{H}} \ .$$

Exactly the same relations should hold for the wavelets Ψ_Γ, $\widetilde{\Psi}_\Gamma$ on the domain Γ. The wavelet transform $\mathbf{T}_{\mathcal{H},J_\mathcal{H}}$ for the space $\mathcal{H}_{J_\mathcal{H}}$ can now be written in matrix representation as

$$\mathbf{T}_{\mathcal{H},J_\mathcal{H}} := \begin{pmatrix} \mathbf{T}_{\Omega,\sigma} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{T}}_{\Gamma,\pi} \end{pmatrix}, \tag{3.1.49}$$

and the diagonal scaling $\mathbf{D}_{\mathcal{H},J_\mathcal{H}}$ accordingly as

$$\mathbf{D}_{\mathcal{H},J_\mathcal{H}} := \begin{pmatrix} \mathbf{D}^{-m}_{\Omega,\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{+s}_{\Gamma,\pi} \end{pmatrix} \ . \tag{3.1.50}$$

Then we can express the system (3.1.26) as

$$\mathbf{D}_{\mathcal{H},J_\mathcal{H}}\mathbf{T}^T_{\mathcal{H},J_\mathcal{H}}\mathbf{L}_{\Phi,J_\mathcal{H}}\mathbf{T}_{\mathcal{H},J_\mathcal{H}}\mathbf{D}_{\mathcal{H},J_\mathcal{H}}\,\mathbf{Y}_{\mathcal{H},J_\mathcal{H}} = \mathbf{D}_{\mathcal{H},J_\mathcal{H}}\mathbf{T}^T_{\mathcal{H},J_\mathcal{H}}\mathbf{F}_{\Phi,J_\mathcal{H}}, \tag{3.1.51}$$

with the single-scale version of the operator $\mathbf{L}_{\Phi,J_\mathcal{H}}$ and the vector $\mathbf{F}_{\Phi,J_\mathcal{H}}$ given as

$$\begin{aligned} \mathbf{L}_{\Phi,J_\mathcal{H}} &:= \begin{pmatrix} \langle\Phi_{\Omega,\sigma}, A\Phi_{\Omega,\sigma}\rangle & \langle\widetilde{\Phi}_{\Gamma,\pi}, B\Phi_{\Omega,\sigma}\rangle^T \\ \langle\widetilde{\Phi}_{\Gamma,\pi}, B\Phi_{\Omega,\sigma}\rangle & \mathbf{0} \end{pmatrix}, \\ \mathbf{F}_{\Phi,\sigma} &:= \begin{pmatrix} \langle\Phi_{\Omega,\sigma}, f\rangle \\ \langle\widetilde{\Phi}_{\Gamma,\pi}, Y\rangle \end{pmatrix}. \end{aligned} \tag{3.1.52}$$

Thus, after the solution vector $\mathbf{Y}_{\mathcal{H},J_\mathcal{H}}$ is determined, we can calculate

$$\mathbf{Y}_{\Phi,J_\mathcal{H}} = \begin{pmatrix} \langle\widetilde{\Phi}_{\Omega,\sigma}, y\rangle \\ \langle\Phi_{\Gamma,\pi}, p\rangle \end{pmatrix} = \mathbf{T}_{\mathcal{H},J_\mathcal{H}}\mathbf{D}_{\mathcal{H},J_\mathcal{H}}\mathbf{Y}_{\mathcal{H},J_\mathcal{H}} \tag{3.1.53}$$

and obtain the solution y_σ as

$$y_\sigma = \langle\widetilde{\Phi}_{\Omega,\sigma}, y\rangle\Phi_{\Omega,\sigma} \in X_\sigma \subset X \ . \tag{3.1.54}$$

3.2 Elliptic Boundary Value Problems

Let $\Omega \subseteq \square \subset I\!R^n$ be a domain bounded by the cube $\square = \square^n = (0,1)^n$. We assume Ω has a Lipschitz continuous boundary $\partial\Omega \in C^{0,1}$ and $\Gamma \subseteq \partial\Omega$ is a subset of $\partial\Omega$ with strictly positive surface measure. We consider the following

Problem 3.14 [Elliptic Boundary Value Problem]
For given functions $f_\Omega \in (H^1(\Omega))'$ and $u \in H^{1/2}(\Gamma)$, search $y \in H^1(\square)$ satisfying

$$\begin{aligned} -\nabla\cdot(\mathbf{a}\nabla y) + a_0\, y &= f_\Omega, && \text{in } \Omega, \\ y &= u, && \text{on } \Gamma, \\ (\mathbf{a}\nabla y)\cdot\mathbf{n} &= 0, && \text{on } \partial\Omega\setminus\Gamma, \end{aligned} \tag{3.2.1}$$

where $\mathbf{n} = \mathbf{n}(x)$ is the outward normal at $x \in \partial\Omega\setminus\Gamma$. Moreover, $\mathbf{a}(x) = (a_{i,j}(x))_{i,j}$ is uniformly positive definite on Ω and $a_0 > 0$.

In order to apply the theory from Section 3.1, we transform the above problem into a weak formulation, expressing the equations (3.2.1) by **bilinear forms**. We can formulate the partial differential equation (3.2.1) by the theory of Section 1.3.2 as follows:
Given $f_\Omega \in (H^1(\Omega))'$ and $u \in H^{1/2}(\Gamma)$, find the solution $y \in H^1(\Omega)$, which solves

$$\begin{aligned} a_\Omega(y,v) &= \langle f_\Omega, v\rangle_\Omega && \text{for all } v \in H^1_{0,\Gamma}(\Omega), \\ y|_\Gamma &= u, \end{aligned} \tag{3.2.2}$$

where the space of the test functions is defined as

$$H^1_{0,\Gamma}(\Omega) := \{v \in H^1(\Omega) \,|\, v|_\Gamma = 0\}\ . \tag{3.2.3}$$

The form $\langle\cdot,\cdot\rangle_\Omega$ is the dual form between function spaces on Ω. The bilinear form $a_\Omega(\cdot,\cdot)$ is given by Section 1.3.2 as

$$a_\Omega(v,w) := \int_\Omega (\mathbf{a}\nabla v\cdot\nabla w + a_0\, vw)\, d\mu\ . \tag{3.2.4}$$

3.2.1 The Fictitious Domain–Lagrange Multiplier Approach

Including the essential boundary conditions in our weak formulation, a standard approach was introduced in [4] and is known as the **Lagrange multiplier method**. We will describe it together with a **fictitious domain** scheme.
In the setting from beginning of Section 3.2, we call $\square \subset I\!R^n$ the **fictitious domain**. We **extend** the problem from the domain Ω onto the fictitious domain $\square$ without loosing that our problem is well-defined. Let the right hand side $f_\Omega \in (H^1(\Gamma))'$ be expanded onto the cube $\square$ such that

$$f := f_\square \in (H^1(\square))', \qquad f_\Omega = f_\square|_\Omega\ . \tag{3.2.5}$$

Moreover, we extend the bilinear form $a_\Omega(\cdot,\cdot)$ to $H^1(\square)\times H^1(\square)$ by defining

$$a(v,w) := a_\square(v,w) := \int_\square (\mathbf{a}\nabla v\cdot\nabla w + a_0\, vw)\, d\mu, \tag{3.2.6}$$

where $\mathbf{a}$ is some appropriate extension of $\mathbf{a}$ from (3.2.1) to $\square$ satisfying $\mathbf{a}$ to be uniformly positive definite. We can now turn to the Dirichlet boundary constraints. To express the essential boundary conditions, we need the **trace operator** from Section 1.2.2,

$$\gamma_0 v = v|_\Gamma\ . \tag{3.2.7}$$

The trace is well-defined for any $v \in H^1(\square)$ since Γ is Lipschitzian as a subset of $\partial\Omega$. Thus, we have $\gamma_0 v \in H^{1/2}(\Gamma)$, cf. Theorem 1.20. We define the **bilinear form** $b(\cdot,\cdot)$ by setting

$$b(v,\tilde q) := \langle\gamma_0 v, \tilde q\rangle_{H^{1/2}(\Gamma)\times(H^{1/2}(\Gamma))'} = \int_\Gamma v|_\Gamma\, \tilde q\, ds, \qquad v \in H^1(\square), \tilde q \in (H^{1/2}(\Gamma))', \tag{3.2.8}$$

which is well-defined because of the above remarks.
Instead of Problem 3.14, consider now the following **saddle point problem** of type Problem 3.1:

Problem 3.15 [Elliptic Boundary Value Problem - Saddle Point Formulation]
For $f \in (H^1(\square))'$ and $u \in H^{1/2}(\Gamma)$, find the solution of

$$\inf_{v \in H^1(\Omega)} \sup_{\tilde{q} \in (H^{1/2}(\Gamma))'} \frac{1}{2} a(v,v) - \langle f, v \rangle_{(H^1(\square))' \times H^1(\square)} + b(v, \tilde{q}) - \langle u, \tilde{q} \rangle_{H^{1/2}(\Gamma) \times (H^{1/2}(\Gamma))'} \ . \tag{3.2.9}$$

In this formulation, the essential boundary conditions are not enforced in $H^1(\Omega)$, but appended by the **Lagrangian multiplier** $\tilde{q} \in (H^{1/2}(\Gamma))'$. By design, this technique allows for a decoupling of the differential operator from the boundary constraints. Consequentially, changing boundary conditions or changing boundaries can be treated by updating the right hand side g or by adapting the trace operator γ_0 to a new domain. Since these actions only involve a lower dimensional manifold, this can be done relatively easy compared to the cost a change of the domain Ω would induce. Specifically, changes to the domain Ω excluding Γ have no effect on the setup, as long as the domain is still bounded by the same fictitious domain $\square$. The topology of the fictitious domain is obviously chosen to be as simple as possible to allow for an easy setup and evaluation of the bilinear form (3.2.6).
In the case of the **Dirichlet problem**, the Lagrange multipliers of the solution $p \in (H^{1/2}(\Gamma))'$ can be shown to be the conormal derivative of y at Γ, $p = \mathbf{n} \cdot (\mathbf{a} \nabla y)$. This is often interpreted as the **stress** of the solution at the boundary.
We can now employ the theory from Section 3.1 and derive the optimality conditions of (3.2.9). This yields the following reformulation of our elliptic boundary value problem:

Problem 3.16 [Elliptic Boundary Value Problem - Optimality Conditions]
Given $(f,u) \in (H^1(\square))' \times H^{1/2}(\Gamma)$, find $(y,p) \in H^1(\square) \times (H^{1/2}(\Gamma))'$ such that

$$\begin{aligned} a(y,v) + b(v,p) &= \langle f, v \rangle_{(H^1(\square))' \times H^1(\square)}, && \text{for all } v \in H^1(\square), \\ b(y, \tilde{q}) &= \langle u, \tilde{q} \rangle_{H^{1/2}(\Gamma) \times (H^{1/2}(\Gamma))'}, && \text{for all } \tilde{q} \in (H^{1/2}(\Gamma))', \end{aligned} \tag{3.2.10}$$

holds.

As formulated in the abstract setting from Section 3.1.1, we can write (3.2.10) in operator form. The linear operator A is defined through the bilinear form (3.2.6) analogously to (3.1.13), i.e.,

$$\langle Av, w \rangle_{(H^1(\square))' \times H^1(\square)} := a(v,w) \ . \tag{3.2.11}$$

Again, we drop the spaces from the dual forms if the exact form can be derived immediately. Note that this operator A is **self-adjoint**, i.e., $A' = A$ because of the symmetry of the bilinear form (3.2.6),

$$\langle Av, w \rangle = a(v,w) = a(w,v) = \langle Aw, v \rangle = \langle w, A'v \rangle = \langle A'v, w \rangle \ . \tag{3.2.12}$$

The operator B is set up accordingly using the bilinear form (3.2.8) and the definition of (3.1.14), i.e.,

$$\langle v, B'\tilde{q} \rangle_{H^1(\square) \times (H^1(\square))'} = \langle Bv, \tilde{q} \rangle_{H^{1/2}(\Gamma) \times (H^{1/2}(\Gamma))'} := b(v, \tilde{q}) \ . \tag{3.2.13}$$

This operator is surjective which implies $\ker B' = \{0\}$.

As in Section 3.1, $\mathcal{H} := H^1(\square) \times (H^{1/2}(\Gamma))'$ is the product space of the solution spaces. The elements solving (3.2.9) will be denoted by $(y,p) \in \mathcal{H}$. The right hand side $(f,g) \in \mathcal{H}'$ will be an element of the dual space $\mathcal{H}' = (H^1(\square))' \times H^{1/2}(\Gamma)$. Problem 3.16 written in operator form reads:

Problem 3.17 [Elliptic Boundary Value Problem - Operator Representation]
Given $(f,u) \in \mathcal{H}'$, find $(y,p) \in \mathcal{H}$, such that

$$L \begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ u \end{pmatrix} \ . \tag{3.2.14}$$

This operator L is now obviously also **self-adjoint**, i.e., $L' = L$. However, inherent in the setup is that, because of the nature of the Saddle Point Problem 3.15, the linear system (3.2.14) is **indefinite**. This means that to actually solve Problem 3.17 for any discretization, we have to use different iterative solvers than for system with positive definite system matrices. The most well-known algorithms for such

indefinite symmetric systems are **Uzawa**-type algorithms, see Section 3.3 for the implementation details of those solvers.
It remains to answer the question whether the solution of Problem 3.17 is really a solution of Problem 3.15, when restricted to the domain Ω. This can be answered positively if $\Gamma = \partial\Omega$, see [37]. In case $\Gamma \subset \partial\Omega$, this is no longer automatically valid and still an open question in general. In fact this depends on the way the right hand side extension is constructed, cf. [53].
This will, however, not pose a problem in Section 4.

3.2.2 The Schur Complement

The solution to Problem 3.17 can be expressed analytically using only the operators A and B and the right hand side data. Because of the ellipticity and continuity of $a(\cdot,\cdot)$, the operator A is invertible and we can introduce the **Schur complement**. This is the operator

$$S := BA^{-1}B', \qquad S : (H^{1/2}(\Gamma))' \to H^{1/2}(\Gamma) \tag{3.2.15}$$

which is positive definite, because B is surjective and A is positive definite. It can be used to define an **energy norm** of the operator L on the space $\mathcal{H}$ as

$$\left\| \begin{pmatrix} v \\ \tilde{q} \end{pmatrix} \right\|_L^2 := \|v\|_A^2 + \|\tilde{q}\|_S^2 = \langle Av, v\rangle_{(H^1(\Box))'\times H^1(\Box)} + \langle S\tilde{q}, \tilde{q}\rangle_{H^{1/2}(\Gamma)\times(H^{1/2}(\Gamma))'} \ . \tag{3.2.16}$$

If B is bounded and the inf-sup condition (3.1.19) holds, then

$$\|\tilde{q}\|_S \sim \|\tilde{q}\|_{(H^{1/2}(\Gamma))'}, \tag{3.2.17}$$

see [21] for a proof. Since we already have $\|\cdot\|_A \sim \|\cdot\|_{H^1(\Box)}$, we get the equivalence of the norm $\|\cdot\|_{\mathcal{H}}$ of (3.1.3) and $\|\cdot\|_L$ from (3.2.16). The Schur complement can also be used to rewrite Problem 3.17 by eliminating y as

$$Sp = BA^{-1}f - u \ . \tag{3.2.18}$$

The above remarks show that S is invertible and can be brought to the other side in this equation. Substituting p into the first equation of (3.2.14) leads to an explicit representation of y as

$$\begin{aligned} y &= A^{-1}B'S^{-1}u + A^{-1}(I - B'S^{-1}BA^{-1})f && (3.2.19)\\ &=: A^{-1}(\widetilde{B}'u + \widetilde{f}) \ . && (3.2.20) \end{aligned}$$

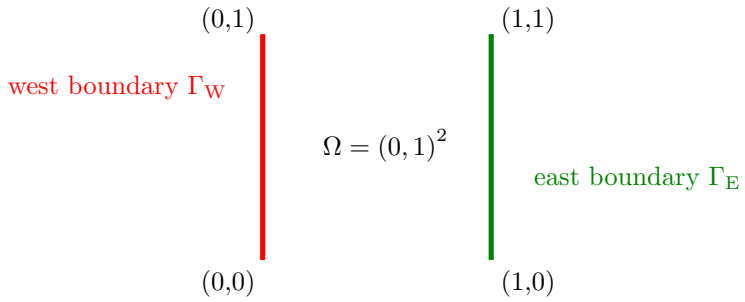
The new operator $\widetilde{B} := S^{-1}B$ obviously inherits the surjectivity property from operator B. The Schur complement also determines the efficiency of the **Uzawa** algorithms of Section 3.3, although of course, on never explicitly calculates A^{-1}.

3.2.3 The Case $\Omega = \Box$, $\Gamma = |$

We will now discuss the elliptic boundary value problem of type Problem 3.14 which will emerge in the setting of our control problem from Section 4.3.
Let $\mathrm{I} = (0,1)$ and $\Omega = \Box = (0,1)^n \subset \mathbb{R}^n$ for a fixed $n \geq 2$. For illustration, we will focus on the case $n = 2$, since it is easier to visualize. This domain has a piecewise smooth boundary $\partial\Omega$, in particular $\partial\Omega \in C^{0,1}$.
The task is now to find a solution to Problem 3.14,

$$\begin{aligned} -\nabla\cdot(\mathbf{a}\nabla y) + a_0\, y &= f, && \text{in } \Omega,\\ y &= u, && \text{on } \Gamma,\\ (\mathbf{a}\nabla y)\cdot\mathbf{n} &= 0, && \text{on } \partial\Omega\setminus\Gamma, \end{aligned} \tag{3.2.21}$$

To solve this problem we clearly do not need to employ of the fictitious domain approach of Section 3.2.1. We assume our Dirichlet boundary Γ to be one of the two opposing faces of the hypercube $\Box$ with $x = (0, x_2, \ldots, x_n)$ and $x = (1, x_2, \ldots, x_n)$ for $0 \leq x_2, \ldots, x_n \leq 1$. These two faces shall be designated Γ_{W} (west) and Γ_{E} (east) respectively, see the following diagram for the case $n = 2$:



The choices of the Dirichlet boundary edges are completely arbitrary and the symmetry of the domain permits immediate transfer of any results to the respective boundary value problems with Γ_N (north) and Γ_S (south) boundaries. This argument obviously also applies to higher dimensions $n > 2$.
We can infer from Section 1.2.2 that the trace operators

$$\gamma_W : H^1(\Omega) \to H^{1/2}(\Gamma_W), \qquad v \mapsto v|_{\Gamma_W} := v(0, x_2, \dots, x_n), \tag{3.2.22}$$
$$\gamma_E : H^1(\Omega) \to H^{1/2}(\Gamma_E), \qquad v \mapsto v|_{\Gamma_E} := v(1, x_2, \dots, x_n), \tag{3.2.23}$$

are well-defined. In the following, we fix one operator and refer to it with the symbol $\gamma_B \in \{\gamma_E, \gamma_W\}$. The Dirichlet boundary $\Gamma \in \{\Gamma_E, \Gamma_W\}$ shall be uniquely determined by this operator.

The ℓ_2-Problem

We now employ the wavelet theory from Section 2 and Section 3.1.2 for our problem. We have constructed wavelet bases $\Psi^{I\!I}, \widetilde{\Psi}^{I\!I}$ in Section 2.3.2 which allow for norm equivalences for the Sobolev spaces ranging from $H^1(\Gamma)'$ to $H^1(\Gamma)$ using the scaled wavelet bases $\Psi^1_\Gamma = \mathbf{D}^{-1}\Psi^{I\!I}$, $\widetilde{\Psi}^1_\Gamma = \mathbf{D}^{+1}\widetilde{\Psi}^{I\!I}$. The wavelet bases $\Psi^1_\Omega := \mathbf{D}^{-1}_\Omega \Psi^{I\!I}_\Omega, \widetilde{\Psi}^1_\Omega := \mathbf{D}^{+1}_\Omega \widetilde{\Psi}^{I\!I}_\Omega$ for the domain Ω are obtained by taking tensor products as introduced in Section 2.4, i.e.,

$$\Psi^{I\!I}_\Omega := \bigotimes_{i=1}^{n} \Psi^{I\!I}, \qquad \widetilde{\Psi}^{I\!I}_\Omega := \bigotimes_{i=1}^{n} \widetilde{\Psi}^{I\!I} \ . \tag{3.2.24}$$

In case $n > 2$ the domain Γ will be constructed by $n-1$ tensor products of the interval I so that we set

$$\Psi^{I\!I}_\Gamma := \bigotimes_{i=2}^{n} \Psi^{I\!I}, \qquad \widetilde{\Psi}^{I\!I}_\Gamma := \bigotimes_{i=2}^{n} \widetilde{\Psi}^{I\!I} \ . \tag{3.2.25}$$

We now present the explicit construction of the linear operator $\mathbf{L}$ of (3.1.26) for our problem. Afterwards, we show that the conditions of Theorem 3.4 are satisfied and therefore an exact solution for our problem exists.
Let $a(\cdot,\cdot)$ be the symmetric continuous elliptic **bilinear form** of Section 1.3.2 given by (3.2.4). Recall the discretized operator $\mathbf{A}$ is called **stiffness matrix** and is given by

$$\mathbf{A} = \langle \Psi^1_\Omega, A\Psi^1_\Omega \rangle = \mathbf{D}^{-1}_\Omega \langle \Psi^{I\!I}_\Omega, A\Psi^{I\!I}_\Omega \rangle \mathbf{D}^{-1}_\Omega = \mathbf{D}^{-1}_\Omega \left(a(\psi, \psi') \right)_{\psi,\psi' \in \Psi^{I\!I}_\Omega} \mathbf{D}^{-1}_\Omega \ . \tag{3.2.26}$$

The trace operator $\mathbf{B}$ in wavelet coordinates is constructed similarly. The bilinear form $b(\cdot,\cdot)$ was introduced in (3.2.8) and the infinite dimensional operator B is, thus,

$$\langle \tilde{q}, Bv \rangle := b(\tilde{q}, v) = \int_\Gamma \gamma_B\, v \cdot \tilde{q}\, ds \ . \tag{3.2.27}$$

The most important detail here is the need for the dual basis $\widetilde{\Psi}^{I\!I}_\Gamma$ since $\tilde{q} \in (H^{1/2}(\Gamma))'$. The wavelet discretized operator $\mathbf{B}$ now has the form

$$\mathbf{B} = \langle \widetilde{\Psi}^{1/2}_\Gamma, B\Psi^1_\Omega \rangle = \mathbf{D}^{1/2}_\Gamma \langle \widetilde{\Psi}^{I\!I}_\Gamma, B\Psi^{I\!I}_\Omega \rangle \mathbf{D}^{-1}_\Omega \ . \tag{3.2.28}$$

The tensor product structure can now be used to further simplify the representation of (3.2.28). Since the trace operator is nothing else than the restriction to a lower dimension along the cartesian product axes, we have

$$\gamma_B \, \Psi_\Omega^{\mathbb{I}} = \left(\gamma_B \, \Psi^{\mathbb{I}}\right) \otimes \bigotimes_{i=2}^{n} \Psi^{\mathbb{I}}, \tag{3.2.29}$$

or in detail for any wavelet $\psi_{\Omega,(j,k)} \in \Psi_\Omega^{\mathbb{I}}$

$$\begin{aligned} \gamma_B \, \psi_{\Omega,(j,k)}(x) &= \gamma_B \, \left(\psi_{j_1,k_1}(x_1) \cdots \psi_{j_n,k_n}(x_n)\right) \\ &= \left(\gamma_B \, \psi_{j_1,k_1}(x_1)\right) \cdot \psi_{j_2,k_2}(x_2) \cdots \psi_{j_n,k_n}(x_n) \\ &= \psi_{j_1,k_1}(x_1)|_\Gamma \cdot \psi_{j_2,k_2}(x_2) \cdots \psi_{j_n,k_n}(x_n) \\ &= \left\{ \begin{array}{c} \psi_{j_1,k_1}(0) \\ \psi_{j_1,k_1}(1) \end{array} \right\} \psi_{j_2,k_2}(x_2) \cdots \psi_{j_n,k_n}(x_n), \qquad \left\{ \begin{array}{l} \text{if } \Gamma = \Gamma_{\mathrm{W}} \\ \text{if } \Gamma = \Gamma_{\mathrm{E}} \end{array} \right. . \end{aligned} \tag{3.2.30}$$

Proposition 3.18 *The trace operator in wavelet coordinates is given by*

$$\mathbf{B} = \left(\psi_{j_1,k_1}|_\Gamma \cdot \delta_{(g_2,\ldots,g_n \, , \, j_2,\ldots,j_n)} \cdot \delta_{(k_2,\ldots,k_n \, , \, l_2,\ldots,l_n)}\right)_{(g,l),(j,k)} \tag{3.2.31}$$

It is uniformly sparse and thus applicable in linear time.

Proof: We can combine (3.2.30) with (3.2.28) and (3.2.27) to calculate the value at any position in the matrix $\mathbf{B}$ as

$$\begin{aligned} \langle \widetilde{\psi}_{\Gamma,(g,l)}, B\psi_{\Omega,(j,k)} \rangle &= \langle \widetilde{\psi}_{\Gamma,(g_2,\ldots,g_n;l_2,\ldots,l_n)}(x_2,\ldots,x_n), B\psi_{\Omega,(j_1,\ldots,j_n;k_1,\ldots,k_n)}(x_1,\ldots,x_n) \rangle \\ &= \int_\Gamma \psi_{j_1,k_1}|_\Gamma \cdot \widetilde{\psi}_{g_2,\ldots,g_n;l_2,\ldots,l_n}(x_2,\ldots,x_n) \cdot \psi_{j_2,\ldots,j_n;k_2,\ldots,k_n}(x_2,\ldots,x_n) \, ds \\ &= \psi_{j_1,k_1}|_\Gamma \cdot \delta_{(g_2,\ldots,g_n \, , \, j_2,\ldots,j_n)} \cdot \delta_{(k_2,\ldots,k_n \, , \, l_2,\ldots,l_n)} \int_\Gamma ds, \end{aligned}$$

where we used biorthogonality $(\mathcal{B})$(2.1.52) on Γ for the last step. The assertion follows because the measure of the boundary Γ is exactly 1. Thus, we have eliminated the need for an explicit representation of the dual wavelets in this construction. ∎

We can conclude straightforward from (3.2.31) that the structure of the trace operator is very similar for any choice of $\gamma_B \in \{\gamma_W, \gamma_E\}$. Indeed, the columns of either one discretized trace operator are just permutated when compared to the other.

Stability Remarks – Stiffness Matrix

We now consider the finite-dimensional problem 3.7 with respect to the wavelet spaces created by $\Psi_{(\Omega,\sigma)} \subset \Psi_\Omega^{\mathbb{I}}$, $\Psi_{(\Gamma,\pi)} \subset \Psi_\Gamma^{\mathbb{I}}$ with the index sets $\mathbb{I}_{\Omega,\sigma} \subset \mathbb{I}_\Omega$, $\mathbb{I}_{\Gamma,\pi} \subset \mathbb{I}_\Gamma$ exactly as in Section 3.1.3.
Recall that if the bilinear form $a(\cdot,\cdot)$ of (3.2.4) is elliptic and continuous, then the Galerkin scheme is $(s,-s)$-stable from Section 2.2.5. The finite-dimensional discretized operator $\mathbf{A}_\sigma$ of (3.2.26) is then given by

$$\mathbf{A}_\sigma = \langle \Psi^1_{(\Omega,\sigma)}, A\Psi^1_{(\Omega,\sigma)} \rangle = \mathbf{D}_\sigma^{-1} \langle \Psi^{\mathbb{I}}_{(\Omega,\sigma)}, A\Psi^{\mathbb{I}}_{(\Omega,\sigma)} \rangle \mathbf{D}_\sigma^{-1} \, . \tag{3.2.32}$$

Remark 3.19 *Numerical studies [52] show that the condition number of the finite discretized differential operators $\mathbf{A}_\sigma$ preconditioned by the application of $\mathbf{D}_{1,\sigma}^{-1}$ of (2.2.14) is indeed* **uniformly bounded**. *Its absolute value can be further reduced by computing the diagonal entries of the unscaled matrix $\langle \Psi^{\mathbb{I}}_{(\Omega,\sigma)}, A\Psi^{\mathbb{I}}_{(\Omega,\sigma)} \rangle$,*

$$\mathbf{D}_a := \left(a(\psi_{j,k}, \psi_{j,k})\right)_{(j,k)\in\mathbb{I}_\Omega} , \tag{3.2.33}$$

and using the matrix with entries

$$\mathbf{D}_{a,\sigma}^{-s} := \left(\left((\mathbf{D}_a)_j\right)^{-s/2} \delta_{(j,j')} \delta_{(k,k')} \right)_{(j,k)\in\mathbb{I}_{\Omega,\sigma},(j',k')\in\mathbb{I}_{\Omega,\sigma}} \tag{3.2.34}$$

as the preconditioning operator in (3.2.32). This operator $\mathbf{D}_{a,\sigma}^{-s}$ can be understood to precondition $\mathbf{A}_\sigma$ in the **energy norm** $\|\cdot\|_A^2 = a(\cdot,\cdot)$ *which explains its effectiveness shown later.*

Stability Remarks – Trace Operator

The **LBB**-condition (3.1.39) is satisfied because the prerequisites of Corollary 3.10 hold. A more direct proof of the validity of the LBB-condition is based on the Trace Theorems from Section 1.2.2 and Theorem 3.11:

Corollary 3.20 *The bilinear form $b(\cdot,\cdot)$ on $H^1(\Omega) \times \left(H^{1/2}(\Gamma)\right)'$ satisfies the* **inf-sup condition** *with the constants from (1.2.28) and (1.2.26)*

$$\inf_{\tilde{q}\in(H^{1/2}(\Gamma))'} \sup_{v\in H^1(\Omega)} \frac{b(v,\tilde{q})}{\|v\|_{H^1(\Omega)}\|\tilde{q}\|_{(H^{1/2}(\Gamma))'}} \geq \frac{1}{C_{T,\Omega}\, C_{E,\partial\Omega}} \ . \tag{3.2.35}$$

Remark 3.21 *This result was proved in [46] in the fictitious domain setting $\Omega \subset \square$ from Section 3.2.1. In that case another constant appears in (3.2.35), which arises from the need to give an upper estimate for the extension $\|\tilde{u}\|_{H^s(\square)}$ by $\|u\|_{H^s(\Omega)}$.*

The **LBB**-condition could also be inferred by this corollary and Theorem 3.11. The required linear projector $\Pi_\sigma : H^1(\Omega) \to X_\sigma = S(\Psi_{(\Omega,\sigma)})$ is naturally given by the L_2-Projector (2.1.44), e.g. $\Pi_\sigma := P_\sigma$. Condition (3.1.40) can then be achieved through biorthogonality (2.1.42).

Numerical Details

Since all our discretized operators are stable, we ascertain from Theorem 3.4 a unique solution $(y_\sigma, p_\pi)^T \in H^1(\Omega) \times \left(H^{1/2}(\Gamma)\right)'$. Furthermore, this implies convergence towards the exact solution, i.e., $(y_\sigma, p_\pi)^T \to (y,p)^T$ with $\sigma,\pi \to \infty$ (cf. (3.1.42) and (3.1.43)).
As mentioned before, the setup of the problem (3.2.21) permits asymptotically optimal preconditioning and the use of fast iterative solvers, see Section 3.3 for implementations. These solvers only need a fixed number of iterations to reduce the residual error by a constant fraction. In every iteration, though, an application of the whole system matrix $\mathbf{L}_{J_\mathcal{H}}$ is needed. Other operations (scalar product, simple floating point operations) are negligible when compared to the amount of work necessary for computing the product $\mathbf{L}_{J_\mathcal{H}}\mathbf{u}_{J_\mathcal{H}}$.
It was shown in Section 3.1.4 how the system matrix can be assembled by using the wavelet transform applied onto the operators in single-scale basis. Because the single-scale operator $\mathbf{A}_{\Phi_\sigma}$, $\mathbf{B}_{\Phi_\sigma,\Phi_\pi}$ are uniformly sparse, the operator $\mathbf{L}_{\Phi,J_\mathcal{H}}$ (3.1.52) can be applied in $\mathcal{O}(\#\mathcal{I}_{\mathcal{H},J_\mathcal{H}})$ floating point operations. Since this is also true for the wavelet transform (3.1.49) and the diagonal matrices (3.1.50), the whole operator $\mathbf{L}_{J_\mathcal{H}}$ can be applied in linear time.
This fact is crucial because this means the above mentioned solvers only need linear time $\mathcal{O}(\#\mathcal{I}_{\mathcal{H},J_\mathcal{H}})$ to reduce the residual error to a fixed fraction of its value.
Also, we can conclude for uniform refinements from Section 2.2.4 together with (2.2.29) and $y \in H^2$ the rate

$$\inf_{v_\sigma \in S(\Psi^1_{(\Omega,\sigma)})} \|y - v_\sigma\|_{H^1} \lesssim 2^{-\sigma}\|y\|_{H^2} \ . \tag{3.2.36}$$

Therefore the (residual) error rate we should be able to achieve is $2^{-\sigma}$, which is equivalent to a factor of $1/2$ per level and thus possible in linear time. The algorithm details will be covered in the next section.

3.3 Algorithms

We now discuss the iterative algorithms used for solving the saddle point problems in Section 3.2. The task is to solve the matrix-vector equation

$$\mathbf{L}_{J_{\mathcal{H}}}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} := \begin{pmatrix}\mathbf{A}_\sigma & \mathbf{B}^T_{\sigma,\pi}\\ \mathbf{B}_{\sigma,\pi} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} = \begin{pmatrix}\mathbf{f}_\sigma\\ \mathbf{u}_\pi\end{pmatrix} \tag{3.3.1}$$

in wavelet coordinates on level $\sigma, \pi \geq j_0$, where we have $\kappa_2(\mathbf{L}_{J_{\mathcal{H}}}) \sim 1$. Let $N_{J_{\mathcal{H}}}$ be the total number of unknowns in the above equation. The matrix $\mathbf{L}_{J_{\mathcal{H}}}$ here is **indefinite**, meaning it has positive and negative eigenvalues. Equation (3.3.1) can be solved using common iteration methods like the CG Solver for the normal equations of (3.3.1), see Section 3.3.1, or Uzawa type algorithms, see Section 3.3.2. The common goal of these algorithms is to produce an element $(\mathbf{y}_\delta, \mathbf{p}_\delta)^T \in \ell_2(I\!I_{X,\sigma} \times I\!I_{Q,\pi})$ for which holds

$$\left\| \mathbf{L}_{J_{\mathcal{H}}}\begin{pmatrix}\mathbf{y}_\delta\\ \mathbf{p}_\delta\end{pmatrix} - \begin{pmatrix}\mathbf{f}_\sigma\\ \mathbf{u}_\pi\end{pmatrix} \right\|_{\ell_2} < \delta\ . \tag{3.3.2}$$

The right choice of the initial starting vector $\mathbf{x}^{(0)}$ can significantly improve the total time the solver needs to calculate a solution up to discretization error accuracy. The simplest starting vector is obviously given by using the vector of only zeros ($\mathbf{0}$) as the initial solution on any level. This is known as **common iteration**. The number of floating point operations necessary to solve (3.3.1) here is

$$\mathcal{O}(N_{J_{\mathcal{H}}} \log N_{J_{\mathcal{H}}}), \tag{3.3.3}$$

see, for example, [60]. Although feasible, a more sophisticated approach than common iteration is given by the **nested iteration** scheme. This can be described as a meta-algorithm acting with the algorithms used to actually solve the system of linear equations. It originates from the fact that the trial spaces are nested (cf. $(\mathcal{R})(2.1.6)$) and thus **prolongation** resp. **restriction** operators exists which can transfer an element in the space S_j to level $j+1$, resp., $j-1$. Hence a calculated solution $\mathbf{x}_j$ is transferred to the next level $j+1$ and used as the starting vector of that level's solver.

Remark 3.22 *In a wavelet framework, we have the space decomposition identity (2.1.12). It can be directly deduced that* **prolongation** *and* **restriction operators** *merely add or delete entries at the end of a vector, or, in the cartesian product setting, also at certain positions inside the vectors. These are obviously extremely easy to implement and involve no floating point operations.*

Algorithm 3.23 $\mathrm{NI}\left[j_0, J, \{Alg_j\}_{j=j_0,\ldots,J}\right] \to \mathbf{x}_J$

(I) `Set` $j := j_0$
$\mathbf{x}_j^{(0)} := \mathbf{0}$

(II) `Repeat`
`Solve` $\mathbf{A}_j\mathbf{x} = \mathbf{b}_j$ `using algorithm` Alg_j `with initial solution` $\mathbf{x}_j^{(0)}$
`obtaining the solution` $\mathbf{x}_j^{(k_j)}$ `up to discretization error accuracy after` k_j `iterations.`
`Prolongate` $\mathbf{x}_j^{(k_j)}$`:`
$\mathbf{x}_{j+1}^{(0)} := \mathbf{x}_j^{(k_j)} \in I\!R^{N_{j+1}}$
`by inserting zeros.`
`Update`
$j \leftarrow j+1$
`Until` $j > J$

(III) `Return` $\mathbf{x}_J^{(k_J)}$

The number of unknowns on the coarsest level j_0 is usually small, depending on the construction. For the wavelets of Section 2.3.2 holds $\#\Delta_{j_0} = 9$. Thus, a direct solver like the **QR**-decomposition can be used on level $j = j_0$. In this way, a very accurate starting vector $\mathbf{x}_{j_0+1}^{(0)}$ is constructed. Since any solution $\mathbf{x}_j^{(k_j)}$ is of discretization error precision, so is the prolongated vector $\mathbf{x}_{j+1}^{(0)}$. The solver thus needs only to improve the quality of the solution by a constant factor on each level. This leads to the following important result.

Theorem 3.24 *The nested iteration algorithm computes the solution on the finest level J with a complexity of $\mathcal{O}(N_J)$ floating point operations.*

Proof: Optimal preconditioning makes it possible to achieve a reduction of the error by a constant factor only requiring a constant number of iteration steps independent of the level. Each step is executed with $\mathcal{O}(N_j)$ operations. Combined with a geometric series argument over the number of unknowns N_j (cf. Proposition 2.11),

$$\mathcal{O}(\sum_{j=j_0}^{J} N_j) = \mathcal{O}(N_J),$$

proves the claim. ∎

Next, we present two classes of iterative solvers which can be used to solve (3.3.1).

3.3.1 Conjugate Gradient Solver

The **Conjugate Gradient** (CG) Solver is the most well-known algorithm used for solving a system of linear equations with a symmetric positive definite system matrix iteratively. We list this algorithm for completeness.

Algorithm 3.25 $\mathrm{CG}\big[\delta, \mathbf{A}, \mathbf{x}^{(0)}, \mathbf{b}\big] \to \mathbf{x}_\delta$

(I) `Set`
$k := 0$
$\mathbf{r}^{(0)} := \mathbf{A}\,\mathbf{x}^{(0)} - \mathbf{b}$
$\mathbf{d}^{(1)} := -\mathbf{r}^{(0)}$
$\mathbf{q}^{(1)} := \mathbf{0}$
$\gamma^{(0)} := \big\langle \mathbf{r}^{(0)}, \mathbf{r}^{(0)} \big\rangle$

(II) `While` $\|\mathbf{r}^k\|_{\ell_2} > \delta$
$k \leftarrow k+1$
$\mathbf{q}^{(k)} \leftarrow \mathbf{A}\,\mathbf{d}^{(k)}$
$\alpha^{(k)} \leftarrow \gamma^{(k-1)} / \big\langle \mathbf{d}^{(k)}, \mathbf{q}^{(k)} \big\rangle$
$\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} + \alpha^{(k)} \mathbf{d}^{(k)}$
$\mathbf{r}^{(k)} \leftarrow \mathbf{r}^{(k-1)} + \alpha^{(k)} \mathbf{q}^{(k)}$
$\gamma^{(k)} \leftarrow \big\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \big\rangle$
$\beta^{(k)} \leftarrow \gamma^{(k)} / \gamma^{(k-1)}$
$\mathbf{d}^{(k)} \leftarrow -\mathbf{r}^{(k)} + \beta^{(k)} \mathbf{d}^{(k-1)}$

(III) `Return` $x^{(k)}$

The CG scheme is known to be **optimal** in the sense that it calculates the solution with the amount of steps equal to the number of unknowns if all calculations were done with infinite precision. Solving a linear equation stemming from a discretization of finite precision, one cannot expect an infinitely precise solution. Rather, we solve the equation up to discretization error accuracy. This can obviously be done in fewer steps than the number of unknowns even though computers only calculate with machine precision and introduce a rounding error in every calculation.
The convergence speed of the above algorithm determines the number of steps necessary to reduce the residual error $\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\|_{\ell_2}$ up to the discretization error. Let $\mathbf{x}^*$ be the exact solution. Then the error $\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_{\mathbf{A}}$ in energy norm decreases by a factor of

$$\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} < 1 \tag{3.3.4}$$

in every step, where $\kappa_2(\cdot)$ is the **spectral condition**. Having optimally preconditioned operators is thus vital of calculating the solution efficiently since then

$$\|\mathbf{x}^* - \mathbf{x}^{(k)}\|_{\mathbf{A}} \sim \|\mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}\|_{\ell_2} = \|\mathbf{r}^{(k)}\|_{\ell_2} \sim \|\mathbf{x}^* - \mathbf{x}^{(k)}\|_{\ell_2},$$

where the constant for each of the upper equivalences is $\|\mathbf{A}\|$ and the constant of the lower ones is $\|\mathbf{A}^{-1}\|^{-1}$. This means that the quality (cf. (2.2.37)) of the equivalences is determined by $\kappa_2(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$. In practice, the residual does not fall strictly with the value of (3.3.4) in every step.

3.3.2 Uzawa Algorithms

The details of this section are borrowed from [8]. Uzawa algorithm are iterative solvers for **saddle point problems**. These algorithms generally have a lower complexity than CG applied to the normalized equation $\mathbf{L}^T\mathbf{L}\,\mathbf{x} = \mathbf{L}^T\mathbf{b}$. Conceptually, it can be written as

$$\left.\begin{array}{rcl} \mathbf{A}\,\mathbf{y}^{(k)} & = & \mathbf{f} - \mathbf{B}^T\mathbf{p}^{(k-1)}, \\ \mathbf{p}^{(k)} & = & \mathbf{p}^{(k-1)} + \alpha(\mathbf{B}\,\mathbf{y}^{(k)} - \mathbf{u}), \end{array}\right\} k = 1, 2, \dots \ . \tag{3.3.5}$$

The stepping parameter α has to be small enough to ensure convergence depending on the spectral properties of $\mathbf{A}$ and $\mathbf{B}$. Since it also determines the convergence speed, α must be chosen as large as possible in applications. It is known (see [21]) that the upper bound for α is given by the norm of the **Schur complement** (3.2.15) as

$$\alpha < \frac{2}{\|\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\|} \ . \tag{3.3.6}$$

Assuming the vector $\mathbf{y}^{(k)}$ is calculated analytically yields the representation for the residual error

$$\|\mathbf{L}\mathbf{x}^{(k)} - \mathbf{b}\|_{\ell_2} = \|\mathbf{B}\,\mathbf{y}^{(k)} - \mathbf{u}\|_{\ell_2} =: \|\mathbf{q}^{(k)}\|_{\ell_2} \tag{3.3.7}$$

in every step.

Uzawa Algorithm

The standard Uzawa algorithm uses the formula

$$\alpha^{(k)} = \frac{\left\langle \mathbf{q}^{(k)}, \mathbf{q}^{(k)} \right\rangle}{\left\langle \mathbf{q}^{(k)}, (\mathbf{B}\,\mathbf{A}^{-1}\,\mathbf{B}^T)\mathbf{q}^{(k)} \right\rangle} \tag{3.3.8}$$

to calculate the stepping parameter in every iterations. The inversion of the matrix $\mathbf{A}$ is done here using the CG solver and is thus only executed approximatively. For equation (3.3.7) to hold, this inner iteration must obviously be carried out with smaller tolerance than the outer iteration.

Algorithm 3.26 UZAWA$\left[\delta, \{\delta_i\}_{i=0,\dots}, \mathbf{L}, \binom{\mathbf{y}^{(0)}}{\mathbf{p}^{(0)}}, \binom{\mathbf{f}}{\mathbf{u}}\right] \to \binom{\mathbf{y}_\delta}{\mathbf{p}_\delta}$

(I) `Set` $k := 0$
`Update` $\mathbf{y}^{(1)} \leftarrow \mathtt{CG}\left[\delta_0, \mathbf{A}, \mathbf{y}^{(0)}, \mathbf{f} - \mathbf{B}^T\,\mathbf{p}^{(0)}\right]$
`Set` $\mathbf{h}^{(1)} := \mathbf{0}$
$\mathbf{j}^{(1)} := \mathbf{0}$
$\mathbf{q}^{(1)} := \mathbf{0}$

(II) `Repeat`
$k \leftarrow k + 1$
$\mathbf{q}^{(k)} \leftarrow \mathbf{u} - \mathbf{B}\,\mathbf{y}^{(k)}$
$\mathbf{j}^{(k)} \leftarrow \mathbf{B}^T\,\mathbf{q}^{(k)}$
$\mathbf{h}^{(k)} \leftarrow \mathtt{CG}\left[\delta_k, \mathbf{A}, \mathbf{h}^{(k-1)}, \mathbf{j}^{(k)}\right]$
$\alpha^{(k)} \leftarrow \left\langle \mathbf{q}^{(k)}, \mathbf{q}^{(k)} \right\rangle / \left\langle \mathbf{j}^{(k)}, \mathbf{h}^{(k)} \right\rangle$
$\mathbf{p}^{(k)} \leftarrow \mathbf{p}^{(k-1)} - \alpha^{(k)}\,\mathbf{q}^{(k)}$
$\mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} - \alpha^{(k)}\,\mathbf{h}^{(k)}$
`Until` $\|\mathbf{q}^{(k)}\|_{\ell_2} \le \delta$

(III) `Return` $\left(\mathbf{y}^{(k)}, \mathbf{p}^{(k)}\right)^T$

The convergence rate of this algorithm is determined by the spectral condition of the Schur complement,

$$\theta_{\text{Uzawa}} := \frac{\sqrt{\kappa_2(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)} - 1}{\sqrt{\kappa_2(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)} + 1} < 1 \ . \tag{3.3.9}$$

Consequently, if the Schur complement has a uniformly bounded condition number, then only a fixed number of iterations is needed to reduce the error by a fixed fraction.
Since the spectral condition of the Schur complement is generally large, it can be very good for performance to introduce **conjugate directions** for the outer iteration as well.

Uzawa Algorithm with Conjugate Directions

The following version, also taken from [8], uses slightly more memory in the form of one additional vector $\mathbf{d}^{(k)}$. The benefits of increases in speed can outweight the drawback of higher memory usage.

Algorithm 3.27 UzawaCD$\left[\delta, \{\delta_i\}_{i=0,\ldots}, \mathbf{L}, \binom{\mathbf{y}^{(0)}}{\mathbf{p}^{(0)}}, \binom{\mathbf{f}}{\mathbf{u}}\right] \rightarrow \binom{\mathbf{y}_\delta}{\mathbf{p}_\delta}$

(I) `Set` $k := 0$
`Update` $\mathbf{y}^{(1)} \leftarrow \mathtt{CG}\left[\delta_0, \mathbf{A}, \mathbf{y}^{(0)}, \mathbf{f} - \mathbf{B}^T\,\mathbf{p}^{(0)}\right]$
`Set` $\mathbf{d}^{(1)} := \mathbf{B}\,\mathbf{y}^{(1)} - \mathbf{u}$
$\mathbf{h}^{(1)} := \mathbf{0}$
$\mathbf{j}^{(1)} := \mathbf{0}$
$\mathbf{q}^{(1)} := -\mathbf{d}^{(1)}$

(II) `Repeat`
$k \leftarrow k+1$
$\mathbf{j}^{(k)} \leftarrow \mathbf{B}^T\,\mathbf{d}^{(k)}$
$\mathbf{h}^{(k)} \leftarrow \mathtt{CG}\left[\delta_k, \mathbf{A}, \mathbf{h}^{(k-1)}, \mathbf{j}^{(k)}\right]$
$\alpha^{(k)} \leftarrow \left\langle \mathbf{q}^{(k)}, \mathbf{q}^{(k)} \right\rangle / \left\langle \mathbf{j}^{(k)}, \mathbf{q}^{(k)} \right\rangle$
$\mathbf{p}^{(k)} \leftarrow \mathbf{p}^{(k-1)} - \alpha^{(k)}\,\mathbf{d}^{(k)}$
$\mathbf{y}^{(k+1)} \leftarrow \mathbf{y}^{(k)} - \alpha^{(k)}\,\mathbf{h}^{(k)}$
$\mathbf{q}^{(k+1)} \leftarrow \mathbf{u} - \mathbf{B}\,\mathbf{y}^{(k+1)}$
$\beta^{(k)} \leftarrow \left\langle \mathbf{q}^{(k+1)}, \mathbf{q}^{(k+1)} \right\rangle / \left\langle \mathbf{q}^{(k)}, \mathbf{q}^{(k)} \right\rangle$
$\mathbf{d}^{(k+1)} \leftarrow -\mathbf{q}^{(k+1)} + \beta^{(k)}\,\mathbf{d}^{(k)}$
`Until` $\|\mathbf{q}^{(k+1)}\|_{\ell_2} \leq \delta$

(III) `Return` $\left(\mathbf{y}^{(k+1)}, \mathbf{p}^{(k)}\right)^T$

The error bounds δ_i for the inner matrix inversion are chosen smaller than δ, ususally by a fixed factor like 1/4 or 1/8 for all δ_i.

