Fast Solvers and Domain Decomposition Preconditioners for Spectral Element Discretizations of Problems in H(curl)

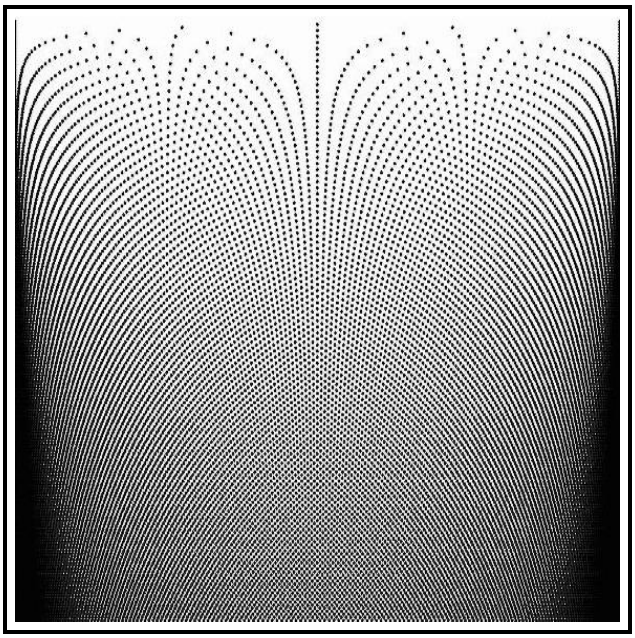
by

Bernhard Hientzsch
Courant Institute of Mathematical Sciences
New York University

Technical Report TR2001-823 November 28, 2001
Department of Computer Science Courant Institute of Mathematical Sciences

Ph.D. thesis (Advisor: Olof B. Widlund) September 2001
Department of Mathematics Courant Institute of Mathematical Sciences

Bernhard Hientzsch 251 Mercer Street, New York, NY 10012
Department of Mathematics Courant Institute of Mathematical Sciences
hientzsc@cims.nyu.edu http://www.math.nyu.edu/~hientzsc/



This is a version of my thesis with some small corrections, and typeset differently from the official version submitted to the Graduate School of Arts and Sciences at New York University.

The URLs given were last checked and found valid in November 2001.

To Lady Mathematics, for all the fun and moments of enlightenment

Acknowledgments

First and foremost I want to thank my advisor and friend, Olof Widlund, for proposing the thesis subject, and for all his support and help in the last six years, four of them as his student.

I also want to thank Yu Chen and Jonathan Goodman for their willingness to serve as readers on short notice.

I thank all the faculty, staff, and students of the Courant Institute who contributed to create a warm and motivating atmosphere. I thank all the professors who transmitted some of their excitement about mathematics to me, I especially want to mention John Rinzel and Bud Mishra. I also want to thank all who fed my neverending interest in all things mathematical and who were as curious as me about mathematics, physics and all that. Thank you, Sávio and Franz.

I have had the privilege of becoming very good friends with four fantastic people during my several stays at the Courant Institute: Alla, Denise, Firas and Sávio. I certainly would be in much worse shape now and my life would be much emptier without them.

I also want to thank all the other friends presently or formely at Courant, a non-exhaustive, alphabetical, list includes: Amiya, Ann, Anuroopa, Breno, Brynja, Cameron, Dan, Ed, Eileen, Eliza, Enrique, Helga, Isela, Itir, Jennifer, José, Jose, Lauraine, Leopoldo, Marcos, Mario and Santina, Nikos, Paulo, Shimn, Suresh and Thelmo. Many thanks also to Juliana and Tal.

I owe many thanks to my parents and to my family. They have always loved and supported me even as they did not always understand me, and would have preferred me being geographically closer to them. I thank especially my mother Erika and my father Eberhard for all their love and help, my brothers and sisters and their families: Ekkehard and Anett; Barbara, Heinz, Richard and Robert; Beate, Tobias, Arne and Liv; Brigitte, Stefan, Sebastian and Johann; for always making me feel welcome and forgiving me my long absences and silences. I also thank all friends in Dresden who still make me miss Dresden after all those years and make me feel like I have never left. Thank you, especially to Tobias, Jana, Olaf and Simone.

Abstract

For problems with piecewise smooth solutions, spectral element methods hold great promise. They combine the exponential convergence of spectral methods with the geometric flexibility of finite elements. Spectral elements are well-established for scalar elliptic problems and problems of fluid dynamics, and recently the first methods for problems in $H(\mathrm{curl})$ and $H(\mathsf{div})$ were proposed. In this dissertation we study spectral element methods for a model problem. We first consider Maxwell's equation and derive the model problem in $H(\mathrm{curl})$. Then we introduce anisotropic spectral Nédélec element discretizations with variable numerical integration for the model problem. We discuss their structure, and their convergence and approximation properties. We also obtain results on the norm of the Nédélec interpolants between Nédélec and Raviart-Thomas spaces of different degree, needed for the computation of the splitting constant for the domain decomposition preconditioner and the numerical analysis of nonlinear equations. We also prove a Friedrichs-like inequality for the model problem for the spectral case.

We present fast direct solvers for the model problem on separable domains, taking advantage of the tensor product discretization and fast diagonalization methods. We use those fast solvers as local solvers in domain decomposition methods for problems that are too large to be solved directly, or posed on non-separable domains, and use them to compute and subassemble the Schur complement system corresponding to the interface. We also apply them in the direct solution of the Schur complement system for general domains.

As an example for the domain decomposition methods that can be implemented with these tools, we introduce overlapping Schwarz methods, both one-level and two-level versions.

We extend the theory for overlapping Schwarz methods to the spectral Nédélec element case. We reduce the proof of the condition number estimate to three basic estimates, and present theoretical and numerical results on those estimates. The technique of the proof works in both the two-dimensional and three-dimensional case.

We also present numerical results for one-level and two-level methods in two dimensions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational electromagnetics concerns the numerical approximation of Maxwell's equations. Maxwell's equations describe the interaction of electromagnetic waves and matter, and form a vector system of time-dependent partial differential equations. There is an increasing need for optimal solvers for Maxwell's equation since devices such as optical devices in integrated optics or photonic crystals have been proposed and need to be modeled. Since for some of those devices the production of prototypes is very expensive and complicated, an accurate numerical model has to be designed and solved, and the solution has to be fast, since it will possibly be used in design optimizations. In the last ten years, computational electromagnetics has become a very important research area in numerical analysis. Besides the design problem mentioned above, areas of interest are also the simulation of antennas, the scattering by complicated objects; used for instance in one of the approaches to inverse scattering – which also needs a very fast direct solver – and the calculation of eddy currents in electric conductors.

The main focus of this thesis is the spectral element discretization of Maxwell's equation, the construction of fast direct solvers for such discretizations, and the construction and analysis of domain decomposition preconditioners for iterative methods for such discretizations.

For the analysis of Maxwell's equations, suitable Sobolev spaces have to be introduced: $H(\mathbf{curl})$ and $H(\mathsf{div})$ are the graph spaces of $\mathbf{curl}$ and div over L^2. Suitable finite element spaces conforming in those continuous spaces were introduced in the late 1970's, in particular the Nédélec or edge element spaces, conforming in $H(\mathbf{curl})$, and the Raviart-Thomas spaces, conforming in $H(\mathsf{div})$. In those approximation spaces only some of the components are forced to be continuous across the interface. We introduce the continuous spaces in chapter 2, and we discuss the hN-extension of the Nédélec and Raviart-Thomas-Nédélec spaces in chapter 7. We discuss the commuting diagram properties which they share with

the continuous setting, the approximation properties, and the properties of the interpolation operators when used as mappings between Nédélec or Raviart-Thomas-Nédélec spaces of different degrees. We also prove a discrete Friedrichs' inequality. To the best of our knowledge, the use of spectral element degrees of freedom for the Raviart-Thomas-Nédélec spaces is new, and so is the study of the mapping properties of the Nédélec interpolation between spaces of different order.

We introduce a standard model problem in $H(\mathbf{curl})$, which we derive from the implicit time-integration of the time-dependent Maxwell's equations in chapter 3, and we present spectral Nédélec element discretizations of the model problem in chapter 8. We only know of one group working on spectral elements for Maxwell's equations, around Ben Belgacem (see, e.g., Ben Belgacem and Bernardi [15]), which seems to use mainly mortar elements. We do not know of any experimental work on spectral elements for the model problem.