4 Control Problems

Control theory in the sense used here is concerned with the optimization of some given objective function under a constraint. First we will need to discuss existence and uniqueness of the minimizing element. After this is done, theory of calculus of variations is employed to deduce the necessary and sufficient conditions to characterize the optimum.
Here, the functional to be minimized will act as an indicator of smoothness and regularity for elements of function spaces. The constraints can generally be any correlation between the functions on which the functional depends. Correlations stated by a partial differential equation, be it the elliptic, parabolic or hyperbolic case, hereby pose a particular challenge for numerics.
One of the first works on this topic was by J. L. Lions [49] in 1970, where a complete overview of the subject was literally put down for the first time. The article [40] of W. Hackbusch in 1980 can be seen as the predecessor to this work because of its emphasis on fast numerical solutions to elliptic control problems. In that work, multigrid methods were used to solve the elliptic partial differential equations of the resulting optimality conditions. The advances of the wavelet construction process during the last 18 years described in Section 2 and their successful application to elliptic partial differential equations allow for employing wavelet techniques for such problems today.
Wavelet approaches to optimal control problems can be found in [46] and [47]. This thesis will complement these papers as it will present numerical results of an important practical example problem on this topic. The algorithms proposed there are proved to work in an application. We also extend the wavelet theory a bit further by rigorously incorporating Riesz operators into the wavelet discretization.

4.1 Introduction

This section gives a brief definition to control problems, providing the most important vocabulary regarding this matter. After the general introduction to control problems with elliptic PDEs follows the specific control problem which is assembled, solved and examined in this thesis.
The general **control problem** refers to the following setting (see [49]):
Given the following ingredients:

- **Controls** u belonging to some space $\mathcal{U}_{ad}$ consisting of all **admissible controls**;
- The **state** $y(u)$ of the system to be controlled is given as the solution of a stationary partial differential equation
$$\mathbf{A}\, y(u) = f(u) \tag{4.1.1}$$
with a given right hand side f and an operator $\mathbf{A}$ called the **model** of the system;
- The **observation** $z(u)$, which is a function of $y(u)$;
- The **cost functional** $\mathcal{J}(u) = \Phi(z(u))$ defined in terms of the observation as a function $z \mapsto \Phi(z) \geq 0$.

The task is to find the function $u \in \mathcal{U}_{ad}$ which minimizes the cost functional

$$\inf_{u \in \mathcal{U}_{ad}} \mathcal{J}(u). \tag{4.1.2}$$

A control $u \in \mathcal{U}_{ad}$ is termed **optimal control**, with respect to $\inf \mathcal{J}(u)$ if it is the infimum of the functional.

Remark 4.1 *Problems in optimal control arise in a number of scientific fields, ranging from biology to physics and economics, see the books [7, 43] for a vast number of diverse examples.*

Depending on the exact form of the functional, there can be either none, exactly one, several or infinitely many optimal controls. Certainly one must identify which situation applies before attempting to apply numeric schemes. Assuming the existence of one unique optimum, it must be characterized in order to be determined. Usually the equations expressing the minimizing element cannot be solved analytically so that numerical methods need to be employed. Discretizations naturally only work on finite subspaces of the spaces in which the control u and the state y exist, so convergence to the solution of the control problem on infinite spaces must be proved. One of the objectives in this setting is to construct algorithms which yield the best approximations to u and y with as little computational work as necessary.

4.2 Linear-Quadratic Control Problems Governed by Elliptic PDEs

This chapter is mostly based on [46] and [47]. Quotations are marked and proofs only given if they differ in details from the sources or if it is helpful for understanding of the overall picture.
Let $\Omega \subseteq \square \subset I\!R^n$ be a domain bounded with a Lipschitz continuous boundary $\partial\Omega \in C^{0,1}$. The spaces X,Q and Y are to be **Hilbert spaces**.
The general type of problem that we are going to discuss here is termed **linear-quadratic elliptic control problem**, because a quadratic functional is minimized under the constrains of a linear elliptic partial differential equation:

Problem 4.2 [Control Problem with Elliptic PDE Constraints]
Let functions f,y_* and a parameter $\omega > 0$ be given. Minimize the **cost functional**

$$\mathcal{J}(y,u) = \frac{1}{2}\|y - y_*\|_Y^2 + \frac{\omega}{2}\|u\|_Q^2, \tag{4.2.1}$$

where the **state** y *and the* **control** u *are coupled by the elliptic problem (the* **model**)

$$\mathcal{L}\, y \quad = \quad f(u), \qquad\qquad \text{in } \Omega\ . \tag{4.2.2}$$

Here the operator $\mathcal{L}$ is a partial differential operator of elliptic type of order $2m$ (cf.(1.3.2)). The smoothness we can expect for the state is thus $y \in H^m(\Omega)$.
The functional (4.2.1) consists of two quadratical terms, which serve completely different purposes. The first term is called **data fitting term**, since it forces y to approximate to the **observation** y_* in the space Y. The second term is entitled **regularization term**, because it enforces regularity of the control. The parameter $\omega > 0$ weights the importance of the two terms.

Remark 4.3 *It should be noted that our problem 4.2 is only well defined when $\omega > 0$. For $\omega \to 0$ the problem becomes ill-defined and (as we shall see) harder to solve. In case $\omega = 0$ the solution state is determined by the right hand side functions f, y_* alone. In general, the control may not be unique in this case, but rather an element of a subspace $\mathcal{U} \subset \mathcal{U}_{ad}$, see [39].*

The observation y_* can either be given on any part of the domain Ω or on its boundary $\partial\Omega$. This means the space Y could either be a Sobolev space on a yet to be specified part of the boundary $\partial\Omega$ or the whole domain Ω. The same holds true for the space Q. The control could either be exerted on the domain Ω or on a patch of the boundary $\Gamma \subset \partial\Omega$. The former setup of **distributed control** is investigated in [11,12]. Here we focus on the latter problem of the practically most improtant **boundary control**.
The above mentioned regularity of y implicates that we have $y|_{\partial\Omega} \in H^{m-1/2}(\partial\Omega)$. This entails that the natural norms in (4.2.1) are fractional Sobolev spaces' norms on the boundary of Ω. So we will have to deal with norms of spaces, that are not only difficult to evaluate but not even uniquely defined, see Section 1.2.
The target is thus not so much to compute the exact quantity of the optimization functional but to control the qualitative behavior of the minimization. We will formulate the control problem in a wavelet setting, which uses norms that are equivalent to the natural norms of (4.2.1).

4.2.1 Dirichlet Problems with Boundary Control

To formalize the above remarks, we choose our partial differential constraints (4.2.2) to be the **boundary value problem** 3.14. The space Q is now a function space on the border patch $\Gamma \subset \partial\Omega$ and Y a space on $\Gamma_Y \subset \partial\Omega$.
The elliptic boundary value problem will be used in the operator formulation (3.2.14). Recall that the elliptic operator $A : X \to X'$ and the trace operator $B : X \to Q$ are well-defined, possibly in a fictitious domain setting (see Section 3.2.1), and Theorem 3.4 holds.
The analytical norms in (4.2.1) will now be replaced by **equivalent** norms from wavelet theory (see Theorem 2.17):

$$\|\cdot\|_{\mathbf{Y}}^2 := (\cdot,\cdot)_{\mathbf{Y}} \sim \|\cdot\|_Y^2, \qquad \|\cdot\|_{\mathbf{Q}}^2 := (\cdot,\cdot)_{\mathbf{Q}} \sim \|\cdot\|_Q^2\ . \tag{4.2.3}$$

Furthermore, let

$$T : X \to Y \tag{4.2.4}$$

be the linear continuous trace operator, which maps the space X on Ω onto Y living on Γ_Y.

Problem 4.4 [Control Problem with Boundary Control]
Given $f \in X'$, $y_* \in Y$, find $(y,p,u) \in X \times Q' \times Q$, such that

$$\mathcal{J}(y,u) = \frac{1}{2}\|Ty - y_*\|^2_{\mathbf{Y}} + \frac{\omega}{2}\|u\|^2_{\mathbf{Q}} \tag{4.2.5}$$

is minimized subject to

$$L\begin{pmatrix} y \\ p \end{pmatrix} := \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ u \end{pmatrix} . \tag{4.2.6}$$

The space of admissible controls in this problem is taken as $\mathcal{U}_{ad} := Q$. This problem will now be solved by the **Lagrangian multiplier method**. Recall that $D^s\mathcal{J}(u;v_1,\ldots,v_n)$ is the $s-th$ variation of $\mathcal{J}$ at u in directions $v_1,\ldots,v_n$ and for $s=1$ the derivative at u in direction v is defined as

$$D\mathcal{J}(u;v) = \langle \delta\mathcal{J}(u), v\rangle := \lim_{t\to 0}\frac{\mathcal{J}(u+tv)-\mathcal{J}(u)}{t} . \tag{4.2.7}$$

In order to give necessary and sufficient criteria for a minimum of the functional (4.2.5), we append the boundary value problem (4.2.6) to the functional and define

$$\textsc{Lagr}(y,p,u,z,\mu) := \frac{1}{2}\|Ty - y_*\|^2_{\mathbf{Y}} + \frac{\omega}{2}\|u\|^2_{\mathbf{Q}} + \left\langle \begin{pmatrix} z \\ \mu \end{pmatrix}, L\begin{pmatrix} y \\ p \end{pmatrix} - \begin{pmatrix} f \\ u \end{pmatrix}\right\rangle \tag{4.2.8}$$

on $X \times Q' \times Q \times X \times Q'$.
The connection between this Lagrangian functional and Problem 4.4 is established by a theorem from [46]:

Theorem 4.5 *Suppose that the following two conditions hold:*

(i) $\mathcal{J} : X \times Q' \times Q \to I\!R$ *is differentiable at* $x^* \in X \times Q' \times Q$,

(ii) $\mathcal{G}(y,p,u) := L\begin{pmatrix} y \\ p \end{pmatrix} - \begin{pmatrix} f \\ u \end{pmatrix} : X \times Q' \times Q \to X' \times Q$ *is a* **submersion** *(see [62]) at* x^*, *i.e.,* $\mathcal{G}$ *is a smooth map and its Jacobian is surjective at every* $x \in X \times Q' \times Q$.

Then, if $\mathcal{J}$ *has a local minimum at* x^* *with respect to the solution space*

$$\mathcal{K} := \{(y,p,u) \in X \times Q' \times Q \,|\, \mathcal{G}(y,p,u) = 0\}, \tag{4.2.9}$$

then Lagr *satisfies the necessary conditions*

$$\delta\textsc{Lagr}(x^*) = 0 . \tag{4.2.10}$$

If $\mathcal{J}$ *is strongly convex, then (4.2.10) is also sufficient for the unique minimum of Problem 4.4 to be attained at* x^*.

Before investigating the implications of this theorem, we reformulate the functional (4.2.5) using identity (3.2.19), for which we now assume the operator A to be invertible,

$$\mathcal{J}(u) = \mathcal{J}(y(u)) = \frac{1}{2}\|TA^{-1}(\widetilde{B}'u + \widetilde{f}) - y_*\|^2_{\mathbf{Y}} + \frac{\omega}{2}\|u\|^2_{\mathbf{Q}} . \tag{4.2.11}$$

Proposition 4.6 *The functional (4.2.11) is twice differentiable on* Q *with derivatives*

$$\begin{aligned} D\mathcal{J}(u;v) &= \left(Ty(u) - y_*, TA^{-1}\widetilde{B}'v\right)_{\mathbf{Y}} + \omega\,(u,v)_{\mathbf{Q}} \\ &= \left(TA^{-1}(\widetilde{B}'u + \widetilde{f}) - y_*, TA^{-1}\widetilde{B}'v\right)_{\mathbf{Y}} + \omega\,(u,v)_{\mathbf{Q}} \end{aligned} \tag{4.2.12}$$

for all $v \in Q$ *and*

$$D^2\mathcal{J}(u;v,w) = \left(TA^{-1}\widetilde{B}'v, TA^{-1}\widetilde{B}'w\right)_{\mathbf{Y}} + \omega\,(v,w)_{\mathbf{Q}} \tag{4.2.13}$$

for all $v,w \in Q$. *In particular,* $\mathcal{J}$ *is* **strictly convex** *on* Q *for* $\omega \neq 0$, *e.g.*

$$D^2\mathcal{J}(u;v,v) > 0, \qquad \text{for all } v \in Q \setminus \{0\} . \tag{4.2.14}$$

Proof: The proof is adapted from [46]. For any $u, v \in Q$ and $t > 0$, one has for $\mathcal{J}$ from (4.2.11):

$$\frac{\mathcal{J}(u+tv) - \mathcal{J}(u)}{t} = \frac{1}{2}\left(2\Big(Ty(u) - y_*, TA^{-1}\widetilde{B}'v\Big)_{\mathbf{Y}} + t\,\|TA^{-1}\widetilde{B}'v\|^2_{\mathbf{Q}}\right) + \frac{\omega}{2}\left(2(u,v)_{\mathbf{Q}} + t\,\|v\|^2_{\mathbf{Q}}\right),$$

yielding

$$D\mathcal{J}(u;v) = \lim_{t\to 0}\frac{\mathcal{J}(u+tv)-\mathcal{J}(u)}{t} = \Big(Ty(u) - y_*, TA^{-1}\widetilde{B}'v\Big)_{\mathbf{Y}} + \omega\,(u,v)_{\mathbf{Q}}$$

and, thus, (4.2.12) upon inserting (4.2.11). Furthermore, let $\widehat{\mathcal{J}}(u;v) := \langle \delta\mathcal{J}(u), v\rangle$. Then

$$\frac{\widehat{\mathcal{J}}(u+tw;v) - \widehat{\mathcal{J}}(u;v)}{t} = \Big(TA^{-1}\widetilde{B}'v, TA^{-1}\widetilde{B}'w\Big)_{\mathbf{Y}} + \omega(v,w)_{\mathbf{Q}}$$

is independent of u, yielding identity (4.2.13) for all $v, w \in Q$. In particular, one has

$$D^2\mathcal{J}(u;v,v) = \|TA^{-1}\widetilde{B}'v\|^2_{\mathbf{Y}} + \omega\|v\|^2_{\mathbf{Q}} \geq \omega\|v\|^2_{\mathbf{Q}} > 0$$

for all nonzero $v \in Q$ and $\omega \neq 0$ (cf. Remark 4.3). ∎

Let us introduce Riesz maps (cf. (2.2.38)) $R_{\mathbf{Y}} : Y \to Y'$, $R_{\mathbf{Q}} : Q \to Q'$ defined by

$$\langle R_{\mathbf{Y}}v, w\rangle := \langle R_{\mathbf{Y}}v, w\rangle_{Y\times Y'} := (v,w)_{\mathbf{Y}}, \qquad \text{for all } v, w \in Y, \tag{4.2.15}$$

$$\langle R_{\mathbf{Q}}v, w\rangle := \langle R_{\mathbf{Q}}v, w\rangle_{Q\times Q'} := (v,w)_{\mathbf{Q}}, \qquad \text{for all } v, w \in Q\ . \tag{4.2.16}$$

The norms in the Lagrangian functional (4.2.8) can be represented by using these operators as

$$\textsc{Lagr}(y,p,u,z,\mu) = \frac{1}{2}\langle R_{\mathbf{Y}}(Ty - y_*), Ty - y_*\rangle + \frac{\omega}{2}\langle R_{\mathbf{Q}}u, u\rangle + \left\langle \begin{pmatrix} z \\ \mu \end{pmatrix}, L\begin{pmatrix} y \\ p \end{pmatrix} - \begin{pmatrix} f \\ u \end{pmatrix}\right\rangle\ . \tag{4.2.17}$$

The first prerequisite of Theorem 4.5 is obviously guaranteed by Proposition 4.6. To check that $\mathcal{G}$ is a submersion, we need to analyze its Jacobian. This is the matrix

$$\left(\begin{array}{c|c} L & \begin{matrix} 0 \\ -I \end{matrix} \end{array}\right) = \left(\begin{array}{cc|c} A & B' & 0 \\ B & 0 & -I \end{array}\right) \in L(X \times Q' \times Q; X' \times Q)\ . \tag{4.2.18}$$

Since we have assured by the formulation of the PDE as a saddle point problem the applicability of Theorem 3.4, we have that L is invertible and, concordantly, this mapping is surjective.

Lemma 4.7 *Let $\mathcal{J}$ be the functional in (4.2.11), let $f \in X'$ and $y_* \in Y$ be given. Then the Euler equations resulting from (4.2.10) for the minimization problem Problem 4.4 are*

$$\begin{aligned} Ay + B'p - f &= 0, \\ By - u &= 0\ . \\ \omega R_{\mathbf{Q}}u - \mu &= 0, \\ T'R_{\mathbf{Y}}Ty - T'R_{\mathbf{Y}}y_* + A'z + B'\mu &= 0, \\ Bz &= 0, \end{aligned} \tag{4.2.19}$$

Proof: According to Theorem 4.5, we need to determine the Euler equations (4.2.10),

$$\delta\textsc{Lagr}(y,p,u,z,\mu;r) = 0 \qquad \text{for } r = y,p,u,z,\mu\ . \tag{4.2.20}$$

Each one of these conditions yields one of the above equations. Since we have shown in Proposition 4.6 that $\mathcal{J}$ is convex, other sufficient conditions need not be taken into account. ∎

The system of equations (4.2.19) are equivalent to the following linear system

$$\begin{aligned} NU &:= \begin{pmatrix} L & E \\ \widehat{E} & L' \end{pmatrix}\begin{pmatrix} y \\ p \\ z \\ \mu \end{pmatrix} \\ &:= \left(\begin{array}{cc|cc} A & B' & 0 & 0 \\ B & 0 & 0 & -\omega^{-1}R_{\mathbf{Q}}^{-1} \\ \hline T'R_{\mathbf{Y}}T & 0 & A' & B' \\ 0 & 0 & B & 0 \end{array}\right)\begin{pmatrix} y \\ p \\ z \\ \mu \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ T'R_{\mathbf{Y}}y_* \\ 0 \end{pmatrix} =: F \end{aligned} \tag{4.2.21}$$

by using the third equation to eliminate $u = \omega^{-1} R_{\mathbf{Q}}^{-1} \mu$ in the last equation.

Remark 4.8 *Note that (4.2.21) can be interpreted as solving simultaneously two systems of saddle point problems, namely, the* **primal system** *(4.2.6),*

$$L\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ u \end{pmatrix}, \tag{4.2.22}$$

and the **adjoint system**

$$L'\begin{pmatrix} z \\ \mu \end{pmatrix} = \begin{pmatrix} A' & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} z \\ \mu \end{pmatrix} = \begin{pmatrix} -T' R_{\mathbf{Y}} (Ty - y_*) \\ 0 \end{pmatrix} \tag{4.2.23}$$

for $(z, \mu) \in X \times Q'$. The variable z is called **adjoint state variable**. *The central third equation in (4.2.19), establishing the coupling between the primal and the adjoint system, is called* **design equation**.

If the adjoint system is solved analytically or numerically using a direct solver, one obtains an explicit expression for the control u in terms of the state y, which involves the Schur complement and the inverse of A. Of course, this scheme has the disadvantage of being too costly in numerical applications because of the properties of the Schur complement (see (3.2.15)). The above formulation (4.2.21) circumvents these drawbacks because A^{-1} does not need to be computed explicitly here. The following proposition shows an important property of this primal/adjoint system characterization.

Proposition 4.9 *We can infer from Proposition 4.6 directly that the first variation of $\mathcal{J}$ is given by*

$$\delta \mathcal{J}(u) = \widetilde{B} (A^{-1})' T' R_{\mathbf{Y}} (Ty - y_*) + \omega R_{\mathbf{Q}} u \ . \tag{4.2.24}$$

An equivalent representation is

$$\delta \mathcal{J}(u) = \omega R_{\mathbf{Q}} u - \mu, \tag{4.2.25}$$

that is, the evaluation of $\delta \mathcal{J}(u)$ is equivalent to solving first (4.2.22), followed by (4.2.23).

Proof: The variable μ in (4.2.23) has by (3.2.18) the explicit form

$$\mu = -S^{-1} B A^{-1} \left(T' R_{\mathbf{Y}} (Ty - y_*) \right),$$

from which (4.2.25) by (4.2.24) follows directly with $\widetilde{B} = S^{-1} B$ and the **self-adjointness** of A. ∎

The next corollary establishes the operator N from (4.2.21) as an isomorphism, which, combined with stable wavelet discretizations, leads to an optimal preconditioned ℓ_2-operator in the next section.

Corollary 4.10 *Let A and B fulfill the prerequisites of Theorem 3.4. Then N is an isomorphism*

$$N : \mathcal{N} \longrightarrow \mathcal{N}', \qquad \text{with } \mathcal{N} := X \times Q' \times X \times Q', \tag{4.2.26}$$

satisfying the norm equivalence

$$\|V\|_{\mathcal{N}} \sim \|NV\|_{\mathcal{N}'} \qquad \text{for any } V \in \mathcal{N} \ . \tag{4.2.27}$$

Proof: The invertibility of N follows from Theorem 4.5 and Proposition 4.6 since $\mathcal{G}$ is, under the conditions on A and B, a submersion. One now has to check the lower and upper estimates of (4.2.27). We have for $V = (y, p, z, \mu) \in \mathcal{N}$ by definition of the norm

$$\begin{aligned} \|NV\|_{\mathcal{N}'}^2 &= \left\| \begin{pmatrix} L\binom{y}{p} + E\binom{z}{\mu} \\ \widehat{E}\binom{y}{p} + L'\binom{z}{\mu} \end{pmatrix} \right\|_{\mathcal{N}'}^2 = \left\| \begin{pmatrix} L\binom{y}{p} + \binom{0}{-\omega^{-1} R_{\mathbf{Q}}^{-1} \mu} \\ T' R_{\mathbf{Y}} T \binom{y}{0} + L'\binom{z}{\mu} \end{pmatrix} \right\|_{\mathcal{N}'}^2 \\ &\lesssim \left\| L\begin{pmatrix} y \\ p \end{pmatrix} \right\|_{X' \times Q}^2 + \|\omega^{-1} R_{\mathbf{Q}}^{-1} \mu\|_Q^2 + \left\| L'\begin{pmatrix} z \\ \mu \end{pmatrix} \right\|_{X' \times Q}^2 + \|T' R_{\mathbf{Y}} T y\|_{X'}^2 \ . \end{aligned} \tag{4.2.28}$$

The last term can now be estimated by the **Trace Theorem** 1.20 and the spectral boundedness of $R_{\mathbf{Y}}$ (2.2.39) as

$$\|T' R_{\mathbf{Y}} T y\|_{X'} \lesssim c_{T,\Omega} \, C_{T,\Omega} \, \|y\|_X \ . \tag{4.2.29}$$

Combining (4.2.29) and (4.2.28) yields

$$\|NV\|_{\mathcal{N}'}^2 \lesssim \left\| L\begin{pmatrix} y \\ p \end{pmatrix} \right\|_{X'\times Q}^2 + \|\omega^{-1} R_{\mathbf{Q}}^{-1}\mu\|_Q^2 + \left\| L'\begin{pmatrix} z \\ \mu \end{pmatrix} \right\|_{X'\times Q}^2 + \|y\|_X^2 \ .$$

Lastly, using the norm equivalence (3.1.21) for $L = L'$ and (2.2.39) for $R_{\mathbf{Q}}$, one obtains

$$\|NV\|_{\mathcal{N}'}^2 \lesssim \|V\|_{\mathcal{N}}^2 + \omega^{-1}\|\mu\|_{Q'}^2 + \|y\|_X^2 \lesssim \|V\|_{\mathcal{N}}^2$$

Since N is linear, continuous and invertible on a product of Hilbert spaces, the **inverse mapping theorem** (see [3]) proves that the inverse mapping N^{-1} exists and is also linear and continuous, i.e., the estimate

$$\|V\|_{\mathcal{N}}^2 \lesssim \|NV\|_{\mathcal{N}'}^2$$

holds. ∎

Thus, we have two problem formulations which consist only of linear isomorphism between Hilbert spaces. The next step is now to create a wavelet representation $\mathbf{N}$ of the isomorphism N.

4.2.2 Reformulation as a Problem in ℓ_2

Let now be biorthogonal wavelet bases with the properties of Section 2 for the spaces X, Q and Y and their duals are at our disposal. We fix the involved spaces as

$$X = H^1(\Omega), \qquad Y = H^{1/2}(\Gamma_Y), \qquad Q = H^{1/2}(\Gamma) \ . \tag{4.2.30}$$

The primal wavelet bases will be indexed accordingly by Ψ_Ω^1, $\Psi_{\Gamma_Y}^{1/2}$ and $\Psi_\Gamma^{1/2}$ with the dual bases as introduced in Section 3.1.2. Obviously, the Riesz basis property (Definition 2.1) and the norm equivalences (2.2.12), (2.2.13) hold for the required ranges, for example, the norm equivalence for the Sobolev range $[-1, 1]$ on the domain Ω is assured. Of course, we associate the wavelet bases with the usual infinite index sets $I\!I_X$, $I\!I_Y$, $I\!I_Q$. The wavelet basis for the product space $\mathcal{N}$ of (4.2.26) is the external product of the involved wavelet bases, i.e.,

$$\Psi_{\mathcal{N}} := \left(\Psi_\Omega^1, \widetilde{\Psi}_\Gamma^{1/2}, \Psi_\Omega^1, \widetilde{\Psi}_\Gamma^{1/2}\right)^T, \qquad I\!I_{\mathcal{N}} := I\!I_X \times I\!I_Q \times I\!I_X \times I\!I_Q \ . \tag{4.2.31}$$

Instead of just discretizing N from (4.2.21), we stress that the present wavelet framework allows for a much more natural approach by formulating the functional (4.2.5) in terms of discrete ℓ_2 norms. These discrete norms must be equivalent to the norms involved in the functional $\mathcal{J}$. This means for any $v = \mathbf{v}^T\Psi_{\Gamma_Y}^{1/2} \in Y$ and $q = \mathbf{q}^T\Psi_\Gamma^{1/2} \in Q$ that we want to employ operators $\mathbf{R_Q}$, $\mathbf{R_Y}$ such that for the norms (4.2.3) holds

$$\|v\|_{\mathbf{Y}} = \|\mathbf{R}_{\mathbf{Y}}^{1/2}\mathbf{v}\|_{\ell_2} = \langle \mathbf{R_Y v}, \mathbf{v}\rangle, \qquad \|q\|_{\mathbf{Q}} = \|\mathbf{R}_{\mathbf{Q}}^{1/2}\mathbf{q}\|_{\ell_2} = \langle \mathbf{R_Q q}, \mathbf{q}\rangle \ . \tag{4.2.32}$$

These **Riesz operators** $\mathbf{R_Q}$, $\mathbf{R_Y}$ are the ℓ_2 pendants of the Riesz operators introduced in (4.2.15) and (4.2.16). The smallest and largest eigenvalues of $\mathbf{R_Q}$ and $\mathbf{R_Y}$ shall be denoted by $c_{\mathbf{Q}}$, $C_{\mathbf{Q}}$ and $c_{\mathbf{Y}}$, $C_{\mathbf{Y}}$ respectively,

$$c_{\mathbf{Y}}\|\mathbf{v}\|_{\ell_2} \le \|\mathbf{R_Y v}\|_{\ell_2} \le C_{\mathbf{Y}}\|\mathbf{v}\|_{\ell_2}, \quad c_{\mathbf{Q}}\|\mathbf{v}\|_{\ell_2} \le \|\mathbf{R_Q v}\|_{\ell_2} \le C_{\mathbf{Q}}\|\mathbf{v}\|_{\ell_2} \qquad \text{for all } \mathbf{v} \in \ell_2 \ . \tag{4.2.33}$$

The right hand side variables (f, y_*) and the unknowns (y, p, u) are now expanded in terms of these weighted wavelet bases as

$$(f, y_*)^T = \qquad (\mathbf{f}^T\widetilde{\Psi}_\Omega^1, \mathbf{y}_*^T\Psi_{\Gamma_Y}^{1/2})^T \qquad = (\mathbf{f}^T\mathbf{D}_\Omega^{+1}\widetilde{\Psi}_\Omega, \mathbf{y}_*^T\mathbf{D}_{\Gamma_Y}^{-1/2}\Psi_{\Gamma_Y})^T, \tag{4.2.34}$$

$$(y, p, u)^T = \ (\mathbf{y}^T\Psi_\Omega^1, \mathbf{p}^T\widetilde{\Psi}_\Gamma^{1/2}, \mathbf{u}^T\Psi_\Gamma^{1/2})^T \ = (\mathbf{y}^T\mathbf{D}_\Omega^{-1}\Psi_\Omega, \mathbf{p}^T\mathbf{D}_\Gamma^{+1/2}\widetilde{\Psi}_\Gamma, \mathbf{u}^T\mathbf{D}_\Gamma^{-1/2}\Psi_\Gamma)^T \ . \tag{4.2.35}$$

The elliptic boundary value problem (4.2.6) can be discretized and preconditioned by the theory of Section 3.1.2, and the operator $\mathbf{L}$ then exactly attains the form (3.1.26). The discrete trace operator $\mathbf{T} : \ell_2 \to \ell_2$ descends from the continuous trace operator T just like the trace operator $\mathbf{B}$ from B does in Definition (3.1.27), i.e.,

$$\mathbf{T} := \langle \widetilde{\Psi}_{\Gamma_Y}^{1/2}, T\Psi_\Omega^1\rangle = \mathbf{D}_{\Gamma_Y}^{+1/2}\langle \widetilde{\Psi}_{\Gamma_Y}, T\Psi_\Omega\rangle\mathbf{D}_\Omega^{-1} \ . \tag{4.2.36}$$

These are all ingredients which are needed for the following reformulation of Problem 4.4:

Problem 4.11 [Control Problem with Boundary Control in Wavelet Coordinates] *Given* $(\mathbf{f}, \mathbf{y}_*)^T \in \ell_2(I\!I_X \times I\!I_Y)$, *find* $(\mathbf{y}, \mathbf{p}, \mathbf{u}) \in \ell_2(I\!I_X \times I\!I_Q \times I\!I_Q)$, *such that*

$$\mathbf{J}(\mathbf{y}, \mathbf{u}) = \frac{1}{2}\|\mathbf{R}_\mathbf{Y}^{1/2}(\mathbf{T}\mathbf{y} - \mathbf{y}_*)\|_{\ell_2}^2 + \frac{\omega}{2}\|\mathbf{R}_\mathbf{Q}^{1/2}\mathbf{u}\|_{\ell_2}^2 \tag{4.2.37}$$

is minimized subject to

$$\mathbf{L}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} \equiv \begin{pmatrix}\mathbf{A} & \mathbf{B}^T\\ \mathbf{B} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ \mathbf{u}\end{pmatrix} . \tag{4.2.38}$$

Remark 4.12 *Problem 4.11 can be viewed as the representation of Problem 4.2 in wavelet coordinates in the following sense. Because we chose our norms* $\|\cdot\|_\mathbf{Y}$, $\|\cdot\|_\mathbf{Q}$ *equivalent to the norms* $\|\cdot\|_Y$ *and* $\|\cdot\|_Q$ *in (4.2.3) are the functionals (4.2.37) and (4.2.1) equivalent, i.e.,*

$$\mathbf{J}(\mathbf{y}, \mathbf{u}) \sim \mathcal{J}(y, u),$$

for any $y = \mathbf{y}^T\Psi_\Omega^1 \in X$, *given* $y_* = \mathbf{y}_*^T\Psi_{\Gamma_Y}^{1/2} \in Y$ *and any* $u = \mathbf{u}^T\Psi_\Gamma^{1/2} \in Q$. *This obviously means we are solving a different, albeit equivalent, problem. At this time, this modelling discrepancy cannot be eliminated because fractional Sobolev space norms are not even uniquely defined, see Section 1.2. We will try to minimize the potential gap with* **Riesz operators** *which model the fractional Sobolev norms accurately. In the case of integer norms* $Y \in \{L_2, H^1\}$,$Q \in \{L_2, H^1\}$ *are the norms evaluated* **exactly**. *It is in this sense that the new functional (4.2.37) captures the essential features of the original minimization functional, see also the discussion in [28].*

A unique minimizer can now be proved for this problem to exist by the same techniques as in Section 4.2.1 . Rewriting $\mathbf{y}$ in terms of $\mathbf{u}$ again with the help of (3.2.20) yields

$$\mathbf{y} = \mathbf{y}(\mathbf{u}) = \mathbf{A}^{-1}(\widetilde{\mathbf{B}}^T\mathbf{u} + \widetilde{\mathbf{f}}) \tag{4.2.39}$$

with the abbreviations

$$\begin{aligned}\widetilde{\mathbf{B}}^T &= \mathbf{B}^T(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T),\\ \widetilde{\mathbf{f}} &= (\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)^{-1}\mathbf{B}\mathbf{A}^{-1})\mathbf{f} .\end{aligned}$$

The functional $\mathbf{J}$ rewritten as a function of the control $\mathbf{u}$ alone reads

$$\mathbf{J}(\mathbf{u}) = \mathbf{J}(\mathbf{y}(\mathbf{u})) = \frac{1}{2}\|\mathbf{R}_\mathbf{Y}^{1/2}(\mathbf{T}\mathbf{A}^{-1}(\widetilde{\mathbf{B}}^T\mathbf{u} + \widetilde{\mathbf{f}}) - \mathbf{y}_*)\|_{\ell_2}^2 + \frac{\omega}{2}\|\mathbf{R}_\mathbf{Q}^{1/2}\mathbf{u}\|_{\ell_2}^2 . \tag{4.2.40}$$

The discrete analogon to Proposition 4.6 now attains the following form:

Proposition 4.13 *The functional (4.2.11) is twice differentiable on* ℓ_2 *with derivatives*

$$\begin{aligned}D\mathbf{J}(\mathbf{u};\mathbf{v}) &= \langle\mathbf{R}_\mathbf{Y}(\mathbf{T}\,\mathbf{y}(\mathbf{u}) - \mathbf{y}_*), \mathbf{T}\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T\mathbf{v}\rangle + \omega\,\langle\mathbf{R}_\mathbf{Q}\mathbf{u}, \mathbf{v}\rangle\\ &= \langle\mathbf{R}_\mathbf{Y}(\mathbf{T}\mathbf{A}^{-1}(\widetilde{\mathbf{B}}'\mathbf{u} + \widetilde{\mathbf{f}}) - \mathbf{y}_*), \mathbf{T}\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T\mathbf{v}\rangle + \omega\,\langle\mathbf{R}_\mathbf{Q}\mathbf{u}, \mathbf{v}\rangle\end{aligned} \tag{4.2.41}$$

for all $\mathbf{v} \in \ell_2$ *and*

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{w}) = \langle\mathbf{R}_\mathbf{Y}(\mathbf{T}\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T\mathbf{v}), \mathbf{T}\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T\mathbf{w}\rangle + \omega\,\langle\mathbf{R}_\mathbf{Q}\mathbf{v}, \mathbf{w}\rangle \tag{4.2.42}$$

for all $\mathbf{v}, \mathbf{w} \in \ell_2$. *Moreover,* $D^2\mathbf{J}$ *satisfies for all* $\mathbf{v}, \mathbf{w} \in \ell_2$ *the estimates*

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{w}) \le C_*\|\mathbf{v}\|_{\ell_2}\|\mathbf{w}\|_{\ell_2} \tag{4.2.43}$$

and

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{v}) \ge c_*\|\mathbf{v}\|_{\ell_2}^2 \tag{4.2.44}$$

with constants

$$c_* := \omega\, c_\mathbf{Q}, \qquad C_* := C_\mathbf{Y}\,(c_{T,\Omega}\,\mathbf{c}_\mathbf{L}^{-1})^2 + \omega\, C_\mathbf{Q} . \tag{4.2.45}$$

In particular, for $\omega \neq 0$, $\mathbf{J}$ *is* **strictly convex**, *i.e.,*

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{v}) > 0, \qquad \text{for all } \mathbf{v} \in \ell_2 \setminus \{\mathbf{0}\} . \tag{4.2.46}$$

Proof: The identities (4.2.41) and (4.2.42) are proved exactly as in Proposition 4.6.
The derivation of the constants c_* and C_* is as follows. Considering the following saddle point problems and relation (3.2.20),

$$\mathbf{L}\begin{pmatrix}\widetilde{\mathbf{y}}\\ \widetilde{\mathbf{p}}\end{pmatrix}=\begin{pmatrix}\mathbf{0}\\ \mathbf{v}\end{pmatrix},\qquad \mathbf{L}\begin{pmatrix}\overline{\mathbf{y}}\\ \overline{\mathbf{p}}\end{pmatrix}=\begin{pmatrix}\mathbf{0}\\ \mathbf{w}\end{pmatrix},$$

we obtain the representations for $\widetilde{\mathbf{y}}$ and $\overline{\mathbf{y}}$ as

$$\widetilde{\mathbf{y}}=\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T\mathbf{v},\qquad \overline{\mathbf{y}}=\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T\mathbf{w}\ .$$

Using these variables, we can give an upper bound for $D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{w})$ as

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{w})\le C_{\mathbf{Y}}\,\|\mathbf{T}\,\widetilde{\mathbf{y}}\|_{\ell_2}\|\mathbf{T}\,\overline{\mathbf{y}}\|_{\ell_2}+\omega\,C_{\mathbf{Q}}\,\|\mathbf{v}\|_{\ell_2}\|\mathbf{w}\|_{\ell_2},$$

where $C_{\mathbf{Q}}$, $C_{\mathbf{Y}}$ are the largest eigenvalues of $\mathbf{R}_{\mathbf{Q}}$ and $\mathbf{R}_{\mathbf{Y}}$. Now we can apply estimates (1.2.25) and (3.1.29) to conclude

$$\|\mathbf{T}\,\widetilde{\mathbf{y}}\|_{\ell_2}\le c_{T,\Omega}\|\widetilde{\mathbf{y}}\|_{\ell_2}\le c_{T,\Omega}\,\mathbf{c}_{\mathbf{L}}^{-1}\|\mathbf{v}\|_{\ell_2},$$

and correspondingly for $\overline{\mathbf{y}}$. By using this in the previous estimate, we obtain (4.2.43),

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{w})\le(C_{\mathbf{Y}}\,(c_{T,\Omega}\,\mathbf{c}_{\mathbf{L}}^{-1})^2+\omega\,C_{\mathbf{Q}})\|\mathbf{v}\|_{\ell_2}\|\mathbf{w}\|_{\ell_2}=:C_*\|\mathbf{v}\|_{\ell_2}\|\mathbf{w}\|_{\ell_2}\ .$$

The lower estimate (4.2.44) is achieved by estimating

$$D^2\mathbf{J}(\mathbf{u};\mathbf{v},\mathbf{v})\ge c_{\mathbf{Y}}\,\|\mathbf{T}\,\widetilde{\mathbf{y}}\|_{\ell_2}^2+\omega\,c_{\mathbf{Q}}\,\|\mathbf{v}\|_{\ell_2}^2\ge\omega\,c_{\mathbf{Q}}\,\|\mathbf{v}\|_{\ell_2}^2,$$

where $c_{\mathbf{Q}}$, $c_{\mathbf{Y}}$ are the smallest eigenvalues of $\mathbf{R}_{\mathbf{Q}}$ and $\mathbf{R}_{\mathbf{Y}}$. The convexity (4.2.46) of $\mathbf{J}(\mathbf{u})$ is obvious for $\omega\neq 0$. ∎

The next step to solving Problem 4.11 is to form as in (4.2.17) the **Lagrangian functional** by appending the conditions (4.2.38) by means of additional Lagrangian multipliers $\mathbf{z}$, $\boldsymbol{\mu}\in\ell_2$ to the minimization functional (4.2.37), i.e.,

$$\mathbf{Lagr}(\mathbf{y},\mathbf{p},\mathbf{u},\mathbf{z},\boldsymbol{\mu}):=\frac{1}{2}\|\mathbf{R}_{\mathbf{Y}}^{1/2}(\mathbf{T}\mathbf{y}-\mathbf{y}_*)\|_{\ell_2}^2+\frac{\omega}{2}\|\mathbf{R}_{\mathbf{Q}}^{1/2}\mathbf{u}\|_{\ell_2}^2+\left\langle\begin{pmatrix}\mathbf{z}\\ \boldsymbol{\mu}\end{pmatrix},\mathbf{L}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix}-\begin{pmatrix}\mathbf{f}\\ \mathbf{u}\end{pmatrix}\right\rangle\ . \tag{4.2.47}$$

This functional is utilized in the following Lemma, which is the ℓ_2-version of Lemma 4.7.

Lemma 4.14 *Let* $\mathbf{J}$ *be the functional in (4.2.40) and let* $\mathbf{f}\in\ell_2$ *and* $\mathbf{y}_*\in\ell_2$ *be given. Then the Euler equations*

$$\delta\mathbf{Lagr}(\mathbf{y},\mathbf{p},\mathbf{u},\mathbf{z},\boldsymbol{\mu};\mathbf{r})=0,\qquad \textit{for } \mathbf{r}=\mathbf{y},\mathbf{p},\mathbf{u},\mathbf{z},\boldsymbol{\mu}, \tag{4.2.48}$$

for the minimization Problem 4.11 can be written shortly as

$$\begin{aligned}\mathbf{L}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix}&=\begin{pmatrix}\mathbf{f}\\ \mathbf{u}\end{pmatrix},\\ \omega\mathbf{R}_{\mathbf{Q}}\mathbf{u}&=\boldsymbol{\mu},\\ \mathbf{L}^T\begin{pmatrix}\mathbf{z}\\ \boldsymbol{\mu}\end{pmatrix}&=\begin{pmatrix}-\mathbf{T}^T\mathbf{R}_{\mathbf{Y}}(\mathbf{T}\mathbf{y}-\mathbf{y}_*)\\ \mathbf{0}\end{pmatrix}.\end{aligned} \tag{4.2.49}$$

Proof: The proof is carried out exactly as in Lemma 4.7. The **primal** and **adjoint systems** result from merely combining the first two and the last two Euler equations. ∎

We can again infer from Proposition 4.13 an analogon to Proposition 4.9.

Proposition 4.15 *The first variation of* $\mathbf{J}$ *is given by*

$$\begin{aligned}\delta\mathbf{J}(\mathbf{u})&=\widetilde{\mathbf{B}}\mathbf{A}^{-T}\mathbf{T}^T\mathbf{R}_{\mathbf{Y}}(\mathbf{T}\mathbf{y}-\mathbf{y}_*)+\omega\,\mathbf{R}_{\mathbf{Q}}\mathbf{u} &(4.2.50)\\ &=\widetilde{\mathbf{B}}\mathbf{A}^{-T}\mathbf{T}^T\mathbf{R}_{\mathbf{Y}}(\mathbf{T}\,\mathbf{A}^{-1}(\widetilde{\mathbf{B}}^T\mathbf{u}+\widetilde{\mathbf{f}})-\mathbf{y}_*)+\omega\,\mathbf{R}_{\mathbf{Q}}\mathbf{u}\\ &=(\widetilde{\mathbf{B}}\mathbf{A}^{-T}\mathbf{T}^T\mathbf{R}_{\mathbf{Y}}\mathbf{T}\,\mathbf{A}^{-1}\widetilde{\mathbf{B}}^T+\omega\,\mathbf{R}_{\mathbf{Q}})\mathbf{u}+\widetilde{\mathbf{B}}\mathbf{A}^{-T}\mathbf{T}^T\mathbf{R}_{\mathbf{Y}}(\mathbf{T}\,\mathbf{A}^{-1}\widetilde{\mathbf{f}}-\mathbf{y}_*)\\ &=:\widehat{\mathbf{A}}\mathbf{u}+\widehat{\mathbf{f}}\ . &(4.2.51)\end{aligned}$$

The matrix $\widehat{\mathbf{A}}$ *is obviously symmetric positive definite. An equivalent representation is*

$$\delta\mathbf{J}(\mathbf{u}) = \omega\,\mathbf{R}_{\mathbf{Q}}\mathbf{u} - \boldsymbol{\mu}, \tag{4.2.52}$$

which will be used in Section 4.4 to construct an inexact gradient method based on the iterative solving of the primal and adjoint systems in (4.2.49).

Lastly, the Euler equations (4.2.49) can be assembled into one linear system of equations by eliminating one variable.

Corollary 4.16 *The operator* $\mathbf{N}$ *defined by*

$$\begin{aligned}
\mathbf{NU} &:= \begin{pmatrix} \mathbf{L} & \mathbf{E} \\ \widehat{\mathbf{E}} & \mathbf{L}^T \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{z} \\ \boldsymbol{\mu} \end{pmatrix} \\
&:= \left(\begin{array}{cc|cc} \mathbf{A} & \mathbf{B}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & -\omega^{-1}\mathbf{R}_{\mathbf{Q}}^{-1} \\ \hline \mathbf{T}^T\mathbf{R}_{\mathbf{Y}}\mathbf{T} & \mathbf{0} & \mathbf{A}^T & \mathbf{B}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{0} \end{array}\right) \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \\ \mathbf{z} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{T}^T\mathbf{R}_{\mathbf{Y}}\mathbf{y}_* \\ \mathbf{0} \end{pmatrix} =: \mathbf{F}
\end{aligned} \tag{4.2.53}$$

is an ℓ_2*-automorphism, i.e., for any* $\boldsymbol{V} \in \ell_2 := \ell_2(I\!I_{\mathcal{N}}) = \ell_2(I\!I_X \times I\!I_Q \times I\!I_X \times I\!I_Q)$ *the equivalence*

$$\|\mathbf{N}\boldsymbol{V}\|_{\ell_2} \sim \|\boldsymbol{V}\|_{\ell_2}\ . \tag{4.2.54}$$

holds.

Proof: We have for $\boldsymbol{V} = (\mathbf{y}, \mathbf{p}, \mathbf{z}, \boldsymbol{\mu}) \in \ell_2$ by the reasoning as in (4.2.28),

$$\begin{aligned}
\|\mathbf{N}\boldsymbol{V}\|_{\ell_2}^2 &= \left\| \begin{pmatrix} \mathbf{L}\binom{\mathbf{y}}{\mathbf{p}} + \mathbf{E}\binom{\mathbf{z}}{\boldsymbol{\mu}} \\ \widehat{\mathbf{E}}\binom{\mathbf{y}}{\mathbf{p}} + \mathbf{L}^T\binom{\mathbf{z}}{\boldsymbol{\mu}} \end{pmatrix} \right\|_{\ell_2}^2 = \left\| \begin{pmatrix} \mathbf{L}\binom{\mathbf{y}}{\mathbf{p}} + \binom{\mathbf{0}}{-\omega^{-1}\mathbf{R}_{\mathbf{Q}}^{-1}\boldsymbol{\mu}} \\ \mathbf{T}^T\mathbf{R}_{\mathbf{Y}}\mathbf{T}\binom{\mathbf{y}}{\mathbf{0}} + \mathbf{L}^T\binom{\mathbf{z}}{\boldsymbol{\mu}} \end{pmatrix} \right\|_{\ell_2}^2 \\
&\le 2\left(\left\| \mathbf{L}\begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} \right\|_{\ell_2}^2 + \|\omega^{-1}\mathbf{R}_{\mathbf{Q}}^{-1}\boldsymbol{\mu}\|_{\ell_2}^2 + \left\| \mathbf{L}\begin{pmatrix} \mathbf{z} \\ \boldsymbol{\mu} \end{pmatrix} \right\|_{\ell_2}^2 + \|\mathbf{T}^T\mathbf{R}_{\mathbf{Y}}\mathbf{T}\,\mathbf{y}\|_{\ell_2}^2 \right),
\end{aligned}$$

and thus, with the constants from estimates (1.2.25), (3.1.29) and (4.2.33),

$$\begin{aligned}
&\le 2\left(\mathbf{C}_{\mathbf{L}} \left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} \right\|_{\ell_2}^2 + \omega^{-1}c_{\mathbf{Q}}^{-1}\|\boldsymbol{\mu}\|_{\ell_2}^2 + \mathbf{C}_{\mathbf{L}} \left\| \begin{pmatrix} \mathbf{z} \\ \boldsymbol{\mu} \end{pmatrix} \right\|_{\ell_2}^2 + (c_{T,\Omega}\,C_{T,\Omega}\,C_{\mathbf{Y}})\,\|\mathbf{y}\|_{\ell_2}^2 \right) \\
&= 2\left((\mathbf{C}_{\mathbf{L}}^2 + (c_{T,\Omega}\,C_{T,\Omega}\,C_{\mathbf{Q}})^2)\|\mathbf{y}\|_{\ell_2}^2 + \mathbf{C}_{\mathbf{L}}^2\|\mathbf{p}\|_{\ell_2}^2 + \mathbf{C}_{\mathbf{L}}^2\|\mathbf{z}\|_{\ell_2}^2 + (\mathbf{C}_{\mathbf{L}}^2 + (\omega\,c_{\mathbf{Q}})^{-2})\|\boldsymbol{\mu}\|_{\ell_2}^2 \right) \\
&\le 2\left(\mathbf{C}_{\mathbf{L}}^2 + \max\left\{ (c_{T,\Omega}\,C_{T,\Omega}\,C_{\mathbf{Q}})^2, (\omega\,c_{\mathbf{Q}})^{-2} \right\} \right) \|\boldsymbol{V}\|_{\ell_2}^2\ .
\end{aligned}$$

The lower estimate follows as before by an application of the **inverse mapping theorem** (see [3]). ∎

Next, we must investigate which stability conditions allow us to construct the finite dimensional operator $\mathbf{N}_{J_{\mathcal{N}}}$ from the ℓ_2-automorphism $\mathbf{N}$.

4.2.3 Stability of the Finite-Dimensional Systems

We now discuss the transformation of Problem 4.11 into a finite-dimensional problem. We are going to use the same notation for the level variables $J_{\mathcal{N}} := (J_X, J_Q, J_Y) \equiv (\sigma, \pi, \tau)$ and $J_{\mathcal{H}} = (\sigma, \pi)$ as in Section 3.1.3, i.e.,

$$I\!I_{X,\sigma} \subset I\!I_X, \qquad I\!I_{Q,\pi} \subset I\!I_Q, \qquad I\!I_{Y,\tau} \subset I\!I_Y\ . \tag{4.2.55}$$

The problem will be well-posed in the Euclidean metric

$$\ell_2(I\!I_{\mathcal{N},J_{\mathcal{N}}}) := \ell_2(I\!I_{X,\sigma} \times I\!I_{Q,\pi} \times I\!I_{X,\sigma} \times I\!I_{Q,\pi})\ .$$

The spaces corresponding to finite index sets are the linear wavelet spaces

$$X_\sigma := S(\Psi_{(\Omega,\sigma)}), \qquad Q_\pi := S(\Psi_{(\Gamma,\pi)}), \qquad Y_\tau := S(\Psi_{(\Gamma_Y,\tau)})\ . \tag{4.2.56}$$

Now, using these finite discretizations, Problem 4.11 reads

Problem 4.17 [Control Problem with Boundary Control in Finite Wavelet Coordinates] *Given* $(\mathbf{f}_\sigma, \mathbf{y}_{*,\tau})^T$, *find* $(\mathbf{y}_\sigma, \mathbf{p}_\pi, \mathbf{u}_\pi)^T$ *such that*

$$\mathbf{J}(\mathbf{y}_\sigma, \mathbf{u}_\pi) = \frac{1}{2}\|\mathbf{R}_{\mathbf{Y},\tau}^{1/2}(\mathbf{T}_{\sigma,\tau}\mathbf{y}_\sigma - \mathbf{y}_{*,\tau})\|_{\ell_2}^2 + \frac{\omega}{2}\|\mathbf{R}_{\mathbf{Q},\pi}^{1/2}\mathbf{u}_\pi\|_{\ell_2}^2 \ . \tag{4.2.57}$$

is minimized subject to

$$\mathbf{L}_{J_\mathcal{H}}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} \equiv \begin{pmatrix}\mathbf{A}_\sigma & \mathbf{B}_{\sigma,\pi}^T\\ \mathbf{B}_{\sigma,\pi} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} = \begin{pmatrix}\mathbf{f}_\sigma\\ \mathbf{u}_\pi\end{pmatrix} \ . \tag{4.2.58}$$

This problem can obviously be dealt with by the same techniques as described in Section 4.2.2 . The steps leading to Proposition 4.13 are all unchanged. Only the boundedness of the second derivative of the functional (4.2.57) requires special attention:

$$D^2\mathbf{J}(\mathbf{u}_\pi; \mathbf{v}_\pi, \mathbf{w}_\pi) = \langle \mathbf{R}_{\mathbf{Y},\tau}(\mathbf{T}_{\sigma,\tau}\mathbf{A}_\sigma^{-1}\widetilde{\mathbf{B}}_{\sigma,\pi}^T\mathbf{v}_\pi), \mathbf{T}_{\sigma,\tau}\mathbf{A}_\sigma^{-1}\widetilde{\mathbf{B}}_{\sigma,\pi}^T\mathbf{w}_\pi\rangle + \omega\,\langle \mathbf{R}_{\mathbf{Q},\pi}\mathbf{v}_\pi, \mathbf{w}_\pi\rangle \tag{4.2.59}$$

This expression is obviously bounded from above and below, if the spectral norms of the matrices $\mathbf{R}_{\mathbf{Y},\tau}^{1/2}\mathbf{T}_{\sigma,\tau}\mathbf{A}_\sigma^{-1}\widetilde{\mathbf{B}}_{\sigma,\pi}^T$ and $\mathbf{R}_{\mathbf{Q},\pi}^{1/2}$ are bounded uniformly. The Riesz maps $\mathbf{R}_{\mathbf{Q},\pi}$ and $\mathbf{R}_{\mathbf{Y},\tau}$ are uniformly stable because of the **Riesz basis** property $(\mathcal{S})$(2.1.5). Specifically, we have for $\mathbf{R}_J \in \{\mathbf{R}_{\mathbf{Q},\pi}, \mathbf{R}_{\mathbf{Y},\tau}\}$ with the definitions of the constants in (2.2.35) that

$$c_{\Psi_J}\|\mathbf{v}_J\|_{\ell_2} \le \|\mathbf{R}_J^{1/2}\mathbf{v}_J\|_{\ell_2} \le C_{\Psi_J}\|\mathbf{v}_J\|_{\ell_2}$$

holds. The remaining matrix $\mathbf{T}_{\sigma,\tau}\mathbf{A}_\sigma^{-1}\widetilde{\mathbf{B}}_{\sigma,\pi}$ is uniformly stable if the **saddle point problem** (4.2.58) and the operator $\mathbf{T}_{\sigma,\tau}$ is uniformly stable. Since we arranged at the beginning of this section Problem 3.14 to be our constraint in the original problem formulation (Problem 4.2), this entails that stability is ensured by an application of the Theorem 3.4. This in turn specifically means satisfying the **discrete ellipticity condition** (3.1.36) and the **LBB**-condition (3.1.39).
Since stability can be ensured, the resulting finite dimensional Euler equations (cf. (4.2.49)),

$$\begin{aligned}
\mathbf{L}_{J_\mathcal{H}}\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\end{pmatrix} &= \begin{pmatrix}\mathbf{f}_\sigma\\ \mathbf{u}_\pi\end{pmatrix},\\
\omega\,\mathbf{R}_{\mathbf{Q},\pi}\mathbf{u}_\pi &= \boldsymbol{\mu}_\pi,\\
\mathbf{L}_{J_\mathcal{H}}^T\begin{pmatrix}\mathbf{z}_\sigma\\ \boldsymbol{\mu}_\pi\end{pmatrix} &= \begin{pmatrix}-\mathbf{T}_{\sigma,\tau}^T\mathbf{R}_{\mathbf{Y},\tau}(\mathbf{T}_{\sigma,\tau}\mathbf{y}_\sigma - \mathbf{y}_{*,\tau})\\ \mathbf{0}\end{pmatrix},
\end{aligned} \tag{4.2.60}$$

then lead to the operator $\mathbf{N}_{J_\mathcal{N}}$ defined as

$$\begin{aligned}
\mathbf{N}_{J_\mathcal{N}}\mathbf{U}_{J_\mathcal{N}} &:= \left(\begin{array}{cc|cc}\mathbf{A}_\sigma & \mathbf{B}_{\sigma,\pi}^T & \mathbf{0} & \mathbf{0}\\ \mathbf{B}_{\sigma,\pi} & \mathbf{0} & \mathbf{0} & -\omega^{-1}\mathbf{R}_{\mathbf{Q},\pi}^{-1}\\ \hline \mathbf{T}_{\sigma,\tau}^T\mathbf{R}_{\mathbf{Y},\tau}\mathbf{T}_{\sigma,\tau} & \mathbf{0} & \mathbf{A}_\sigma^T & \mathbf{B}_{\sigma,\pi}^T\\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{\sigma,\pi} & \mathbf{0}\end{array}\right)\begin{pmatrix}\mathbf{y}_\sigma\\ \mathbf{p}_\pi\\ \mathbf{z}_\sigma\\ \boldsymbol{\mu}_\pi\end{pmatrix}\\
&= \begin{pmatrix}\mathbf{f}_\sigma\\ \mathbf{0}\\ \mathbf{T}_{\sigma,\tau}^T\mathbf{R}_{\mathbf{Y},\tau}\mathbf{y}_{*,\tau}\\ \mathbf{0}\end{pmatrix} =: \mathbf{F}_{J_\mathcal{N}},
\end{aligned} \tag{4.2.61}$$

to be an **automorphism**

$$\|\mathbf{N}_{J_\mathcal{N}}\mathbf{U}_{J_\mathcal{N}}\|_{\ell_2} \sim \|\mathbf{U}_{J_\mathcal{N}}\|_{\ell_2}, \tag{4.2.62}$$

on the space $\ell_2(I\!I_{\mathcal{N},J_\mathcal{N}})$.

4.2.4 Variation of Norms

We have mentioned before that the natural Sobolev index for the **observation space** $H^s(\Gamma_Y)$ and the **control space** $H^t(\Gamma)$ in the formulation of Problem 4.2 is $s = t = 1/2$ for our control problem with Dirichlet boundary control. As we have seen in Section 4.2.1, this leads to a well-defined problem, for which exactly one solution exists.

Remark 4.18 *More generally, we can extend our original problem by replacing the functional (4.2.1) with*

$$\mathcal{J}(y,u) = \frac{1}{2}\|y - y_*\|^2_{H^s(\Gamma_Y)} + \frac{\omega}{2}\|u\|^2_{H^t(\Gamma)}, \qquad 0 \le s,t \le 1, \tag{4.2.63}$$

which means measuring the state and control with coarser or finer norms than the natural ones. Of course, we have to assume at least $u \in H^{1/2}$ in order to have a well-posed problem. Thus, we cannot expect choices $t < 1/2$ will always yield correct or meaningful results. Especially the case $s = t = 0$ is **not** *sufficient to guarantee well-posedness of the control problem.*

In the wavelet context, this change implies the insertion of another diagonal matrix in the minimization functional (4.2.37): Theorem 2.17 established the diagonal matrices $\mathbf{D}^p$ as operators from $H^k \to H^{k-p}$ for any k and p. So, ideally, we have for $v = \mathbf{v}^T\Psi_\Gamma^{1/2} \in H^{1/2}(\Gamma)$ and with the Riesz operator $\mathbf{R}_{H^t}$ for $H^t(\Gamma)$,

$$\|v\|_{H^t(\Gamma)} = \|\mathbf{R}^{1/2}_{H^t}\mathbf{D}_\Gamma^{-1/2+t}\mathbf{v}\|_{\ell_2} \ . \tag{4.2.64}$$

As such, the functional (4.2.63) in wavelet coordinates looks like (4.2.37) with the addition of the new diagonal scaling and proper Riesz operators as

$$\mathbf{J}(\mathbf{y},\mathbf{u}) = \frac{1}{2}\|\mathbf{R}^{1/2}_{H^s}\mathbf{D}^{-1/2+s}_{\Gamma_Y}(\mathbf{T}\mathbf{y} - \mathbf{y}_*)\|^2_{\ell_2} + \frac{\omega}{2}\|\mathbf{R}^{1/2}_{H^t}\mathbf{D}^{-1/2+t}_{\Gamma}\mathbf{u}\|^2_{\ell_2} \ . \tag{4.2.65}$$

Carrying out the construction process described in Section 4.2.2 with this functional leads to the following system of linear equations (cf. (4.2.60)) :

$$\begin{aligned}
\mathbf{T}^T\mathbf{D}^{-1/2+s}_{\Gamma_Y}\mathbf{R}_{H^s}(\mathbf{D}^{-1/2+s}_{\Gamma_Y}\mathbf{T}\mathbf{y} - \mathbf{y}_*) + \mathbf{A}\mathbf{z} + \mathbf{B}^T\boldsymbol{\mu} &= \mathbf{0},\\
\mathbf{B}\,\mathbf{z} &= \mathbf{0},\\
\omega\,\mathbf{D}^{-1/2+t}_{\Gamma}\mathbf{R}_{H^t}\mathbf{D}^{-1/2+t}_{\Gamma}\mathbf{u} - \boldsymbol{\mu} &= \mathbf{0},\\
\mathbf{A}\mathbf{y} + \mathbf{B}^T\mathbf{p} - \mathbf{f} &= \mathbf{0},\\
\mathbf{B}\mathbf{y} - \mathbf{u} &= \mathbf{0}\ .
\end{aligned} \tag{4.2.66}$$

We retain (4.2.49) for $s = t = 1/2$, since the diagonal operators then coincide with the identity. The above equations can be shortened by the definition of a trace operator $\check{T} : H^1(\Omega) \to H^s(\Gamma_Y)$ in wavelet coordinates:

$$\check{\mathbf{T}} := \mathbf{D}^{-1/2+s}_{\Gamma_Y} \circ \mathbf{T} = \mathbf{D}^s_{\Gamma_Y}\langle\widetilde{\Psi}_{\Gamma_Y}, T\Psi_\Omega\rangle\mathbf{D}^{-1}_\Omega \ . \tag{4.2.67}$$

We introduce the following variables and operators in the same fashion:

$$\begin{gathered}
\check{\mathbf{B}} := \mathbf{D}^{-1/2+t}_\Gamma \circ \mathbf{B}, \qquad \check{\mathbf{y}}_* := \mathbf{D}^{-1/2+s}_{\Gamma_Y}\mathbf{y}_*,\\
\check{\mathbf{p}} := \mathbf{D}^{-1/2+t}_\Gamma\mathbf{p}, \qquad \check{\boldsymbol{\mu}} := \mathbf{D}^{-1/2+t}_\Gamma\boldsymbol{\mu}, \qquad \check{\mathbf{u}} := \mathbf{D}^{-1/2+t}_\Gamma\mathbf{u}\ .
\end{gathered}$$

The assembled system then has the form

$$\check{\mathbf{N}}\check{\mathbf{U}} := \left(\begin{array}{cc|cc} \mathbf{A} & \check{\mathbf{B}}^T & \mathbf{0} & \mathbf{0}\\ \check{\mathbf{B}} & \mathbf{0} & \mathbf{0} & -\omega^{-1}\mathbf{R}^{-1}_{H^t}\\ \hline \check{\mathbf{T}}^T\mathbf{R}_{H^s}\check{\mathbf{T}} & \mathbf{0} & \mathbf{A}^T & \check{\mathbf{B}}^T\\ \mathbf{0} & \mathbf{0} & \check{\mathbf{B}} & \mathbf{0} \end{array}\right)\left(\begin{array}{c}\mathbf{y}\\ \check{\mathbf{p}}\\ \mathbf{z}\\ \check{\boldsymbol{\mu}}\end{array}\right) = \left(\begin{array}{c}\mathbf{f}\\ \mathbf{0}\\ \check{\mathbf{T}}^T\mathbf{R}_{H^s}\check{\mathbf{y}}_*\\ \mathbf{0}\end{array}\right) =: \check{\mathbf{F}}, \tag{4.2.68}$$

Note that this operator $\check{\mathbf{N}}$ does no longer have a uniformly bounded condition number. This can be seen by analyzing the new saddle point operator $\check{\mathbf{L}}$ of the upper left and lower right block:

$$\check{\mathbf{L}} := \begin{pmatrix}\mathbf{A} & \check{\mathbf{B}}^T\\ \check{\mathbf{B}} & \mathbf{0}\end{pmatrix} = \begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}^{-1/2+t}_\Gamma\end{pmatrix}\begin{pmatrix}\mathbf{A} & \mathbf{B}^T\\ \mathbf{B} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}^{-1/2+t}_\Gamma\end{pmatrix} \ . \tag{4.2.69}$$

Proposition 4.19 *The operator $\check{\mathbf{L}}$ does not have uniformly bounded condition for $t \neq 1/2$.*

Proof: Depending on the value of t relative to $1/2$ the diagonal matrices $\mathbf{D}^{-1/2+t}_\Gamma$ have either no smallest ($t < 1/2$) or no largest ($t > 1/2$) uniformly bounded eigenvalue. Since $\mathbf{L}$ has uniformly bounded

eigenvalues and the diagonal matrices do not, $\check{\mathbf{L}}$ cannot have a bounded condition. ∎

The operator $\check{\mathbf{L}}$ is obviously still invertible and a unique solution exists for given right hand side. In fact, the equations

$$\check{\mathbf{L}}\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ \mathbf{g}\end{pmatrix}, \qquad \mathbf{L}\begin{pmatrix}\check{\mathbf{y}}\\ \check{\mathbf{p}}\end{pmatrix} = \begin{pmatrix}\check{\mathbf{f}}\\ \check{\mathbf{g}}\end{pmatrix},$$

are algebraically (although not numerically) equivalent, which resolves the problem of unboundedness of the condition. However, this does not work with (4.2.68) though, since $\mathbf{N}$ then reads

$$\left(\begin{array}{cc|cc} \mathbf{A} & \mathbf{B}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & -\omega^{-1}\check{\mathbf{R}}_{H^t}^{-1} \\ \hline \check{\mathbf{T}}^T\mathbf{R}_{H^s}\check{\mathbf{T}} & \mathbf{0} & \mathbf{A}^T & \mathbf{B}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{0} \end{array}\right)\begin{pmatrix}\mathbf{y}\\ \mathbf{p}\\ \mathbf{z}\\ \boldsymbol{\mu}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ \mathbf{0}\\ \check{\mathbf{T}}^T\mathbf{R}_{H^s}\check{\mathbf{y}}_*\\ \mathbf{0}\end{pmatrix}, \tag{4.2.70}$$

abbreviating

$$\check{\mathbf{R}}_{H^t} := \check{\mathbf{R}}_{H^t(\Gamma)} := \mathbf{D}_\Gamma^{+1/2-t}\mathbf{R}_{H^t}\mathbf{D}_\Gamma^{+1/2-t}\ . \tag{4.2.71}$$

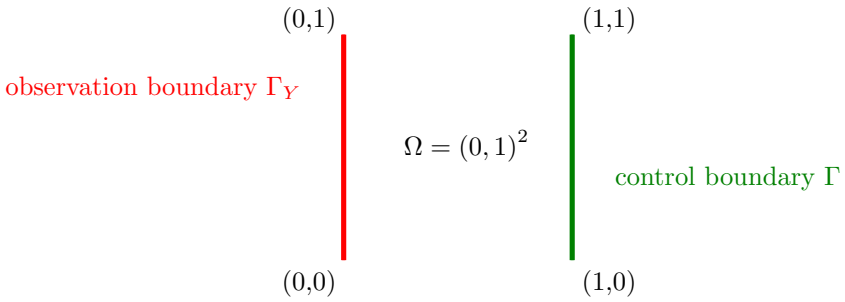
Now the upper right and lower left blocks are still scaled and induce increasing eigenvalues. This change obviously also affects the finite discretization operator $\check{\mathbf{N}}_{J_\mathcal{N}}$. However, in applications, this effect is rarely seen because of the nature of wavelet discretizations, explained as follows.
A uniformly bounded condition number means that the extremal eigenvalues of any symmetric positive definite operator $\mathbf{A}_\sigma$ are constant for all levels. That implies that these extremal eigenvalues ideally refer to wavelets on the lowest possible level $j = j_0$. The additional eigenvalues $\{\lambda_i\}$ which arise on higher level discretizations $j > j_0$ are now enclosed by the extremal eigenvalues: $\lambda_{\min} \le \lambda_i \le \lambda_{\max}$ for all i.
The diagonal scaling $\mathbf{D}^{\pm l}$ in general has entries of magnitude $\sim 2^{\pm jl}$, which enforce ill conditioning of our matrix. The eigenvalues arising from the higher levels $j > j_0$ will be scaled by the much higher (or lower) values, depending on the sign of the exponent l. Assuming $\lambda_i \approx 1$ and for some constant $c > 1$,

$$\frac{\lambda_{\max}}{\lambda_{\min}} = \frac{c}{1/c} = c^2,$$

we can expect so see a significant change of the condition number on levels $j \gtrsim \log_2(c)/|l|$.
Our diagonal scalings $\mathbf{D}_{\Gamma_\mathcal{Y}}^{-1/2+s}$ and $\mathbf{D}_\Gamma^{-1/2+t}$ have exponent values of between $-1/2$ and $1/2$ which are beneficial for an even higher lower bound. In this application, the perturbation of the condition number is rarely observed, because it usually applies for high levels $j \gg j_0$ only.

4.3 A Control Problem with Dirichlet Boundary Control

Now the optimal control problem we are going to discuss in Section 5 is presented. Let $\Omega = \square = (0,1)^n$ be the unit cube in $\mathbb{R}^n$. The control boundary Γ is selected as the face with coordinates $x = (1, x_2, \ldots, x_n)$ or $\Gamma = \Gamma_{\mathrm{W}}$ with the definition of Section 3.2.3. In addition, we choose the state boundary Γ_Y to be the opposite face Γ_{E} with coordinates $x = (0, x_2, \ldots, x_n)$. Furthermore, we set $\Gamma_N := \partial\Omega \setminus \Gamma$. The following graphics depicts the case $n = 2$:



The two corresponding **trace operators** are denoted by $B = \gamma_E : H^1(\Omega) \to H^{1/2}(\Gamma)$ and $T = \gamma_W : H^1(\Omega) \to H^{1/2}(\Gamma_Y)$, see Section 3.2.3 for the definition. This setup is used in the following

Problem 4.20 [Control Problem with Dirichlet Boundary Control]
For given $f \in (H^1(\Omega))'$ and $y_{\Gamma_Y} \in H^s(\Gamma_Y)$, minimize

$$\mathcal{J}(y,u) = \frac{1}{2}\|Ty - y_{\Gamma_Y}\|^2_{H^s(\Gamma_Y)} + \frac{\omega}{2}\|u\|^2_{H^t(\Gamma)}, \tag{4.3.1}$$

where the **state** *y and the* **control** *u are coupled by the following* **elliptic boundary value problem** *(confer with Section 3.2.3)*

$$\begin{aligned} -\nabla\cdot(\mathbf{a}\nabla y) + a_0\, y &= f \quad \text{in } \Omega, \\ y &= u \quad \text{on } \Gamma, \\ (\mathbf{a}\nabla y)\cdot\mathbf{n} &= 0 \quad \text{on } \Gamma_N\ . \end{aligned} \tag{4.3.2}$$

The term $\mathbf{n} = \mathbf{n}(x)$ denotes the outward normal at every point $x \in \Gamma_N$. As before, $\mathbf{a}(x) = (a_{i,j}(x))_{i,j}$ is uniformly positive definite on Ω and $a_0 > 0$. Since we have the regularity $y|_{\partial\Omega} \in H^{1/2}(\partial\Omega)$, we can adjust the Sobolev norms in the functional (4.3.1). The problem is still well-posed if the value of s is chosen to be smaller than $1/2$. Values $s > 1/2$ are only admissible if the solution state y lives in a higher smoothness space than $H^1(\Omega)$. This can be assured if the right hand side is given $f \in L_2$. Because we have to guarantee at least $u \in H^{1/2}(\Gamma)$ for (4.3.2) are values $t \geq 1/2$ feasible. Choosing $t < 1/2$ may not yield meaningful results.

The ℓ_2-Problem

The constrains (4.3.2) of Problem 4.20 are formulated with the theory of Section 3.2.3 as a **saddle point problem**. The wavelet bases used in the setup of this problem are, of course, those of Section 2.3.2 in tensor product constructions,

$$\Psi^{I\!I}_\Omega := \bigotimes_{i=1}^n \Psi^{I\!I}, \qquad \widetilde{\Psi}^{I\!I}_\Omega := \bigotimes_{i=1}^n \widetilde{\Psi}^{I\!I}\ . \tag{4.3.3}$$

In case $n > 2$ the domains Γ and Γ_Y are constructed by $n-1$ tensor products of the interval I such that

$$\Psi^{I\!I}_{\Gamma_Y} = \Psi^{I\!I}_\Gamma := \bigotimes_{i=2}^n \Psi^{I\!I}, \qquad \widetilde{\Psi}^{I\!I}_{\Gamma_Y} = \widetilde{\Psi}^{I\!I}_\Gamma := \bigotimes_{i=2}^n \widetilde{\Psi}^{I\!I}\ . \tag{4.3.4}$$

The construction of the stiffness matrix $\mathbf{A}$ and the trace operators $\mathbf{T}$ and $\mathbf{B}$ was shown in Paragraph 3.2.3. As it was proved there, the resulting operator L is an isomorphism and in the correctly scaled wavelet discretizations are also ℓ_2-isomorphisms. Because these tensorized primal wavelet bases allow the norm equivalences described in Theorem 2.17 for up to $\gamma = 3/2$, they are well suited for our example problem. Now we can employ the theory of Section 4.2 to obtain the representer given by equations (4.2.66) for our problem. In case of the natural constraints in (4.3.1), i.e., $s = t = 1/2$, Corollary 4.16 applies and yields the ℓ_2-isomorphism $\mathbf{N}$ defined in (4.2.53).

Stability Remarks

Stability of the **saddle point operator** $\mathbf{L}_{J_\mathcal{H}}$ was examined in Paragraph 3.2.3. Recall that stability is ensured by the **LBB-condition** (3.1.39) and the **discrete ellipticity condition** (3.1.36). These, in turn, depend on the specific control boundary $\Gamma \subset \partial\Omega$ and the actual values of the coefficient matrix $\mathbf{a}$ and a_0 in (4.3.2). Corollary 3.6 then gives the desired result.
The stability remarks for general Riesz operators were discussed in Section 4.2.3 and apply here to the Riesz operators $\mathbf{R}_{H^s,\tau}$ and $\mathbf{R}_{H^t,\pi}$. Thus, the operator $\mathbf{N}_{J_\mathcal{N}}$ defined in (4.2.61) fulfills the isomorphism relation (4.2.62) in case $s = t = 1/2$.

4.3.1 Implementation Details of the Riesz Operators

There are several possible ways to construct Riesz operators for our use. The ideal construction given by (2.2.38),

$$R_\mathcal{H} = (\Psi, \Psi)_\mathcal{H}, \tag{4.3.5}$$

cannot be applied for $\mathcal{H} \in \{H^s(\Gamma), H^t(\Gamma_Y)\}$ with $s, t \notin \mathbb{Z}$, because the inner product is not uniquely defined, see Section 1.2. We could always use the approximations

$$R_{H^s} \approx R_{L_2}, \qquad R_{H^s} \approx Id, \tag{4.3.6}$$

where the last equivalence is identical to using no Riesz operator at all. In many cases, these approximations might suffice. However, we are going to use the diagonal scaling based construction (2.2.44),

$$\widehat{\mathbf{R}}_{H^s} = \mathbf{D}^{-s}\widehat{\mathbf{D}}^{+s}\big(\Psi^{I\!I}, \Psi^{I\!I}\big)_{L_2}\widehat{\mathbf{D}}^{+s}\mathbf{D}^{-s}, \qquad (\widehat{\mathbf{D}}^{+s})_{(j,j')(k,k')} = 2^{+js}\delta_{(j,j')}\delta_{(k,k')}, \tag{4.3.7}$$

because it leads to better results in actual calculations, see Section 5.2.3. We are also going to use the new interpolating Riesz operator construction from [11].
One detail, which has not yet been discussed, is the application of the **inverse** of the **Riesz operator** $\widehat{\mathbf{R}}_{H^s}$ in (4.2.53). Although we have the mathematical equality

$$\begin{aligned}\widehat{\mathbf{R}}_{H^s}^{-1} &= \mathbf{D}^{+s}\widehat{\mathbf{D}}^{-s}\Big(\widetilde{\Psi}^{I\!I}, \widetilde{\Psi}^{I\!I}\Big)_{L_2}\widehat{\mathbf{D}}^{-s}\mathbf{D}^{+s} \\ &= \mathbf{D}^{+s}\widehat{\mathbf{D}}^{-s}\,\widetilde{\mathbf{T}}^T\left(\widetilde{\Phi}, \widetilde{\Phi}\right)_{L_2}\widetilde{\mathbf{T}}\,\widehat{\mathbf{D}}^{-s}\mathbf{D}^{+s},\end{aligned}$$

it cannot be constructed in this way, because the **dual generators** $\widetilde{\Phi}$ are not explicitly given (see Section 2.3). However, we can use the correlation of the primal and dual bases (2.1.60) and their wavelet transforms (see Figure 2.1) to express the above as

$$\widehat{\mathbf{R}}_{H^s} = \mathbf{D}^{+s}\widehat{\mathbf{D}}^{-s}\,\mathbf{T}^{-1}\,(\Phi, \Phi)_{L_2}^{-1}\,\mathbf{T}^{-T}\,\widehat{\mathbf{D}}^{-s}\mathbf{D}^{+s}\ . \tag{4.3.8}$$

Our primal generators ϕ_j, defined in (2.3.25), have a very small support $\sim 2^{-j}$, and the resulting matrix

$$\mathbf{M}_\Phi = (\Phi, \Phi)_{L_2} \tag{4.3.9}$$

is, thus, very sparse. Specifically, it is a **band matrix** with band width 1 everywhere except for a 4×4 block at the matrix's upper left and lower right corner. These blocks stem from the increased support of the generators at the interval borders.
Since $\mathbf{M}_{\Phi_J}$ must be an ℓ_2-isomorphism, each inversion can be calculated by the CG scheme (Algorithm 3.25) very quickly. Each application of $\mathbf{M}_{\Phi_J}^{-1}$ can be carried out in $\mathcal{O}(\#I\!I_{\Gamma,J})$ operations every

time. It is, however, possible to analytically calculate the **LU**-decomposition of $\mathbf{M}_{\Phi_J}$ and use this information to program a routine which does the job also in $\mathcal{O}(\#I_{\Gamma,J})$ but with a smaller constant factor than the CG solver. Thus, the application of the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ and its inverse $\widehat{\mathbf{R}}_{H^s}^{-1}$ can both be carried out in linear time with respect to the number of unknowns.
This still holds true in higher dimensions since the tensor product structure yields

$$\begin{aligned} \mathbf{M}_{\Phi_J\otimes\cdots\otimes\Phi_J}^{-1} &= (\mathbf{M}_{\Phi_J}\otimes\cdots\otimes\mathbf{M}_{\Phi_J})^{-1} \\ &= \mathbf{M}_{\Phi_J}^{-1}\otimes\cdots\otimes\mathbf{M}_{\Phi_J}^{-1}\ . \end{aligned} \tag{4.3.10}$$

Each application of $\mathbf{M}_{\Phi_J}^{-1}$ is now implemented by the above mentioned LU-decomposition scheme.
These thoughts conclude the construction of the involved operators for our control problem in wavelet discretization.

4.4 Algorithms

There are several possible ways to determine the solutions $\mathbf{y}_\sigma \in \ell_2(\mathbb{I}_{\Omega,\sigma})$ and $\mathbf{u}_\pi \in \ell_2(\mathbb{I}_{\Gamma,\pi})$ of Problem 4.17. The main task is to propose algorithms which are **computationally efficient** in the sense that the finite-dimensional equations are solved with optimal complexity, i.e., proportional to the number of unknowns $N_J := \#\mathbb{I}_{\mathcal{N},J_\mathcal{N}}$.

Remark 4.21 *In the following, we treat the representation of Problem 4.17. The case of Section 4.2.4 is handled in complete analogy, and we spare us here the extra terms the change of the norms induce. The whole change can easily be implemented by using the* $\check{\mathbf{T}}, \check{\mathbf{B}}$ *version of the Trace operators, (4.2.67), and of the Riesz operator* $\check{\mathbf{R}}_{H^t}$*, (4.2.71).*

4.4.1 An All-In-One Solver

One way to obtain the solutions $\mathbf{y}_\sigma$, $\mathbf{u}_\pi$ is by solving the linear equation (4.2.61),

$$\begin{aligned}\mathbf{N}_{J_\mathcal{N}}\mathbf{U}_{J_\mathcal{N}} &= \left(\begin{array}{cc|cc} \mathbf{A}_\sigma & \mathbf{B}^T_{\sigma,\pi} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{\sigma,\pi} & \mathbf{0} & \mathbf{0} & -\omega^{-1}\mathbf{R}^{-1}_{H^t,\pi} \\ \hline \mathbf{T}^T_{\sigma,\tau}\mathbf{R}_{H^s,\tau}\mathbf{T}_{\sigma,\tau} & \mathbf{0} & \mathbf{A}^T_\sigma & \mathbf{B}^T_{\sigma,\pi} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{\sigma,\pi} & \mathbf{0} \end{array}\right)\left(\begin{array}{c}\mathbf{y}_\sigma \\ \mathbf{p}_\pi \\ \mathbf{z}_\sigma \\ \boldsymbol{\mu}_\pi\end{array}\right) \\ &= \left(\begin{array}{c}\mathbf{f}_\sigma \\ \mathbf{0} \\ \mathbf{T}^T_{\sigma,\tau}\mathbf{R}_{H^s,\tau}\mathbf{y}_{\Gamma_Y,\tau} \\ \mathbf{0}\end{array}\right) = \mathbf{F}_{J_\mathcal{N}}\ . \end{aligned} \tag{4.4.1}$$

This can be done with a direct solver like **LU**-decomposition or **QR**-decomposition on levels $j \approx j_0$. For higher levels $j \gg j_0$ this technique is not feasible because of the cubic complexity $\mathcal{O}(N^3)$ in the number of unknowns N of these algorithms.
Rather, we use the **nested iteration** scheme, Algorithm 3.23, with a direct solver on level $j = j_0$ and the conjugate gradient solver, Algorithm 3.25, for levels $j > j_0$ applied to the normal equation $\mathbf{N}^T_{J_\mathcal{N}}\mathbf{N}_{J_\mathcal{N}}\mathbf{U}_{J_\mathcal{N}} = \mathbf{N}^T_{J_\mathcal{N}}\mathbf{F}_{J_\mathcal{N}}$. This is necessary because the matrix $\mathbf{N}_{J_\mathcal{N}}$ is not symmetric and nor positive definite. The squared matrix $\mathbf{N}^T_{J_\mathcal{N}}\mathbf{N}_{J_\mathcal{N}}$, however, has both properties and, consequently, the CG algorithm can be applied. We quote the following corollary from [46].

Corollary 4.22 *The vector* $\mathbf{U}_{J_\mathcal{N}}$ *solves (4.4.1) if and only if* $\mathbf{U}_{J_\mathcal{N}}$ *solves the system*

$$\mathbf{P}_{J_\mathcal{N}}\mathbf{U}_{J_\mathcal{N}} := \mathbf{N}^T_{J_\mathcal{N}}\mathbf{N}_{J_\mathcal{N}}\mathbf{U}_{J_\mathcal{N}} = \mathbf{N}^T_{J_\mathcal{N}}\mathbf{F}_{J_\mathcal{N}}\ . \tag{4.4.2}$$

Moreover, the matrix $\mathbf{P}_{J_\mathcal{N}}$ *defines an automorphism of* $\ell_2(\mathbb{I}_{\mathcal{N},J_\mathcal{N}})$ *and one has*

$$\boldsymbol{V}^T_{J_\mathcal{N}}\mathbf{P}_{J_\mathcal{N}}\boldsymbol{V}_{J_\mathcal{N}} = \|\mathbf{N}_{J_\mathcal{N}}\boldsymbol{V}_{J_\mathcal{N}}\|^2_{\ell_2} \sim \|\boldsymbol{V}_{J_\mathcal{N}}\|^2_{\ell_2}\ . \tag{4.4.3}$$

The drawback, however, is that the matrix $\mathbf{P}_{J_\mathcal{N}}$ possesses an absolute higher condition number than the matrix $\mathbf{N}_{J_\mathcal{N}}$,i.e., it holds

$$\kappa(\mathbf{P}_{J_\mathcal{N}}) = \left(\kappa(\mathbf{N}_{J_\mathcal{N}})\right)^2\ . \tag{4.4.4}$$

Since the convergence speed (3.3.4) of the CG algorithm depends on this condition number, the convergence rate expected for the system (4.4.2) is expected to be lower than for (4.4.1). Note that the nested iteration scheme combined with the conjugate gradient scheme used on the normalized problem still has complexity $\mathcal{O}(\#\mathbb{I}_{\mathcal{N},J_\mathcal{N}})$. The constants in this estimate might yet be very high and this whole scheme is thus not very attractive for applications.
One approach to lower the costs of each execution is based on a permutation of the rows of the operator

$\mathbf{N}_{J_\mathcal{N}}$, yielding a symmetric operator, i.e.,

$$\mathbf{N}'_{J_\mathcal{N}}\mathbf{U}'_{J_\mathcal{N}} := \left(\begin{array}{cc|cc} \mathbf{T}^T_{\sigma,\tau}\mathbf{R}_{H^s,\tau}\mathbf{T}_{\sigma,\tau} & \mathbf{0} & \mathbf{A}^T_\sigma & \mathbf{B}^T_{\sigma,\pi} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{\sigma,\pi} & \mathbf{0} \\ \hline \mathbf{A}_\sigma & \mathbf{B}^T_{\sigma,\pi} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{\sigma,\pi} & \mathbf{0} & \mathbf{0} & -\omega^{-1}\mathbf{R}^{-1}_{H^t,\pi} \end{array}\right)\left(\begin{array}{c} \mathbf{z}_\sigma \\ \boldsymbol{\mu}_\pi \\ \mathbf{y}_\sigma \\ \mathbf{p}_\pi \end{array}\right)$$
$$= \left(\begin{array}{c} \mathbf{T}^T_{\sigma,\tau}\mathbf{R}_{H^s,\tau}\mathbf{y}_{\Gamma_Y,\tau} \\ \mathbf{0} \\ \mathbf{f}_\sigma \\ \mathbf{0} \end{array}\right) =: \mathbf{F}'_{J_\mathcal{N}} \ . \tag{4.4.5}$$

Now we could apply the CG algorithm to $\mathbf{N}'_{J_\mathcal{N}}\mathbf{U}'_{J_\mathcal{N}} = \mathbf{F}'_{J_\mathcal{N}}$ directly if this operator is found to be positive definite. This should lower the total amount of steps needed to calculate $\mathbf{U}'_{J_\mathcal{N}}$ up to discretization accuracy.
However, the positivity is not assured. As a remedy, we present next an algorithm for **iteratively** solving the systems of equations (4.2.49).

4.4.2 An Inexact Gradient Solver

A simple gradient algorithm is sufficient to minimize the functional $\mathbf{J}$ depending only on $\mathbf{u}$ derived in (4.2.11). The algorithm starts out with a vector $\mathbf{x}_\pi$ and calculates in each step the gradient of the functional at the current point to determine the direction in which to proceed. After it has walked in this direction with a certain step size ρ_i, the cycle begins anew until a minimum (or a certain proximity to a minimum) is reached. This algorithm reads

Algorithm 4.23 BASIC$[\delta, \{\rho_i\}, \mathbf{x}_\pi] \to \mathbf{u}^\delta_\pi$

(I) `Set` $k := 0$
$\mathbf{u}^{(0)}_\pi := \mathbf{x}_\pi$

(II) `Repeat`
$k \leftarrow k+1$
$\mathbf{u}^{(k)}_\pi \leftarrow \mathbf{u}^{(k-1)}_\pi - \rho_{k-1}\,\delta\mathbf{J}(\mathbf{u}^{(k-1)}_\pi)$
`Until` $\|\mathbf{u}^{(k)}_\pi - \mathbf{u}^{(k-1)}_\pi\|_{\ell_2} \le \rho_{k-1}\,\delta$

(III) `Return` $\mathbf{u}^{(k)}_\pi$

The convergence of the gradient algorithm can abstractly be determined by characteristics of the functional (4.2.37). A general result for twice differentiable, strictly convex functionals like our functional (4.2.37) can be found in [17]. The possible range of the step size parameters ρ_i, for which this algorithm converges, is determined by the following proposition from [46].

Proposition 4.24 *Algorithm 4.23 converges for any initial guess $\mathbf{x}_\pi$ for ρ_i satisfying*

$$0 < \rho_* \le \rho_i \le \rho^* < 2\frac{c_*}{(C_*)^2}, \tag{4.4.6}$$

where c_* and C_* are the constants defined in (4.2.45). The parameters ρ_* and ρ^* are fixed chosen values.

To bring Algorithm 4.23 into a computationally more efficient form, the alternative characterization of $\delta\mathbf{J}(\mathbf{u})$ in (4.2.52) is used. To this end, consider the independent systems of (4.2.60),

$$\mathbf{L}_{J_\mathcal{H}}\left(\begin{array}{c} \mathbf{y}^{(i+1)}_\sigma \\ \mathbf{p}^{(i+1)}_\pi \end{array}\right) = \left(\begin{array}{c} \mathbf{f}_\sigma \\ \mathbf{u}^{(i)}_\pi \end{array}\right), \tag{4.4.7}$$

and

$$\mathbf{L}_{J_{\mathcal{H}}}^T \begin{pmatrix} \mathbf{z}_\sigma^{(i+1)} \\ \boldsymbol{\mu}_\pi^{(i+1)} \end{pmatrix} = \begin{pmatrix} -\mathbf{T}_{\sigma,\tau}^T \mathbf{R}_{H^s,\tau}(\mathbf{T}_{\sigma,\tau}\mathbf{y}_\sigma^{(i+1)} - \mathbf{y}_{\Gamma_Y,\tau}) \\ \mathbf{0} \end{pmatrix}. \tag{4.4.8}$$

The importance of these equations is that the evaluation of $\delta\mathbf{J}(\mathbf{u}_\pi^i)$ is equivalent to

$$\delta\mathbf{J}(\mathbf{u}_\pi^{(i)}) = \omega\, \mathbf{R}_{H^t,\pi}\mathbf{u}_\pi^{(i)} - \boldsymbol{\mu}_\pi^{(i+1)}, \tag{4.4.9}$$

where $\boldsymbol{\mu}_\pi^{(i+1)}$ is part of the solution of (4.4.8). The proof of this correlation can be found in [46]. Thus, Algorithm 4.23 can be implemented directly using other algorithms to calculate the solutions of (4.4.7) and (4.4.8). This can be done, for example, by direct solvers like **QR**- and **LU**-decomposition for $\mathbf{L}_{J_{\mathcal{H}}}$ on the lower levels $j \approx j_0$. However, for higher levels we have to resort to fully iterative solvers like the CG (Algorithm 3.25) or Uzawa algorithms (Algorithm 3.26, Algorithm 3.27).
These algorithms require a tolerance limit which determines the precision of the calculated solutions. Because the tolerance cannot - and shall not - be enforced too strictly, it is, for example, restricted by **machine precision**, we call the following algorithm an **inexact gradient** method. This notation pays tribute to the fact that the inversion of the matrix $\mathbf{L}_{J_{\mathcal{H}}}$ by iterative solvers is not exactly the application of the matrix $\mathbf{L}_{J_{\mathcal{H}}}^{-1}$. The inner tolerances $\varepsilon_y(k+1, J_{\mathcal{H}})$ and $\varepsilon_\mu(k+1, J_{\mathcal{H}})$ are specified in Theorem 4.26 below.

Algorithm 4.25 INEXACT GRADIENT$[\delta, \{\rho_i\}, \mathbf{x}_\pi] \to \mathbf{u}_\pi$

(I) `Set` $k := 0$
$\mathbf{u}_\pi^{(k)} := \mathbf{x}_\pi$

(II) `Repeat`

(II.1) `Calculate the approximate solution` $(\widetilde{\mathbf{y}}_\sigma^{(k+1)}, \widetilde{\mathbf{p}}_\pi^{(k+1)})^T$ `of (4.4.7) with right hand side` $(\mathbf{f}_\sigma, \mathbf{u}_\pi^{(k)})^T$ `and initial guess` $(\widetilde{\mathbf{y}}_\sigma^{(k)}, \widetilde{\mathbf{p}}_\pi^{(k)})^T$ `until`

$$\left\| \mathbf{L}_{J_{\mathcal{H}}} \begin{pmatrix} \widetilde{\mathbf{y}}_\sigma^{(k+1)} \\ \widetilde{\mathbf{p}}_\pi^{(k+1)} \end{pmatrix} - \begin{pmatrix} \mathbf{f}_\sigma \\ \mathbf{u}_\pi^{(k)} \end{pmatrix} \right\|_{\ell_2} < \varepsilon_y(k+1, J_{\mathcal{H}})$$

(II.2) `Calculate the approximate solution` $(\widetilde{\mathbf{z}}_\sigma^{(k+1)}, \widetilde{\boldsymbol{\mu}}_\pi^{(k+1)})^T$ `of (4.4.8) with right hand side` $(-\mathbf{T}_{\sigma,\tau}^T \mathbf{R}_{H^s,\tau}(\mathbf{T}_{\sigma,\tau}\widetilde{\mathbf{y}}_\sigma^{(k+1)} - \mathbf{y}_{\Gamma_Y,\tau}), \mathbf{0})^T$ `and initial guess` $(\widetilde{\mathbf{z}}_\sigma^{(k)}, \widetilde{\boldsymbol{\mu}}_\pi^{(k)})^T$ `until`

$$\left\| \mathbf{L}_{J_{\mathcal{H}}}^T \begin{pmatrix} \widetilde{\mathbf{z}}_\sigma^{(k+1)} \\ \widetilde{\boldsymbol{\mu}}_\pi^{(k+1)} \end{pmatrix} - \begin{pmatrix} -\mathbf{T}_{\sigma,\tau}^T \mathbf{R}_{H^s,\tau}(\mathbf{T}_{\sigma,\tau}\widetilde{\mathbf{y}}_\sigma^{(k+1)} - \mathbf{y}_{\Gamma_Y,\tau}) \\ \mathbf{0} \end{pmatrix} \right\|_{\ell_2} < \varepsilon_\mu(k+1, J_{\mathcal{H}})$$

(II.3) `Update` $\mathbf{u}_\pi^{(k)}$ `by`
$\mathbf{r}_\pi^{(k)} := \mathbf{u}_\pi^{(k)} - \omega^{-1}\mathbf{R}_{H^t,\pi}^{-1}\widetilde{\boldsymbol{\mu}}_\pi^{(k+1)}$
$\mathbf{u}_\pi^{(k+1)} \leftarrow \mathbf{u}_\pi^{(k)} - \rho_k\, \mathbf{r}_\pi^{(k)}$
`and`
$k \leftarrow k+1$

`Until` $\|\mathbf{r}_\pi^{(k-1)}\|_{\ell_2} \le \delta$

(III) `Return` $\mathbf{u}_\pi^{(k)}$

Before we specify further implementation details of this gradient scheme, recall from [8] that the convergence speed θ_{grad} of the gradient method 4.23 depends on the spectral condition number of the symmetric

positive definite matrix $\widehat{\mathbf{A}}_J$ in (4.2.51), i.e.,

$$0 \leq \theta_{\text{grad}} := \frac{\kappa_2(\widehat{\mathbf{A}}_J) - 1}{\kappa_2(\widehat{\mathbf{A}}_J) + 1} < 1 \ . \tag{4.4.10}$$

This means that for every iteration of Algorithm 4.23 there is some $0 < \theta \leq \theta_{\text{grad}}$ such that the error is reduced in each iteration step by a fixed fraction, i.e.,

$$\|\mathbf{u}_\pi^{(k+1)} - \mathbf{u}_\pi^*\|_{\ell_2} \leq \theta \, \|\mathbf{u}_\pi^{(k)} - \mathbf{u}_\pi^*\|_{\ell_2}, \tag{4.4.11}$$

where $\mathbf{u}_\pi^*$ is the exact element of $\ell_2(I\!I_\pi)$ minimizing $\mathbf{J}(\cdot)$ from (4.2.40). Let $\tau(\mathbf{u}_\pi^*)$ be an estimate for the discretization error on level π, i.e.,

$$\|\mathbf{u}_\pi^{(0)} - \mathbf{u}_\pi^*\|_{\ell_2} \leq \tau(\mathbf{u}_\pi^*) \ . \tag{4.4.12}$$

The following result from [46] shows the dependence of the tolerances involved in the above formulation.

Theorem 4.26 *If the tolerances $\varepsilon_y(i+1, J_\mathcal{H})$ and $\varepsilon_\mu(i+1, J_\mathcal{H})$ in Algorithm 4.25 are selected at each stage according to*

$$\varepsilon_y(i+1, J_\mathcal{H}) := \frac{1}{2} \frac{\mathbf{c_L}}{\mathbf{C_L}\rho_i} \frac{\theta^i}{(i+1)^2} \tau(\mathbf{u}_\pi^*), \tag{4.4.13}$$

$$\varepsilon_\mu(i+1, J_\mathcal{H}) := \frac{1}{2} \frac{1}{\mathbf{C_L}\rho_i} \frac{\theta^i}{(i+1)^2} \tau(\mathbf{u}_\pi^*), \tag{4.4.14}$$

then Algorithm 4.25 converges for any initial guess $\mathbf{x}_\pi$ for ρ_i satisfying

$$0 < \rho_* \leq \rho_i \leq \rho^* < 2\frac{c_*}{(C_*)^2}, \tag{4.4.15}$$

where the constants c_* and C_* are defined in (4.2.45) with $\mathbf{c_L}$ and $\mathbf{C_L}$ from (3.1.29).