We derive the discretization for arbitrary degrees, possibly different in different directions, and arbitrary degrees of numerical integration. We present subassembly procedures on rectangular domains for H^1-, $H(\mathrm{curl})$-, and $H(\mathsf{div})$-conforming discretizations.

We study fast direct solvers on rectangular domains, and direct solvers for the Schur complement system on the element interfaces for non-separable domains in chapter 9.

For systems too large to fit into the memory of a single machine, or so large that direct solvers are not competitive in terms of storage or computing time, we consider iterative methods. Recently, efficient preconditioners for the finite element method for the model problem have become the subject of extensive research. Some of the most promising methods are domain decomposition and multigrid solvers and preconditioners. In a domain decomposition approach, a problem on a large domain is solved approximately by solving problems over smaller subregions and combining the local solutions appropriately. One can easily design iterative schemes which start from an initial guess, and solve local problems, in parallel or in sequence, in each step. These basic iterative methods can also be used to construct preconditioners for the discretizations, that are then accelerated by Krylov subspace methods. When problems with a large number of subdomains are to be solved, a coarse problem has to be added to improve convergence, especially to make the convergence independent of the number of subregions. Domain decomposition methods are by design parallel methods, and can be easily implemented on parallel computers and have been shown to lead to scalable preconditioners.

Many domain decomposition methods can be viewed in terms of the abstract Schwarz framework, see chapter 5. A Schwarz preconditioner is defined by a collection of subspaces with exact or inexact solver provided for each of them, where the union of the collection of the subspaces equals the original space. The algorithms we consider here are two-level algorithms, where we work with discretizations on a fine mesh of elements and a coarse mesh of subregions.

We study the model problem

$$a(\mathbf{u},\mathbf{v}) := \eta_1(\mathbf{u},\mathbf{v})_{L^2} + \eta_2(\mathbf{curl}\,\mathbf{u},\mathbf{curl}\,\mathbf{v})_{L^2} = \mathbf{f}(\mathbf{v})$$

on $H(\mathbf{curl})$. The domain Ω is a bounded connected polygon or polyhedron. Essential, natural, and Silver-Müller boundary conditions can be considered.

The study and analysis of preconditioners for Nédélec and Raviart-Thomas-Nédélec discretization started only recently, even for the $h-$version of the elements. Two-level overlapping Schwarz preconditioners for $H(\mathrm{div})$ were developed by Arnold, Falk and Winther [6], They were further investigated by Toselli in the $H(\mathbf{curl})$ case in [96, 98] and by Hiptmair and Toselli [60] for both $H(\mathrm{div})$ and $H(\mathbf{curl})$.

Multigrid and multilevel methods for $H(\mathrm{div})$ and $H(\mathbf{curl})$ were considered in Arnold, Falk and Winther [8, 7], Hiptmair and Toselli [60] and Hiptmair [59, 58, 57]. Iterative substructuring methods are treated in Alonso and Valli [4]; Toselli [96]; Toselli, Widlund, and Wohlmuth [101]; and Wohlmuth, Toselli, and Widlund [104], a Neumann-Neumann solver is considered in Toselli [97] and FETI preconditioners are proposed in Toselli and Rapetti [100], and Toselli and Klawonn [99]. We are not aware of any work on domain decomposition preconditioners for spectral element discretizations for Maxwell's equations or the model problem.

We present an implementation of a two-level additive overlapping method in chapter 10, and a proof of a condition number estimate for this method for the two- and three-dimensional case in chapter 11.

In the following, we will denote by A the representation of the bilinear form $a(\cdot,\cdot)$ on the spectral element space, and by B the additive Schwarz preconditioner. Denoting by h the size of the small elements, by H the size of the subregions, by δ the size of the overlap, by N_C the number of colors needed to color the subregions so that no two regions of the same color overlap, and by N the degree of the spectral Nédélec elements, we prove a condition number estimate of the form

$$\kappa(BA)) \leq C(N_c+1)\frac{\max(\eta_1,\eta_2)}{\min(\eta_1,\eta_2)}\left(1+N_c\left(1+\left(\frac{H}{\delta}\right)^2\right)\right)$$

for generous or fixed overlap, and of the form

$$\kappa(BA) \leq C(N_c+1)N\frac{\max(\eta_1,\eta_2)}{\min(\eta_1,\eta_2)}\left(1+N_c\left(1+\left(\frac{H}{\delta}\right)^2\right)\right)$$

for minimal overlap. Both the power of $\frac{H}{\delta}$ and of N can most probably be improved; for the first one would have to extend the small overlap type of proof of Dryja and Widlund

[44] to $H(\mathbf{curl})$; for the second we present a different treatment of the partition of unity that could improve the estimate.