Computational Work in a Nested Iteration Scheme

The above Algorithm 4.25 is optimal in the sense that it only requires a fixed amount of iterations to reduce the error $\|\mathbf{u}_\pi^{(i)} - \mathbf{u}_\pi^*\|_{\ell_2}$ by any constant factor. This behavior will now be examined in a nested iteration scheme (Algorithm 3.23).

Let us assume that after i_j steps of Algorithm 4.25 we have determined the element $\mathbf{u}_j^{(i_j)}$ on level j which satisfies

$$\|\mathbf{u}_j^{(i_j)} - \mathbf{u}_j^*\|_{\ell_2} \leq \frac{1}{2}\tau(\mathbf{u}_j^*), \tag{4.4.16}$$

with $\tau(\mathbf{u}_j)$ defined in (4.4.12) and $\mathbf{u}_j^*$ being the exact solution of level j. Now prolongate $\mathbf{u}_j^{(i_j)}$ to level $j+1$ to be used as initial guess for the iteration on that level. This can be done easily in the wavelet framework by adding zeros at the correct positions of the vector $\mathbf{u}_j^{(i_j)}$. In view of Remark 3.22, we write

$$\mathbf{u}_{j+1}^{(0)} := \mathbf{u}_j^{(i_j)} \ . \tag{4.4.17}$$

The number i_{j+1} of iterations necessary to reduce the error analogous to (4.4.16),

$$\|\mathbf{u}_j^{(i_{j+1})} - \mathbf{u}_{j+1}^*\|_{\ell_2} \leq \frac{1}{2}\tau(\mathbf{u}_{j+1}^*), \tag{4.4.18}$$

can now be shown to be independent of j, see [46]. Since we have, thus, by construction

$$\frac{\varepsilon_\mu(i+1, j)}{\varepsilon_\mu(i, j)} = \frac{\varepsilon_y(i+1, j)}{\varepsilon_y(i, j)} \approx \frac{\tau(\mathbf{u}_{j+1})}{\tau(\mathbf{u}_j)} \longrightarrow \text{const} < 1,$$

we only need to reduce the error on each level by a constant factor. We know that this can be done in a constant amount of steps i_j for each j. The error bounds given by (4.4.13) and (4.4.14) can, thus, be achieved with a number of steps independent of the discretization and level, if we prolongate $\widetilde{\mathbf{y}}_j^{(i_j)}$ and

$\widetilde{\boldsymbol{\mu}}_j^{(i_j)}$ for use in the next iteration of the nested iteration scheme.
If the matrices in (II.1) and (II.2) from Algorithm 4.25 have optimally bounded spectral condition numbers, we know that the Uzawa algorithms Algorithm 3.26 and Algorithm 3.27 have a convergence rate independent of the discretization and level. Thus, only a uniformly bounded number of Uzawa steps and, consequently, computational work of order $\mathcal{O}(\#\mathbb{I}_{\mathcal{H},J_{\mathcal{H}}})$ is needed in each calculation.
As we have mentioned in Section 4.3.1, the inversion of the Riesz operator can be done in linear time. Step (II.3) can therefore be neglected for the overall complexity at this point.
Using a direct solver like **LU**- or **QR**-decomposition on level $j = j_0$ and **prolongation** operators for every step, we can conclude by a geometric series argument that the overall accumulated complexity is always proportional to the number of unknowns on the current level, i.e., $\mathcal{O}(\#\mathbb{I}_{\mathcal{N},J_{\mathcal{N}}})$. To summarize this, we quote from [46].

Theorem 4.27 *If in each iteration of Algorithm 4.25 the systems in steps* (II.1) *and* (II.2) *are solved up to tolerances (4.4.13) and (4.4.14) and these solutions are taken as initial guesses for the next higher level, then Algorithm 4.25 is an asymptotically optimal method in the sense that it provides the solution up to the discretization error on levels $J_{\mathcal{N}}$ in an overall amount of $\mathcal{O}(\#\mathbb{I}_{\mathcal{N},J_{\mathcal{N}}})$ operations, where $\#\mathbb{I}_{\mathcal{N},J_{\mathcal{N}}}$ is the total number of unknowns in steps* (II.1)*,* (II.2) *and* (II.3)*.*

5 Numerical Results

5.1 Numerical Evaluation of the Wavelet Bases

Wavelet theory offers a lot of features for theoretical and practical use, all of which has been outlined in Section 2. The characteristic mostly payed attention to is the spectral **condition number** of the involved system operator. These condition numbers were shown to be bounded uniformly, i.e., independent of the discretization and level under certain circumstances, see (2.2.31) and Section 3.1.3. The actual values of these constant bounds depend strongly on the type of wavelets and the preconditioner used in the discretization. Wavelet theory permits many tweaks to further reduce its absolute value, like improved diagonal preconditioners (like (3.2.34)) or simple basis transformations (see Section 2.3.3).
We have several different wavelet constructions and optimizations thereof available for using in applications. We will now show a list of these and the abbreviations used on the following pages to denote them:

DKU This MRA provides the groundwork for the following types. It was introduced by Dahmen, Kunoth and Urban in [30]; their construction is outlined in Section 2.3 . Images of the primal and dual wavelets were presented in Section 2.3.2 . The primal generators are defined in (2.3.25). When displaying functions expanded using this basis, their coefficient vectors must be transformed to the nodal basis first.

DKU-Orth This MRA is implemented with the orthogonal basis transformation (2.3.39) which is specifically useful for reducing the condition numbers of the stiffness matrices.

B As mentioned in Section 2.3.2, the wavelets of this MRA are scaled according to Lemma 2.41, using the value defined in (2.3.28). This scaling was first introduced by Carsten Burstedde in [11], hence the name "B"[1]. The generator bases are taken as in the DKU-case.

B-Nodal This derivation of the B-type is constructed by applying a basis transformation of the form (2.3.30). The primal boundary generators are transformed to the nodal basis.

B-SVD This derivative of the DKU-type is constructed by applying the block singular value decomposition transformation defined in (2.3.35). The parameter q is chosen differently for varying spatial dimensions to achieve best effect in any case.

B-Orth This MRA implements the orthogonal basis transformation (2.3.39) specifically designed for reducing the condition numbers of the stiffness matrices.

P This MRA was designed by Miriam Primbs [56] with the focus on particularly low condition numbers of the fast wavelet transform. They should offer all the same traits as the DKU type; but construction is still "work in progress". These wavelets have just been included for testing purposes and the results should be considered preliminary. The primal generator basis consists merely of the linear nodal hat functions restricted to the interval $(0, 1)$.

P-Orth Orthogonal basis transformation applied to the P-type MRA.

The actual computational results do not only depend on the type of wavelets and the applied basis transformations but also on the kind of preconditioner used, as we will see in the numerical experiments. Certain basis transformations and the right preconditioner combined can yield improvements in the absolute values of the condition numbers by several orders of magnitude.

Remark 5.1 *This section focuses on the properties of the building blocks used for the control problem of Section 4.3. We like to emphasize that no operators are explicitly assembled in any application, as this would spoil the sparsity of the matrices in wavelet discretization. The images of operators presented on the next pages are made by consecutively calculating the matrix products with the unity vectors, one at a time. This technique is employed for illustration purposes only and never in calculations.*

[1] A more appropriate name might have been "DKU-B", but we traded accurateness for readability here.

5.1.1 The Fast Wavelet Transform

The spectral condition numbers of the multiscale transformation matrices $\mathbf{T}_J$ are uniformly bounded according to Theorem 2.13 in order to achieve the **Riesz basis** property for the wavelet bases. We show the computed numbers for the different types described in Section 5.1 in Table 5.1. Since $\mathbf{T}_J$ is not symmetric, we calculate $\sqrt{\kappa_2(\mathbf{T}_J^T\mathbf{T}_J)}$ as an approximation. Note that the condition number of the Fast Wavelet Transform does not allow direct transfer to the condition numbers of any operators: a lower bound does not imply a lower bound for the condition of the stiffness matrix.

J	$\#\Delta_J$	$\kappa_2(\mathbf{T}_{\text{DKU}})$	$\kappa_2(\mathbf{T}_{\text{B}})$	$\kappa_2(\mathbf{T}_{\text{B-Nodal}})$	$\kappa_2(\mathbf{T}_{\text{P}})$
3	9	1.0000e+00	1.0000e+00	1.0000e+00	
4	17	4.7433e+00	4.6401e+00	3.8924e+00	1.0000e+00
5	33	6.2211e+00	6.0239e+00	5.2485e+00	1.5666e+00
6	65	8.1548e+00	6.8603e+00	6.3985e+00	2.0994e+00
7	129	9.4736e+00	7.3961e+00	7.1528e+00	2.4861e+00
8	257	1.0236e+01	7.7073e+00	7.5827e+00	2.7362e+00
9	513	1.0648e+01	7.8764e+00	7.8133e+00	2.9024e+00
10	1.025	1.0863e+01	7.9652e+00	7.9334e+00	3.0145e+00
11	2.049	1.0974e+01	8.0108e+00	7.9948e+00	3.0922e+00
12	4.097	1.1030e+01	8.0340e+00	8.0260e+00	3.1474e+00
14	16.385	1.1072e+01	8.0516e+00	8.0496e+00	3.2174e+00
16	65.535	1.1083e+01	8.0561e+00	8.0556e+00	3.2569e+00
$\sim$	2^J	1	1	1	1

Table 5.1: Comparison of the condition numbers of several fast wavelet transform matrices in 1D. Because of definition (2.1.28) the wavelet transform on the minimum level $j_0 = 3$ is the identity and its condition must be exactly 1 in any case. By comparing the columns $\mathbf{T}_{\text{B}}$ and $\mathbf{T}_{\text{B-Nodal}}$, we can see the negative impact of the boundary adapted primal generators Φ_{j_0} of (2.3.25) on the condition numbers on small levels. The after-effect of this small adaption in the case of $\mathbf{T}_{\text{B-Nodal}}$ vanishes for higher levels as more wavelets are added and the relative importance of the single-scale basis Φ_{j_0} thus decreases. The P-type wavelet transform was designed to exhibit very small absolute condition numbers.

The condition numbers of the tensorized fast wavelet transform matrices are simply powers of the numbers in Table 5.1, e.g., in 2D all numbers are squared.

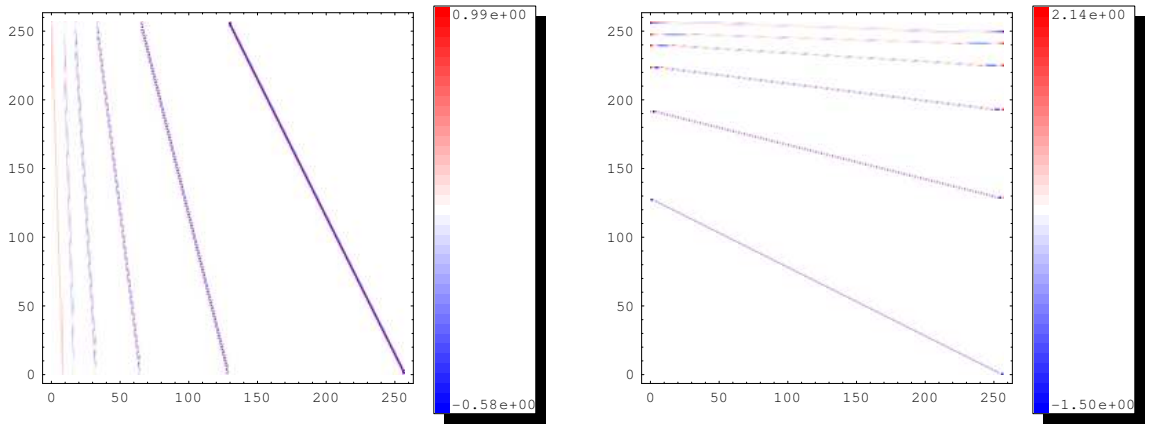


Figure 5.1: General form of the Fast Wavelet Transform $\mathbf{T}_J$ and $\mathbf{T}_J^{-1}$ for the B-type MRA in 1D. The left graphic shows the fast wavelet transform from level $j = j_0 = 3$ up to $J = 8$ and the right graphics its inverse.

5.1.2 Stiffness Matrices

The spectral properties of the stiffness matrix $\mathbf{A}$ are generally considered to be the primary criteria for the effectiveness of a wavelet basis. It is of special interest in our application since it largely determines the condition of the **Schur complement** (3.2.15) and, thus, the speed of Algorithms 3.26 and 3.27.

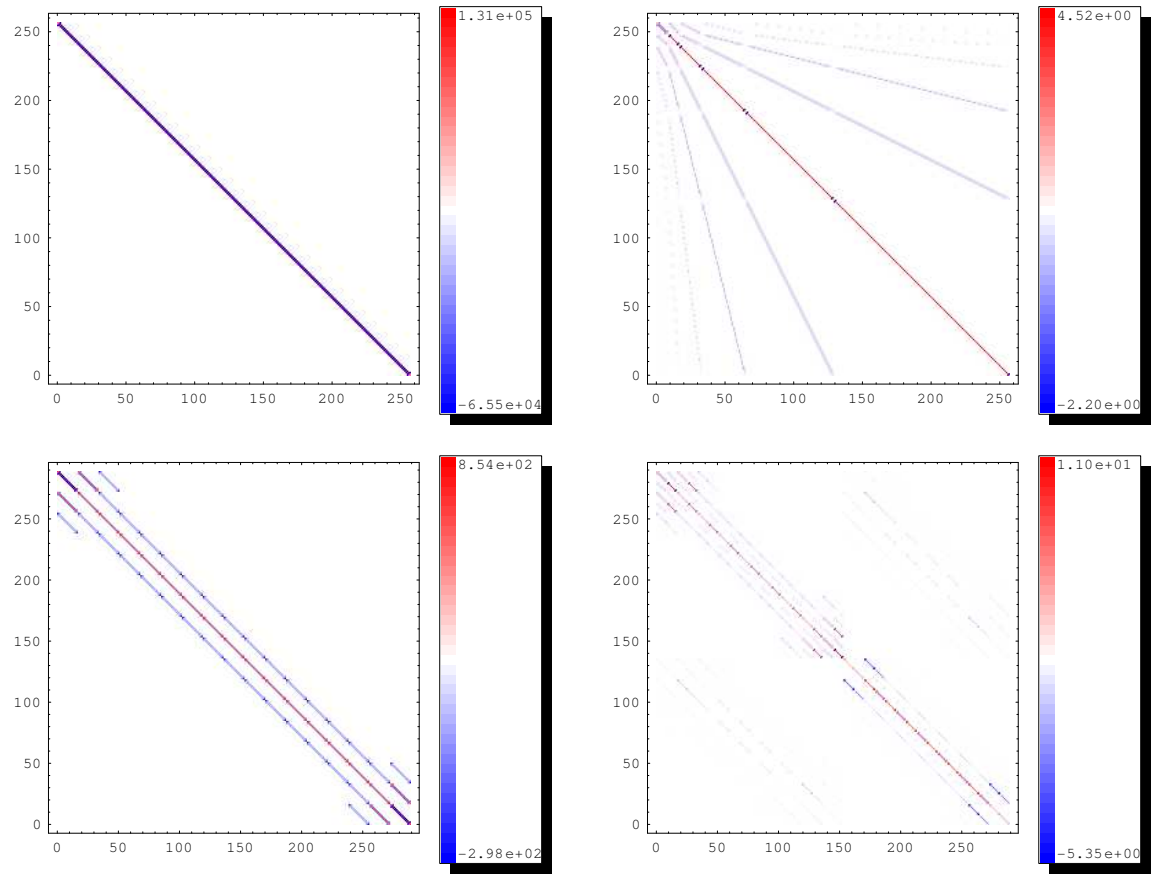


Figure 5.2: General form of the **stiffness matrix** (with B-type MRA). In the first row, we display the stiffness matrix on level $j = 8$ in 1D, assembled in the generator basis on the left and transformed into the properly scaled wavelet basis on the right. The second row presents the 2D stiffness matrix on level $j = 4$ (left: single-scale basis; right: in wavelet basis).

In Tables 5.2, 5.3 and 5.4 we show the condition numbers of the stiffness matrix for several wavelet constructions and the standard preconditioner $\mathbf{D}_1^{-1}$ from (2.2.14). The column $\kappa_2\left(\mathbf{A}_{\Phi_j}\right)$ depicts the condition numbers for the stiffness matrix assembled in the **single-scale basis** Φ_j of (2.3.25) with no preconditioning used. We see that the DKU-type wavelets exhibit the best condition numbers for this preconditioner. Although the border block transformation (2.3.37) (columns $\mathbf{A}_{\text{B-SVD}}$) proves to be well suited for 1D, the tensor product construction does not apply for higher dimensions. The B-Orth and P-Orth constructions are not listed in these tables because there can be no change in condition numbers compared to the B− and P−type, respectively, when using the standard preconditioner.
Tables 5.5, 5.6 and 5.7 show the results for several wavelet types and the preconditioner $\mathbf{D}_a^{-1}$. This preconditioner generally produces lower condition numbers as it is literally adapted for "itself".
Note that there is no difference in this case between the B- and DKU-types as the diagonal of the stiffness matrix incorporates the scaling (2.3.28) and, thus, annihilates its effect when used for preconditioning. The same holds true for B-Orth and DKU-Orth. The B-Orth- and P-Orth-types now present perfectly conditioned stiffness matrices for $j = j_0$ in 1D. The effect carries over quite well to higher dimensions. The observed increase of the condition numbers for higher levels $j > j_0$ now ideally depends on the characteristics of the wavelets only. The P-Orth-type exhibits exceptionally good results, although the numbers of the P-type in 1D are inferior to the other constructions. Yet here the values in 2D and 3D

seem to be powers of the 1D condition numbers, see Tables 5.5 – 5.7, and, thus, give the lowest absolute values.
In summary, we see that there is ample room for obtaining absolute small condition numbers for optimized constructions of wavelets.

J	$\#\Delta_J$	$\kappa_2(\mathbf{A}_{\Phi_j})$	$\kappa_2(\mathbf{A}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{A}_{\mathrm{B}})$	$\kappa_2(\mathbf{A}_{\text{B-SVD}})$	$\kappa_2(\mathbf{A}_{\mathrm{P}})$
3	9	2.3221e+02	2.3221e+02	2.3221e+02	6.7509e+01	
4	17	9.8139e+02	2.3780e+02	3.5062e+02	1.0521e+02	1.0794e+03
5	33	4.0480e+03	2.5716e+02	3.9257e+02	1.1799e+02	1.0845e+03
6	65	1.6449e+04	2.6705e+02	4.3359e+02	1.3099e+02	1.1909e+03
7	129	6.6319e+04	2.7606e+02	4.6673e+02	1.4128e+02	1.2402e+03
8	257	2.6631e+05	2.8266e+02	4.9278e+02	1.4937e+02	1.2836e+03
10	1.025	4.2736e+06	2.9249e+02	5.3141e+02	1.6130e+02	1.3399e+03
12	4.097	6.8427e+07	2.9924e+02	5.5797e+02	1.6948e+02	1.3761e+03
14	16.385	1.0950e+09	3.0413e+02	5.7698e+02	1.7532e+02	1.4008e+03
16	65.535	1.7521e+10	3.0781e+02	5.9102e+02	1.7963e+02	1.4187e+03
$\sim$	2^J	2^{2J}	1	1	1	1

Table 5.2: (Spectral) Condition numbers of 1D **stiffness matrices** using $\mathbf{D}_1^{-1}$ from (2.2.14) for preconditioning. The value of the parameter for the case B-SVD is optimized as $q = 1.41789$.

J	$\#\Delta_J$	$\kappa_2(\mathbf{A}_{\Phi_j})$	$\kappa_2(\mathbf{A}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{A}_{\mathrm{B}})$	$\kappa_2(\mathbf{A}_{\text{B-SVD}})$	$\kappa_2(\mathbf{A}_{\mathrm{P}})$
3	81	5.3201e+02	5.3201e+02	5.3201e+02	2.1497e+02	
4	289	2.4219e+03	6.4240e+02	1.0965e+03	6.9637e+02	1.1437e+03
5	1.089	1.0330e+04	7.8598e+02	1.5029e+03	1.0172e+03	1.3180e+03
6	4.225	4.2652e+04	9.3794e+02	1.8667e+03	1.2812e+03	1.9569e+03
7	16.641	1.7331e+05	1.1070e+03	2.2022e+03	1.5115e+03	2.5998e+03
8	66.049	6.9872e+05	1.2640e+03	2.5249e+03	1.7361e+03	3.1898e+03
9	263.169	2.8058e+06	1.4005e+03	2.8005e+03	1.9266e+03	3.6910e+03
10	1.050.625	1.1245e+07	1.5143e+03	3.0301e+03	2.0853e+03	4.1110e+03
11	4.198.401	4.5025e+07	1.6095e+03	3.2214e+03	2.2173e+03	4.4582e+03
12	16.785.409	1.8018e+08	1.6888e+03	3.3808e+03	2.3272e+03	4.7455e+03
$\sim$	2^{2J}	2^{2J}	1	1	1	1

Table 5.3: Condition numbers of 2D **stiffness matrices** using $\mathbf{D}_1^{-1}$ of (2.2.14) for preconditioning. The value of parameter q in case B-SVD is optimized as $q = 0.9895$.

J	$\#\Delta_J$	$\kappa_2(\mathbf{A}_{\Phi_j})$	$\kappa_2(\mathbf{A}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{A}_{\mathrm{B}})$	$\kappa_2(\mathbf{A}_{\text{B-SVD}})$	$\kappa_2(\mathbf{A}_{\mathrm{P}})$
3	729	1.2385e+03	1.2385e+03	1.2385e+03	4.2818e+03	
4	4.913	5.7509e+03	1.2885e+04	3.4338e+03	2.6037e+05	1.2119e+03
5	35.937	2.5338e+04	2.7003e+04	7.1069e+03	6.8250e+05	1.6603e+03
6	274.625	1.0630e+05	4.4642e+04	1.1601e+04	1.1667e+06	3.0151e+03
7	2.146.689	4.3536e+05	6.3673e+04	1.6556e+04	1.6672e+06	4.6059e+03
8	16.974.593	1.7621e+06	8.2879e+04	2.1188e+04	>2.0e+06	6.1706e+03
$\sim$	2^{3J}	2^{2J}	1	1	1	1

Table 5.4: Condition numbers of 3D **stiffness matrices** using $\mathbf{D}_1^{-1}$ of (2.2.14) for preconditioning. The value of parameter q in case B-SVD is 1.85. There was no reducing effect observed (compared to B-type) for any value.

J	$\#\Delta_J$	$\kappa_2(\mathbf{A}_{\text{DKU}})$	$\kappa_2(\mathbf{A}_{\text{B}})$	$\kappa_2(\mathbf{A}_{\text{B-Orth}})$	$\kappa_2(\mathbf{A}_{\text{P}})$	$\kappa_2(\mathbf{A}_{\text{P-Orth}})$
3	9	2.2872e+02	2.2872e+02	1.0000e+00		
4	17	2.4399e+02	2.4399e+02	7.3431e+00	1.0243e+03	1.0000e+00
5	33	2.5497e+02	2.5497e+02	8.9868e+00	1.0508e+03	2.1545e+00
6	65	2.6201e+02	2.6201e+02	1.1113e+01	1.1515e+03	2.9969e+00
7	129	2.6714e+02	2.6715e+02	1.2212e+01	1.2005e+03	3.6346e+00
8	257	2.7075e+02	2.7074e+02	1.3225e+01	1.2424e+03	4.1105e+00
10	1.025	2.7532e+02	2.7532e+02	1.4470e+01	1.2961e+03	4.8232e+00
12	4.097	2.7789e+02	2.7789e+02	1.5258e+01	1.3300e+03	5.2797e+00
14	16.385	2.7939e+02	2.7939e+02	1.5782e+01	1.3528e+03	5.5875e+00
16	65.535	2.8029e+02	2.8029e+02	1.6145e+01	1.3689e+03	5.8064e+00
$\sim$	2^J	1	1	1	1	1

Table 5.5: Condition numbers of 1D **stiffness matrices** using for preconditioning $\mathbf{D}_a^{-1}$ of (3.2.34) and $\mathbf{D}_{\{O,a\}}^{-1}$ from (2.3.45) for the cases that include the orthogonal basis transformation.

J	$\#\Delta_J$	$\kappa_2(\mathbf{A}_{\text{DKU}})$	$\kappa_2(\mathbf{A}_{\text{B}})$	$\kappa_2(\mathbf{A}_{\text{B-Orth}})$	$\kappa_2(\mathbf{A}_{\text{P}})$	$\kappa_2(\mathbf{A}_{\text{P-Orth}})$
3	81	5.1949e+02	5.1949e+02	1.8982e+01		
4	289	6.2705e+02	6.2705e+02	1.0664e+02	1.0243e+03	2.8084e+00
5	1.089	6.5216e+02	6.5216e+02	1.4229e+02	1.2234e+03	6.4460e+00
6	4.225	6.8300e+02	6.8300e+02	1.8203e+02	1.5235e+03	9.4769e+00
7	16.641	7.0367e+02	7.0367e+02	2.1625e+02	1.8212e+03	1.2958e+01
8	66.049	7.2046e+02	7.2046e+02	2.4572e+02	2.0178e+03	1.6415e+01
9	263.169	7.3363e+02	7.3363e+02	2.6797e+02	2.2319e+03	1.9333e+01
10	1.050.625	7.4451e+02	7.4451e+02	2.8627e+02	2.4054e+03	2.1745e+01
11	4.198.401	7.5366e+02	7.5366e+02	3.0054e+02	2.5463e+03	2.3926e+01
12	16.785.409	7.6162e+02	7.6162e+02	3.1194e+02	2.7773e+03	2.5795e+01
$\sim$	2^{2J}	1	1	1	1	1

Table 5.6: Condition numbers of 2D **stiffness matrices** using for preconditioning $\mathbf{D}_a^{-1}$ of (3.2.34) and $\mathbf{D}_{\{O,a\}}^{-1}$ from (2.3.45) for the cases that include the orthogonal basis transformation.

J	$\#\Delta_J$	$\kappa_2(\mathbf{A}_{\text{DKU}})$	$\kappa_2(\mathbf{A}_{\text{B}})$	$\kappa_2(\mathbf{A}_{\text{B-Orth}})$	$\kappa_2(\mathbf{A}_{\text{P}})$	$\kappa_2(\mathbf{A}_{\text{P-Orth}})$
3	729	1.1029e+03	1.1029e+03	3.6268e+02		
4	4.913	1.9175e+03	1.9175e+03	1.8235e+03	1.0243e+03	7.8982e+00
5	35.937	2.8338e+03	2.8338e+03	2.6919e+03	2.0611e+03	3.2286e+01
6	274.625	3.3821e+03	3.3821e+03	3.5333e+03	2.4346e+03	4.4821e+01
7	2.146.689	4.0410e+03	4.0410e+03	4.2428e+03	2.8897e+03	6.1539e+01
8	16.974.593	4.6128e+03	4.6128e+03	4.8142e+03	3.3803e+03	8.0971e+01
$\sim$	2^{3J}	1	1	1	1	1

Table 5.7: Condition numbers of 3D **stiffness matrices** using for preconditioning $\mathbf{D}_a^{-1}$ of (3.2.34) and $\mathbf{D}_{\{O,a\}}^{-1}$ from (2.3.45) for the cases that include the orthogonal basis transformation.

5.1.3 The Saddle Point Operator

The saddle point operator $\mathbf{L}_J$ is given by (3.1.35). We only consider the case $\Omega = \square^2$ and $\Gamma = \Gamma_E$. The operator $\mathbf{L}$ is indefinite by nature and its condition number $\kappa_2(\mathbf{L}_J)$ can actually become smaller than the one of the operator $\mathbf{A}_J$ which it inherits. The actual computed values can be found in Table 5.8 and Table 5.9. Although this might seem contradictory, for the **Schur complement** $\mathbf{S}_J = \mathbf{B}_J\mathbf{A}_J^{-1}\mathbf{B}_J^T$ we have

$$\kappa_2(\mathbf{S}_J) \leq \kappa_2(\mathbf{A}_J)\,\kappa_2(\mathbf{B}_J\mathbf{B}_J^T)\ . \tag{5.1.1}$$

As it was pointed out in Section 3.3.2, $\kappa_2(\mathbf{S})$ determines the convergence speed of the Uzawa type algorithms Algorithm 3.26 and Algorithm 3.27. The corresponding results can be found in Table 5.11 and Table 5.12. It should be noted that, although the number of unknowns only grows proportional to 2^J, the complexity of the computation $\mathbf{A}_J^{-1}\mathbf{x}$ for every vector $\mathbf{x}$ is generally $\mathcal{O}(2^{2J})$. This inner inversion was performed computationally with the CG solver. The presence of the inverse operator $\mathbf{A}_J^{-1}$ is responsible for the convergence from above to the limit, e.g., $\kappa_2(\mathbf{S}_B)$ in Table 5.12.

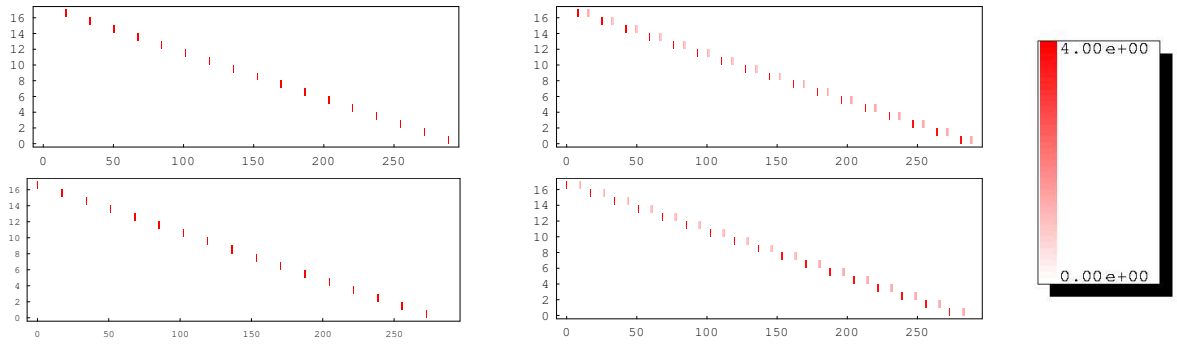


Figure 5.3: The trace operators. We show the discretizations of the trace operators γ_E (first row) and γ_W (second row), on the left assembled in the generator basis and on the right in wavelet coordinates on level $j = 4$. Due to the extreme sparsity, the wavelet representations do not greatly differ on this level. More saturated values indicate higher absolute values.

The results of the theoretical ill-conditioning, outlined in (4.2.69), can be found in Table 5.10. The higher the difference t, the lower is the level j the effect can be observed at.

Iteration Results

We are now going to investigate the efficiency of the Uzawa and CG solvers for a model problem. Consider Problem 3.14 with the right hand side $f \equiv 1$ and $u \equiv 0$ in $n = 2$ dimensions. In this case, the exact solution $y^*_{a_0}$ to (3.2.21) in dependence of a_0 is

$$y^*_{a_0}(x_1, x_2) = \frac{1}{a_0} - \frac{\cosh(\sqrt{a_0}\,x_1)}{a_0\,\cosh(\sqrt{a_0})}\ . \tag{5.1.2}$$

Thus, with increasing values of a_0, the solution attains $y \approx \frac{1}{a_0}$ which tends to zero. For $a_0 \to 0$, we obtain the limit

$$\lim_{a_0 \to 0} y^*_{a_0}(x_1, x_2) = \frac{1}{2}(1 - x_1^2)\ .$$

We show that the computed solutions are consistent with the theory for this right hand side in Figure 5.4 and Figure 5.5.

The iteration results are presented in Table 5.13 through Table 5.15. There we show the iteration numbers for a nested iteration scheme for the elliptic boundary value problem (3.2.21) for $a_0 = 1$. The **nested iteration** is started with a **QR** decomposition on the coarsest level j_0. Then the respective iterative solver is used on the higher levels. The conjugate gradient solver 3.25 is applied to the operator $\mathbf{L}_J^T\mathbf{L}_J$ because $\mathbf{L}_J$ is indefinite. The Uzawa algorithms 3.26 and 3.27 have to solve an equation of the form $\mathbf{A}_J\mathbf{x}_J = \mathbf{b}_J$ in every step, for which we use the CG Solver. Here the terminating condition of the inner

iteration is $\delta_i = \delta/8$. All calculations are executed up to a residual error accuracy of $\frac{1}{100}\,2^{-J}$.
The practical impact of the condition numbers on the iteration schemes can also be observed in these tables. The higher the absolute values of the condition numbers of the operators are, the higher the iteration numbers will be. The norm $\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$ in these tables is equivalent to the $H^1(\Omega)$ norm and the asymptotic decline of this error proportional to norm of the residual error $\|\mathbf{r}_J^{(k_J)}\|$ is clearly visible in all cases.
The spikes in the CG iteration diagram indicate the prolongation in the nested iteration scheme. These spikes can be very high because the matrix $\mathbf{L}_{J+1}^T\mathbf{L}_{J+1}$ is not directly an extension of $\mathbf{L}_J^T\mathbf{L}_J$ and the condition numbers are squared when compared to $\mathbf{L}_J$.

The results indicate that substantial improvements in execution costs can be achieved through proper preconditioning and the basis transformations of Section 2.3.3 . The Uzawa algorithms generally produce the solution up to discretization error accuracy with fewer iteration steps. The dominating factor for the cost of the Uzawa algorithms here seems not to be the number of the outer iterations but the number of internal conjugate gradient steps (#CG-It). This is affirmed by the generally low condition number of the Schur complement $\mathbf{S}_J$, see Table 5.11 and Table 5.12. The condition numbers of the stiffness matrices $\mathbf{A}_J$ vary much larger in magnitude, see Table 5.3 and Table 5.6. Since the UzawaCD algorithm uses conjugate directions to calculate the next descent direction, it calculates the solution with fewer steps than the Uzawa algorithm.

In summary, the UzawaCD algorithm together with the wavelet P-Orth with $\mathbf{D}_{\{O,a\}}$ preconditioner is the fastest of our test candidates, see Table 5.15. This comes as no surprise because this is the construction with the smallest absolute value of the condition number of the stiffness matrix. We will use this MRA and B with $\mathbf{D}_a$ for comparison in the next section for the numerical experiments with our control problem.

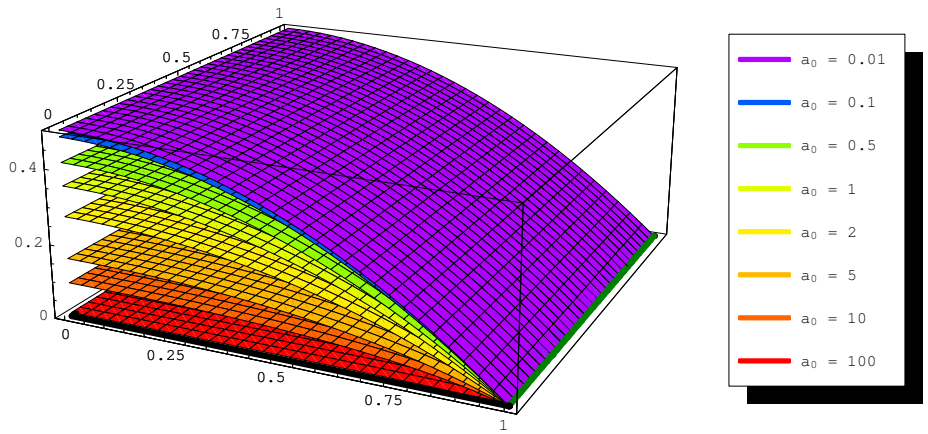


Figure 5.4: Solutions of the boundary value problem Problem 3.14 for $f \equiv 1$ and the Dirichlet boundary Γ as the right facing edge (green), while varying the parameter a_0 of the elliptic PDE.

J	$\#\mathbb{I}_{\mathcal{H},J}$	$\kappa_2(\mathbf{L}_{\Phi_j})$	$\kappa_2(\mathbf{L}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{L}_{\mathrm{B}})$	$\kappa_2(\mathbf{L}_{\text{B-SVD}})$	$\kappa_2(\mathbf{L}_{\mathrm{P}})$
3	90	1.1271e+04	1.6182e+02	1.6182e+02	1.2120e+02	
4	306	8.9747e+04	3.3627e+02	3.3392e+02	4.7790e+02	3.3110e+02
5	1.122	7.1713e+05	5.0524e+02	4.5946e+02	7.9721e+02	3.8359e+02
6	4.290	5.7350e+06	6.9318e+02	5.7168e+02	1.1202e+03	5.7086e+02
7	16.770	4.5875e+07	9.3544e+02	6.7476e+02	1.3782e+03	7.5665e+02
8	66.306	3.6699e+08	1.1396e+03	7.7374e+02	1.6136e+03	9.2678e+02
9	263.682	2.9359e+09	1.3055e+03	8.5821e+02	1.8098e+03	1.0713e+03
10	1.051.650	2.3487e+10	1.4441e+03	9.2859e+02	1.9682e+03	1.1926e+03
11	4.200.450	>1.0e+11	1.5596e+03	9.8722e+02	2.0978e+03	1.2928e+03
12	16.789.506	>1.0e+11	1.6534e+03	1.0361e+03	2.2046e+03	1.3757e+03
$\sim$	2^{2J}	2^{2J}	1	1	1	1

Table 5.8: Condition numbers of 2D **saddle point matrices** $\mathbf{L}_J$ using $\mathbf{D}_1^{-1}$ of (2.2.14) for preconditioning. The values for the B-SVD type were selected as $q = 1.59$ for the 2D transformations and $q = 0.92$ for the 1D transformations, which produced the best results for this matrix.

J	$\#\mathbb{I}_{\mathcal{H},J}$	$\kappa_2(\mathbf{L}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{L}_{\mathrm{B}})$	$\kappa_2(\mathbf{L}_{\text{B-Orth}})$	$\kappa_2(\mathbf{L}_{\mathrm{P}})$	$\kappa_2(\mathbf{L}_{\text{P-Orth}})$
3	90	1.5806e+02	1.5805e+02	1.8826e+01		
4	306	1.9034e+02	1.9023e+02	1.0504e+02	3.8093e+02	4.3877e+00
5	1.122	1.9966e+02	1.9957e+02	1.3996e+02	4.3259e+02	6.9351e+00
6	4.290	2.1118e+02	2.1109e+02	1.8059e+02	4.9442e+02	1.0213e+01
7	16.770	2.3184e+02	2.3174e+02	2.1449e+02	5.6455e+02	1.3956e+01
8	66.306	2.5298e+02	2.5289e+02	2.4407e+02	6.2836e+02	1.7350e+01
9	263.682	2.7063e+02	2.7054e+02	2.6725e+02	6.8222e+02	2.0170e+01
10	1.051.650	2.8549e+02	2.8549e+02	2.8543e+02	7.5436e+02	2.2485e+01
11	4.200.450	2.9587e+02	2.9548e+02	3.0012e+02	7.9262e+02	2.4582e+01
12	16.789.506	3.0514e+02	3.0278e+02	3.1172e+02	8.0243e+02	2.6381e+01
$\sim$	2^{2J}	1	1	1	1	1

Table 5.9: Condition numbers of 2D **saddle point matrices** using for preconditioning $\mathbf{D}_a^{-1}$ of (3.2.34) and $\mathbf{D}_{\{O,a\}}^{-1}$ from (2.3.45) for the cases that include the orthogonal basis transformation.

J	$\#\mathcal{I}_{\mathcal{H},J}$	$t = 0.0$	$t = 0.1$	$t = 0.2$	$t = 0.3$	$t = 0.4$	$t = 0.5$
3	90	1.6182e+02	1.6280e+02	1.6441e+02	1.6685e+02	1.7038e+02	1.8854e+02
4	306	3.3392e+02	3.3629e+02	3.3983e+02	3.5575e+02	6.0615e+02	1.0410e+03
5	1.122	4.5946e+02	4.6272e+02	4.6754e+02	7.2119e+02	1.4184e+03	2.8118e+03
6	4.290	5.7168e+02	5.7572e+02	6.0231e+02	1.3444e+03	3.0450e+03	6.9545e+03
7	16.770	6.7476e+02	6.7952e+02	9.2786e+02	2.3863e+03	6.2307e+03	1.6370e+04
8	66.306	7.7374e+02	7.7920e+02	1.3912e+03	4.1250e+03	1.2405e+04	3.7493e+04
9	263.682	8.5821e+02	8.6427e+02	2.0211e+03	6.9092e+03	2.3918e+04	8.3119e+04
10	1.051.650	9.2859e+02	9.3515e+02	2.8688e+03	1.1302e+04	4.5017e+04	1.7981e+05
$\sim$	2^{2J}	1	$2^{0.2J}$	$2^{0.4J}$	$2^{0.6J}$	$2^{0.8J}$	2^{J}

Table 5.10: Condition numbers for the saddle point operator $\check{\mathbf{L}}$ of (4.2.69) using B-type wavelets and the standard $\mathbf{D}_1^{-1}$ preconditioner. Thus, from left to right, the ill-conditioning increases as the order of the operator is no longer matched properly by the preconditioner.

J	$\#\Delta_J$	$\kappa_2(\mathbf{S}_{\Phi_j})$	$\kappa_2(\mathbf{S}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{S}_{\mathrm{B}})$	$\kappa_2(\mathbf{S}_{\text{B-SVD}})$	$\kappa_2(\mathbf{S}_{\mathrm{P}})$
3	9	1.9674e+01	1.9674e+01	1.9674e+01	1.6209e+01	
4	17	4.0547e+01	1.8558e+01	2.8748e+01	1.6293e+01	3.6460e+01
5	33	8.3098e+01	1.7494e+01	2.8068e+01	1.7816e+01	3.0990e+01
6	65	1.6852e+02	1.7089e+01	2.8157e+01	2.0635e+01	2.9275e+01
7	129	3.3953e+02	1.6945e+01	2.8179e+01	2.2207e+01	2.8857e+01
8	257	6.8164e+02	1.6930e+01	2.8186e+01	2.2990e+01	2.8673e+01
9	513	1.3659e+03	1.6981e+01	2.8187e+01	2.3476e+01	2.8647e+01
10	1025	2.7342e+03	1.7054e+01	2.8188e+01	2.3949e+01	2.8632e+01
$\sim$	2^{J}	2^{J}	1	1	1	1

Table 5.11: Condition numbers for the **Schur complement** $\mathbf{S}_J = \mathbf{B}_J\mathbf{A}_J^{-1}\mathbf{B}_J^T$ (using the 2D stiffness matrix) with $\mathbf{D}_1^{-1}$ of (2.2.14) for preconditioning.

J	$\#\Delta_J$	$\kappa_2(\mathbf{S}_{\mathrm{DKU}})$	$\kappa_2(\mathbf{S}_{\mathrm{B}})$	$\kappa_2(\mathbf{S}_{\text{B-Orth}})$	$\kappa_2(\mathbf{S}_{\mathrm{P}})$	$\kappa_2(\mathbf{S}_{\text{P-Orth}})$
3	9	2.0715e+01	2.0715e+01	3.1626e+00		
4	17	1.9365e+01	2.1241e+01	7.0851e+00	3.5125e+01	1.5103e+00
5	33	1.8381e+01	1.9475e+01	8.7944e+00	3.0029e+01	2.1803e+00
6	65	1.8030e+01	1.8906e+01	1.0140e+01	2.8328e+01	2.7710e+00
7	129	1.7898e+01	1.8698e+01	1.0873e+01	2.7614e+01	3.2519e+00
8	257	1.7859e+01	1.8700e+01	1.1246e+01	2.7220e+01	3.5551e+00
9	513	1.7848e+01	1.8701e+01	1.1507e+01	2.7109e+01	3.7321e+00
10	1025	1.7844e+01	1.8701e+01	1.1740e+01	2.7051e+01	3.8319e+00
$\sim$	2^{J}	1	1	1	1	1

Table 5.12: Condition numbers for the **Schur complement** $\mathbf{S}_J = \mathbf{B}_J\mathbf{A}_J^{-1}\mathbf{B}_J^T$ (using the 2D stiffness matrix) with the $\mathbf{D}_a^{-1}$ and the $\mathbf{D}_{\{O,a\}}^{-1}$ preconditioner for the cases that include the orthogonal basis transformation.

	CG : B with $\mathbf{D}_1^{-1}$		
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J
4	5.6487e−04	4.7422e−03	45
5	3.1043e−04	3.8214e−03	113
6	1.4738e−04	2.0646e−03	325
7	7.5883e−05	4.4920e−04	523
8	3.7548e−05	3.8146e−04	243
9	1.9036e−05	2.4909e−04	371
10	9.5618e−06	3.9558e−05	789
11	5.9958e−06	2.8246e−05	360
12	2.4233e−06	2.7455e−05	161

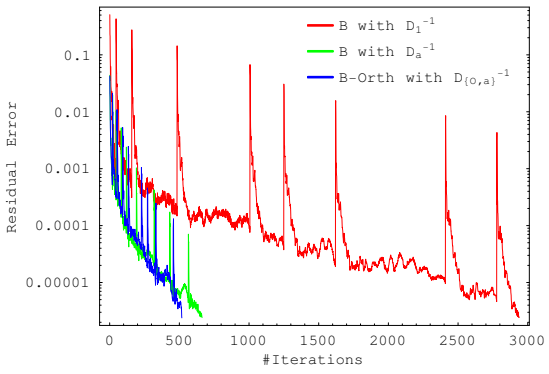
	CG : P with $\mathbf{D}_1^{-1}$		
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J
5	2.8130e−04	3.0671e−03	12
6	1.5476e−04	2.9036e−03	21
7	7.4634e−05	2.7491e−03	38
8	3.7297e−05	2.3370e−03	127
9	1.7281e−05	2.3363e−03	22
10	8.4102e−06	2.3336e−03	43
11	4.8034e−06	2.3331e−03	25
12	2.4343e−06	1.1038e−05	538

	CG : B with $\mathbf{D}_a^{-1}$		
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J
4	4.3002e−04	1.1500e−02	20
5	2.8184e−04	1.0260e−02	26
6	1.3954e−04	8.3695e−03	36
7	7.3407e−05	7.6103e−03	37
8	3.7124e−05	5.7727e−03	72
9	1.9496e−05	2.3433e−03	127
10	9.7468e−06	1.4911e−03	112
11	4.4289e−06	6.1467e−04	134
12	2.4160e−06	4.6495e−04	99

	CG : P with $\mathbf{D}_a^{-1}$		
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J
4	2.4971e−04	5.0744e−03	5
5	1.3546e−04	4.8952e−03	8
6	6.4746e−05	4.7220e−03	11
7	3.8581e−05	4.5031e−03	19
8	1.7681e−05	4.2817e−03	27
9	9.1325e−06	4.2256e−03	23
10	4.0448e−06	3.7678e−03	63
11	2.2040e−06	3.7676e−03	4

	CG : B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$		
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J
4	5.1968e−04	8.8956e−03	19
5	2.9849e−04	6.6252e−03	33
6	1.3916e−04	5.4242e−03	40
7	7.0229e−05	4.5182e−03	41
8	3.7944e−05	1.8353e−03	94
9	1.8049e−05	1.7032e−03	43
10	8.9740e−06	1.4470e−03	59
11	4.6834e−06	5.3712e−04	126
12	2.3927e−06	4.7899e−04	63

	CG : P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$		
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J
5	3.0752e−04	1.1555e−03	6
6	1.3496e−04	4.8133e−04	9
7	7.4880e−05	2.8920e−04	10
8	3.1705e−05	1.2468e−04	13
9	1.9246e−05	7.5909e−05	13
10	8.9419e−06	4.1422e−05	15
11	4.6677e−06	2.5221e−05	16
12	1.9952e−06	1.4810e−05	18



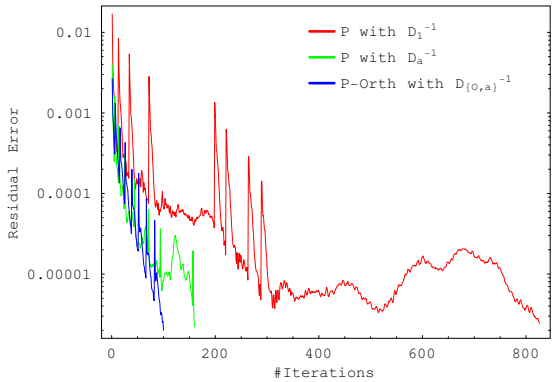


Table 5.13: Nested Iteration with conjugate gradient solver in 2D. We show the CG iteration statistics for different preconditioners applied for the B- and P-type wavelets. The plots show the norm of the residual in the CG scheme plotted against the overall number of iterations.

	Uzawa : B with $\mathbf{D}_1^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
4	2.0296e−04	1.6149e−03	1	43
5	1.8834e−04	4.3111e−04	1	83
6	1.5504e−04	1.8866e−04	1	107
7	7.4934e−05	9.4536e−05	2	99
8	3.7281e−05	4.8934e−05	4	94.5
9	1.3394e−05	1.8290e−05	4	104.2
10	6.5711e−06	9.1808e−06	2	136.5
11	4.3812e−06	7.6332e−06	2	54.5
12	2.2016e−06	3.8892e−06	4	125.5

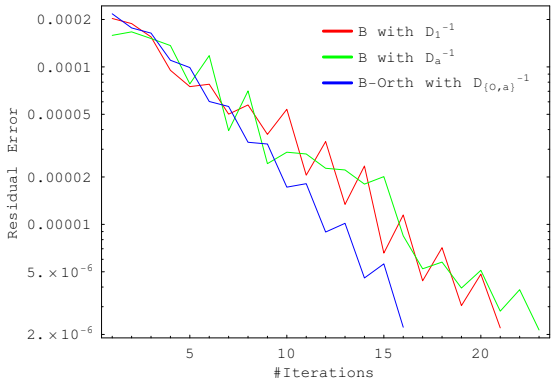
	UzawaCD : B with $\mathbf{D}_1^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
4	2.0261e−04	1.6178e−03	1	59
5	1.8866e−04	4.3311e−04	1	83
6	1.5363e−04	1.8619e−04	1	107
7	7.3606e−05	9.3540e−05	2	96.5
8	3.3588e−05	6.8745e−05	1	147
9	6.8239e−06	1.1969e−05	3	108.3
10	4.9550e−06	7.5775e−06	1	184
11	4.1588e−06	5.9895e−06	2	138
12	1.9515e−06	3.3798e−06	1	215

	Uzawa : B with $\mathbf{D}_a^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
4	1.5858e−04	2.9582e−03	1	39
5	1.6689e−04	9.0077e−04	1	44
6	1.5146e−04	3.4334e−04	1	56
7	7.8024e−05	1.7061e−04	2	42
8	2.4297e−05	9.0301e−05	4	48
9	1.8020e−05	5.0470e−05	5	45
10	8.4589e−06	2.7610e−05	2	63
11	3.9417e−06	9.4177e−06	3	57.3
12	2.1349e−06	5.6642e−06	4	54.2

	UzawaCD : B with $\mathbf{D}_a^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
4	1.6290e−04	2.9763e−03	1	39
5	1.6649e−04	8.9899e−04	1	51
6	1.5115e−04	3.4439e−04	1	56
7	2.6663e−05	1.0652e−04	3	46.6
8	2.6715e−05	7.9748e−05	1	73
9	1.4532e−05	5.5952e−05	2	56
10	8.9953e−06	2.6454e−05	1	83
11	1.9569e−06	1.2782e−05	3	55.3
12	1.1004e−06	4.6270e−06	1	99

	Uzawa : B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
4	2.1736e−04	2.0109e−03	1	17
5	1.7676e−04	7.5719e−04	1	23
6	1.1019e−04	3.1186e−04	2	25.5
7	6.0302e−05	1.2500e−04	2	30
8	3.3219e−05	7.5304e−05	2	32.5
9	1.7234e−05	3.4912e−05	2	35.5
10	8.9406e−06	1.8629e−05	2	39.5
11	4.5679e−06	9.8691e−06	2	43
12	2.2233e−06	5.4892e−06	2	46

	UzawaCD : B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
4	2.2864e−04	1.9692e−03	1	17
5	1.8729e−04	7.2636e−04	1	23
6	7.4811e−05	2.9628e−04	2	24.5
7	3.3135e−05	1.0026e−04	1	33
8	2.7375e−05	7.4556e−05	1	37
9	8.9554e−06	3.3004e−05	2	34
10	2.9872e−06	1.7632e−05	1	49
11	3.3246e−06	1.0837e−05	1	55
12	1.4526e−06	2.8286e−06	1	61



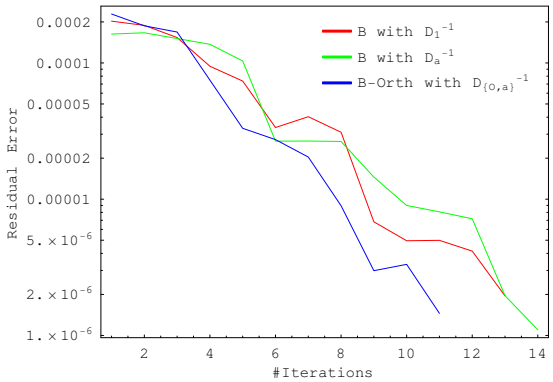


Table 5.14: Nested Iteration for B-type wavelets and the two variants of the Uzawa algorithm in 2D. The number of outer iterations k_J depends on $\kappa_2(\mathbf{S}_J)$; the average number of inner CG steps (#CG-It) is determined by $\kappa_2(\mathbf{A}_J)$. The plots show the norm of the residual against the steps of the outer iteration.

	Uzawa : P with $\mathbf{D}_1^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
5	3.9524e−05	7.4145e−04	1	44
6	2.3067e−05	2.3722e−04	1	55
7	1.5911e−05	8.0410e−05	1	68
8	2.1835e−05	9.4294e−05	1	80
9	7.7497e−06	3.5843e−05	1	89
10	8.9384e−06	1.9734e−05	1	96
11	3.3437e−06	1.4435e−05	3	70
12	1.8742e−06	2.7474e−06	3	75

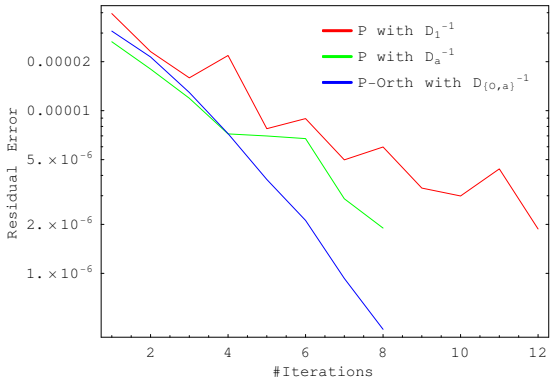
	UzawaCD : P with $\mathbf{D}_1^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
5	3.9189e−05	7.4086e−04	1	44
6	2.2687e−05	2.3758e−04	1	55
7	1.5965e−05	8.0053e−05	1	68
8	2.2570e−05	9.7303e−05	1	80
9	7.7541e−06	2.2282e−05	1	89
10	2.5141e−06	2.7462e−05	2	75.5
11	2.1194e−06	6.3392e−06	1	102
12	1.6905e−06	2.9487e−06	1	108

	Uzawa : P with $\mathbf{D}_a^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
5	2.6525e−05	5.2482e−03	1	27
6	1.8015e−05	3.7965e−04	1	25
7	1.1925e−05	2.2121e−04	1	28
8	7.1981e−06	5.4815e−05	1	39
9	6.9900e−06	6.2382e−05	1	42
10	6.7340e−06	2.8834e−05	1	48
11	2.8741e−06	2.1869e−05	1	47
12	1.8977e−06	1.0888e−05	1	49

	UzawaCD : P with $\mathbf{D}_a^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
5	2.6706e−05	9.7772e−04	1	27
6	1.7972e−05	3.7986e−04	1	32
7	1.1813e−05	2.0452e−04	1	36
8	7.0957e−06	5.3354e−05	1	39
9	6.9222e−06	4.7468e−05	1	42
10	4.5773e−06	5.2134e−05	1	48
11	1.8169e−06	3.1928e−06	1	47
12	7.861e−07	2.5546e−06	2	48

	Uzawa : P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
5	3.0839e−05	4.2243e−04	1	9
6	2.1386e−05	1.5717e−04	1	12
7	1.2937e−05	5.7105e−05	1	14
8	7.2229e−06	2.1973e−05	1	17
9	3.7761e−06	8.4314e−06	1	20
10	2.1157e−06	3.4671e−06	1	22
11	9.3078e−07	8.0748e−07	1	25
12	4.5284e−07	5.1167e−07	1	27

	UzawaCD : P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$			
J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{y}^*-\mathbf{y}_J^{(k_J)}\|$	k_J	$\frac{\#\text{CG-It}}{k_J}$
5	3.1228e−05	4.8165e−04	1	9
6	2.7074e−05	1.6185e−04	1	12
7	1.1496e−05	5.6115e−05	1	14
8	6.0063e−06	2.1205e−05	1	17
9	3.3574e−06	8.2464e−06	1	20
10	1.3615e−06	3.0685e−06	1	22
11	6.4769e−07	1.2522e−06	1	25
12	2.5721e−07	3.7431e−07	1	27



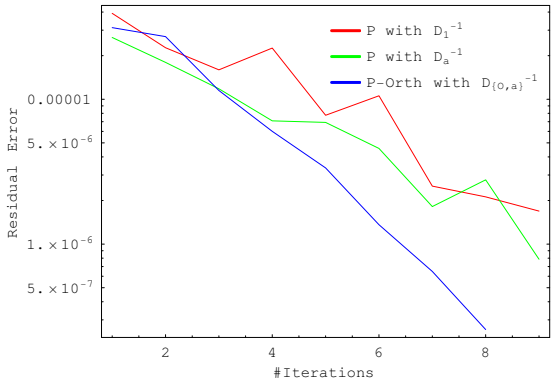


Table 5.15: Nested Iteration for P-type wavelets and the two variants of the Uzawa algorithm. The setup as otherwise identical to Table 5.14. Here the nested iteration starts with level 5 because $j_0 = 4$, on this level the QR decomposition was utilized.

5.2 Control Problem

The following numerical experiments will be concerned with the control problem described in Section 4.3. We will always use the problem setup of Problem 4.20 in weak form and specify the following parameters to achieve a well-defined problem statement:

- the right hand sides $f \in (H^1(\Omega))'$, $y_{\Gamma_Y} \in H^{1/2}(\Gamma_Y)$,
- the boundaries Γ_Y and Γ, fixed by the trace operators B and T,
- the magnitude of the Sobolev norms in the functional controlled by the parameters s and t,
- the weighting parameter ω in the functional (4.3.1),
- the factor $a_0 \geq 0$ of the identity part of the elliptic partial differential equation; the coefficients $\mathbf{a}(x) = (a_{i,j}(x))_{i,j} := \delta_{i,j}$ of the elliptic partial differential equation will not be varied.

We are not only interested in the state y and the control u for a given set of parameters but also in the impact of the different parameters onto the performance of the numerical schemes.

Remark 5.2

(i) Because of the results of Section 5.1.2 and Section 5.1.3, we focus on the cases of B-type wavelets with the $\mathbf{D}_a$ preconditioner and P-Orth-type wavelets with the $\mathbf{D}_{\{O,a\}}$ preconditioner.

(ii) We focus entirely on uniform refinements in our numerical experiments. It is possible with my software to choose different levels of resolution on the boundaries and the domain using these full grids, in accordance with the stability condition of Section 3.1.3.

(iii) The execution times given at any instance were measured on a `Dell Precision(TM) 670` system featuring two `Intel(R) Xeon(TM) CPU 3.20GHz` processors and `2 GB RAM`. Although this machine has 64-bit extensions, these were not used because they did not yield any improvements. The operating system used was `Linux 2.6`, the distribution `Fedora Core 3`.

(iv) All numerical calculations were executed with `double precision`.

5.2.1 Model Problems

We will now describe the model problems we will be referring to in the next sections.

(P1) For the dimension $n = 2$, using the right hand sides

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{2}(1 + 4\pi^2 + (1 + 5\pi^2)\cos(\pi x_1))\cos(2\pi x_2), \\ y_{\Gamma_Y}(x_2) &= \cos(2\pi x_2), \end{aligned}$$

leads to the exact solution of the state

$$y^*(x_1, x_2) = \frac{1}{2}\cos(2\pi x_2)(1 + \cos(\pi x_1)) \ . \tag{5.2.1}$$

The exact observation and control take on the forms

$$y^*_{\Gamma_Y}(x_1, x_2) = y|_{\Gamma_Y}(x_1, x_2) = \cos(2\pi x_2), \qquad u^*(x_2) = y|_{\Gamma}(x_1, x_2) \equiv 0 \ . \tag{5.2.2}$$

This means that the functional (4.3.1) can be minimized optimally since the norms can attain the value zero exactly and our calculations should yield the exact solutions stated above. This problem setup will be used to verify the validity of our theoretical problem formulation with wavelets and to show the convergence rates of the algorithms set forth in Section 4.4, see Figure 5.6 for a graphic presentation of the right hand side functions and Figure 5.5 for the solution.

(P2) The second problem setup shall be the obligatory

$$f(x_1, x_2) \equiv 1, \qquad y_{\Gamma_Y}(x_2) \equiv 1 \ .$$

These functions can be used easily in any problem dimension $n \geq 2$. It is compliant with the problem statement as the smoothness of the observation satisfies $y_{\Gamma_Y} \in H^{1/2}(\Gamma_Y)$ and the homogeneous **Neumann boundary conditions** are satisfied. The analytical solution is not known, though.

(P3) Another example for the dimension $n = 2$ are the functions

$$\begin{aligned} f(x_1, x_2) &= \left(1 + e^{-10|x_1 - \frac{1}{2}|}\right)\left(1 + e^{-10|x_2 - \frac{1}{2}|}\right), \\ y_{\Gamma_Y}(x_2) &= 16\, x_2^2\, (x_2 - 1)^2 \ . \end{aligned}$$

The right hand side f incorporates a cusp, see Figure 5.7 for the plots of these functions.

(P4) The last example for $n = 2$ shall be

$$\begin{aligned} f(x_1, x_2) &= 1 - x_1, \\ y_{\Gamma_Y}(x_2) &= \begin{cases} 1/2 & 0 \leq x_2 < 1/2 \\ 1 & 1/2 \leq x_2 \leq 1 \end{cases} \ . \end{aligned}$$

Since discontinuous piecewise constant functions are not in $H^{1/2}$, we cannot expect $y|_{\Gamma_Y}$ to attain this y_{Γ_Y} exactly. Also, the solution to the boundary value problem in Section 3.2.3 with $f(x_1, x_2) = 1 - x_1$ is not $y(x_2) = 1 - x_2$ as this solution does not satisfy the homogeneous Neumann boundary conditions on the observation boundary Γ_Y, see Figure 5.8 for the plots of these functions.

We have stressed before that our problem formulation Problem 4.4 is an **optimal control problem**, i.e., a unique solution (y^*, u^*) exists if the right hand side data is compliant with the problem formulation. Assuming this, we will terminate our solution methods when the residual error is matched to a constant multiple of the discretization error $h_J = 2^{-J}$. Since all involved operators are uniformly conditioned ℓ_2-automorphism, we can expect the state $y_J = \mathbf{y}_J^T \Psi_\Omega^1 \in H^1(\Omega)$ and the control $u_J = \mathbf{u}_J^T \Psi_\Gamma^{1/2} \in H^{1/2}(\Gamma)$ also to be determined up to an error of h_J with respect to the exact solution, i.e., we can expect

$$\|\mathbf{y}_J - \mathbf{y}^*\| \lesssim h_J \quad \text{and} \quad \|\mathbf{u}_J - \mathbf{u}^*\| \lesssim h_J, \qquad J \to \infty \ .$$

Of course this is an asymptotic result and might not be observed on low levels $j \approx j_0$ if the condition numbers of the involved operators are far greater than 1.

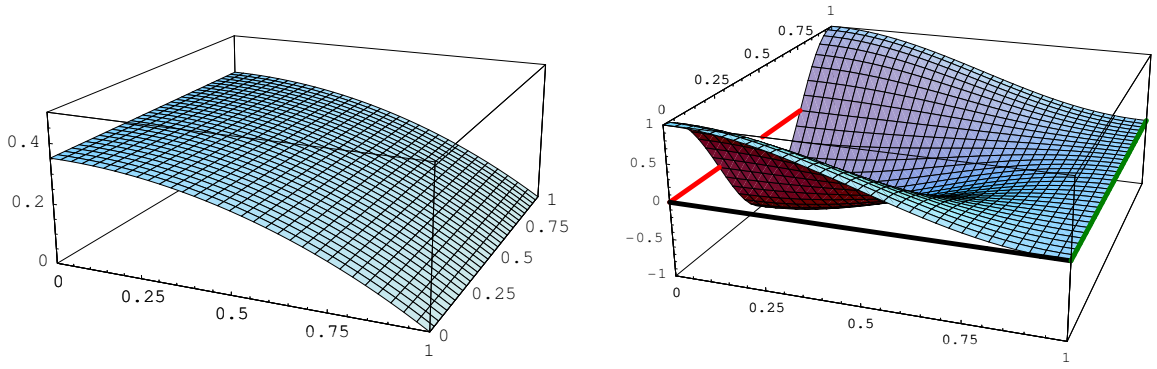


Figure 5.5: Left: Computed solution of the boundary value problem for $f \equiv 1$ on level $J = 5$ and the trace operator for the facing right edge and $a_0 = 1$.
Right: Plot of the exact solution y^* for (P1) on level $J = 5$.

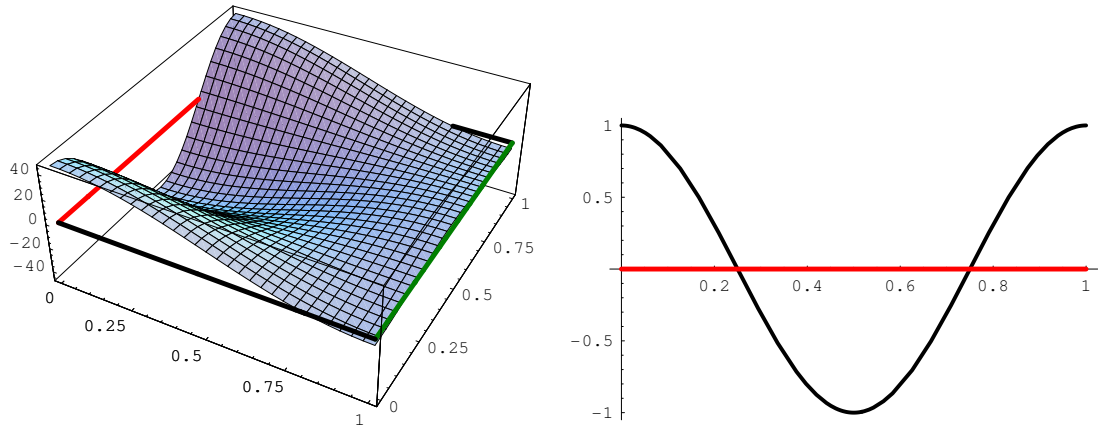


Figure 5.6: Plot of the right hand side $f(x_1, x_2) = \frac{1}{2}(1 + 4\pi^2 + (1 + 5\pi^2)\cos(\pi x_1))\cos(2\pi x_2)$ and $y_{\Gamma_Y}(x_2) = \cos(2\pi x_2)$ used in problem (P1). The far edge in the left picture is the observation boundary, indicated by the red color. The facing edge is the control boundary and therefore plotted in green color.

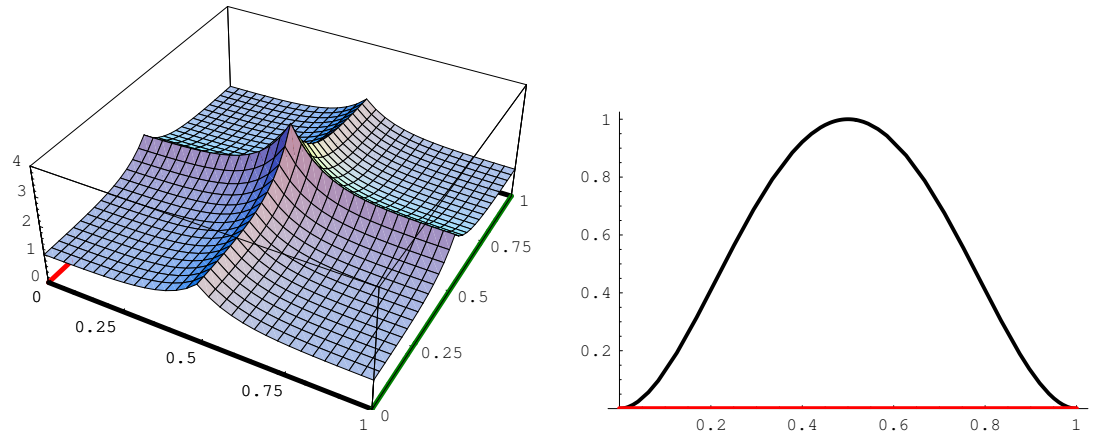


Figure 5.7: Plot of the right hand side $f(x_1, x_2) = \left(1 + e^{-10|x_1 - \frac{1}{2}|}\right)\left(1 + e^{-10|x_2 - \frac{1}{2}|}\right)$ and $y_{\Gamma_Y}(x_2) = 16x_2^4 - 32x_2^3 + 16x_2^2$ of problem (P3) .

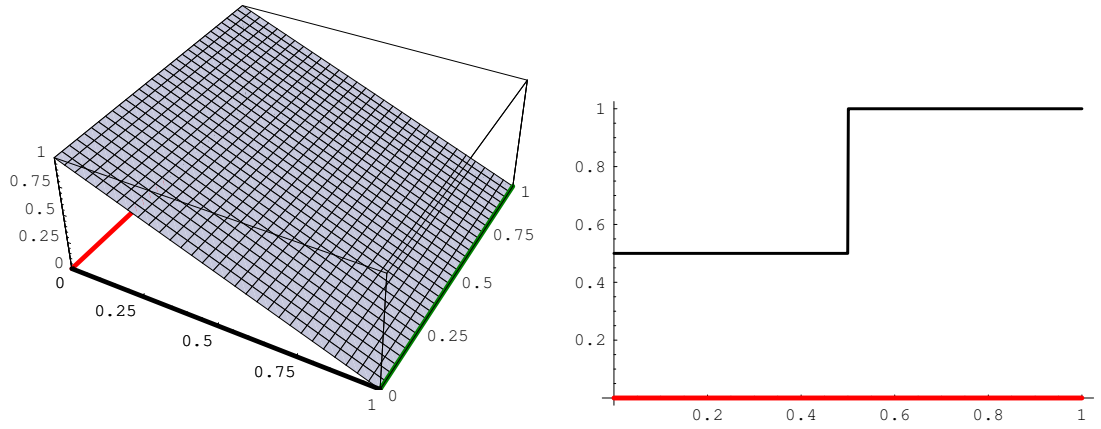


Figure 5.8: Plot of the right hand side $f(x_1, x_2) = 1 - x_1$ and $y_{\Gamma_Y}(x_2) = \frac{1}{2}\chi_{[0,1/2)} + \chi_{[1/2,1]}$ of setup (P4) .

5.2.2 An Exact Solution – Convergence Rates

We are now going to investigate model problem (P1), for which an exact solution exists. For this part we use the natural norm equivalences, i.e., $R_{H^s(\Gamma_Y)} = R_{H^t(\Gamma)} = \mathbf{I}$, and $\omega = 1$. This means we have by (4.2.49) the relation $\mathbf{u} = \boldsymbol{\mu}$ and the Inexact Gradient algorithm will stop when this holds since we then have reached the unique minimization point of the functional by identity (4.2.52).

We will show that our wavelet discretization yields the correct solution and the algorithms of Section 4.4.1 and Section 4.4.2 calculate the solution up to discretization error accuracy in linear time with respect to the number of unknowns $\#\mathcal{I}_{\mathcal{N},J_{\mathcal{N}}}$. This is the case since the number of iterations necessary, of the outer and possibly the inner solvers, are all uniformly bounded.

Remark 5.3 *As Figure 5.9 shows, neither incarnation of the matrix $\mathbf{N}'_4$ from (4.4.5) is positive definite and this holds for all levels. Since the matrix $\mathbf{N}_4$ is not symmetric, its eigenvalues are not even real valued. Thus, the All-In-One Solver described in Section 4.4.1 has to be applied in the form of Corollary 4.22 and (4.4.4) holds for the operator $\mathbf{P}_J = \mathbf{N}_J^T\mathbf{N}_J$.*

The iteration statistics for the **All-In-One Solver** can be found in Table 5.16 and Table 5.17. These show a nested iteration execution starting with a QR decomposition on the lowest level j_0 and then using the CG Solver for $j > j_0$. The starting vector for these cases was $\mathbf{y}_{j_0} = \mathbf{1}$ in wavelet coordinates. The stopping criterion for the residual error is $\frac{1}{100}2^{-J}$. The total values of the iteration numbers k_J are quite high because of identity (4.4.4) and the absolute values of the condition numbers of operator $\mathbf{N}_J$, listed in Table 5.20 and Table 5.21. The last column shows the execution time needed on this level divided by the number of CG steps. Neglecting all other numeric operations during the CG step, this is roughly the time needed for one multiplication of the matrix $\mathbf{P}_J$ from (4.4.2) or twice the time needed for $\mathbf{N}_J$ on the respective levels. This number grows proportional to the number of unknowns which grows with 2^{nJ} for uniform refinements in n dimensions.

The results for the **Inexact Gradient** algorithm are listed in the Tables 5.18 and 5.19. In every step of the nested iteration the residual $\mathbf{r}_J^{(k_J)}$ is computed to an accuracy of $\delta_J := \frac{1}{100}2^{-J}$ and the internal solvers for the primal (4.4.7) and adjoint (4.4.8) systems compute the residual error to below $\frac{1}{4}\delta_J$. The value of the step size parameter of the Inexact Gradient algorithm was $\rho = 0.75$ in all these computations. Here k_J denotes the number of **outer iterations**, i.e., the number of iterations of the **Inexact Gradient algorithm**. The mean number of steps the inner solution algorithm needed to compute the solutions of the **primal** (4.4.7) and **adjoint** (4.4.8) saddle point problems is given in the columns titled $\frac{\#\text{P-It}}{k_J}$ and $\frac{\#\text{A-It}}{k_J}$, respectively. The execution times given in the last column are the mean times which the inexact gradient spent on the respective level, i.e., the time needed for one step in the gradient scheme. This way the proportionality of the complexity to the number of unknowns can be observed more easily.

For both solution strategies it holds that the uniform boundedness of the number of iterations and overall complexity might not be observed on low levels $j \approx j_0$ if the (qualitatively uniformly bounded) spectral condition numbers of the involved operators are quantitatively large away from 1. This can be especially easily observed for the B-type MRA.

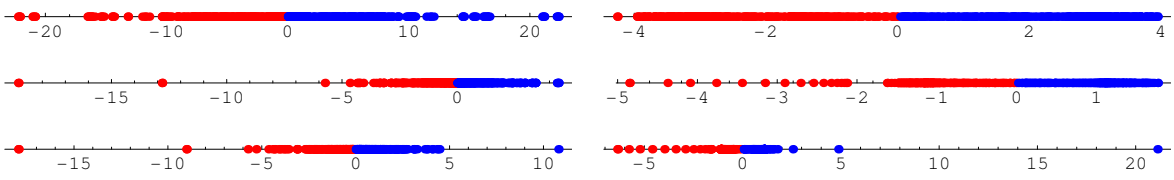


Figure 5.9: Spectrum of $\mathbf{N}'_4$ with Riesz operator $\widehat{\mathbf{R}}_{H^s}$. Negative eigenvalues are marked in red color and positive ones in blue color. The left column shows the spectrum of the matrix $\mathbf{N}'_4$ for the B wavelets and the right column for the P wavelets. The diagonal preconditioner is chosen in every row as follows: the standard $\mathbf{D}_1$ in the first row, the $\mathbf{D}_a$ preconditioner in the second and the $\mathbf{D}_{\{O,a\}}$ preconditioner with the orthogonal basis transformation in the third row.

The linear convergence rates for the state and control, i.e.,

$$\|\mathbf{r}_J^{(k_J)}\| \sim \|\mathbf{y}_J - \mathbf{y}^*\|, \|\mathbf{u}_J - \mathbf{u}^*\| \sim \mathcal{O}(h_J) \tag{5.2.3}$$

are, nevertheless, recognizable in all examples. The functional value $\mathbf{J}(\mathbf{y}, \mathbf{u})$ clearly exhibits superlinear convergence properties here. This is not trivially deducible by definition (4.2.37), since the functional is not linear in $\mathbf{u}$, not even in the condensed form (4.2.40). However, within the current setting, we can conclude the following result.

Proposition 5.4 *If the optimal solution $y^* = (\mathbf{y}^*)^T \Psi_\Omega^1 \in H^1(\Omega)$ satisfies the boundary value problem (4.2.6) with $By^* = u = 0$ and additionally $Ty^* = y_{\Gamma_Y}$, then the value of the functional converges quadratically, i.e.,*

$$|\mathbf{J}(\mathbf{y}_J, \mathbf{u}_J) - \mathbf{J}(\mathbf{y}^*, \mathbf{u}^*)| \lesssim h_J^2 \ .$$

Proof: We show the quadratic convergence for each term of the functional separately. Because of the last assumption, we can conclude $u^* \equiv 0$ and thus follows for the regularization term

$$\|\mathbf{R}_{H^t,J}^{1/2} \mathbf{u}_J\|^2 = \|\mathbf{R}_{H^t,J}^{1/2} (\mathbf{u}_J - \mathbf{u}^*)\|^2 \sim \|\mathbf{u}_J - \mathbf{u}^*\|^2 \lesssim h_J^2$$

where we used the Riesz operator property (2.2.39). The data fitting term can be estimated by (1.2.25) as

$$\begin{aligned} \|\mathbf{R}_{H^s,J}^{1/2} (\mathbf{T}_J \mathbf{y}_J - \mathbf{y}_{\Gamma_Y})\|^2 &\sim \|\mathbf{T}_J \mathbf{y}_J - \mathbf{y}_{\Gamma_Y}\|^2 \\ &\sim \|\mathbf{T}_J \mathbf{y}_J - \mathbf{T}_J \mathbf{y}^*\|^2 \\ &= \|\mathbf{T}_J (\mathbf{y}_J - \mathbf{y}^*)\|^2 \\ &\lesssim \|\mathbf{y}_J - \mathbf{y}^*\|^2 \\ &\lesssim h_J^2 \ . \end{aligned}$$

Thus follows the assertion by noting $\mathbf{J}(\mathbf{y}^*, \mathbf{u}^*) = 0$. ∎

Of course we calculate the value of the functional in our experiments with respect to the finite approximation $\mathbf{y}_{\Gamma_Y,J}$ of $\mathbf{y}_{\Gamma_Y}$, but the error induced here is also of order $\mathcal{O}(h_J)$, which is yields the same convergence rate of order 2.

In summary, the Inexact Gradient algorithm clearly outperforms the All-In-One Solver. Not only determines it the solution $(\mathbf{y}_J, \mathbf{u}_J)$ in a fraction of the time, it also calculates the solution on any level with a higher accuracy. The step size parameter ρ determines the convergence speed of the outer iteration and must be picked carefully. The main cause of long execution times are, nevertheless, the inner solvers for the saddle point problems. The Uzawa algorithms benefit from the extremely low condition numbers of the **Schur complement** $\mathbf{S}_J$, see Table 5.11 and 5.12, and are thus preferred to the CG Solver here.

We are going to discuss more of the numerical properties of the **Inexact Gradient** algorithm in Section 5.3. Next, we look deeper into the matter of **Riesz operators** and their impact on our problem formulation.

J	$\|\mathbf{r}_J^{(k_J)}\|$	$\mathbf{J}(\mathbf{y}_J^{(k_J)}, \mathbf{u}_J^{(k_J)})$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	$\|\mathbf{u}^* - \mathbf{u}_J^{(k_J)}\|$	k_J	$\frac{\text{Time}}{k_J}$
3	–	5.1292e−03	5.4256e+00	0.0000e+00	–	0.04s
4	5.8956e−04	1.5192e−03	1.9636e−01	9.8632e−03	149	<0.01s
5	2.7714e−04	1.3893e−04	6.2264e−02	2.0845e−03	389	<0.01s
6	1.4774e−04	1.0128e−04	4.2498e−02	2.3818e−03	145	0.02s
7	7.8121e−05	1.8623e−05	1.7267e−02	2.6103e−04	469	0.07s
8	3.7391e−05	1.6358e−05	1.6285e−02	2.4649e−04	177	0.28s
9	1.9497e−05	9.3391e−07	3.7424e−03	2.9933e−04	809	1.11s
10	9.7201e−06	5.7878e−07	2.9215e−03	2.3759e−04	332	4.27s
11	4.8070e−06	1.2247e−07	1.3603e−03	1.1507e−05	625	16.50s
$\sim$	$\leq \frac{1}{100}2^{-J}$	2^{-2J}	2^{-J}	2^{-J}	1	2^{2J}

Total Execution Time: 3h 32m 32.81s

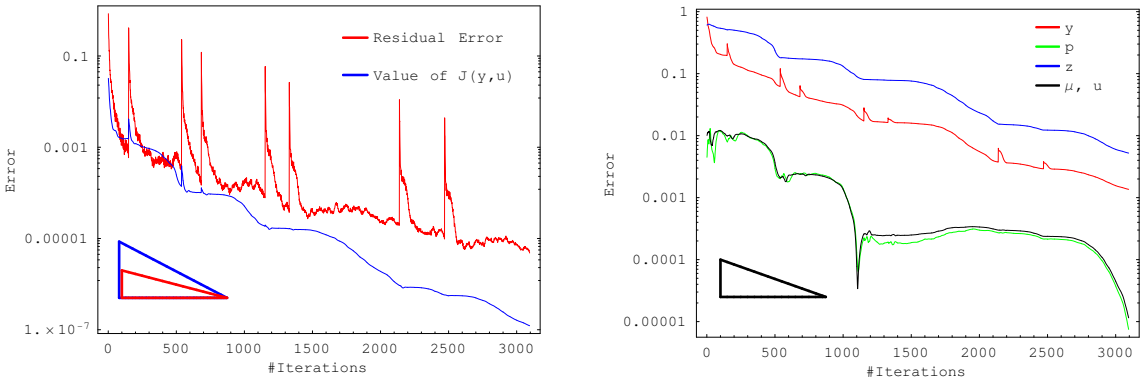


Table 5.16: Nested iteration history for the All-In-One solver and B-type MRA with $\mathbf{D}_a$ preconditioner. On level $j_0 = 3$ we used the QR decomposition so a residual error is not needed. The mean number of CG iterations is 386.75 and we used this value for the abscissa of the triangle in the iteration plots. In the left image, the blue triangle shows the slope of the function which declines exponentially $\sim 2^{-2x}$ with every 386.75 steps. The red triangle and the black triangle on the right both show the slope for a decline proportional to $\sim 2^{-x}$.

J	$\|\mathbf{r}_J^{(k_J)}\|$	$\mathbf{J}(\mathbf{y}_J^{(k_J)}, \mathbf{u}_J^{(k_J)})$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	$\|\mathbf{u}^* - \mathbf{u}_J^{(k_J)}\|$	k_J	$\frac{\text{Time}}{k_J}$
4	–	2.2217e−04	2.8043e+00	0.0000e+00	–	4.26s
5	2.7326e−04	1.4117e−05	2.2886e−02	6.8601e−05	24	<0.01s
6	1.3271e−04	1.3068e−06	6.6313e−03	1.7387e−05	29	0.02s
7	7.2257e−05	2.1405e−07	2.0559e−03	3.9113e−06	35	0.08s
8	3.3238e−05	7.2402e−08	7.6950e−04	9.5511e−07	39	0.28s
9	1.9461e−05	1.6449e−09	2.2335e−04	4.7287e−07	63	1.17s
10	9.7406e−06	6.4133e−10	8.5196e−05	9.9898e−08	42	4.58s
11	4.5715e−06	3.8173e−10	3.9305e−05	4.7749e−08	43	17.71s
$\sim$	$\leq \frac{1}{100} 2^{-J}$	2^{-2J}	2^{-J}	2^{-J}	1	2^{2J}

Total Execution Time: 17m 50.94s

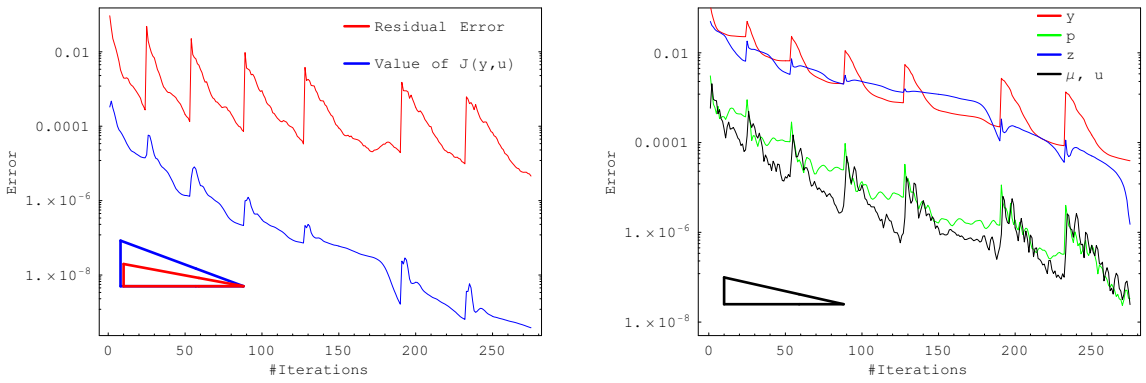
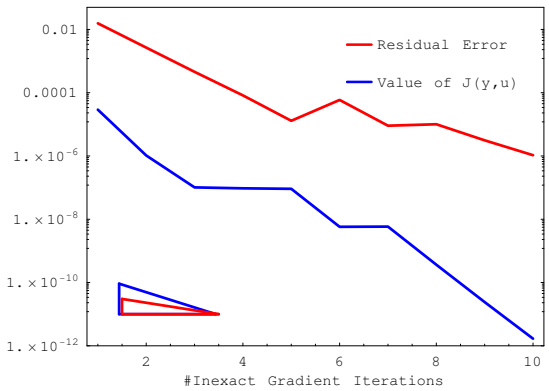


Table 5.17: Nested iteration history for the All-In-One solver and P-Orth-type MRA with the $\mathbf{D}_{\{O,a\}}$ preconditioner. On level $j_0 = 4$ we used the QR decomposition so a residual error is not calculated. The mean number of CG iterations is 39.28 and we used this value for the abscissa of the triangle in the iteration plots. In the left image, the blue triangle shows the slope of the function which declines exponentially $\sim 2^{-2x}$ with every 39.28 steps. The red triangle and the black triangle on the right both show the slope for a decline proportional to $\sim 2^{-x}$.

J	$\|\mathbf{r}_J^{(k_J)}\|$	$\mathbf{J}(\mathbf{y}_J^{(k_J)}, \mathbf{u}_J^{(k_J)})$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	$\|\mathbf{u}^* - \mathbf{u}_J^{(k_J)}\|$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	$\frac{\text{Time}}{k_J}$
3	1.1318e−04	5.1326e−03	5.4628e−01	1.0045e−02	3	–	–	0.18s
4	1.1318e−04	3.7867e−02	8.6126e−01	1.0045e−02	0	–	–	–
5	1.1318e−04	4.7120e−02	9.0306e−01	1.0045e−02	0	–	–	–
6	1.1318e−04	4.8889e−02	9.1217e−01	1.0045e−02	0	–	–	–
7	1.2946e−05	9.1714e−08	3.2439e−03	5.1238e−05	5	7	7.4	11.99s
8	9.0663e−06	5.8518e−09	9.6882e−04	1.0276e−05	2	3	3	20.26s
9	1.0067e−05	3.6600e−10	3.0640e−04	3.4016e−06	1	3	2	84.71s
10	3.1232e−06	2.4057e−11	9.7450e−05	1.1360e−06	1	4	3	334.21s
11	1.0593e−06	1.6810e−12	6.7432e−06	5.6649e−07	1	2	1	916.95s
$\sim$	$\leq \frac{1}{100}2^{-J}$	2^{-2J}	2^{-J}	2^{-J}	1	1	1	2^{2J}

Total Execution Time: 24m 3.15s



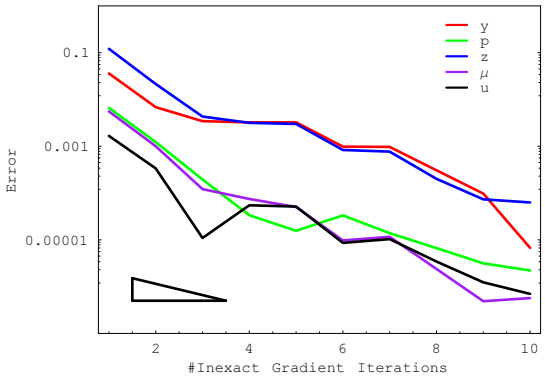
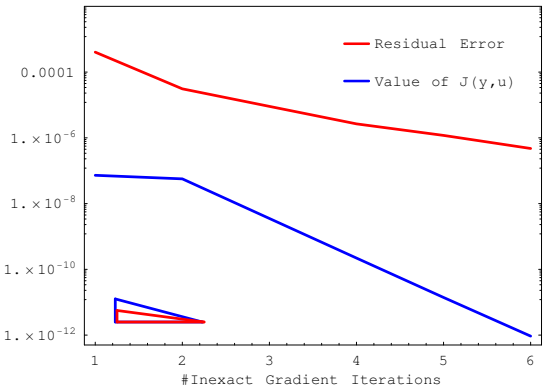


Table 5.18: Nested iteration history for the Inexact Gradient solver and B-type MRA with $\mathbf{D}_a$ preconditioner for problem setup (P1) . On level $j_0 = 3$ we used the QR decomposition for solving the primal and adjoint saddle point systems. The mean number of outer iterations is $10/8 = 1.25$ and we used this value for the abscissa of the triangles in the iteration plots. Thus, the mean value of UzawaCD iterations for the primal system is thus 5 and 4.9 for the adjoint system.

J	$\|\mathbf{r}_J^{(k_J)}\|$	$\mathbf{J}(\mathbf{y}_J^{(k_J)}, \mathbf{u}_J^{(k_J)})$	$\|\mathbf{y}^* - \mathbf{y}_J^{(k_J)}\|$	$\|\mathbf{u}^* - \mathbf{u}_J^{(k_J)}\|$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	$\frac{\text{Time}}{k_J}$
4	9.6557e−05	2.2227e−04	7.9564e−02	2.8949e−04	1	–	–	1.67s
5	9.6557e−05	2.9981e−03	2.9722e−01	2.8949e−04	0	–	–	–
6	9.6557e−05	3.5819e−03	3.2906e−01	2.8949e−04	0	–	–	–
7	3.0961e−05	5.6535e−08	1.9414e−03	8.9811e−06	2	2.5	1.5	1.17s
8	9.0566e−06	3.5273e−09	6.1788e−04	3.0162e−06	1	2	1	4.62s
9	2.6599e−06	2.2037e−10	2.0489e−04	1.1733e−06	1	1	1	16.93s
10	1.1861e−06	1.3961e−11	6.7546e−05	3.5632e−07	1	2	1	74.69s
11	4.7470e−07	9.3086e−13	2.5547e−06	1.3639e−07	1	1	1	258.74s
$\sim$	$\leq \frac{1}{100} 2^J$	2^{2J}	2^J	2^J	1	1	1	2^{2J}

Total Execution Time: 6m 3.82s



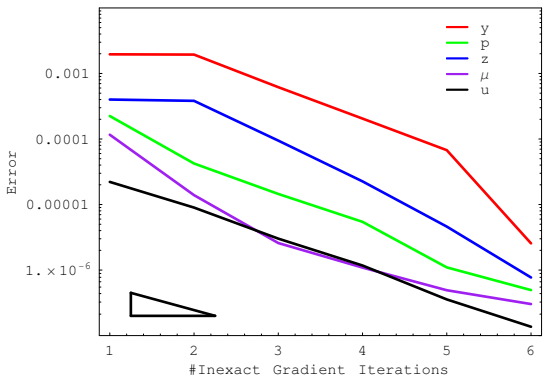


Table 5.19: Nested iteration history for the Inexact Gradient solver and P-Orth-type MRA with the $\mathbf{D}_{\{O,a\}}$ preconditioner. On level $j_0 = 4$ we used the QR decomposition for solving the primal and adjoint saddle point systems. The mean number of outer iterations is $6/7 \approx 0.85$ and we used this value for the abscissa of the triangles in the iteration plots. So the mean value of UzawaCD iterations for the primal system is thus 1.83 and 1.16 for the adjoint system. The asymptotic convergence of Order $\mathcal{O}(h_j)$ for all algorithm variables in clearly visible in the right plot.

5.2.3 Riesz Operators Examined

We now show why it is necessary to introduce Riesz operators in the first place into our problem formulation. Secondly, we show that sufficiently accurate Riesz operators for our control problem are given by scaled versions of the constructions $\widehat{\mathbf{R}}_{H^s}$ (2.2.44) and $\widetilde{\mathbf{R}}_{H^s}$ (2.2.53).

Numerical Impact of Riesz Operators

We now investigate the qualitative impact of Riesz operators onto our control problem formulation. It will be illustrative to study Figure 5.10. The first row shows a plot of the state $y = y(u)$ for problem setup (P2) without a Riesz operator and different diagonal matrices used for preconditioning. Although these plots show states $y(u)$, which are legitimate solutions within certain constant bounds to our problem, they are not as smooth as one would expect the solution to be with given right hand sides $f = y_{\Gamma_Y} \equiv 1$.
The second row shows the states y again for different preconditioning operators but using $\mathbf{R}_{H^s} \approx \mathbf{M}_{L_2}$. According to (2.2.49), this should be a good approximation for the exact Riesz operator $\mathbf{R}_{H^s}$. Yet the solution depends on the preconditioner, which should not be the case. To explain this, we have to distinguish between the diagonal preconditioner matrix $\mathbf{D} \in \{\mathbf{D}_a, \mathbf{D}_1, ...\}$ and the diagonal matrix $\widehat{\mathbf{D}}^{+2s} = \mathbf{D}_1^{+2s}$ used in definition of the Riesz operator (2.2.43):
Generally, preconditioning should never alter the computed solution, but only the rate at which this solution can be obtained. The matrix $\widehat{\mathbf{D}}^{+2s}$ is part of the model which determines how we measure the Sobolev norms by weighting the wavelet coefficients differently. Altering this matrix should imply a noticeable change in the computed solution.
It is easy to see that for a function $v = (\mathbf{v}^s)^T\Psi^s \in H^s$, e.g., with $\mathbf{v}^s := \mathbf{D}^{+s}\langle v, \widetilde{\Psi}\rangle$ and $\mathbf{v} := \langle v, \widetilde{\Psi}\rangle$, it holds

$$\begin{aligned}
\|\mathbf{v}^s\|_{\ell_2}^2 &= \mathbf{v}^T\mathbf{D}^{+s}\mathbf{D}^{+s}\mathbf{v}, \\
\|\mathbf{M}_{L_2}^{1/2}\mathbf{v}^s\|_{\ell_2}^2 &= \mathbf{v}^T\mathbf{D}^{+s}\mathbf{M}_{L_2}\mathbf{D}^{+s}\mathbf{v}, \\
\|\widehat{\mathbf{R}}_{H^s}^{1/2}\mathbf{v}^s\|_{\ell_2}^2 &= \mathbf{v}^T\mathbf{D}^{+s}\mathbf{D}^{-s}\widehat{\mathbf{D}}^{+s}\mathbf{M}_{L_2}\widehat{\mathbf{D}}^{+s}\mathbf{D}^{-s}\mathbf{D}^{-s}\mathbf{v} = \mathbf{v}^T\widehat{\mathbf{D}}^{+s}\mathbf{M}_{L_2}\widehat{\mathbf{D}}^{+s}\mathbf{v} \ .
\end{aligned}$$

Thus, the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ yields a result independent of the preconditioner $\mathbf{D}$ and is, therefore, preferably the choice for our investigations. The last row in Figure 5.10 shows smooth state functions independent of the preconditioner.

The choice of a Riesz operator can have quite a significant influence on the spectral condition of the operator $\mathbf{N}_J$, as can be seen in Table 5.22. A general increase in condition numbers on every level is recognizable for more involved Riesz operator constructions independent of the utilized preconditioner. The effect of the Riesz operator onto the Inexact Gradient algorithm for problem setup (P1) is examined in Tables 5.23 and 5.24. The results show no negative impact on the efficiency of the Inexact Gradient algorithm. The iteration numbers vary more by their distribution over the levels of the nested iteration and not much in their magnitude. The overall cost of the Inexact Gradient algorithm stays the same and is thus much better suited for using different Riesz operator then the All-In-One solver $\mathbf{N}_J$.

Normalization with Respect to Constant Functions

The coherence of the norm equivalences (2.2.50) can be seen from the concrete values of Tables 5.25 and 5.26 for the constant function and the sine function, respectively. These only serve as examples for two important classes of functions: Constant functions are merely determined by the coefficients of the single scale basis. In contrast, the (infinitely differentiable) sine function has non-zero wavelet coefficients on any level $j \geq j_0$.
Note that the orthogonal basis transformation from Section 2.3.3 constructs the stiffness matrix $\mathbf{A}'_J$ (2.3.44) with very small absolute condition numbers, see Table 5.5, that is,

$$\mathbf{A}'_j \approx \mathbf{I}, \qquad \text{for all } j \geq j_0 \ .$$

Assuming $v = \mathbf{v}^T\Psi \in H^1$, we can conclude

$$\|v\|_{H^1} = \| \left(\mathbf{A}'\right)^{1/2} \mathbf{D}^{+1}\mathbf{v}\|_{\ell_2} \approx \|\mathbf{D}^{+1}\mathbf{v}\|_{\ell_2},$$

with probably smaller constants than in the usual equivalence relation $\|v\|_{H^1} \sim \|\mathbf{D}^{+1}\mathbf{v}\|_{\ell_2}$ for other MRAs. The third table in Table 5.25 shows optimal results for constant functions and the third table in Table 5.26 shows values with error less than 10^{-2} for the sine function.

The Riesz operator $\widehat{\mathbf{R}}_{H^s}$ can be improved by a normalization regarding constant functions. Since constant functions $v = c \in \mathbb{R}$ have the same values in the norm of Sobolev spaces H^s for $s \geq 0$, i.e., $\|v\|_{H^s} = \|v\|_{L_2}$, we should expect this to be reflected by our Riesz operator. As can be seen in Table 5.25, this is not the case for the operator $\widehat{\mathbf{R}}_{H^s}$: the value $\|\widehat{\mathbf{R}}_{H^s}\mathbf{v}\|_{\ell_2}$ grows exponentially in s. Exactness for the class of constant functions can be achieved by a simple normalization factor:
Consider the wavelet expansion of the constant function $v = \mathbf{v}^T\Psi \in H^s$. We then have by definition

$$\|\mathbf{v}\|_{H^s}^2 = \|\mathbf{v}\|_{L_2}^2 = \mathbf{v}^T\mathbf{M}_{L_2}\mathbf{v}, \tag{5.2.4}$$

but it also holds

$$\|\widehat{\mathbf{R}}_{H^s}^{1/2}\mathbf{v}^s\|_{\ell_2}^2 = \mathbf{v}^T\widehat{\mathbf{D}}^{+s}\mathbf{M}_{L_2}\widehat{\mathbf{D}}^{+s}\mathbf{v} \ . \tag{5.2.5}$$

As seen in Table 5.26, the matrices $\widehat{\mathbf{D}}^{+s}$ can have a significant impact on the computed norm.
Now note that the wavelet expansion coefficients of constant functions are exactly zero (the vectors $\mathbf{d}_j$ in (2.1.27)), except for the single-scale expansion coefficients of the minimum level j_0 ($\mathbf{c}_{j_0}$ in (2.1.27)). Thus, the diagonal preconditioner matrix $\widehat{\mathbf{D}}^{+s}$ can in this case effectively be written as a scaling of the identity matrix on the lowest level j_0,

$$\widehat{\mathbf{D}}^{+s}(\mathbf{c}_{j_0}, \mathbf{d}_{j_0}, \ldots, \mathbf{d}_{J-1})^T = 2^{+j_0 s}\, I_{j_0}\, \mathbf{c}_{j_0} = 2^{+j_0 s}\mathbf{c}_{j_0} \ . \tag{5.2.6}$$

To counter this effect, we introduce a simple scaling factor into our Riesz operator,

$$q_s = 2^{2j_0 s}, \tag{5.2.7}$$

and define the normalized Riesz operator

$$\mathring{\widehat{\mathbf{R}}}_{H^s} := q_{-s}\widehat{\mathbf{R}}_{H^s} \ . \tag{5.2.8}$$

Thus, the higher the lowest level j_0 in the MRA, the worse the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ fulfills the norm equivalences of Lemma 2.33. The (for every s) constant factor q_{-s} obviously does not change the spectral elements of $\widehat{\mathbf{R}}_{H^s}$.

The condition numbers of the **Riesz operators** $\mathring{\widehat{\mathbf{R}}}_J$ constructed in Section 2.2.6 can be found in Table 5.27. As can be seen there, the condition numbers are uniformly bounded for any value of $0 \leq s \leq 1$. This normalization will now prove essential in cases of constant traces of the state y and the control u itself. Consider $y|_{\Gamma_Y}$, y_{Γ_Y} and u known to be constant beforehand. Then we must expect the same solution (y, u) for any Sobolev smoothness indices $s, t \geq 0$. In the top row of Figure 5.11, we show the solution y of our control problem for the right hand sides of (P1) with the Sobolev smoothness parameters $s = t = 0$ for the B-type MRA. We used the exact Riesz operator $\mathbf{M}_{L_2}$ on both the observation and the control boundary. As it can be seen, the traces $y|_{\Gamma_Y}$ and $y|_{\Gamma}$ are exactly constant. The wavelet coefficient diagrams show the distinctive pattern for constant functions that emerges from the boundary adapted generators, see (2.3.25). The other three rows of diagrams show the same results for the cases $(s = 0, t = 1)$, $(s = 1, t = 0)$, $(s = 1, t = 1)$ using the respective exact Riesz operators $\mathbf{M}_{L_2}$ and $\mathbf{A}_{H^1}$ in each case. The computed solutions are all identical, which is consistent with constant traces $y|_{\Gamma_Y}$ and $y|_{\Gamma}$. Since this is true for all combinations $s, t \in \{0, 1\}$, it must also hold for $s, t \in [0, 1]$.

In Figure 5.12, we present the computed solution states y for fixed $t = 1/2$ and selected values $s \in [0, 3/2]$ with the Riesz operator $\widehat{\mathbf{R}}_{H^s}$,$\widehat{\mathbf{R}}_{H^t}$ for $H^s(\Gamma_Y)$ and $H^t(\Gamma)$, respectively. The graphic shows the states moving upward for increasing values of s. When the given observation y_{Γ_Y} is not constant, such a behavior is generally expected because $y|_{\Gamma_Y}$ is forced toward y_{Γ_Y} in a stronger norm.
This moving effect can be explained by the results in the columns for the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ in Table 5.25. In case $s = 0$, the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ simplifies to the mass matrix $\mathbf{M}_{L_2}$ and the values are exact. For $s > 0$, the attained term $\mathbf{v}^T\widehat{\mathbf{R}}_{H^s}\mathbf{v}$ is scaled irregularly with q_s. Therefore, the **data fitting term** in the functional $\mathbf{J}(\mathbf{y}, \mathbf{u})$ is weighted with a higher factor for $s > 1/2$ which emphasizes its importance supererogatorily. In general, the **data fitting term** is scaled with a factor q_s and the **regularization**

term is scaled with q_t. For constant traces and observation, we can conclude for the functional (4.2.65) with the Riesz operators $\mathbf{R}_{H^s} = \widehat{\mathbf{R}}_{H^s}$ and $\mathbf{R}_{H^t} = \widehat{\mathbf{R}}_{H^t}$,

$$\begin{aligned}\mathbf{J}(\mathbf{y},\mathbf{u}) &= \frac{1}{2}\|\widehat{\mathbf{R}}_{H^s}^{1/2}\mathbf{D}_{\Gamma_Y}^{-1/2+s}(\mathbf{T}\mathbf{y}-\mathbf{y}_{\Gamma_Y})\|_{\ell_2}^2 + \frac{\omega}{2}\|\widehat{\mathbf{R}}_{H^t}^{1/2}\mathbf{D}_{\Gamma}^{-1/2+t}\mathbf{u}\|_{\ell_2}^2 \\ &= \frac{q_s}{2}\|\mathbf{M}_{L_2}^{1/2}\mathbf{D}_{\Gamma_Y}^{-1/2}(\mathbf{T}\mathbf{y}-\mathbf{y}_{\Gamma_Y})\|_{\ell_2}^2 + \frac{\omega\, q_t}{2}\|\mathbf{M}_{L_2}^{1/2}\mathbf{D}_{\Gamma}^{-1/2}\mathbf{u}\|_{\ell_2}^2 \ .\end{aligned}$$

Since the functional can be scaled with any positive factor without changing the attained solution, this can be interpreted as choosing a different weighting parameter

$$\omega' := \frac{\omega\, q_t}{q_s} = \frac{\omega}{q_{s-t}}, \qquad \omega' \neq \omega \text{ for } s \neq t \ .$$

Numerical experiments confirm that the solution only depends on the value of the **difference** $s-t$ in this case for any values of s and t: Figure 5.13 (left side) shows the same solution state for the cases $s-t=1/2$ with $\omega = 1$ and $s = t = 1/2$ with $\omega = 1/q_{1/2} = 1/8$.
Here, however, this behavior must be considered incorrect and the solution to this problem is the normalization for constant functions. In Figure 5.13 (right side) we show the state y computed with the normalized Riesz operator $\mathring{\widehat{\mathbf{R}}}_{H^s}$,$\mathring{\widehat{\mathbf{R}}}_{H^t}$ for the same scenario of varying smoothness. All computed solutions are identical. Specifically, they are identical to the solutions for the exact case $s = t = 0$.
To summarize, we arrive at the following conclusion:

Proposition 5.5 *The normalization technique improves the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ to yield the exact norm for constant functions for any $s \geq 0$. The Riesz map $\mathring{\widehat{\mathbf{R}}}_{H^s}$ inherits from $\widehat{\mathbf{R}}_{H^s}$ the properties of linear complexity, exactness for $s = 0$ and equivalence for $s \geq 0$.*

Remark 5.6 *The Riesz operator $\widetilde{\mathbf{R}}_{H^s}$ of (2.2.53) can also be normalized with respect to constant functions by a simple normalization factor $\widetilde{q}_s$,*

$$\widetilde{q}_s := \frac{2^{-2j_0 s}}{(1-s)+s\,2^{-2j_0}}, \qquad 0 \leq s \leq 1 \ . \tag{5.2.9}$$

We denote the normalized Riesz operator by

$$\mathring{\widetilde{\mathbf{R}}}_{H^s} := \widetilde{q}_{-s}\widetilde{\mathbf{R}}_{H^s}, \tag{5.2.10}$$

see [11] for details.

For the modelling impact of the Riesz operator constructions and their effectiveness, see Figure 5.14. There we investigate the accurateness of the different Riesz operators for Problem (P3) with fixed $t = 0$ and $\omega = 10^{-4}$. The target state $y_{\Gamma_Y}(x_2) = 16x_2^2(1-x_2)^2$ is plotted with black color in each graphic.
We anticipate a convergence towards the target state for higher Sobolev indices s because the norm recognizes not only absolute values but also derivatives, i.e., curvature, for increasing s.
In this example we used the $\mathbf{D}_a$ preconditioner with the B-type MRA. Thus, in the top row with the Riesz operator $\mathbf{I}$ and $\mathbf{M}_{L_2}$, the coefficients of $\mathbf{D}_a$ determine the norm equivalence. The middle row incorporates the Riesz operators $\widehat{\mathbf{R}}_{H^s}$ and $\mathring{\widehat{\mathbf{R}}}_{H^s}$. This means effectively that the norm equivalence is now determined by the operator $\mathbf{D}_1$. The normalization factor for $\mathring{\widehat{\mathbf{R}}}_{H^s}$ seems a bit stringent, effectively canceling out the importance of the parameter s. The normalized interpolating Riesz operator $\mathring{\widetilde{\mathbf{R}}}_{H^s}$ in the last row shows much more accurate results. Specifically, it is exact for integer s.

Remark 5.7 *A completely different approach is given by the characterization of Sobolev spaces based upon* **Fourier transformation** *as outlined in Section 1.2. A Riesz operator for H^s is given by the Gramian matrix with respect to the inner product (1.2.12). This construction is not used here but it should be considered in the future.*

Finally, we can formulate our final remark of this section:

Remark 5.8 *These examples show that a perfect Riesz operator does not (yet) exist for fractional Sobolev spaces. In most applications even using no Riesz operator might be sufficient. Since, however, the exact Riesz operators for integer smoothness parameters are known, they should be incorporated to attain precise results. For fractional smoothness parameters, different constructions should be tested.*

J	$\#\mathcal{I}_{\mathcal{N},J}$	$\kappa_2(\mathbf{N}_{\Phi_j})$	$\kappa_2(\mathbf{N}_{\mathrm{B}})$ with $\mathbf{D}_a^{-1}$	$\kappa_2(\mathbf{N}_{\text{B-Orth}})$ with $\mathbf{D}_{\{O,a\}}^{-1}$
3	180	4.8096e+04	5.5078e+04	1.4284e+04
4	612	7.2443e+05	1.0841e+05	7.6472e+04
5	2.244	1.1467e+07	1.2573e+05	1.0342e+04
6	8.580	1.8297e+08	1.2512e+05	1.1123e+05
7	33.540	≈2.9e+09	1.2198e+05	1.1294e+05
8	132.612	>1.0e+10	1.2413e+05	1.1324e+05
9	527.364	—’’—	1.2622e+05	1.1347e+05
10	2.103.300	—’’—	≈1.27e+05	1.1369e+05
$\sim$	2^{2J}	2^{4J}	1	1

Table 5.20: Condition numbers for the operator $\mathbf{N}_J$ with the $\mathring{\widehat{\mathbf{R}}}_J$ Riesz operator for the B-type MRA with different preconditioners and basis transformations. The values are computed by $\kappa_2\left(\mathbf{N}_J\right) = \sqrt{\kappa_2\left(\mathbf{N}_J^T\mathbf{N}_J\right)}$.

J	$\#\mathcal{I}_{\mathcal{N},J}$	$\kappa_2(\mathbf{N}_{\mathrm{P}})$ with $\mathbf{D}_1^{-1}$	$\kappa_2(\mathbf{N}_{\mathrm{P}})$ with $\mathbf{D}_a^{-1}$	$\kappa_2(\mathbf{N}_{\text{P-Orth}})$ with $\mathbf{D}_{\{O,a\}}^{-1}$
4	612	1.1232e+04	2.1948e+04	2.2466e+04
5	2.244	1.7817e+04	4.6474e+04	4.1120e+04
6	8.580	2.3520e+04	5.8008e+04	4.7253e+04
7	33.540	2.7210e+04	6.4409e+04	4.9003e+04
8	132.612	2.9477e+04	6.8140e+04	4.9344e+04
9	527.364	3.0869e+04	7.0448e+04	4.9361e+04
10	2.103.300	3.1743e+04	7.1956e+04	4.9378e+04
11	8.400.900	≈3.22e+04	7.2623e+04	4.9394e+04
$\sim$	2^{2J}	1	1	1

Table 5.21: Condition numbers for the operator $\mathbf{N}_J$ with the $\mathring{\widehat{\mathbf{R}}}_J$ Riesz operator for the P-type MRA with different preconditioners and basis transformations. The values are computed as $\kappa_2\left(\mathbf{N}_J\right) = \sqrt{\kappa_2\left(\mathbf{N}_J^T\mathbf{N}_J\right)}$.

	B with $\mathbf{D}_a^{-1}$			
J	$\mathbf{R}_J = \mathbf{I}$	$\mathbf{R}_J = \mathbf{M}_{L_2}$	$\mathbf{R}_J = \widehat{\mathbf{R}}_{H^s}$	$\mathbf{R}_J = \mathring{\widehat{\mathbf{R}}}_{H^s}$
3	6.4233e+02	1.9598e+03	2.0699e+03	5.5078e+04
4	7.9723e+02	2.5169e+03	2.9799e+03	1.0841e+05
5	8.6385e+02	2.9201e+03	3.5960e+03	1.2573e+05
6	9.2382e+02	3.1572e+03	3.9314e+03	1.2512e+05
7	9.7015e+02	3.2997e+03	4.1260e+03	1.2194e+05
8	1.0108e+03	3.3899e+03	4.2464e+03	1.2413e+05
9	1.0383e+03	3.4498e+03	4.3251e+03	≈1.26e+05
10	1.0588e+03	3.4914e+03	4.3791e+03	
11	1.0740e+03	3.5213e+03	4.4176e+03	
$\sim$	1	1	1	1

	P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$			
J	$\mathbf{R}_J = \mathbf{I}$	$\mathbf{R}_J = \mathbf{M}_{L_2}$	$\mathbf{R}_J = \widehat{\mathbf{R}}_{H^s}$	$\mathbf{R}_J = \mathring{\widehat{\mathbf{R}}}_{H^s}$
4	1.0006e+01	3.4643e+01	3.0214e+02	2.2466e+04
5	1.6898e+01	7.6618e+01	3.3192e+02	4.1120e+04
6	2.4689e+01	9.1031e+01	3.1784e+02	4.7253e+04
7	3.5607e+01	9.5867e+01	3.0186e+02	4.9003e+04
8	4.4758e+01	9.7392e+01	2.8970e+02	4.9344e+04
9	5.1437e+01	9.7929e+01	2.8159e+02	4.9361e+04
10	5.6012e+01	9.8258e+01	2.7651e+02	4.9378e+04
11	5.9209e+01	9.8539e+01	2.7354e+02	4.9394e+04
$\sim$	1	1	1	1

Table 5.22: Impact of the different Riesz operators onto the condition number of the operator $\mathbf{N}_J$ for different constructions of the wavelet bases. The smoothness parameters $s = t = 1/2$ remain fixed while the Riesz operator is varied. An increase in condition numbers on every level is recognizable for more involved Riesz operator constructions (from left to right columns) independent of the utilized preconditioner.

Riesz Operator	$\mathbf{I}$			$\mathbf{M}_{L_2}$			$\mathring{\widehat{\mathbf{R}}}_{H^s}$			$\mathring{\widetilde{\mathbf{R}}}_{H^s}$		
Iterations	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j=3$	2	–	–	2	–	–	1	–	–	2	–	–
$j=4$	3	5.6	5.3	4	3.25	3.75	0	–	–	4	2.75	1.5
$j=5$	0	–	–	0	–	–	0	–	–	0	–	–
$j=6$	4	5.75	5.5	4	3.75	5.25	0	–	–	4	3.75	1.75
$j=7$	0	–	–	0	–	–	3	5.6	1	0	–	–
$j=8$	0	–	–	0	–	–	0	–	–	3	4	1
$j=9$	3	5.6	5.3	4	4	3.75	5	4.6	0.6	0	–	–
$j=10$	1	2	1	1	3	2	4	3	0.25	3	3.6	0.6
Overall Sum	13	114		13	100		13	83		16	67	
$\mathbf{J}(\mathbf{y},\mathbf{u})$	2.9652e−11			2.1002e−11			1.7468e−11			6.8324e−12		

Table 5.23: Influence of the choice of the Riesz operator onto the Inexact Gradient algorithm for problem setup (P1) . Here we used the B-type MRA with $\mathbf{D}_a^{-1}$ for preconditioning and $\rho = 0.5$. The stopping criterion is $\frac{1}{100}h_j$ on every level. The nested iteration starts with the **QR**-solver on level $j_0 = 3$ and then the UzawaCD algorithm is used on the higher levels. Left out entries indicate that no iteration steps were executed.

Riesz Operator	$\mathbf{I}$			$\mathbf{M}_{L_2}$			$\mathring{\widehat{\mathbf{R}}}_{H^s}$			$\mathring{\widetilde{\mathbf{R}}}_{H^s}$		
Iterations	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j=4$	1	–	–	1	–	–	1	–	–	1	–	–
$j=5$	0	–	–	2	1.5	1	0	–	–	0	–	–
$j=6$	2	2	1	0	–	–	0	–	–	0	–	–
$j=7$	0	–	–	0	–	–	0	–	–	0	–	–
$j=8$	2	1.5	1	3	2.6	1.3	0	–	–	3	2.3	0.3
$j=9$	1	2	1	0	–	–	4	2.75	0.25	0	–	–
$j=10$	1	1	1	1	2	2	5	2	0.2	3	2	0.3
Overall Sum	7	16		7	21		10	23		7	12	
$\mathbf{J}(\mathbf{y},\mathbf{u})$	1.5749e−11			2.1002e−11			5.1101e−12			5.6629e−12		

Table 5.24: Influence of the choice of the Riesz operator onto the Inexact Gradient algorithm for problem setup (P1) . Here we used the P-Orth-type MRA with $\mathbf{D}_{\{O,a\}}^{-1}$ for preconditioning and $\rho = 0.5$. The stopping criterion is $\frac{1}{100}h_j$ on every level. The nested iteration starts with the **QR**-solver on level $j_0 = 4$ and then the UzawaCD algorithm is used on the higher levels. Left out entries indicate that no iteration steps were executed.

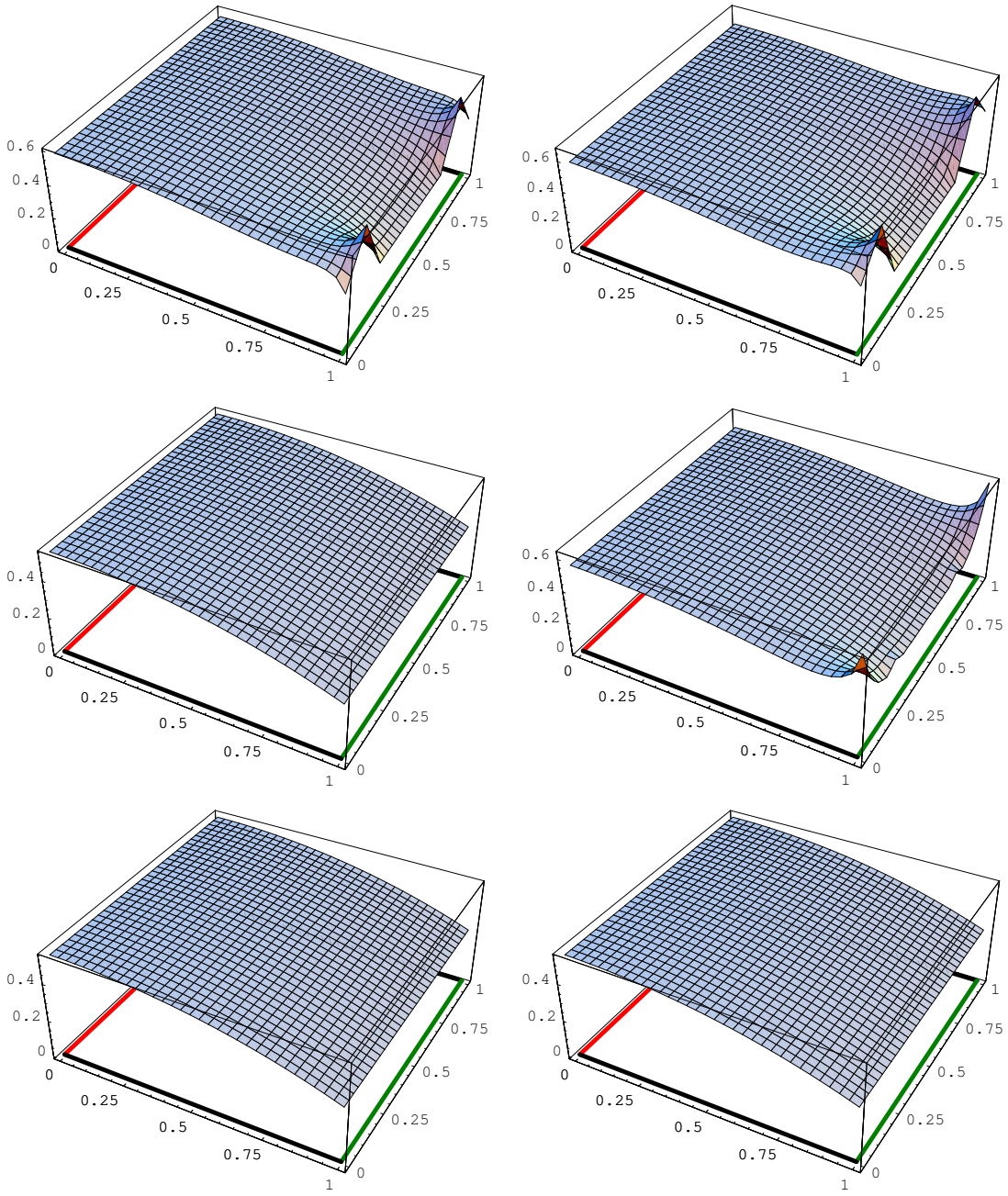


Figure 5.10: Plots of the state y for varying preconditioners and Riesz operators. The right hand sides were set according to (P2) with $\omega = 1$ and $s = t = 1/2$. In the left column, we have always used $\mathbf{D}_1$ as the preconditioning operator while we have chosen $\mathbf{D}_a$ for the three images in the right column. From top to bottom, the Riesz operators change from $\mathbf{R}_{H^s} = \mathbf{I}$ via $\mathbf{M}_{L_2}$ to $\widehat{\mathbf{R}}_{H^s}$ of (2.2.44). This means we have used effectively no Riesz operator in the first row and only the Riesz operator suitable for L_2 in the middle row. Only the last row shows the same smooth solution independent of the preconditioner. In every image, the observation boundary is on the far (red) edge and the control boundary on the front right (green) edge. All solutions were computed using the QR algorithm on $\mathbf{NU} = \mathbf{F}$ (assembled on level $j = 5$) to rule out rounding errors as a cause for the boundary irregularities. The alleged independence of the solution y with the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ of the preconditioner is clearly demonstrated.

Calculated Norm	L_2	
Riesz Operator	$\mathbf{I}$	$\mathbf{M}_{L_2}$
DKU with $\mathbf{D}_1^{-1}$	9.3541e−01	1.0000e+00
DKU with $\mathbf{D}_a^{-1}$	9.3541e−01	1.0000e+00
B with $\mathbf{D}_1^{-1}$	9.3541e−01	1.0000e+00
B with $\mathbf{D}_a^{-1}$	9.3541e−01	1.0000e+00
B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	9.3541e−01	1.0000e+00
P with $\mathbf{D}_1^{-1}$	1.0307e+00	1.0000e+00
P with $\mathbf{D}_a^{-1}$	1.0307e+00	1.0000e+00
P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	1.0307e+00	1.0000e+00

Calculated Norm	$H^{1/2}$			
Riesz Operator	$\mathbf{I}$	$\widehat{\mathbf{R}}_{H^{1/2}}$	$\mathring{\widehat{\mathbf{R}}}_{H^{1/2}}$	$\widetilde{\mathbf{R}}_{H^{1/2}}$
DKU with $\mathbf{D}_1^{-1}$	2.6457e+00	2.8284e+00	1.0000e+00	2.0155e+00
DKU with $\mathbf{D}_a^{-1}$	3.0183e+00	2.8284e+00	1.0000e+00	2.2647e+00
B with $\mathbf{D}_1^{-1}$	2.6457e+00	2.8284e+00	1.0000e+00	2.0155e+00
B with $\mathbf{D}_a^{-1}$	3.0183e+00	2.8284e+00	1.0000e+00	2.2647e+00
B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	9.6671e−01	2.8284e+00	1.0000e+00	1.0032e+00
P with $\mathbf{D}_1^{-1}$	4.1231e+00	4.0000e+00	1.0000e+00	2.8339e+00
P with $\mathbf{D}_a^{-1}$	4.8195e+00	4.0000e+00	1.0000e+00	3.3400e+00
P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	1.0152e+00	4.0000e+00	1.0000e+00	1.0004e+00

Calculated Norm	H^1				
Riesz Operator	$\mathbf{I}$	$\widehat{\mathbf{R}}_{H^1}$	$\mathring{\widehat{\mathbf{R}}}_{H^1}$	$\widetilde{\mathbf{R}}_{H^{1/2}}$	$\mathbf{A}_{H^1}$
DKU with $\mathbf{D}_1^{-1}$	7.4833e+00	8.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00
DKU with $\mathbf{D}_a^{-1}$	9.8361e+00	8.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00
B with $\mathbf{D}_1^{-1}$	7.4833e+00	8.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00
B with $\mathbf{D}_a^{-1}$	9.8361e+00	8.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00
B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	1.0000e+00	8.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00
P with $\mathbf{D}_1^{-1}$	1.6492e+01	1.6000e+01	1.0000e+00	1.0000e+00	1.0000e+00
P with $\mathbf{D}_a^{-1}$	2.2642e+01	1.6000e+01	1.0000e+00	1.0000e+00	1.0000e+00
P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	1.0000e+00	1.6000e+01	1.0000e+00	1.0000e+00	1.0000e+00

Table 5.25: Norms of the constant function $f \equiv 1$ in wavelet discretization with various Riesz operators. The shown results are independent of the level of spatial dimension since all wavelet coefficients are zero for constant functions. The exact value of $\|f\|_{H^s}$ is 1 for any of the spaces L_2 or $H^s, s > 0$. The second column in the middle table and the third column in the last table show that the normed Riesz operator $\mathring{\widehat{\mathbf{R}}}_{H^s}$ yields the same optimal results as the exact Riesz operators for L_2 and H^1.

Calculated Norm	L_2	
Riesz Operator	$\mathbf{I}$	$\mathbf{M}_{L_2}$
DKU with $\mathbf{D}_1^{-1}$	7.1638e−01	7.0709e−01
DKU with $\mathbf{D}_a^{-1}$	7.1638e−01	7.0709e−01
B with $\mathbf{D}_1^{-1}$	7.1634e−01	7.0709e−01
B with $\mathbf{D}_a^{-1}$	7.1634e−01	7.0709e−01
B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	7.1634e−01	7.0709e−01
P with $\mathbf{D}_1^{-1}$	7.0937e−01	7.0709e−01
P with $\mathbf{D}_a^{-1}$	7.0937e−01	7.0709e−01
P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	7.0937e−01	7.0709e−01

Calculated Norm	$H^{1/2}$			
Riesz Operator	$\mathbf{I}$	$\widehat{\mathbf{R}}_{H^{1/2}}$	$\mathring{\widehat{\mathbf{R}}}_{H^{1/2}}$	$\widetilde{\mathbf{R}}_{H^{1/2}}$
DKU with $\mathbf{D}_1^{-1}$	2.0264e+00	1.9999e+00	7.0710e−01	1.5298e+00
DKU with $\mathbf{D}_a^{-1}$	2.4130e+00	1.9999e+00	7.0710e−01	1.7538e+00
B with $\mathbf{D}_1^{-1}$	2.0262e+00	1.9999e+00	7.0710e−01	1.5298e+00
B with $\mathbf{D}_a^{-1}$	2.4128e+00	1.9999e+00	7.0710e−01	1.7539e+00
B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	1.1475e+00	1.9999e+00	7.0710e−01	1.0707e+00
P with $\mathbf{D}_1^{-1}$	2.8375e+00	2.8283e+00	7.0709e−01	2.0420e+00
P with $\mathbf{D}_a^{-1}$	3.3755e+00	2.8283e+00	7.0709e−01	2.4042e+00
P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	1.0662e+00	2.8283e+00	7.0709e−01	1.0431e+00

Calculated Norm	H^1				
Riesz Operator	$\mathbf{I}$	$\widehat{\mathbf{R}}_{H^1}$	$\mathring{\widehat{\mathbf{R}}}_{H^1}$	$\widetilde{\mathbf{R}}_{H^1}$	$\mathbf{A}_{H^1}$
DKU with $\mathbf{D}_1^{-1}$	5.7332e+00	5.6569e+00	7.0712e−01	2.3312e+00	2.3312e+00
DKU with $\mathbf{D}_a^{-1}$	8.1291e+00	5.6569e+00	7.0712e−01	2.3312e+00	2.3312e+00
B with $\mathbf{D}_1^{-1}$	5.7318e+00	5.6569e+00	7.0712e−01	2.3312e+00	2.3312e+00
B with $\mathbf{D}_a^{-1}$	8.1291e+00	5.6569e+00	7.0712e−01	2.3312e+00	2.3312e+00
B-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	2.3595e+00	5.6569e+00	7.0712e−01	2.3312e+00	2.3312e+00
P with $\mathbf{D}_1^{-1}$	1.1350e+01	1.1313e+01	7.0799e−01	2.3312e+00	2.3312e+00
P with $\mathbf{D}_a^{-1}$	1.6062e+01	1.1313e+01	7.0799e−01	2.3312e+00	2.3312e+00
P-Orth with $\mathbf{D}_{\{O,a\}}^{-1}$	2.3384e+00	1.1313e+01	7.0799e−01	2.3312e+00	2.3312e+00

Table 5.26: Norms of the function $f(x) = \sin(\pi x)$ in 1D on level $j = 8$. The exact value of its L_2 norm is $\frac{1}{\sqrt{2}} \approx 7.07106\text{e}{-}01$ and that of its H^1-norm is $\sqrt{\frac{1}{2} + \frac{\pi^2}{2}} \approx 2.3312\text{e}{+}00$.

J	$\#\Delta_J$	$s = 0.0$	$s = 0.25$	$s = 0.5$	$s = 0.75$	$s = 1.0$
3	9	1.9733e+01	2.1118e+01	2.2902e+01	2.5136e+01	2.7883e+01
4	17	2.5478e+01	2.8669e+01	3.3057e+01	3.9036e+01	4.7182e+01
5	33	2.9589e+01	3.4051e+01	3.9893e+01	4.7534e+01	5.7592e+01
6	65	3.1998e+01	3.7074e+01	4.3613e+01	5.2067e+01	6.3093e+01
7	129	3.3446e+01	3.8854e+01	4.5771e+01	5.4673e+01	6.6233e+01
8	257	3.4361e+01	3.9964e+01	4.7106e+01	5.6273e+01	6.8154e+01
9	513	3.4970e+01	4.0695e+01	4.7980e+01	5.7316e+01	6.9399e+01
10	1.025	3.5391e+01	4.1198e+01	4.8579e+01	5.8029e+01	7.0249e+01
11	2.049	3.5695e+01	4.1559e+01	4.9006e+01	5.8535e+01	7.0851e+01
12	4.097	3.5919e+01	4.1825e+01	4.9320e+01	5.8908e+01	7.1293e+01
14	16.385	3.6223e+01	4.2183e+01	4.9742e+01	5.9406e+01	7.1882e+01
16	65.535	3.6413e+01	4.2405e+01	5.0004e+01	5.9714e+01	7.2246e+01
$\sim$	2^J	1	1	1	1	1

1D Riesz operators

J	$\#\Delta_J$	$s = 0.0$	$s = 0.25$	$s = 0.5$	$s = 0.75$	$s = 1.0$
3	81	3.8941e+02	3.8881e+02	3.8882e+02	3.8944e+02	3.9067e+02
4	289	6.4915e+02	6.4958e+02	6.6703e+02	6.9539e+02	7.3445e+02
5	1.089	8.7551e+02	8.4270e+02	8.7823e+02	9.4457e+02	1.0219e+03
6	4.225	1.0239e+03	9.7847e+02	1.0665e+03	1.1875e+03	1.2917e+03
7	16.641	1.1187e+03	1.0937e+03	1.2660e+03	1.4266e+03	1.5296e+03
8	66.049	1.1807e+03	1.2138e+03	1.4608e+03	1.6410e+03	1.7204e+03
9	263.169	1.2229e+03	1.3477e+03	1.6656e+03	1.8198e+03	1.8641e+03
10	1.050.625	1.2526e+03	1.4906e+03	1.8457e+03	1.9623e+03	1.9694e+03
11	4.198.401	1.2741e+03	1.6358e+03	1.9951e+03	2.0733e+03	2.0458e+03
12	16.785.409	1.2902e+03	1.7795e+03	2.1262e+03	2.1585e+03	2.1012e+03
$\sim$	2^{2J}	1	1	1	1	1

2D Riesz operators

Table 5.27: Dependence of the condition number of the Riesz operator $\mathring{\mathbf{R}}_{H^s}$ on the diagonal preconditioner. The table shows the condition numbers of the Riesz operator $\mathring{\mathbf{R}}_{H^s}$ while varying the norm index s and using $\mathbf{D}_a$ (with the B-type wavelets) for preconditioning. In the first column ($s = 0$) $\mathring{\mathbf{R}}_{H^s}$ coincides with the mass matrix $\mathbf{M}_{L_2}$. If we had chosen $\mathbf{D}_1$ for preconditioning, all columns would read like the first one as the diagonal scalings cancel each other out. Since it only holds $\mathbf{D}_a = \Theta(\mathbf{D}_1)$, we generally see a slight increase for ascending values of s. The exact Riesz operator for H^1 is the **stiffness matrix** $\mathbf{A}_{H^1}$.

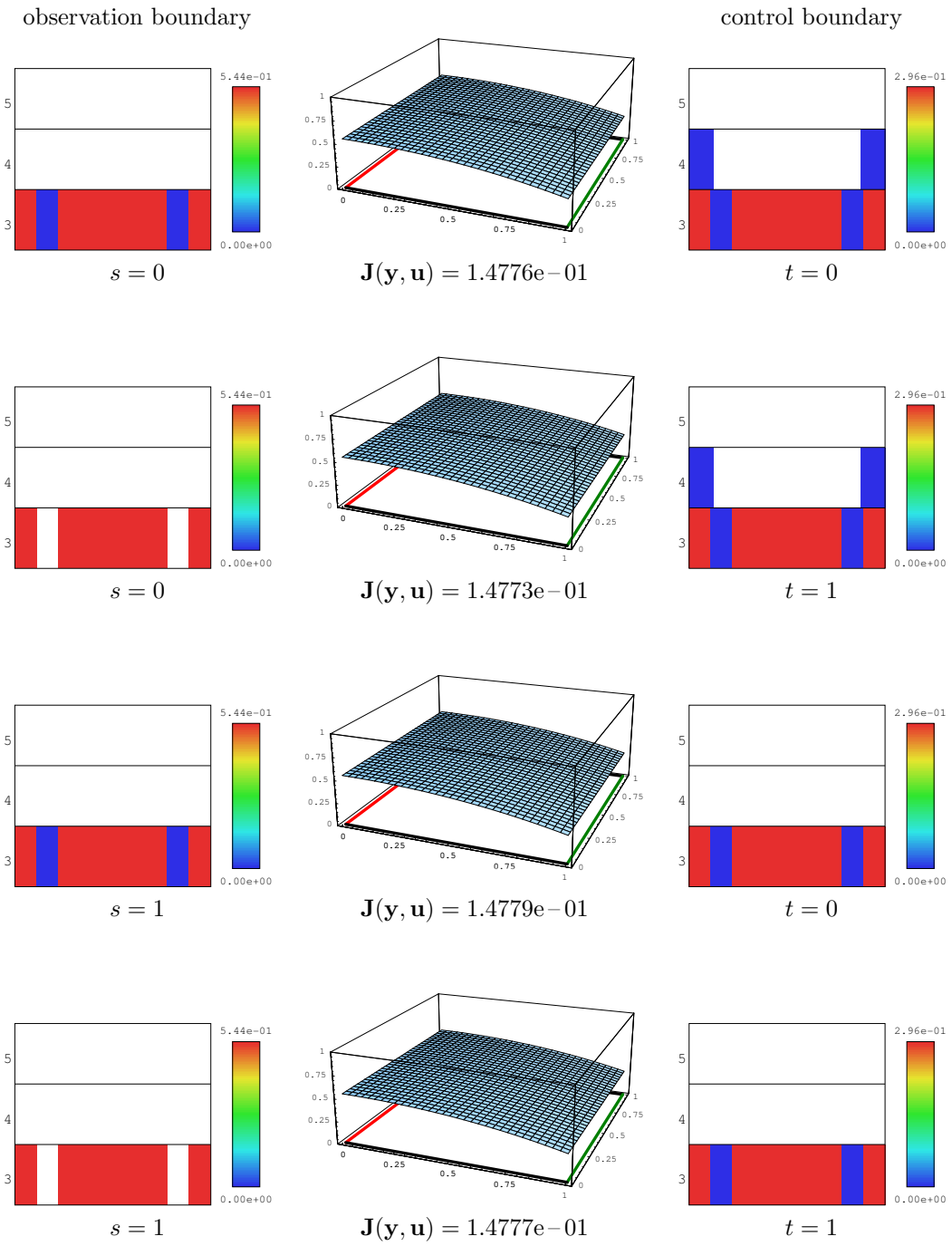


Figure 5.11: Solutions to the control problem with $f = y_{\Gamma_Y} \equiv 1$ for the values $s, t \in \{0, 1\}$. These are calculated up to discretization error accuracy with nested iteration algorithm and the B-type wavelets. In these cases, we use the exact Riesz operators $\mathbf{M}_{L_2}$ for L_2 and $\mathbf{A}_{H^1}$ for H^1. Therefore, the four states y shown in the middle column are the exact solutions. These are exactly equal up to the precision that can be expected by the numerical experiments. On the left and right hand side, we show the wavelet coefficients for the trace of the state $y|_{\Gamma_Y}$ on the observation boundary Γ_Y and control u on Γ. White blocks indicate values equal to zero up to machine precision while blue blocks are zero up to discretization error. These diagrams show that $y|_{\Gamma_Y}$ and u are constant. This is also emphasized by the value of the functional $\mathbf{J}(\mathbf{y}, \mathbf{u})$: in each of these cases, the functional attains the same value, although the norms are different.

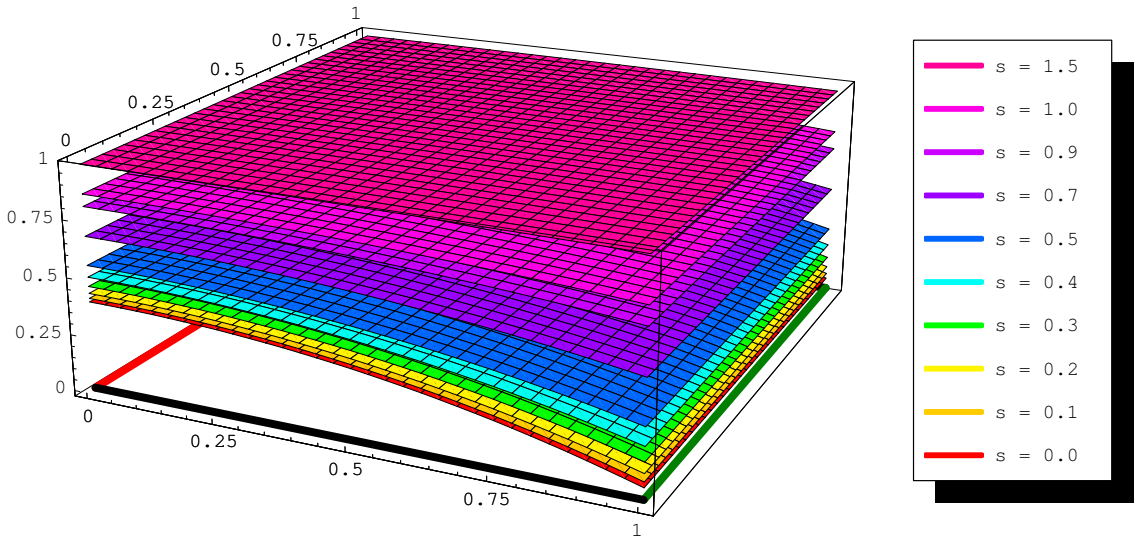


Figure 5.12: Varying the Sobolev index in the norm of the data fitting term in functional (4.3.1) for Problem (P2) with the Riesz operator $\widehat{\mathbf{R}}_{H^s}$. The other parameters are chosen as $\omega = 1$ and $t = 1/2$. Higher values of s indicate stronger norms which should not make a difference in this case. Without the normalization factor, $\mathbf{y}|_{\Gamma_Y}$ is forced more strongly towards $\mathbf{y}_{\Gamma_Y} \equiv 1$ for increasing values of s and the solution generally shifts upwards.

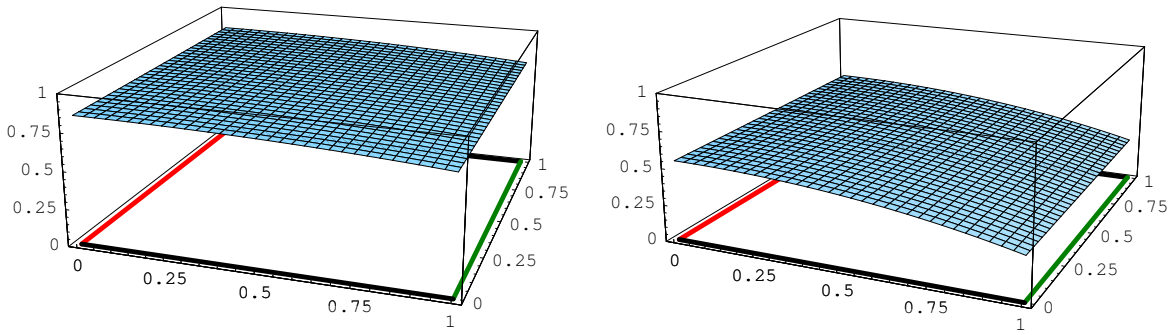


Figure 5.13: Left: Solution states for the control problem 4.20 and (P2) with the Riesz operator $\widehat{\mathbf{R}}_{H^s}$ with DKU wavelets. The parameter values $(s = 1, t = 1/2, \omega = 1)$, $(s = 1/2, t = 0, \omega = 1)$ and $(s = 1/2, t = 1/2, \omega = 1/8)$ all yield the same solution displayed here.
Figure 5.13: Right: Varying the Sobolev index in the norm of the data fitting term in functional (4.3.1) for Problem (P2) with the normalized Riesz operator $\mathring{\widehat{\mathbf{R}}}_{H^s}$. The other parameters are chosen as $\omega = 1$ and $t = 1/2$. The solutions are all identical for any value of s.

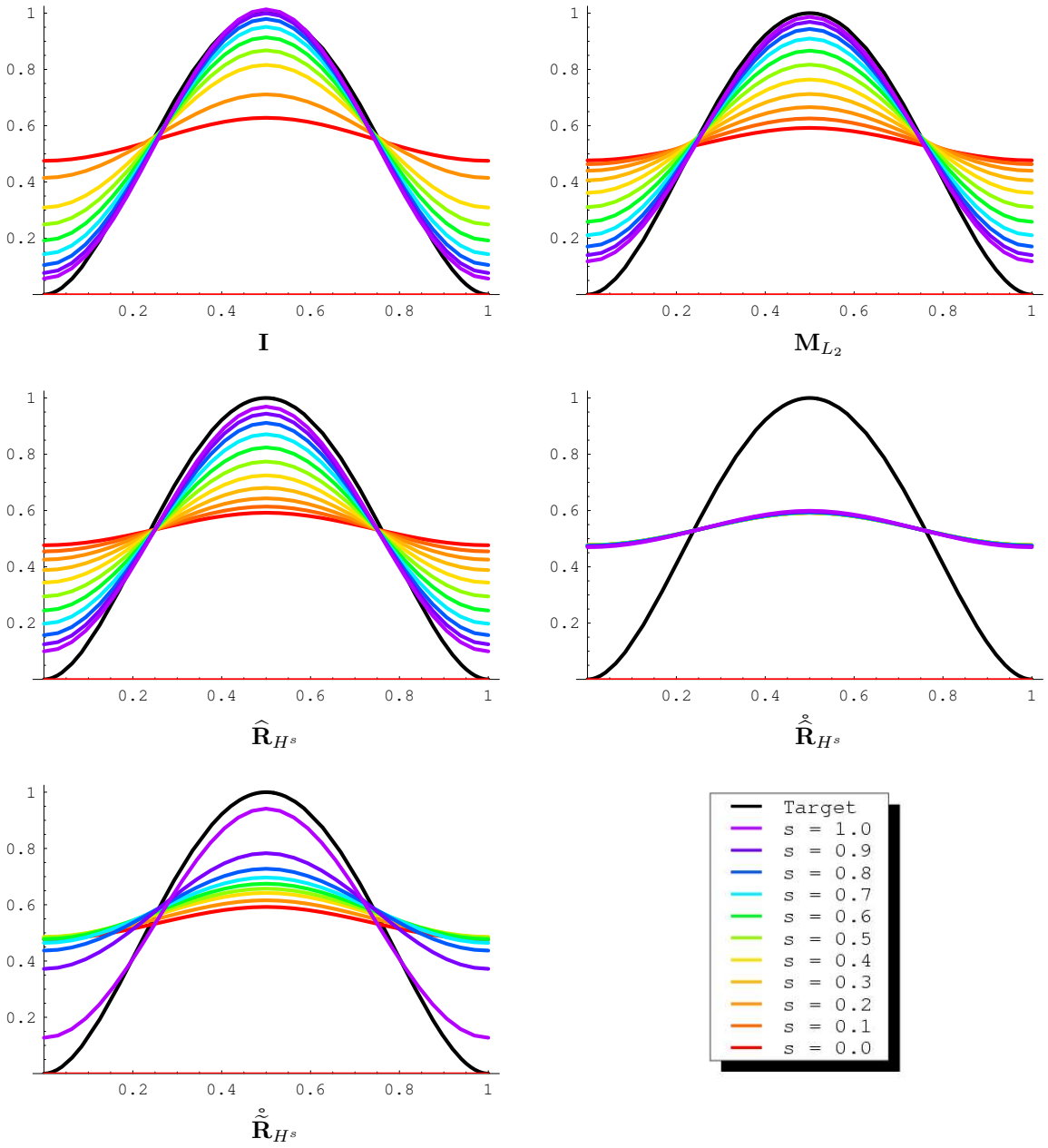


Figure 5.14: Traces of the solution states $\mathbf{y}$ on the observation boundary Γ_Y for varying smoothness indices s and fixed smoothness parameter $t = 0$ for Problem (P3) with $\omega = 10^{-4}$. Here we used the B-type wavelets with $\mathbf{D}_a$ preconditioner. The top left picture depicts the case of incorporating no Riesz operator. Introducing the mass matrix $\mathbf{M}_{L_2}$ (top right) does not change the solution significantly but the trace norm is in this case exactly evaluated for $s = 0$. The middle left picture corresponds to the case where the diagonal matrix $\mathbf{D}_1$ is used as the weighting operator for the wavelet coefficients. In the middle right diagram the normalization factor q_s is included. For the bottom left the normalized interpolating Riesz operator $\mathring{\widetilde{\mathbf{R}}}_{H^s}$ is used. Thus, for $s = 0$ and $s = 1$ the exact norms are evaluated in the numerical experiments.

5.3 Further Numerical Experiments

5.3.1 Robustness of the Inexact Gradient Scheme

We are now going to investigate some of the properties of the Inexact Gradient Algorithm 4.25.

Convergence Speed Subject to ρ

The step size parameter ρ is the key parameter for ensuring and improving the convergence of the **Inexact Gradient** algorithm. It is known by Theorem 4.26 that for convergence the step size ρ can only be chosen in a limited range of values depending on the problem parameters. We show the iteration numbers of the outer iteration for setup (P3) with $\omega = 1$ and the $\mathring{\widetilde{\mathbf{R}}}_{H^s}$ Riesz operator with $s = t = \frac{1}{2}$ in Tables 5.28 and 5.29. The nested iteration starts with a QR-decomposition on level 4 to calculate a starting vector of $\frac{1}{100}2^{-j_0}$ precision. The UzawaCD algorithm is then used to reduce the error by a factor of 2 on every level. The results clearly indicate that the overall number of iterations (which determines the overall complexity) has a minimum for $\rho = 1$ in this case. The optimal value for the problem setup (P1) is empirically found to be $\rho = \frac{3}{4}$.

	$\rho = \frac{1}{100}$	$\rho = \frac{1}{20}$	$\rho = \frac{1}{10}$	$\rho = \frac{1}{4}$	$\rho = \frac{1}{2}$	$\rho = \frac{3}{4}$	$\rho = 1$	$\rho = \frac{5}{4}$	$\rho = \frac{4}{3}$	$\rho = \frac{41}{30}$
$j = 3$	192	39	19	8	4	2	4	12	24	44
$j = 4$	373	78	40	18	0	0	0	0	0	0
$j = 5$	302	69	35	14	7	5	0	15	28	45
$j = 6$	286	50	19	10	5	5	6	0	0	0
$j = 7$	251	44	45	10	0	4	2	10	15	8
$j = 8$	247	59	27	12	7	0	2	0	0	0
$j = 9$	297	62	31	12	0	4	2	0	11	15
$j = 10$	296	63	33	12	7	3	2	4	4	5
Overall Sum	2244	464	249	96	30	23	18	41	84	117

Table 5.28: Iteration numbers of the **Inexact Gradient** algorithm employing the nested iteration strategy of Problem (P3) with the B-type MRA with $\mathbf{D}_a$ preconditioner. We used the QR decomposition on level $j_0 = 3$ and the UzawaCD algorithm for the higher levels. The algorithm did not converge for $\rho = \frac{7}{5}$.

	$\rho = \frac{1}{100}$	$\rho = \frac{1}{20}$	$\rho = \frac{1}{10}$	$\rho = \frac{1}{4}$	$\rho = \frac{1}{2}$	$\rho = \frac{3}{4}$	$\rho = 1$	$\rho = \frac{5}{4}$	$\rho = \frac{4}{3}$	$\rho = \frac{41}{30}$
$j = 4$	236	47	23	9	4	3	5	14	30	55
$j = 5$	232	52	31	13	0	0	0	0	0	0
$j = 6$	284	86	48	20	11	0	0	15	25	31
$j = 7$	320	65	32	13	7	6	0	6	8	10
$j = 8$	355	68	30	14	7	5	7	0	6	7
$j = 9$	351	72	33	11	0	4	2	7	6	0
$j = 10$	343	56	25	12	10	4	2	5	3	5
Overall Sum	2121	446	222	92	39	22	16	47	78	108

Table 5.29: Iteration numbers of the **Inexact Gradient** algorithm employing the nested iteration strategy of Problem (P3) with the P-Orth-type MRA with $\mathbf{D}_{\{O,a\}}$ preconditioner. We used the QR decomposition on level $j_0 = 4$ and the UzawaCD algorithm for the higher levels. The algorithm did not converge for $\rho = \frac{3}{2}$.

The algorithm clearly proves to be more stable with respect to the step size parameter bounds (4.4.15) at the lower bound. Very small values of ρ imply long convergence times while too high values cause the program to interrupt abnormally because the residual becomes infinite, i.e., larger than 1.79769e+308 for double precision data structures. Also, the convergence is of almost equal complexity for a range of values, here $\rho \in [\frac{1}{2}, \frac{5}{4}]$ provides almost the same quality of convergence. In these two cases, the qualitative effectiveness of the step size parameter ρ is identical for our problem although this cannot be guaranteed since the bounds in (4.4.15) depend on properties of the selected MRA.

A good strategy for determining the step size parameter is testing several values for ρ on the lowest level and determining a good value by binary searching. This value can then be used in the nested iteration because the convergence speed of the step size parameter is largely, i.e., asymptotically, independent of the level of resolution.
Another case study of this kind is given in Table 5.32 for a different setup.

Convergence Speed with Extreme Values of ω

The admissible range of values for the step size parameter ρ depends by Theorem 4.26 not only on the quantitative properties of the involved operators in the wavelet discretization but also on the value of the weighting parameter ω.

The case $\omega \gg 1$ forces the control towards $u \equiv 0$ for any norm in the functional (4.3.1). This means that the state $y = y(u)$ is determined by the elliptic boundary value problem (3.2.1) with $u = 0$. Thus, the control problem is very well-posed in the sense that the solution is easily determined. The case $\omega \ll 1$ is more interesting because the control is less penalized and this can have a considerable impact on the solution state $y(u)$. In the limit, $\omega \equiv 0$, the control problem is no longer well-posed because the control u can effectively become very unsmooth, e.g., $u \notin L_2(\Gamma)$ is possible.

We show the iteration numbers for values $\omega \ll 1$ for setup (P2) in Table 5.30. The best iteration numbers are observed for $\rho = 2\,\omega$. Since the solution to (P2) with $\omega = 0$ is $y \equiv 1$, whereby the **data fitting term** attains exactly zero, we expect and observe

$$\mathbf{J}(\mathbf{y}, \mathbf{u}) = \frac{\omega}{2}\, 1^2 \qquad \text{for } \omega \to 0 \ .$$

Since the magnitude of the step size parameter ρ also has an impact on the tolerances $\varepsilon_y(i+1, J_{\mathcal{H}})$ and $\varepsilon_\mu(i+1, J_{\mathcal{H}})$ of Theorem 4.26, these have to be chosen stricter for $\omega \to 0$. Thus, the complexity of the numerical scheme grows proportionally with the exponent of the parameter ω, i.e., the inner iteration numbers of the Uzawa algorithms grow with ω approaching zero. Convergence speed is nevertheless optimal, i.e., the number of iterations is uniformly bounded.
In the case $\omega \gg 1$ (cf. Table 5.31) the algorithm converges very quickly and stable since $\mathbf{u} \equiv \mathbf{0}$ is being attained. The solution $\mathbf{y}(\mathbf{u})$ is then determined by (5.1.2) for $a_0 = 1$. The functional therefore converges towards

$$\mathbf{J}(\mathbf{y}, \mathbf{u}) = \frac{1}{2}\left(\frac{1}{\cosh(1)}\right)^2 \approx 2.09987\text{e}-1 \ .$$

The overall complexity grows when deviating from the optimal value of the step size parameter. The experiments in Tables 5.30 and 5.31 show that one can choose at least values with a **relative tolerance** of 50% for ρ and still have good performance. The **absolute** size of the interval of admissible step sizes, however, is reduced significantly.

In summary, losing the well-posed problem formulation for $\omega = 0$ cannot be undone by any means. The Inexact Gradient algorithm still converges for $\omega \ll 1$ although the step size parameter ρ must be chosen proportional to the weighting parameter ω here. The numerical complexity also increases because the residual tolerances must also be chosen proportional to ω. Since with optimal preconditioned system matrices the residual error can be reduced by any constant factor with a uniformly bounded number of steps, this effect does increase the number of steps by a constant. The whole algorithm is still of optimal complexity. This changes if ω approaches **machine precision** $\varepsilon = 2.2204\text{e}-16$. At this point the numerical scheme must break down because numerical results can never be obtained better than machine precision. In the computer system, the parameter ω is then simply indistinguishable from the value zero.

	$\omega = 10^{-6}$								
Step size ρ	$1\,10^{-6}$			$2\,10^{-6}$			$3\,10^{-6}$		
Iterations	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j = 3$	31	–	–	10	–	–	13	–	–
$j = 4$	10	7.3	6.7	3	8.3	7.3	4	8.25	7.75
$j = 5$	3	6.6	4.6	2	6.5	5	2	7.5	4.5
$j = 6$	3	7.3	4	0	–	–	0	–	–
Residual Error	1.0351e−04			1.0994e−04			1.3478e−04		
$\mathbf{J}(\mathbf{y}, \mathbf{u})$	4.9999e−07								

	$\omega = 10^{-5}$								
Step size ρ	$1\,10^{-5}$			$2\,10^{-5}$			$3\,10^{-5}$		
Iterations	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j = 3$	27	–	–	9	–	–	12	–	–
$j = 4$	10	7.2	6.5	3	7.6	7.3	4	8.25	7.25
$j = 5$	3	7	4.6	0	–	–	0	8	7.3
$j = 6$	3	7	4	2	9	6.5	3	–	–
Residual Error	1.0159e−04			1.2947e−04			5.3074e−05		
$\mathbf{J}(\mathbf{y}, \mathbf{u})$	4.9998e−06								

	$\omega = 10^{-3}$								
Step size ρ	$1\,10^{-3}$			$2\,10^{-3}$			$3\,10^{-3}$		
Iterations	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j = 3$	18	–	–	7	–	–	8	–	–
$j = 4$	10	7.4	6.7	3	7	6	4	8.25	7.75
$j = 5$	3	6.6	5	0	–	–	2	8	7
$j = 6$	3	7.3	4.3	0	–	–	2	8	6.5
Residual Error	1.0406e−04			1.4632e−04			4.9586e−05		
$\mathbf{J}(\mathbf{y}, \mathbf{u})$	4.9881e−04								

Table 5.30: Iteration numbers of the Inexact Gradient algorithm for varying values of ω with the B-type MRA using $\mathbf{D}_a$ and right hand sides of (P2) . The nested iteration starts with the QR decomposition for the saddle point matrices on level $j_0 = 3$, then the UzawaCD algorithm is used.

	$\omega = 10^{+3}$								
Step size ρ	0.5			1			1.5		
Iterations	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j = 3$	1	–	–	1	–	–	1	–	–
$j = 4$	2	0.5	0.5	1	1	1	2	1	0.5
$j = 5$	0	–	–	1	1	1	0	–	–
$j = 6$	2	1.5	0.5	1	3	1	2	1.5	0.5
Residual Error	7.9413e−05			1.3809e−07			7.8965e−05		
$\mathbf{J}(\mathbf{y}, \mathbf{u})$	2.0977e−01								

	$\omega = 10^{+6}$								
Step size ρ	0.5			1			1.5		
Iterations	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j = 3$	1	1	1	1	1	1	1	1	1
$j = 4$	1	1	1	1	1	1	1	1	1
$j = 5$	1	1	1	1	1	1	1	1	1
$j = 6$	1	1	1	1	1	1	1	1	1
Residual Error	1.5902e−07			3.4202e−10			1.5812e−07		
$\mathbf{J}(\mathbf{y}, \mathbf{u})$	2.0997e−01								

	$\omega = 10^{+9}$								
Step size ρ	0.5			1			1.5		
Iterations	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	k_j	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$
$j = 3$	1	1	1	1	1	1	1	1	1
$j = 4$	1	1	1	1	1	1	1	1	1
$j = 5$	1	1	1	1	1	1	1	1	1
$j = 6$	1	1	1	1	1	1	1	1	1
Residual Error	1.5902e−10			3.4250e−13			1.5812e−10		
$\mathbf{J}(\mathbf{y}, \mathbf{u})$	2.0997e−01								

Table 5.31: Iteration numbers of the Inexact Gradient algorithm for varying values of ω with the B-type MRA using $\mathbf{D}_a$ and right hand sides of (P2) . The nested iteration starts with the QR decomposition for the saddle point matrices on level $j_0 = 3$, then the UzawaCD algorithm is used. Here the overall complexity stays almost constant when choosing other step sizes.

5.3.2 Parameter Studies

We now present some results about the impact of our model parameters onto the solution states and control. Due to the vast number of possible combinations for tweaking the parameters a_0, ω, s, t and the Riesz and Trace operators we only show some selected results to present a certain anticipated effect. The numerical results concerning the Sobolev smoothness indices s, t are presented in Section 5.2.3.

Varying the Weight ω

The summands in the functional $\mathcal{J}(y,u)$ (4.3.1) can be weighted with respect to each other by the parameter ω. In applications, the value of ω is usually chosen to be very small, e.g., $\omega = 10^{-6}$. This means the emphasis is laid on state y being forced towards y_{Γ_Y} in the H^s-norm while the control can take on very high values in the H^t-norm. Thus, especially if the control is measured in a weak norm, i.e., $t < 1/2$, it can become very unsmooth, depending on the right hand side f in the PDE (4.3.2).
In Figure 5.15 we show this effect for Problem (P2) . As ω approaches zero, the control problem becomes ill-defined as the coupling between the primal and adjoint system is weakened. Since the control and the state are constant (cf. Section 5.2.3), we see an increase in the absolute value of the control $\mathbf{u}$ for $\omega \to 0$. The control $\mathbf{u}$ is no longer penalized in the functional $\mathbf{J}(\mathbf{y}, \mathbf{u})$ and can take on values far from zero. The state approaches the exact solution $\mathbf{y} \equiv 1$ of Problem 4.11 for $\mathbf{f} \equiv 1$ and $\mathbf{y}_{\Gamma_Y} = \mathbf{u} \equiv 1$. For $\omega \gg 1$ the control is penalized with a higher weight and must achieve lower values in the norm. In the limit, the state $\mathbf{y}$ is therefore determined with the control $\mathbf{u}$ being zero. As such, the state converges toward the solution of the unconstrained boundary value problem $\mathbf{L}(\mathbf{y}, \mathbf{p})^T = (\mathbf{f}, \mathbf{0})^T$, which can be seen in Figure 5.5.

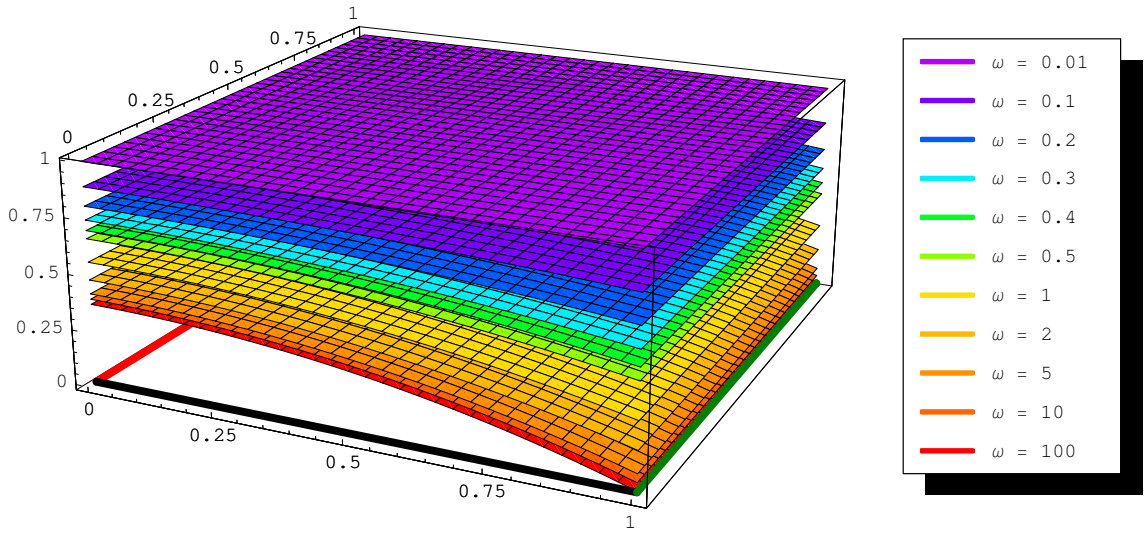


Figure 5.15: Dependence of the solution state $\mathbf{y}$ for Problem (P2) of the weighting parameter ω of the functional (4.3.1).

Another case study of the parameter ω for a different setup can be found in Figure 5.17.

Varying the Parameter a_0 in the PDE

The parameter a_0 in the constraint elliptic boundary value problem controls the impact of the mass term versus the Laplacian. To understand the effects when changing its value in the control problem, it is beneficial to understand the effects for the boundary value problem Problem 3.14. We focus on the case $f \equiv 1$ here, so the exact solution for the boundary value problem is given by (5.1.2).
Considering now the problem setup (P2) of our control problem, it is clear that for $a_0 \to \infty$ the state y must behave like $\sim \frac{1}{a_0}$. Thus, the absolute value and the curvature of the solution state y tends towards

zero. We depict this study in Figure 5.16 for $\omega = 1/5$ and $s = t = 1/2$ and the $\mathring{\mathring{\mathbf{R}}}_{H^s}$ Riesz operator. This means that the control $\mathbf{u}$ is allowed a $\sqrt{5}$ times higher norm (which, thus, determines this as its value). For $a_0 > 1$ the solution state $\mathbf{y}$ moves towards $\mathbf{y} \equiv 0$. In this case, the control $\mathbf{u}$ can attain higher values because of the value of ω and, thus, the surfaces bend upwards. For $a_0 < 1$, the solution takes on values of higher absolute value because the curvature is increasing and $\omega < 1$.
Obviously, this parameter also has an impact on the condition number of the stiffness matrix $\mathbf{A}_J$ and the saddle point operator $\mathbf{L}_J$. This means that the step size parameter ρ of the **Inexact Gradient** algorithm must be adapted accordingly to each a_0. We show the iteration number for this setup in Table 5.32. There the solutions are calculated with the B-type wavelets and $\mathbf{D}_a$ preconditioner up to $\frac{1}{100}2^{-j}$ in each iteration. The iteration is started with the QR solver on level $j = 3$ and then the Uzawa algorithm is used on the higher levels.
The results indicate that the algorithm converges quickest when the value of ρ satisfies $a_0/\rho \approx$ const. The upper bound for the step size ρ can be observed clearly; the algorithm does not converge any more. The lower bound is indicated by increasing iteration numbers, which ultimately should result in an infinite number of steps needed to obtain the solution as $\rho \to 0$.

The Trace Operators

So far, we have changed neither the observation boundary Γ_Y nor the control boundary Γ in our problem setup. Obviously, only two different choices are interesting to consider: We have already extensively addressed the case of boundaries at **opposite** sides of the domain. We are now going to investigate the case of **adjacent** boundaries.
We now choose for the observation boundary the south boundary $\Gamma_Y = \Gamma_{\mathrm{S}}$. Since control and observation share the point $(1, 0)$, the solution is forced to attain values close to the observation y_{Γ_Y} and close to 0 at the same time. This problem setup incorporates a dilemma if $y_{\Gamma_Y}(1, 0) \neq 0$.

We depict this case in Figure 5.17 for problem setup (P2) with the P-type wavelets using the $\mathring{\mathring{\mathbf{R}}}_{H^s}$ Riesz operator for $s = t = 1/2$. The weighting parameter ω is varied from 10^{-4} to 10^{1}. For very small values of ω the control can attain values far from zero and the solution converges towards $y \equiv 1$. For values $\omega \approx 1$ the above mentioned dilemma is visible in form of a spike at the point $(1, 0)$. Using higher values $\omega \gg 1$ results in an overweighting of the control and, thus, the solution state tends towards zero just as before (cf. Figure 5.15).

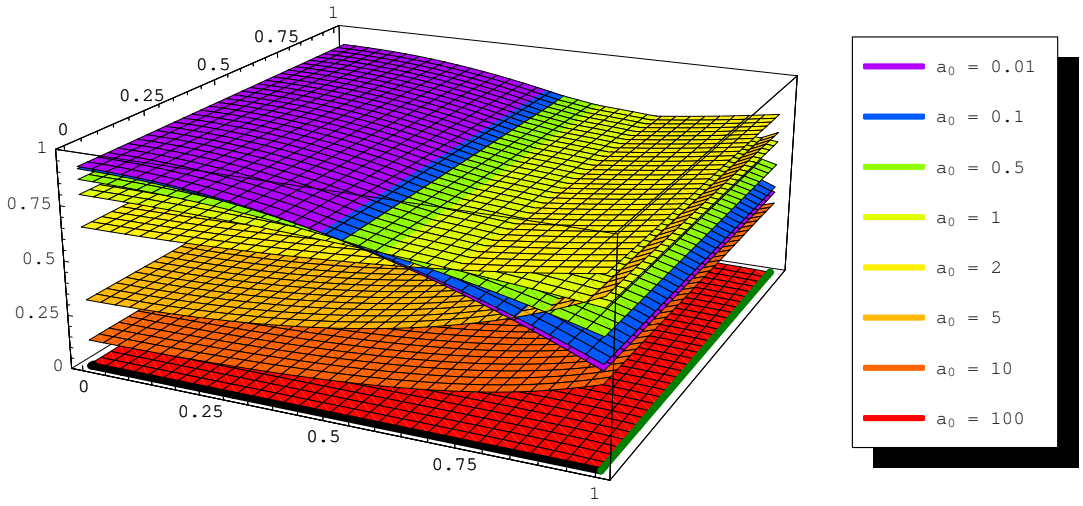


Figure 5.16: Solution for the control problem with $f = y_{\Gamma_Y} \equiv 1$ and $\omega = 1/5$ and $s = t = 1/2$. For $a_0 > 1$ the solution state y moves towards $y \equiv 0$. For $a_0 < 1$ the solution takes on values of higher absolute value because the curvature is increasing and $\omega < 1$.

		$a_0 = \frac{1}{100}$	$a_0 = \frac{1}{10}$	$a_0 = \frac{1}{2}$	$a_0 = 1$	$a_0 = 2$	$a_0 = 5$	$a_0 = 10$	$a_0 = 100$
$\rho = \frac{1}{200}$	$j = 4$	134	133	129	223	516	817	904	0
	$j = 5$	138	134	204	337	721	879	1010	401
	$j = 6$	139	980	682	1235	795	752	667	429
$\rho = \frac{1}{20}$	$j = 4$	12	13	9	22	52	81	90	0
	$j = 5$	13	14	20	34	72	87	100	40
	$j = 6$	15	93	24	136	79	75	67	43
$\rho = \frac{1}{4}$	$j = 4$	0	0	0	4	10	16	17	0
	$j = 5$	3	2	0	7	14	17	19	8
	$j = 6$	3	2	15	15	15	15	13	9
$\rho = \frac{1}{2}$	$j = 4$	–	–	–	0	0	0	0	0
	$j = 5$	–	–	–	5	0	0	8	0
	$j = 6$	–	–	–	6	0	9	7	5
$\rho = 1$	$j = 4$	–	–	–	–	–	2	2	2
	$j = 5$	–	–	–	–	–	2	2	0
	$j = 6$	–	–	–	–	–	2	2	2

Table 5.32: Iteration numbers for the **Inexact Gradient** algorithm for Problem (P2) with fixed $\omega = 1/5$, cf. Figure 5.16, on levels 4 through 6. Left out entries indicate that the algorithm did not converge, i.e., the residual error tends towards infinity.

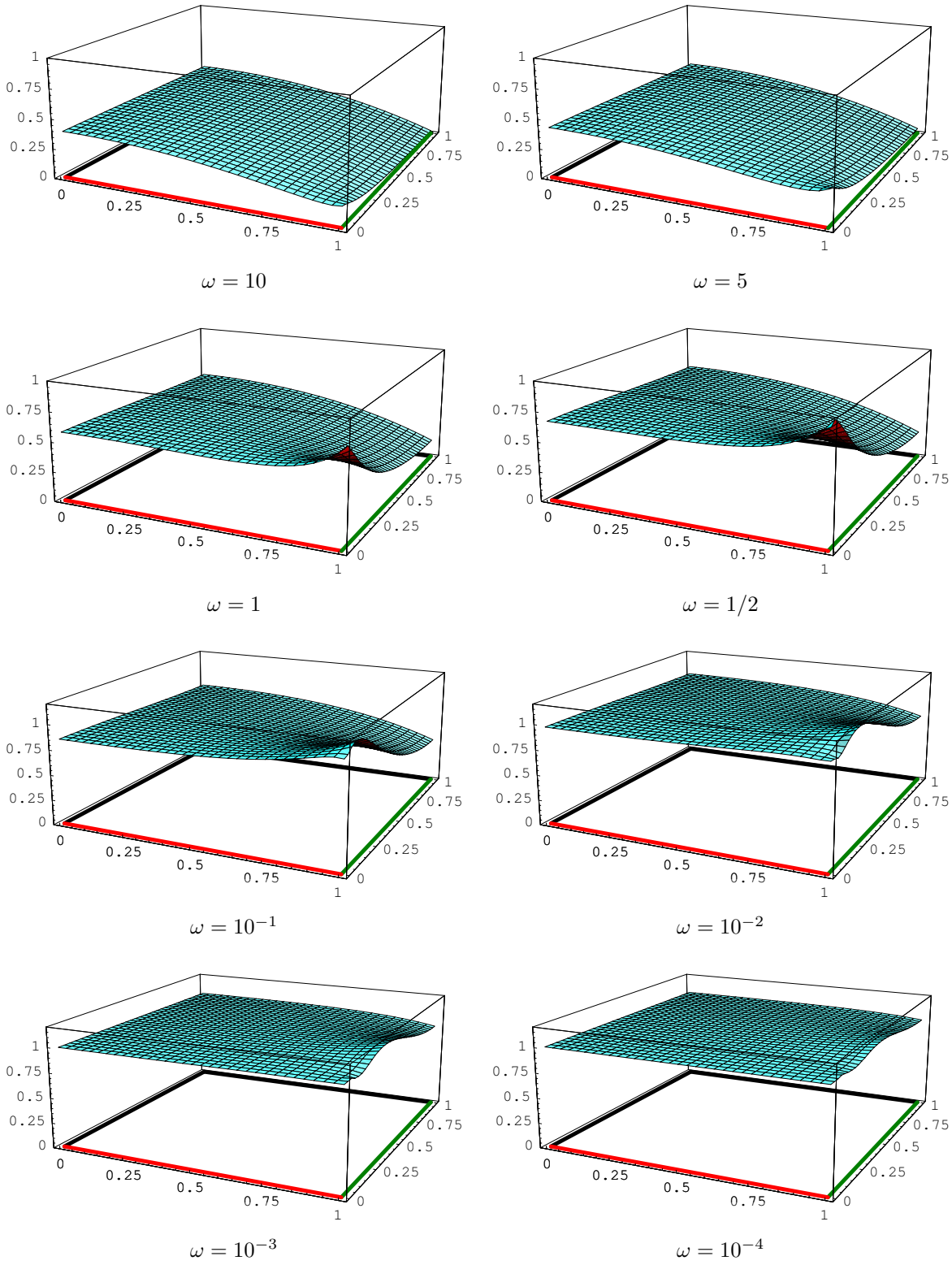


Figure 5.17: Solution states of Problem (P2) for adjacent control and observation boundaries and different values of ω. The observation boundary Γ_Y is pictured in red color and the control boundary Γ in green color.

5.3.3 Future Considerations

We are going to briefly touch a few topics which should be considered in the future.

Changes in the Geometry

The current setting allows a change of the domain by the fictitious domain approach outlined in Section 3.2.1. Another possibility for different setups include trace operators which act only on a part of the faces and edges of the domain $\square^n$, see Figure 5.18 for an example for the boundary value problem. Such constructions would better accommodate real life applications, e.g., modelling the heat transfer in a water basin where the heater can only cover a small part of the boundary area.

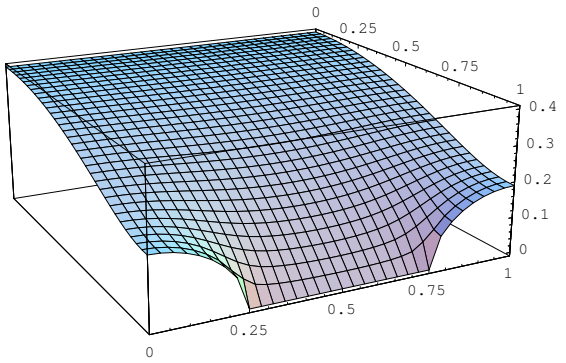


Figure 5.18: Solution to the boundary value problem with $f \equiv 1$ and $u \equiv 0$ on $\Gamma = (\frac{1}{4}, \frac{3}{4})$ calculated on level $j = 5$.

3D Problems

The complete setup of Section 4.3 was done independent of the dimension n. We focused primarily on the case $n = 2$ because this is the easiest to visualize. Now we shall consider problem (P2) in three dimensions and test the **Inexact Gradient** method on this problem. Recall from Section 5.2.2 that this has turned out to be the most efficient iterative solution method.
In Tables 5.33 and 5.34, we show the iteration results for the **Inexact Gradient** algorithm with $\rho = 0.75$ with the stopping criterion of $\frac{1}{100}h_J = \frac{1}{100}2^{-J}$ on every level. Since in 3D even on very small levels $j = 3, 4$ the resulting system matrices have $10^3 - 10^4$ number of rows and columns, we use only the UzawaCD algorithm in the nested iteration. Recall also that the number of unknowns and the complexity grows exponentially like 2^{3J}.

J	N_J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{J}(\mathbf{y}_J^{(k_J)}, \mathbf{u}_J^{(k_J)}) - \mathbf{J}(\mathbf{y}^*, \mathbf{u}^*)\|$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	$\frac{\text{Time}}{k_J}$
3	1.701	1.0129e−03	2.2286e−04	6	9.0	5.2	0.9s
4	10.693	2.0818e−04	1.9259e−05	5	8.4	0.4	6.2s
5	74.151	9.4074e−05	2.6159e−06	5	13.2	1.8	91.6s
6	561.925	6.3949e−05	1.9533e−06	5	12.6	1.0	710.6s
7	4.342.789	2.0094e−05	3.9370e−07	5	15.0	0.4	5964.0s

Table 5.33: Iteration histories for (P2) in 3D. Here the B-type MRA with $\mathbf{D}_a$ and the UzawaCD solver is utilized. The stopping criterion for the residual error is $10^{-2}h_J$.

The uniform boundedness of the iteration numbers is clearly demonstrated. The time required on every level to reduce the error by a factor of 2 grows proportional to 2^3 as expected. Here, as before, the orthogonal basis transformation with the P-type MRA exhibits better numerical properties when compared to the B-type wavelets with the diagonal of the stiffness matrix preconditioner.

J	N_J	$\|\mathbf{r}_J^{(k_J)}\|$	$\|\mathbf{J}(\mathbf{y}_J^{(k_J)}, \mathbf{u}_J^{(k_J)}) - \mathbf{J}(\mathbf{y}^*, \mathbf{u}^*)\|$	k_J	$\frac{\#\text{P-It}}{k_J}$	$\frac{\#\text{A-It}}{k_J}$	$\frac{\text{Time}}{k_J}$
4	10.693	1.6139e−04	1.6635e−05	8	3.1	1.5	0.5s
5	74.151	1.4530e−04	6.4201e−06	5	2.8	0.2	3.6s
6	561.925	9.8076e−05	1.4213e−06	5	2.8	0.2	32.8s
7	4.342.789	3.0278e−05	5.1901e−07	6	3.6	0.5	336.8s

Table 5.34: Iteration histories for (P2) in 3D. Here the P-Orth-type MRA with $\mathbf{D}_{\{O,a\}}$ and the UzawaCD solver is utilized. The stopping criterion for the residual error is $10^{-2}h_J$.

Nonsmooth Data and Adaptivity

Problem setup (P4) has been designed with an inherent flaw: the target observation y_{Γ_Y} is not in $H^{1/2}(\Gamma_Y)$. Therefore, the setup is ill-defined by construction. Specifically, piecewise constant but globally discontinuous functions are in the spaces $H^s, s < 1/2$. Since we allow coarser norms in our data fitting term in the control functional, this problem should manifest itself when we let the Sobolev smoothness parameter s tend to $1/2$. Of course we can still choose the parameter $s \in [0, 1]$ freely since the expansion $\mathbf{y}_{\Gamma_Y,J}^T \Psi_{\Gamma_Y}^s$ is only a **finite dimensional approximation** of y_{Γ_Y} and, thus, continuous, but the norm of the gradient will be very high.
The non-regularity of the problem data embodies itself during computations through higher iteration numbers of the Inexact Gradient algorithm, see Table 5.35. The results clearly show that the overall complexity of the algorithm increases, albeit only by a small constant factor over all levels. The problem is still solved in optimal complexity. The observation target and the traces of some computed solutions on the observation boundary are depicted in Figure 5.19.

Up to this point, throughout this thesis, we have only investigated linear techniques with full grids. This is a problem for high dimensional problems $n > 2$ because the computer power and memory requirements grow exponentially like 2^{nJ}. **Adaptivity** in the wavelet context for nonsmooth data or singularities in the domain can be achieved by best $N-$term approximations choosing N indices which minimize the approximation error. Such methods are much more involved and we will only show that our problem setup could benefit from adaptive methods for nonsmooth data.
Adaptivity is generally advised if the functions in our control problem are composed of smooth and rough parts. Then adaptive methods provide wavelet indices primarily for the rough parts of the functions. This

	$s = 0.0$	$s = 0.1$	$s = 0.2$	$s = 0.3$	$s = 0.4$	$s = 0.45$	$s = 0.475$
$j = 4$	16	17	19	20	22	22	23
$j = 5$	22	23	23	23	23	24	24
$j = 6$	12	12	12	12	13	13	13
$j = 7$	11	11	12	12	12	13	13

	$s = 0.5$	$s = 0.6$	$s = 0.7$	$s = 0.8$	$s = 0.9$	$s = 1.0$
$j = 4$	23	25	26	28	29	30
$j = 5$	24	24	25	25	26	27
$j = 6$	14	15	15	18	21	24
$j = 7$	12	15	14	15	16	30

Table 5.35: Iteration numbers for the (P4) with P-Orth-type MRA and the $\mathbf{D}_{\{O,a\}}$ preconditioner for increasing smoothness order in the data fitting term. Here we used $\omega = \frac{1}{4}$ and $\rho = \frac{1}{8}$ and the $\mathring{\mathbf{R}}_{H^s}$ Riesz operator. We use the **QR**-decomposition for the saddle point problems on level $j_0 = 4$ and the UzawaCD algorithm on the higher levels.

strategy can be extremely successful when dealing with control problems with elements of dual Sobolev spaces, see [11]. Although our solutions $y \in H^1(\Omega)$ and $u \in H^{1/2}(\Gamma)$ have minimal guaranteed smoothness, solutions of partial differential equations often exhibit singular phenomena like cusps, especially if the domain is not convex and has corners.
For our setup, we present the wavelet coefficients for a case where the observation has a spike in Figure 5.20. For such cases, adaptive methods are known to produce results up to a certain error with fewer computational complexity, see [6]. The theoretical background on adaptive methods for linear-quadratic elliptic control problems is given in [28] and experiments have been conducted in [11].

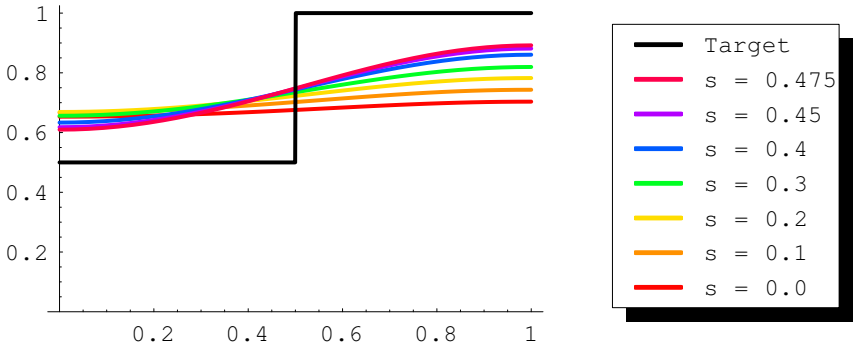


Figure 5.19: Target state and computed states on the observation boundary Γ_Y with the $\mathring{\mathring{\mathbf{R}}}_{H^s}$ Riesz operator. The asymmetry is due to the fact that the center generator on each level is being associated to the right part of the interval.

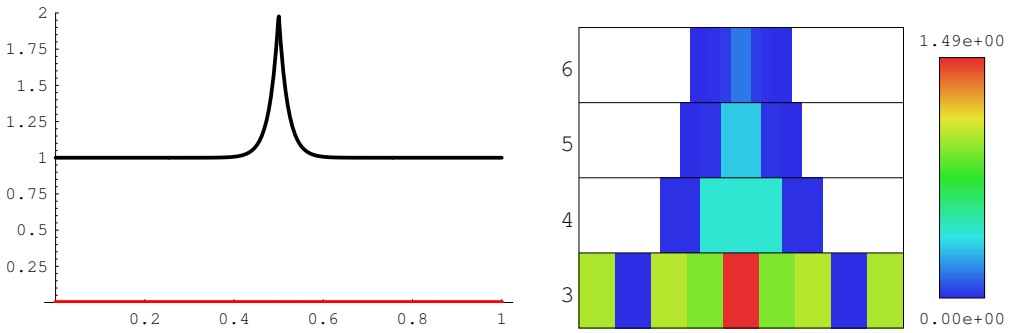


Figure 5.20: An observation $y_{\Gamma_Y}(x_2) = 1 + \exp(-50\,|x_2 - 1/2|)$ and its wavelet coefficients diagram.

6 Résumé and Outlook

In this thesis I have presented the numerical realization of the theoretical framework on elliptic PDE-constrained control problems given in [47] and several additional features and improvements for the practically relevant case of Dirichlet boundary control. It is demonstrated that the natural norm equivalences made available by wavelet methods are an excellent tool for the treatment of fractional Sobolev norms arising in this case of boundary control. I have shown that the amount of computational work in a nested iteration scheme is proportional to the total number of unknowns of the involved linear systems. That is, the condition numbers of the linear operators are uniformly bounded with absolute small constants and sufficiently accurate solutions can be obtained with iterative solvers within a uniformly bounded small absolute number of iterations. The **Inexact Gradient** algorithm 4.25 proves to be numerically preferable to the **All-In-One Solver** (4.4.1).

In summary, the theoretically predicted convergence of the inexact gradient scheme is confirmed for all setups for which the problem is well-posed. We observe uniformly bounded iteration steps over all levels of resolution when the step parameter ρ is chosen within certain bounds predicted by the underlying theory.

I have also developed effective, yet inexpensive, basis transformations aimed at further reducing the condition numbers of the stiffness matrix and operators constructed around it. The actual benefit depends highly on the characteristics of the wavelets itself, but their principle can be applied to most wavelet constructions in the same fashion.

Furthermore, I have shown the numerical effect of introducing **Riesz operators** into the problem statement. Although they are not widely regarded as a central ingredient in the modelling process of a particular problem, I have given an example where not using Riesz operators leads to wrong results. By the means of the Riesz operators the standard norm equivalences between the ℓ_2-norm of weighted wavelet coefficients and Sobolev spaces $H^s, s \in \mathbb{R}$, can be sharpened significantly. Riesz operators should therefore always be utilized if a particularly precise evaluation of a Sobolev norm is needed.

Lastly, I have presented numerical experiments which show the interrelation of the various problem parameters. Due to the vast amount of possible combinations of the parameter set, it was not possible to test all parameter variations and their effects on the modelling and on the numerical schemes. Rather I set the focus on the investigation of some specific exemplary cases which exhibit a certain (anticipated) effect. Specifically, the smoothness parameters s and t can induce a certain change in the state and control which cannot be achieved with fixed norm indices varying only the weighting parameter ω.

There are still some instances to improve the results given in this thesis. First, the **Inexact Gradient** algorithm could be replaced by an **Inexact CG** algorithm which would relegate the need to explicitly choose values for the step size parameter ρ. This should result in a general performance gain in the nested iteration scheme.

Secondly, wavelet constructions can further be improved for their use in numerical analysis. It was demonstrated in Section 5 how a small specialized basis transformation can have a tremendous reduction effect on the condition numbers of the involved operators. Also, the tensor product construction of Section 2.4 is not ideal in the sense that degrees of freedom are unnecessarily distributed evenly on the whole domain Ω. **Adaptive wavelet methods** as proposed in [18, 28] would only insert wavelet coefficients locally to obtain the solution within a desired accuracy tolerance. Meanwhile, the computational work stays proportional to the number of essential, e.g. non-zero, wavelet coefficients which can be lower by several orders of magnitude than the cardinality of unknowns in the uniform discretizations used in this thesis. The theory of adaptive wavelet discretizations has been brought forward to the matter of linear-quadratic, elliptic control problems in [28]; a first adaptive implementation was carried out in [11].

The **Riesz operator** constructions should generally be of further interest for numerical analysis. As it is not possible to create one (exact) Riesz operator for H^s, additional constructions should be carried out and tested for numerical applicability.

A Software Documentation

A.1 General Information

The software was written completely in C++ . It relies extensively on the **Standard Template Library** and uses the **Matrix Template Library** [54] as a backend for memory management for matrix and vector operations. Additionally, we used the very neat `Singleton` and `Factory` classes of the **Loki** library, available from [51], and the threading code was implemented using the **Common-C++** (see [15]) library. The source code of these libraries can be found at the websites referenced in the bibliography.
The software was developed and executed using several flavors of the **Linux** operating system. Although all source code is available, it might not compile or run on other Unix® and Microsoft® Windows® systems.
The software is distributed under a controlled open source license. Access to the software is given after the license agreement has been accepted in written form. For more information, see

`http://www.iam.uni-bonn.de/~pabel/` .

I would like to emphasize that this software is not meant to be a general wavelet code library nor is this an official release. It is just distributed "as is" without any warranty.

A.2 Source Code Structure

The complete program is separated into several subdirectories and many files. The directory structure is as follows

`./`	main directory; holds the configure scripts;
`./src`	source root directory; holds the problem classes which implement the system matrices' multiplication routines;
`./src/diagassemble`	source code of a helper program which assembles the diagonal of the stiffness matrix and saves it to disk for later use;
`./src/loki`	the **Loki** library source files. These are included since they are not common among the system libraries of a **Linux** distribution. The library consists almost completely of template classes and is compiled in where needed. The documentation to this library can be found in the excellent book [2];
`./src/problems`	here the source code of anything related to a problem's implementation can be found. This includes right hand side objects, solvers and action object definitions;
`./src/psgen`	source code of a helper program which visualizes the absolute values of the wavelet coefficients in a nice diagram;
`./src/wavelets`	holds all low level classes like the implementation of the wavelet transforms and operators like the stiffness matrix and the diagonal preconditioners. The entire source is split into several C++ namespaces to ease management of the different MRA types;

The results published in this paper were achieved by compiling the program with `gcc-3.4.x` and the following extra compiler options :

```
-O3 -g0 -fno-exceptions -funroll-loops -frerun-loop-opt
```

On the following pages, we show the major classes of the software modules and describe their functions and the most important parts of their interfaces. This documentation is not meant to be complete. It serves to give any interested reader an introduction into the structure of the program to allow for dealing with the software.

Helper Classes (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets`	
`wavelets.h`	forward declarations and global typedefs
`typedef {double,float} FP_TYPE`	defines the global floating point data type
`typedef mtl::dense1D<FP_TYPE> Vektor`	a normal vector type
`typedef mtl::matrix<FP_TYPE>::type Matrix`	a general rectangle matrix type
`typedef unsigned int length_t`	type of variables denoting lengths (of vectors), (refinement) levels and (spatial) dimensions
`typedef unsigned int level_t`	
`typedef unsigned int dim_t`	
`typedef MultiIndex<length_t> lengths_t`	types for saving the levels and lengths of a `TensorVektor` and an `ExtendedVektor`
`typedef MultiIndex<level_t> levels_t`	
`typedef MultiIndex<levels_t> levels_array_t`	
`enum TRANSFORM_MODE {NORMAL, TRANSPOSED}`	flag for matrix multiplication
`enum WAVELET_PRECOND {NONE, FIRST, LAST}`	flag for signaling the fast wavelet transform to apply the diagonal preconditioner
`enum TRANSFORM_ORDER {ASCENDING, DESCENDING}`	flag for order of fast wavelet transform
`typedef TensorVektorT<FP_TYPE> TensorVektor`	definition of the `TensorVektor` class
`tensorvektor.h`	
`template<typename T> class TensorVektorT`	class that holds a `Vektor` instance and provides means (accessors, iterators) for tensorized matrix and vector computations
`template<typename T> class SubTensorVektorT`	class that holds only pointers to its data but is otherwise as functional as a `TensorVektorT<T>`
`extendedvektor.h`	
`class ExtendedVektor`	holds several `TensorVektor`s and provides transparent access for all numeric and administrative functions; an `ExtendedVektor` is a **system vector**, i.e. $\mathbf{U}_J$ and $\mathbf{F}_J$ in (4.2.61).
`multiindex.h`	
`template<typename T> class MultiIndex`	data structure to store a **multi-index** of type `T` and provide functions like $\lvert\cdot\rvert$
`template<typename NumT, typename ConT> class CompressedVektor`	class for vectors whose values are almost identical for a contiguous range; only stores the value once together with the number of recurrences; used for $\mathbf{D}_a$ of (3.2.34)
`timediff.h`	
`class TimeDiff`	class for recording the user and system execution times of the current process and all its' children
`class TimeDiffSimple`	
`vektorproxy.h`	
`public VektorProxy`	proxy class for `TensorVektor`; will return an object only when demanded explicitly
`class InstanceVektor : public VektorProxy`	return a copy of a previously given instance;
`class FileVektor : public VektorProxy`	data read from a file; or will stop execution
`class NoVektor : public VectorProxy`	upon requesting the data

Tensor Functions and Operators (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets`	
`tensorfunction.h`	
`struct EmptyType {}`	an empty class; used as placeholder
`struct LowerDimensionsFirst` `struct HigherDimensionsFirst`	policy objects which control in what order the tensorfunction applies to the data

`template<typename`
`payload = EmptyType`
`typename tensor = SubTensorVektor`
`typename dimensionPolicy =`
`LowerDimensionsFirst>`
`class TensorFunction`
	the base class of tensorized functions, e.g. a function that is aware of the tensorized structure of the data and can make use of this information; it can be configured by the template parameters, i.e. for the information it needs (the `payload`); `TensorFunction`s can be tensorized and added, thus easily imitating the structure (2.4.15)

`void operator()( tensor* STV,` `payload = payload() )`	the apply method

`template<typename payload = EmptyType`
`typename tensor = SubTensorVektor`
`typename dimensionPolicy =`
`LowerDimensionsFirst>`
`class WavTensorFunction :`
`public TensorFunction<payload, tensor,`
`dimensionPolicy>`
	derived class specifically designed for use with the `Wavelet` class; examples include the **fast wavelet transform** `FWT` and basis transformations matrices

`wavelet.h`	
`struct FWTPayload`	payload type for the FWT; this includes the values for the levels and the mode
`class FWT : public WavTensorFunction<` `struct FWTPayload>`	base class for all wavelet transform classes

`void operator()(`
`SubTensorVektor* STV,`
`TRANSFORM_ORDER o,`
`TRANSFORM_MODE m,`
`FLAG f )`
	this apply operator takes as arguments the vector to work on, the order (de- or ascending), the mode and a flag; this operator is also used for prolongation and restriction in the sense of (2.1.11) by supplying the correct arguments

`struct BasisTraFoPayload`	payload type for basis transformations of type shown in Section 2.3.3

`class BasisTraFo : public`
`WavTensorFunction<`
`struct BasisTraFoPayload>`
	base class for all basis transformation classes

`void operator()(`
`SubTensorVektor* STV,`
`level_t j,`
`TRANSFORM_MODE m )`
	the arguments are the vector to work on, the level of the transformation and the mode

`struct OperatorPayload`	holds only a `TRANSFORM_MODE` value
`class Operator : public TensorFunction<` `struct OperatorPayload>`	base class for operators (i.e. the single-scale mass matrix)

`void operator()(`
`SubTensorVektor* STV,`
`TRANSFORM_MODE m )`
	this apply method only takes an extra mode argument to be put in the payload

Operators and Preconditioners (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets`	
`operators.h`	
`class SpecialOutputPolicy`	dimension policy object for `OutputToFile` class
`class OutputToFilePayload`	payload type which holds information about the file and the coordinates
`class OutputToFile : public TensorFunction< OutputToFilePayload, SubTensorVektor, SpecialOutputPolicy>`	an object that writes a `TensorVektor` to file; writes one entry of the form `x y z ... value` per line to file; data of this form can be displayed easily by programs like **gnuplot** and **Mathematica**
`wavelet.h`	
`class DiagPrecondPayload`	holds data that is necessary for execution of a diagonal preconditioner, i.e. the exponent value
`class DiagPrecond`	base class for all implementations of diagonal preconditioners like $\mathbf{D}_a$ and $\mathbf{D}_1$
`void operator()( SubTensorVektor* STV, double o, FP_TYPE f = 1. )`	the arguments are the data to work on, the order (exponent) and optionally a constant multiplicative factor
`preconditioners.h`	holds the implementations of all preconditioners
`class DiagPowerPrecond : public DiagPrecond`	intermediate base class for all implementations of diagonal preconditioners based on power of 2, e.g. $\mathbf{D}_1$
`class DiagP2Precond : public DiagPrecond`	implementation of $\mathbf{D}_1$
`class DiagP2ModPrecond : public DiagPrecond`	modified implementation of $\mathbf{D}_1$, e.g. scaling mimics the diagonal of stiffness matrix preconditioner
`class DiagSMPrecond : public DiagPrecond`	implementation of $\mathbf{D}_a$; the actual values are read from a file which was created by the `DiagAssemble` target and saved in the `CompressedVektor` format this class is also used in the construction process of the `CompressedVektor` data and for saving it to disk
`class HybridPrecond : public DiagPrecond`	hybrid preconditioning implements different preconditioners for data of different spatial dimensions, i.e. $\mathbf{D}_1$ for all 1D data and $\mathbf{D}_a$ for all 2D data; combinations can lead to different condition numbers for example for the operator $\mathbf{L}_J$

Riesz and Trace Operators (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets`	
`rieszoperator.h`	
`class RieszOperator`	interface class for all Riesz operator implementations
`void setOrder(double s)`	sets the order of Riesz operator, e.g. the desired **Sobolev space** regularity
`void operator()( TensorVektor* TV ) const`	computes $\mathbf{R}$ `TV` and saves the result in the same vector
`void Solve( TensorVektor* TV ) const`	computes $\mathbf{R}^{-1}$ `TV` and saves the result in the same vector
`class Identity : public RieszOperator`	this class performs no operations in its' methods; acts as if no Riesz operator was used
`class GramMatrix : public RieszOperator`	applies the **mass matrix**, see (2.2.49); works as the L_2 Riesz operator
`class PowerDiagScale : public RieszOperator`	implements the Riesz operator $\widehat{\mathbf{R}}_{H^s}$, see (5.2.8)
`class NormedPowerDiagScale : public RieszOperator`	implements the Riesz operator $\mathring{\widehat{\mathbf{R}}}_{H^s}$, see (2.2.44)
`traceoperator.h`	implementation of the trace operators
`class TraceOperator`	interface class for all trace operator objects
`void operator()( TensorVektor* output, const TensorVektor* input, TRANSFORM_MODE T )`	apply operator which calculates `output` $= B^{(T)}$ `input`

`class East : public TraceOperator`
`class West : public TraceOperator`
`class North : public TraceOperator`
`class South : public TraceOperator`
	implementation of the trace operators onto the edges of the unity cube; see (3.2.22) and (3.2.23)

Wavelet Class Interface (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets`	
`wavelet.h`	
`class Wavelet`	interface class for all wavelet flavors
`level_t getMinimumLevel() const`	returns j_0
`length_t singleScaleQuantity( level_t j) const`	returns $\#\Delta_j$
`length_t multiScaleQuantity( level_t j) const`	returns $\#\nabla_j$
`length_t dimSingleScaleQuantity( const levels_t& j) const`	calculates $\#\Delta_{\mathbf{j}}^{\square}$, see (2.4.1), for the **multi-index j**
`length_t dimMultiScaleQuantity( const levels_t& j) const`	calculates $\#\Delta_{\mathbf{j}+1}^{\square} - \#\Delta_{\mathbf{j}}^{\square}$
`void direct(TensorVektor* output, const TensorVektor* input, TRANSFORM_ORDER O = ASCENDING, TRANSFORM_MODE M = NORMAL, WAVELET_PRECOND P = FIRST, double s = 1.) const`	interface method to the `FWT` and `DiagPrecond` classes; this method applies the standard diagonal scaling and the fast wavelet transform in the specified modes; default arguments calculate `output` $= \mathbf{TD}^{-1}$ `input` regardless of the dimension of the input data
`void inverse(TensorVektor* output, const TensorVektor* input, TRANSFORM_ORDER O = DESCENDING, TRANSFORM_MODE M = TRANSPOSED, WAVELET_PRECOND P = LAST, double s = 1.) const`	analogon for the inverse FWT: `output` $= \mathbf{D}^1\mathbf{T}^{-T}$ `input` $= \mathbf{D}^1\widetilde{\mathbf{T}}$ `input`
`void applyStiffnessMatrix( TensorVektor* TV) const`	when called, will apply the **single-scale stiffness matrix** (as shown in (2.4.16)); the correct structure of **TensorFunction**s is automatically created for the dimension of the input data
`void applyMassMatrix( TensorVektor* TV) const`	computes the single-scale product of the mass matrices, cf. (2.4.11)
`void applyInverseMassMatrix( TensorVektor* TV) const`	computes the inverse of the above function
`void prolongate(TensorVektor* TV) const`	prolongates the vector (given in wavelet coordinates) to the next levels by inserting empty data at the correct memory addresses
`void restrict(TensorVektor* TV) const`	restricts the vector (given in wavelet coordinates) by deleting data
`void exportValues(TensorVektor* V) const`	transfers the vector from the current single-scale basis to the nodal basis
`void importValues(TensorVektor* V) const`	imports a vector of coefficients given in the nodal basis into the single-scale basis

Wavelet Implementations (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets::Wavlet_P_24_3_CDFBASE`	
`wavelet_p_24_3_cdfbase.h`	holds all classes for the implementation of the wavelets described in Section 2.3.2; the different "flavors"(B and DKU) only differ in the definition of four constants, which are saved in the files listed below
`class Wavelet_P_24_3_cdfbase : public Wavelet`	implementation of the `Wavelet` interface for the Wavelet construction of [19]; uses the classes listed below f
`class SM : public Operator`	class implements the 1D single-scale stiffness matrix
`class MM : public Operator`	class implements the 1D single-scale mass matrix
`class iMM : public Operator`	class implements the 1D single-scale inverse of the mass matrix
`class dFWT : public FWT` `class iFWT : public FWT`	implementation of the fast wavelet transform and its inverse;
`class TransformToNodalBasis : public BasisTraFo` `class TransformFromNodalBasis : public BasisTraFo`	basis transformation objects; the primal generator basis is not the nodal basis here; this simple basis transformation performs the changes; also used in `importValues()/exportValues()`
`class dGeneratorOrthogonalizer : public BasisTraFo` `class iGeneratorOrthogonalizer : public BasisTraFo`	these classes implement the orthogonalization of the generator basis with respect to the operator $-\Delta + I$ on the minimum level j_0, see (2.3.44)
`class dTransC : public BasisTraFo` `class iTransC : public BasisTraFo`	the SVD based basis transformation, see (2.3.37)
`wavelet_p_24_3_dku.h` `class Wavelet_P_24_3_dku : public Wavelet_P_24_3_cdfbase`	
`wavelet_p_24_3_cb.h`	
Namespace `Wavelets::Wavlet_P_24_3_CQBASE`	
`wavelet_p_24_3_cqbase.h` `class Wavelet_P_24_3_cqbase : public Wavelet`	these files hold the implementation of the wavelet construction of Miriam Primbs, see [56]
`wavelet_p_24_4_mp.h` `class Wavelet_P_24_4_mp : public Wavelet_P_24_3_cqbase`	

Problem and Action Descriptions (Path `./src/problems`)	
Files/Classes/Functions	Comments
Namespace `Wavelets::Problems`	

`actions.h`
`class Action`
`void operator() (LinearProblem* WP)`
	interface definition for all action implementations; `Action`s implement algorithms that can work on an instantiation of the `LinearProblem` class

`class NoOp : public Action`	Executes nothing, program will exit normally
`class Solve : public Action`	when applied, solves the equation described by the problem with an exact or iterative solver
`class NestedIteration : public Action`	uses **nested-iteration** to solve the system of equations; a different solver (i.e. a different `Solve` object) can be used for any step of the iteration
`class Assemble : public Action`	assembles the system matrix in a `Matrix` object by multiplying all unit vectors; can save the resulting matrix to file (either as values, in a Matlab spy mode or Mathematica compatible format)
`class Condition : public Action`	computes the spectral condition of the system matrix by power iteration and inverse power iteration (optionally with spectral shifting)
`class EigenValues : public Action`	computes the eigenvalues of a symmetric problem matrix by assembling the matrix with `Assemble` and reducing it to a tri-diagonal form then applying **QR**-decomposition and multiplicative disposition

Problem Implementation Descriptions (Path `./src/problems`)	
Files/Classes/Functions	Comments
Namespace `Wavelets::Problems`	
`implbase.h`	
`class Problem`	base class for all problems, linear or not
`void exec()`	must be implemented in the derived class and will be called upon execution
`class LinearProblem : public Problem`	base class describing the interface for all linear problems. derived classes can do command line option parsing, but have to construct a `LinearImplementation` which does the real work
`LinearProblemImplementation* Impl()`	the derived class must supply an instance of an `LinearImplementation` derived class; this object will be used by the `Action` objects on page 146
`class LinearProblemImplementation`	base class for linear problems. A linear problem is a system equation like (4.4.1) or (3.1.35) of which the system matrix must be implemented in the corresponding derived class
`void operator()( ExtendedVektor* output, const ExtendedVektor* input, TRANSFORM_MODE mode ) const`	method that must implement the system matrix multiplication (and the transposed multiplication for non-symmetric matrices)
`class SaddlePointProblemImplementation : public LinearProblemImplementation`	this class specifies an interface that is necessary to make use of the **Uzawa** algorithms (Algorithm 3.26 and Algorithm 3.27)
`void applyStiffnessMatrix()( ExtendedVektor* output, const ExtendedVektor* input ) const`	method that must implement the stiffness matrix multiplication
`void applyLagrangeMatrix()( ExtendedVektor* output, const ExtendedVektor* input, TRANSFORM_MODE mode ) const`	method that must implement the Lagrange matrix multiplication
`class StiffnessMatrixProblem : public LinearProblem`	during the execution of the Algorithm 3.26 algorithms, this class is used to handle the multiplication with the inverse of the stiffness matrix
`class StiffnessMatrixProblemImplementation : public LinearProblemImplementation`	implements the inversion of the stiffness matrix

Right Hand Sides (Path `./src/problems`)	
Files/Classes/Functions	Comments
Namespace `Wavelets::Problems`	
`rhs.h`	
`class RightHandSide`	objects of this kind can fill a `TensorVektor` for use as a right hand side function
`void create( const Wavelet* W, TensorVektor* TV ) const`	main function which fills the `TensorVektor` object and transforms the vector into the wavelet expansion of the specified `Wavelet` instance; will use `Wavelet::importValues()` and the `Wavelet::direct()` interface, if necessary
`class NoRHS : public RightHandSide`	acts as a placeholder object until a proper `RightHandSide` object is available; will stop the program, if asked to serve data
`class GivenVektor : public RightHandSide`	will save `TensorVektor` object and return a copy of it upon request
`class NumericFunction : public RightHandSide`	class that holds a `Function` object and will use this to fill a vector with the values of that function
`class Function`	interface definition for an object that represents a real-valued function of the unity line, square, etc...
`void create(TensorVektor*) const`	method that must implement the creation of a `TensorVektor` object with the expansion coefficient of a real-valued function expanded in the nodal basis
`class Zero : public Function`	fills the vector with zeros
`class Const : public Function`	fills the whole vector with a constant value, works for any dimension; implements f and y_{Γ_Y} for Problem (P2)
`class Linear : public Function`	implements a function of the form $m\,x + b$
`class Cos2Y : public Function`	implements the function $\cos(2\pi y)$, used in y_{Γ_Y} of Problem (P1) ; can only be used for 1D
`class CosX2CosY : public Function`	implements the function $\frac{1}{2}(1 + 4\pi^2 + (1 + 5\pi^2)\cos(\pi x))\cos(2\pi y)$, used as f of Problem (P1) ; can only be used for 2D
`class ExpAbs : public Function`	implements functions of the form $c + e^{-f\lvert y-\frac{1}{2}\rvert}$, and tensorproducts thereof; used for both y_{Γ_Y} and f in Problem (P3)
`class Skip : public Function`	implementation of a discontinuous, but otherwise constant, function

Solvers (Path `./src/wavelets`)	
Files/Classes/Functions	Comments
Namespace `Wavelets::Solver`	
`solvers.h`	holds several algorithms for solving linear equations
`class ExactSolver`	exact solvers are only used on very small levels since they require large amounts of memory (for the system matrix) and time
`void operator()( Matrix* M, ExtendedVektor* lsg, ExtendedVektor* rhs )`	algorithm will compute the solution of the equation `M x = rhs` and save the result in the vector `lsg`
`class LU : public ExactSolver`	the **LU** decomposition
`class QR : public ExactSolver`	the **QR** decomposition
`class IterativeSolver`	base class for all iterative solvers, which manages statistics (i.e. changes in the norm of the residual in course of the calculations)
`class LinearProblemSolver : public IterativeSolver`	base class for all solvers specifically designed to solve `LinearProblem`s
`void operator()( LinearProblem* WP, ExtendedVektor* lsg, ExtendedVektor* rhs, FP_TYPE tol, VektorProxy* sol )`	algorithm will compute the solution of the equation `M x = rhs` up to the prescribed tolerance `tol` and save this in the vector `lsg`; during execution, the computed solution can be compared to the (exact) solution `sol`, if it was supplied
`class CG : public LinearProblemSolver`	implementation of Algorithm 3.25
`class CG_DH : public LinearProblemSolver`	a different implementation of Algorithm 3.25, written according to [34]
`class SaddlePointProblemSolver : public IterativeSolver`	base class for all solvers specifically designed to solve `SaddlePointProblem`s
`void operator()( SaddlePointProblem* SPP, ExtendedVektor* lsg, ExtendedVektor* rhs, FP_TYPE tol, VektorProxy* sol )`	the algorithm will compute the solution of the equation `M x = rhs` up to the prescribed tolerance `tol` and saves this in the vector `lsg`; during execution, the computed solution can be compared to the (exact) solution `sol`, if it was supplied
`class Uzawa : public SaddlePointProblemSolver`	implementation of Algorithm 3.26
`class UzawaCD : public SaddlePointProblemSolver`	implementation of Algorithm 3.27

Problem Descriptions (Path `./src`)	
Files/Classes/Functions	Comments
Namespace Global ::	
`controlproblem.h`	
`class ControlProblem : public LinearProblem`	implementation of the problem described in Section 4.3
`CPImpl : public LinearProblem`	base class for the following real implementations; does storage tasks
`class AllInOne : public CPImpl`	implementation of the **All-In-One Solver** method of Section 4.4.1; the resulting matrix multiplication of the matrix $\mathbf{N}_J$ (4.4.1) is swapped out and split up into the `MatMultNUpper` and `-Lower` classes to support multithreading
`class MatMultNBase`	
`class MatMultNLower : public MatMultNBase`	
`class MatMultNUpper : public MatMultNBase`	
`class CPThread : public ost::Thread`	
`class CoupledSP : public CPImpl`	implementation of the problem of Section 4.3 using the coupled-saddlepointproblem approach of Section 4.4.2; the systems (4.4.7) and (4.4.8) are implemented as independent `SaddlePointProblems` and can be configured differently, e.g. for different solvers
`class SPBase : public SaddlePointProblemImplementation`	
`class PrimalSP : public SPBase`	
`class AdjointSP : public SPBase`	
`class CSPSolver : public IterativeSolver`	implements the **Inexact Gradient** algorithm, Algorithm 4.25
`bdryvalueprob.h`	
`class BoundaryValueProblem : public LinearProblem`	implementation of the problem outlined in Section 3.2.3; used for testing purposes
`class BVPImpl : public SaddlePointProblemImplementation`	implementation of the system matrix (3.1.26)
`class SchurComplement : public LinearProblemImplementation`	implementation of the **Schur complement** (3.2.15)
`testproblem.h`	
`class TestProblem`	has no special meaning, incorporates only test and trial code

Problem Descriptions (Path `./src/diagassemble`)	
Files/Classes/Functions	Comments
Namespace Global ::	
`diagassemble.h`	
`class DiagAssemble`	assembles the diagonal of the stiffness matrix $\mathbf{D}_a$; this can either be done sequentially by using all unity vectors or by honoring the repetitive structure of the wavelet discretization and the tensorproduct construction; execution can be carried out in multiple processes simultaneously
`class DASig`	
`class DAThread : public ost::Thread`	
`assembler.h`	
`normal.h`	
`signature.h`	

A.3 Command Line Options

A.3.1 controlproblem

The general structure is as follows:

```
./controlproblem <ProblemName> --op <Action> --level <I,J> --dim <N> [ other options ]
```

Irregular command line options will be ignored silently. The `ProblemName` parameter can be the name of the class or one of the following abbreviations:

`BoundaryValueProblem`	`BVP`
`ControlProblem`	`CP`

The mandatory command line options are :

`--level` comma separated list of positive integers

`--dim` positive integer, most likely 1,2 or 3

`--op` name of a class derived from `class Action` listed on page 146

Depending on the problem and the operation, many more command line options are recognized.

Common options:	`--normal`	apply system matrix twice on each application
	`--load-start`	`ilename` of starting vector for solvers
	`--no-fwt`	do not use wavelet transform and preconditioners
MRA options	`--mra`	type of the MRA to use, see Section 5.1
	`--prec`	`Preconditioner`, e.g., `P2`, `SMD`
	`--SMDiag-path`	path to pre-calculated diagonals of the stiffness matrix
	`--btf`	`BasisTransformation`, e.g. `Orth`, `SVD`

`ControlProblem`	`--omega`	value of the parameter ω, default 1
	`--f`	a name of a `Right Hand Side` object, see page 148
	`--y`	specifies y_{Γ_Y}
	`--T`	order of the norm on the observation bdry, default 1/2
	`--ObsB`	name of observation boundary, default "West"
	`--B`	order of the norm on the control boundary, default 1/2
	`--CoB`	name of control boundary, default "East"
	`--CoBRieszOp`	type of the Riesz operator to use for the control
	`--ObsBRieszOp`	type of the Riesz operator to use for the observation
	`--RieszOp`	type of the Riesz operator to use for both of the above
	`--CSP`	enables coupled saddlepoint problem mode
	the All-In-One mode uses the following options:	
	`--symmetric`	use symmetric form $\mathbf{N}'_J$
	`--multiThread`	enable multithreading
	the CSP mode uses the following options:	
	`--rho`	value of the parameter ρ, default 1/2
	`--P-*`	these options will be used for the primal system
	`--A-*`	these options will be used for the adjoint system
`BoundaryValue-Problem`	`--f`	a name of a `RightHandSide` object, see page 148
	`--g`	a name of a `RightHandSide` object, see page 148
	`--order`	order of the border preconditioner, default 1/2
	`--side`	`TraceOperator`: East, West, North, South
	`--Schur`	calculates the **Schur complement** $\mathbf{S}_J$ instead
	`--SMP-Solver`	`solver` for the calculation of $\mathbf{A}_J^{-1}$

The action objects make use of the following parameters :

`Solve`	`--quiet`	shorten output of messages
	`--verbose`	opposite of `--quiet`, takes precedence
	`--no-stats`	do not compute statistics (iterative solvers only)
	`--yes-stats`	opposite of `--no-stats`, takes precedence
	`--max-steps`	`int`: maximum number of iterations (iterative solvers only)
	`--save-RAM`	save memory (by writing unused data to disk)
	`--output`	`filename`: solution vector in nodal basis
	`--no-output`	do not save solution vector
	`--statsfile`	`filename`: output file for statistics
	`--solver`	`name`: name of the solver to use
	`--exact-sol`	`filename`: loads the exact solution from specified file
	`--tmp-sol`	save solution in wavelet discretization to file
`NestedIteration`	`--ni`	`int`: nested-iteration steps to conduct
	`--solvers`	`names`: comma separated list of solvers applied sequentially
	`--output`	`filename`: solution vector in nodal basis (of last iteration)
	`--no-output`	do not save solution vectors
	`--statsfile`	`filename`: statistics
	`--prefix`	`path` common prefix for all file operations
	`--load-sols`	loads the exact solutions from specified files
	`--tmp-sols`	saves solutions in wavelet discretization to files
	`--no-stats`	do not compute statistics (iterative solvers only)
	`--yes-stats`	opposite of `--no-stats`, takes precedence
`Assemble`	`--spy`	print 'X' and 'O' instead of real values
	`--rhs`	assemble the right hand side vector
	`--Mathematica`	print matrix in a Mathematica compatible format
	`--output`	`filename`: where to write the matrix to
`Condition`	`--max-it`	`int`: limit maximum number of iterations
	`--no-max-it`	no maximum number of iterations
	`--tol`	`value`: tolerance to achieve for eigenvalues
	`--save-EV`	save the computed eigenvectors to file
	`--load-pow-EV`	`filename`: load start vector from file for power iteration
	`--load-inv-EV`	`filename`: load start vector from file for inverse iteration
	`--skip-pow-it`	skips power iteration
	`--skip-inv-it`	skips inverse power iteration
	`--shift`	`value`: spectral shifting (for inverse power iteration)
`Eigenvalues`	`--max-it`	`int`: maximum number of iteration for algorithm
	`--no-max-it`	no maximum number of iterations
	`--output`	`filename`: name of the file to write the eigenvalues to
	`--no-output`	do not save eigenvalues

The right hand side implementations accept the following parameters:

`Const`	`--*-value`	value of the constant function
`Linear`	`--*-ord`	ordinate for linear function
	`--*-slope`	slope of linear function
`ExpAbs`	`--*-factor`	the factor in the exponent of f of $(P3)$
	`--*-addend`	the addend in f of $(P3)$
`Skip`	`--*-v1`	value before the skip
	`--*-v2`	value after the skip

The asterisk must be replaced with the name of the variable that is being addressed, i.e., 'f' or 'y'.

A.3.2 diagassemble/diagassemble

The helper program `diagassemble` can calculate the diagonal of the stiffness matrix for any implemented wavelet base and save them to disk. These files are then later loaded for preconditioning purposes.
This can either be done in the traditional fashion of inserting every wavelet consecutively ("normal assemble") or by utilizing the tensor product setting and the repetetive structure of the wavelet basis ("signature assemble"). The first one can utilize multi-threading for speed gains. The latter one does not this as the overall amount of complexity in this case can be reduced to be proportional to J^n (J=level, n=dimension) and not the number of unknowns $N_J \sim 2^{nJ}$. (In other words: signature assemble is fast.)
The general command line structure is as follows:

```
./diagassemble --level <I,J> --dim <N> [ other options ]
```

Irregular command line options will be ignored silently. The mandatory command line options are :

`--level` comma separated list of positive integers

`--dim` positive integer, most likely 1,2 or 3

The program normaly runs a "normal assemble"and can be configured with these options:

`--multiThread` enable multithreading

`--signature` use tensor structure instead of choosing every unit vector

Additionally, the MRA options of page 151 are recognized.

A.3.3 psgen/wavpsgen

This helper program takes as input wavelet coefficient vectors and outputs a wavelet coefficient diagram in `eps` format. The `eps` generation code is based on code from Carsten Burstedde's `BWP` framework, see [11].
The general command line structure is as follows:

```
./wavpsgen --input <file.tmp> --output <file.eps> [ other options ]
```

Irregular command line options will be ignored silently. The mandatory command line options are :

`--input` input filename (wavelet coefficients)

`--output` output filename (eps file)

The dimensions of the picture can be controlled with the following options:

`--width` width of the eps picture (default: 600)

`--height` height of the eps picture (default: 400)

B Notation

The following notations and definitions should hold anywhere in this document unless explicitly specified otherwise. We have taken care to avoid using any mathematical symbol more than once for different purposes. Due to the small number of Latin and Greek characters it may not always be possible to guarantee absolute uniqueness across the whole document.

B.1 General Notation

keywords	are emphasized and often referenced in the index
`computer`	the typewriter font indicates programming language or computer reference
a.e.	almost everywhere, i.e., valid everywhere except for a domain of measure zero
$\wedge, \vee$	logical *and*, logical *or*
$\mathcal{H}, \mathcal{N}$	a Hilbert space, a product space of Hilbert spaces
$X, Y, ...$	spaces are denoted by capital letters
$u, v, w, ...$	elements of spaces
$\mathbf{u}, \mathbf{v}, \mathbf{w}, ...$	elements of sequence spaces, especially of ℓ_2
$\mathbf{u}_J, \mathbf{v}_J, \mathbf{w}_J, ...$	elements of finite sequence spaces, especially of $\ell_2(\Delta_J)$
$\mathbf{A}, \mathbf{B}, \mathbf{C}, ...$	discretized operators in wavelet form, i.e. matrices of possibly infinite size
$\|\cdot\|, \|\cdot\|_X$	norms in some space X
$(\cdot,\cdot), (\cdot,\cdot)_X$	inner product of X
$\|\cdot\|, \|\cdot\|_X$	seminorms
$\|\cdot\|$	the absolute value function in $\mathbb{R}$; any (equivalent) norm in $\mathbb{R}^n$
$\langle\cdot,\cdot\rangle, \langle\cdot,\cdot\rangle_{X\times X'}$	dual form on the space X, i.e. for $u \in X, v \in X' : \langle u, v\rangle_{X\times X'} := v(u)$
$\langle\cdot,\cdot\rangle$	scalar product of $\mathbb{R}^n$, i.e. for $v, w \in \mathbb{R}^n : \langle v, w\rangle := v^T w$
Ω	an open bounded set in $\mathbb{R}^n$ with boundary $\partial\Omega \supset \Gamma$
I; $\square$, $\square^n$	the interval $(0,1)$; the open unity cube in $\mathbb{R}^n$, $(0,1)^n$
$x_1, \ldots, x_n$	Euclidian coordinates in $\mathbb{R}^n$
$\delta_{(i,j)}$	Kronecker delta for numbers or multi-indices $= \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$
χ_A	characteristic function on some interval/domain A
$f\|_A$	restriction of a function $f : \Omega \to \mathbb{R}$ to a subdomain $A \subseteq \Omega$
$d\mu; ds$	the Lebesque measure in $\mathbb{R}^n$; surface measure
$\lesssim$ $(\gtrsim)$	lesser (greater) or equal to except for a positive constant which is independent of any parameters of the arguments, see (1.1.4)
$\sim$	stands for both $\lesssim$ and $\gtrsim$, see (1.1.4)
$\#$	cardinality of a set
$\lfloor\cdot\rfloor$ $(\lceil\cdot\rceil)$	the highest (smallest) non-negative integer smaller (higher) than or equal to the argument

$\kappa(\cdot), \kappa_2(\cdot)$ (spectral) condition of an operator, matrix or function set

$A^{\updownarrow}$ matrix whose rows and columns are reversed, i.e. $(A^{\updownarrow})_{i,j} := (A)_{n-i,n-j}$

$(\alpha_i, \ldots, \alpha_n)$ **multi-index**:
The n-dimensional **multi-index** α is a n-tuple of non-negative integers α_i with $i = 1, \ldots, n$. Two multi-indices are equal, if and only if all indices are equal. The length is denoted by $|\alpha|$ and defined as $\alpha_1 + \cdots + \alpha_n$. For $x \in \mathbb{R}^n$ are these shorthand expressions defined:

$$x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$$

$$\partial^\alpha = D^\alpha = \frac{\partial^{|\alpha|}}{\partial x^\alpha} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}}$$

$o(\cdot), \mathcal{O}(\cdot), \Theta(\cdot)$ **Landau symbols**:
Let n be an integer that tends to infinity and x be a variable that approaches some limit $x^* \geq x_0$. Also let g be a positive function and f another function. Then are the Landau symbols defined as:

$$\begin{array}{lll} f = \mathcal{O}(g) & :\Longleftrightarrow & |f(x)| < k\, g(x) \qquad \text{for all } x_0 \leq x \leq x^* \text{ and a constant } k > 0 \\ f = o(g) & :\Longleftrightarrow & |f(x)|\,/g(x) \to 0 \qquad \text{for all } x_0 \leq x \leq x^* \\ f = \Theta(g) & :\Longleftrightarrow & f = \mathcal{O}(g) \wedge g = \mathcal{O}(f) \end{array}$$

$[t_0, \ldots, t_n]\, f$ **divided difference**:
For any $f \in C^n$ and knot points $-\infty < t_0 \leq t_1 \leq \ldots \leq t_{n-1} \leq t_n < \infty$ are the divided differences recursively defined as

$$\begin{array}{ll} [t_0]\, f = f(t_0), & \\ [t_0, \ldots, t_n]\, f = \frac{[t_0,\ldots,t_{n-1}]f - [t_1,\ldots,t_n]f}{t_n - t_0}, & t_n \neq t_0, \\ [t_0, \ldots, t_n]\, f = f^{(n)}(t_0)/n!, & t_0 = \cdots = t_n\ . \end{array}$$

$\phi^d(x)$ **cardinal B-Spline**:
The cardinal B-spline ϕ^d of order $d \in \mathbb{N}$ is given as

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

where $x_+^d := (\max\{0, x\})^d$.

$\mathcal{F}(\cdot), \hat{(\cdot)}$ Fourier transform, see (1.2.9)

B.2 Special Mathematical Symbols

n spatial dimension

$s; m + \sigma$ Sobolev smoothness indices: $s \in \mathbb{R}$; $m \in \mathbb{N}_0$, $0 < \sigma < 1$; see Section 1.2.2

$\gamma_j(\cdot)$ general **trace operator** of order j, see (1.2.23)

$\mathcal{L}, \mathcal{A}$ linear partial differential operator of order 2m, see (1.3.2)

N_J generic number of unknowns, may be different in every section

h, h_j discretization error on level j

j_0 minimum level in a **multiresolution analysis** (**MRA**), see $(\mathcal{R})$(2.1.6)

J maximum level of resolution in a given context, see Section 2.1.3

d ($\tilde{d}$) primal (dual) order of **polynomial exactness**, see $(\mathcal{P})$(2.2.3) (see $(\widetilde{\mathcal{P}})$(2.2.4))

γ ($\tilde{\gamma}$) primal (dual) **regularity**, range of smoothness for norm equivalences, see (2.2.7)

S_j ($\widetilde{S}_j$)	closed subspace of primal (dual) Hilbert space, see (2.1.7) (see (2.1.43))
Φ_j ($\widetilde{\Phi}_j$)	primal (dual) **single-scale basis** (also called **generators**), see (2.1.7) (see (2.1.43))
Δ_j	index set for single-scale bases Φ_j and $\widetilde{\Phi}_j$, see (2.1.7) and (2.1.43)
$\phi_{j,k}$ ($\widetilde{\phi}_{j,k}$)	primal (dual) single-scale function on level j located at position $k \in \Delta_j$
W_j ($\widetilde{\mathrm{W}}_j$)	primal (dual) detail spaces, see (2.1.13) (see (2.1.47))
Ψ_j ($\widetilde{\Psi}_j$)	primal (dual) complement basis for space W_j ($\widetilde{\mathrm{W}}_j$), see (2.1.13)
∇_j	index set for complement bases Ψ_j and $\widetilde{\Psi}_j$, see (2.1.13)
$\psi_{j,k}$ ($\widetilde{\psi}_{j,k}$)	primal (dual) **wavelet function** on level j located at position $k \in \nabla_j$, see (2.1.14)
$\Psi_{(J)}$ ($\widetilde{\Psi}_{(J)}$)	primal (dual) wavelet basis up to level J, see (2.1.28)
$\mathcal{S}$ ($\widetilde{\mathcal{S}}$)	primal (dual) **multiresolution analysis**(**MRA**) of $\mathcal{H}$ ($\mathcal{H}'$), see Definition 2.2
$\mathbf{M}_j$ ($\widetilde{\mathbf{M}}_j$)	two-level transformation matrix from level j to $j+1$, see (2.1.18)
$\mathbf{M}_{j,0}$ ($\widetilde{\mathbf{M}}_{j,0}$)	left part of $\mathbf{M}_j(\widetilde{\mathbf{M}}_j)$; matrix of dimensions $\#\Delta_{j+1} \times \#\Delta_j$, see (2.1.11)
$\mathbf{M}_{j,1}$ ($\widetilde{\mathbf{M}}_{j,1}$)	right part of $\mathbf{M}_j(\widetilde{\mathbf{M}}_j)$; matrix of dimensions $\#\Delta_{j+1} \times \#\nabla_j$, see (2.1.17)
P_j ($\widetilde{P}_j$)	primal (dual) projector onto the space Φ_j ($\widetilde{\Phi}_j$), see (2.1.44) (see (2.1.45))
$\mathbf{T}_J$ ($\widetilde{\mathbf{T}}_J$)	primal (dual) **fast wavelet transform**, see (2.1.31) (see (2.1.61))
$\mathbf{G}_j$	inverse of $\mathbf{M}_j$, see (2.1.20)
$\mathbf{G}_{j,0}$	upper part of $\mathbf{G}_j$; matrix of dimensions $\#\Delta_j \times \#\Delta_{j+1}$, see (2.1.24)
$\mathbf{G}_{j,1}$	lower part of $\mathbf{G}_j$; matrix of dimensions $\#\nabla_j \times \#\Delta_{j+1}$, see (2.1.24)
$\mathbb{I}$	infinite index set, see (2.1.39)
$\lambda := (j,k) \in \mathbb{I}$	wavelet index in short notation; $\lvert\lambda\rvert := j$ for $n = 1$ $\lvert\lambda\rvert := \max\{j_1, \ldots, j_n\}$ for $n > 1$, see (2.2.24)
$\Psi^{\mathbb{I}}$ ($\widetilde{\Psi}^{\mathbb{I}}$)	primal (dual) **wavelets** associated to the index set $\mathbb{I}$, see (2.1.39)
$\mathbf{M}_{\mathcal{H}}$ ($\widetilde{\mathbf{M}}_{\mathcal{H}'}$)	primal (dual) **Gramian matrices** using the inner product $(\cdot,\cdot)$ of space $\mathcal{H}$ ($\mathcal{H}'$), see (2.1.58) (see (2.1.59))
$\mathbf{A}_{H^1}, \mathbf{A}_J$	**stiffness matrix**, see (2.3.38)
$\mathbf{S}_{H^1}, \mathbf{S}_J$	**Laplace matrix**, see (2.4.13)
$\mathbf{D}^{\pm s}$	any diagonal matrix that can be used for shifting wavelet coefficient vectors in the Sobolev scale by $\pm s$, see (2.2.12) and (2.2.13)
$\mathbf{D}_1^{\pm s}$	diagonal matrix consisting of powers of 2, see (2.2.14)
$\mathbf{D}_a^{\pm s}$	inverse diagonal of stiffness matrix, see (3.2.34)
$\mathbf{D}_{\{O,X\}}^{\pm s}$	special preconditioner for basis transformed wavelet bases, see (2.3.45)
Ψ^s ($\widetilde{\Psi}^s$)	scaled wavelet bases constituting **Riesz bases** for H^{+s} ($H^{+s'}$), see Corollary 2.22
$\mathbb{I}_J$	finite subset of $\mathbb{I}$ created by eliminating all indices with $\lvert\cdot\rvert > J$, see (2.2.24)
Ψ_J^s	scaled version of finite wavelet basis (2.2.26), see (2.2.24)

$R_{\mathcal{H}}$ ($\mathbf{R}_{\mathcal{H}}$)	(wavelet discretized) Riesz operator for space $\mathcal{H} \in \{L_2, H^s, \ldots\}$, see (2.2.38)
$\widehat{\mathbf{D}}^{+2s}$	diagonal matrix used in the construction of Riesz operator $\widehat{\mathbf{R}}_{H^s}$, see (2.2.42)
$\widehat{R}_{H^s}$ ($\widehat{\mathbf{R}}_{H^s}$)	specifically constructed (wavelet discretized) Riesz operator for H^s, see (2.2.44)
$\mathring{\widehat{\mathbf{R}}}_{H^s}$	Riesz operator for H^s normalized w.r.t. constant functions, see (5.2.8)
$\widetilde{\mathbf{R}}_{H^s}$ ($\mathring{\widetilde{\mathbf{R}}}_{H^s}$)	wavelet discretized (normalized) interpolating Riesz operator for H^s, see (2.2.53)
$\mathring{\widehat{\mathbf{R}}}_{H^s}$	wavelet discretized Riesz operator $\widehat{\mathbf{R}}_{H^s}$ normalized w.r.t. constant functions, see (5.2.8)
$\mathbf{C}_j$	basis transformation working on the boundary functions only, see (2.3.30)
O	orthogonal basis transformation, see (2.3.39)
$\Phi_{(\square,j)}, \Psi_{(\square,J)}, ..$	**tensor product** analogons to their respective 1D objects, see Section 2.4.1
$\mathbf{n} = \mathbf{n}(x)$	outward normal at any point $x \in \partial\Omega$
c_X, C_X	lower and upper constants used in norm equivalences for an operator $X \in \{\mathbf{A}, \mathbf{L}, \ldots\}$
B ($\mathbf{B}$)	**trace operator** (in wavelet discretization), see (3.2.27) (see (3.2.28))
L ($\mathbf{L}$)	**saddle point matrix** (in wavelet discretization), see (3.1.20) (see (3.1.26))
$J_{\mathcal{H}}$	levels of resolution of the finite dimensional saddle point problem (see Section 3.1.3); $J_{\mathcal{N}} = (\sigma, \pi) \equiv$ levels on the domain and boundary
σ	level of resolution on the domain Ω (see Section 3.1.3)
π	level of resolution on the boundary Γ (see Section 3.1.3)
y ($\mathbf{y}$), u ($\mathbf{u}$)	**state** and **control** variables (in wavelet discretization), see Section 4.1
$\mathcal{J}(y,u)(\mathbf{J}(\mathbf{y},\mathbf{u}))$	cost functional (in wavelet discretization), see Section 4.1 and (4.2.37)
y_*, y_{Γ_Y}	**observation** in control theory, see Section 4.1 and Problem 4.20
ω	weighting parameter in the cost functional, see (4.2.1)
T ($\mathbf{T}$)	**trace operator** (in wavelet discretization), see (4.2.4) (see (4.2.36))
c_*, C_*	constants bounding the second variation of the cost functional, see (4.2.45)
s, t	smoothness indices in the cost functional, see (4.3.1)
N ($\mathbf{N}$)	All-In-One operator (in wavelet discretization) , see (4.2.21) (see (4.2.53))
$J_{\mathcal{N}}$	levels of resolution of the finite dimensional control problem (see Section 4.2.3); $J_{\mathcal{N}} = (\sigma, \pi, \tau) \equiv$ levels on the domain, control boundary, observation boundary
σ	level of resolution on the domain Ω (see Section 4.2.3)
π	level of resolution on the control boundary Γ (see Section 4.2.3)
τ	level of resolution on the observation boundary Γ_Y (see Section 4.2.3)
ρ, ρ_i	step size parameter in the **Inexact Gradient** algorithm (see Section 4.4.2)
$:= \quad \leftarrow$	initial definition and update of variables in the algorithm specifications

B.3 Spaces

$\mathbb{N}$ the natural numbers $= \{1, 2, \ldots\}$

$\mathbb{N}_0$ the natural numbers including zero $= \{0, 1, 2, \ldots\}$

$\mathbb{Z}$ integers $= \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

$\mathbb{Q}$ the rational numbers

$\mathbb{R}, \mathbb{R}^n$ the real numbers, the n-dimensional Euclidean Vector space

$\mathbb{R}_+, \mathbb{R}^n_+$ the positive real numbers, $\{(x_1, \ldots, x_n) \,|\, x_i > 0 \text{ for } 1 \le i \le n\}$

ℓ_2 = sequence space of all sequences for which the ℓ_2-norm is finite, i.e. let $\mathbf{c} \in \mathbb{R}^{\mathbb{N}} := \{x = (x_i)_{i\in\mathbb{N}} : x_i \in \mathbb{R} \text{ for } i \in \mathbb{N}\}$, then

$$\mathbf{c} \in \ell_2 \quad \Longleftrightarrow \quad \|\mathbf{c}\|_{\ell_2} := \left(\sum_{k\in\mathbb{R}} |c_k|^2\right)^{1/2} < \infty \ .$$

B.4 Function Spaces

$C^k(\Omega)$ $= \{\phi : \Omega \to \mathbb{R} \,|\, \text{all derivatives } \partial^\alpha\phi \text{ of order } |\alpha| \le k \text{ are continuous in } \Omega\}$

$C^k(\bar\Omega)$ $= \{\phi \in C^k(\Omega) \,|\, \text{all derivatives } \partial^\alpha\phi \text{ of order } |\alpha| \le k \text{ have continuous extensions to } \bar\Omega\}$

$C^k_0(\Omega)$ $= \{\phi \in C^k(\Omega) \,|\, \operatorname{supp}\phi \subset\subset \Omega$, i.e. ϕ has compact support fully contained in $\Omega\}$

$C^{k,1}(\Omega)$ = Lipschitz continuous functions =

$$\{f \in C^k(\Omega) \,|\, |D^s f(x) - D^s f(y)| \le L|x-y| \ \ \forall\, \alpha \,\textbf{multi-index}, |s| \le k, 0 < L < \infty\}$$

$C^{k,\alpha}(\Omega)$ = Hölder continuous functions of order $0 < \alpha < 1$ =

$$\{f \in C^k(\Omega) \,|\, \sup_{x\neq y\in\Omega} \frac{|D^s f(x) - D^s f(y)|}{|x-y|^\alpha} < \infty \ \ \forall\, \alpha \,\textbf{multi-index}, |s| \le k\}$$

$C^\infty(\Omega)$ = space of infinitely differentiable functions on Ω with values in $\mathbb{R} = \bigcap_{k\in\mathbb{N}} \{C^k(\Omega)\}$

$\mathcal{D}(\Omega), C^\infty_0(\Omega)$ = space of infinitely differentiable functions with values in $\mathbb{R}$ and with compact support fully contained in Ω

$\mathcal{D}'(\Omega)$ = dual space of $\mathcal{D}(\Omega)$ = space of distributions on Ω

$L_2(\Omega)$ = space (equivalence class) of all real valued square Lebesque-integrable functions on the domain Ω

$H^m(\Omega)$ = Sobolev space of order $m \in \mathbb{N}$ =

$$\{\phi \in L_2(\Omega) \,|\, \frac{\partial\phi}{\partial x_i} \in L_2(\Omega), \ldots, D^\alpha\phi \in L_2(\Omega) \ \ \forall\, \alpha \,\textbf{multi-index}, |\alpha| \le m\}$$

$H^m_0(\Omega)$ $= \{\phi \in H^m(\Omega) \,|\, D^\alpha\phi = 0 \text{ on } \partial\Omega, |\alpha| \le m-1\}$

$H^s(\Omega)$ = Sobolev space of fractional order s on Ω

$H^{-s}(\Omega)$ = Dual Space of $H^s_0(\Omega)$

References

[1] R. Adams. *Sobolev Spaces*. Academic Press, 1978.

[2] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley, 2001.

[3] H. Alt. *Lineare Funktionalanalysis*. Springer, 2002. (In German).

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

[5] I. Babuška. The finite element method with penalty. *Mathematics of Computation*, 27:221–228, 1973.

[6] 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.

[7] 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, July 2003.

[8] D. Braess. *Finite Elemente – Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, 2003. (In German).

[9] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *RAIRO Anal. Numér.*, 8(R-2):129–151, 1974.

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

[11] C. Burstedde. *Fast Optimised Wavelet Methods for Control Problems Constrained by Elliptic PDEs*. PhD Dissertation, Institut für Angewandte Mathematik, Universität Bonn, December 2005.

[12] C. Burstedde and A. Kunoth. Fast iterative solution of elliptic control problems in wavelet discretizations. *To appear in: J. Comp. Appl. Maths.*

[13] C. Canuto, A. Tabacco, and K. Urban. The wavelet element method, part i: Construction and analysis. *Appl. Comp. Harm. Anal.*, 6:1–52, 1999.

[14] J. Carnicer, W. Dahmen, and J. Peña. Local decompositions of refinable spaces. *Appl. Comp. Harm. Anal.*, 3:127–153, 1996.

[15] GNU Common C++. http://www.gnu.org/software/commoncpp/.

[16] C. Chui and E. Quak. Wavelets on a bounded interval. In D. Braess and L.L. Schumaker, editors, *Numerical Methods of Approximation Theory*, volume 6, pages 155–189. Birkhäuser, 1992.

[17] P. Ciarlet. *Introduction to Numerical Linear Algebra and Optimization*. Cambridge University Press, 1989.

[18] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations – convergence rates. *Math. Comp.*, 70:27–75, 2001.

[19] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure an Appl. Math.*, 45:485–560, 1992.

[20] S. Dahlke, W. Dahmen, and K. Urban. Adaptive wavelet methods for saddle point problems – optimal convergence rates. *SIAM J. Numer. Anal.*, 40:1230–1262, 2002.

[21] S. Dahlke, R. Hochmuth, and K. Urban. Adaptive wavelet methods for saddle point problems. *Mathematical Modelling and Numerical Analysis (M2AN)*, 34:1003–1022, 2000.

[22] W. Dahmen. Decomposition of refinable spaces and applications to operator equations. *Numer. Algor.*, 5:229–245, 1993.

[23] W. Dahmen. Some remarks on multiscale transformations, stability and biorthogonality. In *Wavelets, Images and Surface Fitting*, pages 157–188. AK Peters, Wellesley, MT, 1994.

[24] W. Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl.*, 4:341–362, 1996.

[25] W. Dahmen. Wavelet and multiscale methods for operator equations. *Acta Numerica*, pages 55–228, 1997.

[26] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63:315–344, 1992.

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

[28] 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.

[29] W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal spline-wavelets on the interval – stability and moment conditions. http://www.iam.uni-bonn.de/∼kunoth/data/databw.html. Online Resources.

[30] 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.

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

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

[33] C. de Boor. *A Practical Guide to Splines*. Springer, revised edition, December 2001.

[34] P. Deuflhard and A. Hohmann. *Numerical Analysis: A First Course in Scientific Computation*. de Gruyter, 1995.

[35] R. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[36] M. Fortin. An analysis of the convergence of mixed finite element methods. *RAIRO Anal. Numér.*, 11:341–354, 1977.

[37] R. Glowinski, T.W. Pan, and J. Periaux. A fictitious domain method for dirichlet problem and application. *Comp. Math. Appl. Mechs. Eng.*, 111:282–303, 1994.

[38] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd. edition, 1996.

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

[40] W. Hackbusch. Fast solution of elliptic control problems. *J. Optim. Theory Appl.*, 31:565–581, 1980.

[41] W. Hackbusch. *Elliptic Differential Equations. Theory and Numerical Treatment*. Springer, 2003.

[42] H.O. Kim and J.K. Lim. New characterizations of riesz bases. *Appl. Comp. Harm. Anal.*, 4:222–229, 1997.

[43] Bruce J. Kirby. *Optimal Control Theory and its Applications I/II*. Lecture Notes in Economics and Mathematical Systems. Springer, 1974.

[44] J. Krumsdorf. Finite element wavelets for the numerical solution of elliptic partial differential equations on polygonal domains. Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, 2004.

[45] A. Kunoth. *Multilevel Preconditioning*. PhD Dissertation, Fachbereich Mathematik, Freie Universität Berlin, January 1994.

[46] A. Kunoth. *Wavelet Methods – Elliptic Boundary Value Problems and Control Problems*. Advances in Numerical Mathematics. Teubner Verlag, 2001.

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

[48] A. Kunoth and J. Sahner. Wavelets on manifolds: An optimized construction. *SFB 611 Preprint No. 163, Universitt Bonn*, page 29, April 2005. To appear in Math. Comp.

[49] J. Lions. *Optimal Control of Systems governed by Partial Differential Equations*. Springer, 1971.

[50] J. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications*, volume I,II and III. Springer, 1972.

[51] Loki development on sourceforge.net. http://sourceforge.net/projects/loki-lib/.

[52] R. Masson. *Wavelet Methods in numerical simulation for elliptic and saddle point problems*. PhD Dissertation, University of Paris VI, January 1999.

[53] M. Mommer. *Towards a Fictitious Domain Method with Optimally Smooth Solutions*. PhD Dissertation, RWTH Aachen, June 2005.

[54] Matrix template library. http://www.lsc.nd.edu/research/mtl/.

[55] P. Oswald. *Multilevel Finite Element Approximation, Theory and Applications*. Teubner Verlag, 1994.

[56] M. Primbs. *Stabile biorthogonale Spline-Waveletbasen auf dem Intervall*. PhD Dissertation, Universität Duisburg-Essen, Campus Duisburg, January 2006.

[57] K. Scherer. Approximation mit Wavelets I/II. Lecture courses at the Universität Bonn, Summer Semester 2001 / Winter Semester 2002.

[58] R. Stevenson. Locally supported, piecewise polynomial biorthogonal wavelets on non-uniform meshes. *Const. Approx.*, 19(4):477–508, 2003.

[59] L.F. Villemoes. Sobolev regularity of wavelets and stability of iterated filter banks. In Y. Meyer and S. Roques, editors, *Progress in Wavelet Analysis and Applications*, pages 243–251. Editions Frontières, 1993.

[60] J. Vorloeper. Multiskalenverfahren und Gebietszerlegungsmethoden. Diplomarbeit, RWTH Aachen, 1999. (In German).

[61] Eric W. Weisstein et al. Robin boundary conditions. http://mathworld.wolfram.com/RobinBoundaryConditions.html. From MathWorld – A Wolfram Web Resource.

[62] Eric W. Weisstein et al. Submersion. http://mathworld.wolfram.com/Submersion.html. From MathWorld – A Wolfram Web Resource.

[63] Wikipedia et al. Haar wavelet. http://en.wikipedia.org/wiki/Haar_wavelet. Version of December 2005.

Index