The proof is an extension of Toselli's proof in [98] to the spectral case. We have reduced it to the proof of three required estimates, for which we present both numerical and theoretical results.

In the course of the work on this thesis, we have also developed direct solvers for the model problem using a computational Helmholtz decomposition to reduce the solution of the model problem to scalar and vector Helmholtz and Laplace solves, and we have worked on a generalization of the restricted additive Schwarz method for the Poisson and the Helmholtz equation. Unfortunately we lack both space and time to present the results in the context of this thesis. We hope to present this work in future publications.

Chapter 2

Function spaces and regularity results

In this chapter we present the function spaces and regularity results that we will need later in this thesis. In the first subsection, we introduce the standard Sobolev spaces H^s and $W^{s,p}$ and some of their properties. The next three sections are dedicated to a short introduction to the graph spaces $H(\mathsf{div}, \Omega)$, $H(\mathrm{curl}, \Omega)$ in two dimensions and $H(\mathbf{curl}, \Omega)$ in three dimensions. In the fifth section we present orthogonal decompositions of $(L^2(\Omega))^n$ and of the graph spaces that generalize the Helmholtz decomposition of smooth vector fields into divergence-free and curl-free parts. In the next section, we present some regularity results for the Laplace operator that are needed later in the chapter. Section 7 presents a discussion of imbedding theorems for the intersection of $H(\mathsf{div})$ and $H(\mathbf{curl})$. We end the chapter with a discussion of the regularity of **curl** potentials in the last section.

For a general theory of the classical Sobolev spaces see Adams [1], Nečas [73], or Grisvard [52]. For a theory of $H(\mathsf{div})$ and $H(\mathbf{curl})$ we refer to Dautray and Lions [34], and Girault and Raviart [48]. For an introduction to the case of non-smooth domains see Grisvard [52], and Amrouche, Bernardi, Dauge, and Girault [5] and references therein.

Let $\Omega \subset \mathbb{R}^n$ be an open, bounded and connected set, with a Lipschitz continuous boundary $\partial\Omega$ and exterior normal $\mathbf{n}$. Given a generic vector $\mathbf{u} \in \mathbb{R}^n$, we denote its Cartesian components by u_i, $i = 1, \cdots, n$. Any definition of inner products or norms can be extended from the scalar case to the vector case in a straightforward way, i.e., for X^n we use

$$(\mathbf{u}, \mathbf{v})_{X^n} := \sum_{i=1}^{n} (u_i, v_i)_X$$

$$||\mathbf{u}||^2_{X^n} := \sum_{i=1}^{n} ||u_i||^2_X$$

2.1 Sobolev spaces

$L^p(\Omega)$ is the space of Lebesgue measurable functions u with $||u||_{L^p(\Omega)} = ||u||_{0,p,\Omega} < \infty$ where

$$||u||^p_{0,p,\Omega} = \int_\Omega |u|^p$$

$$||u||_{0,\infty,\Omega} = \text{ess sup}_\Omega |u|$$

$L^p(\Omega)$ is a Banach space, and for $p = 2$ it is a Hilbert space with the inner product

$$(u, v)_{0,\Omega} = (u, v)_0 := \int_\Omega uv$$

$L^p_0(\Omega)$ is the subspace of $L^p(\Omega)$ of functions with mean zero, i.e., $\int_\Omega u = 0$.

The Sobolev space $W^{k,p}(\Omega)$ for k integer consists of all locally summable functions u such that for each multi-index α with $|\alpha| \leq k$, $D^\alpha u \in L^p(\Omega)$. Its norm is defined by

$$||u||_{W^{k,p}(\Omega)} = ||u||_{k,p,\Omega} := \left(\sum_{|\alpha|\leq k} ||D^\alpha u||^p_{0,p,\Omega} \right)^{1/p}$$

$$||u||_{k,\infty,\Omega} := \max_{|\alpha|\leq k} ||D^\alpha u||_{0,\infty,\Omega}$$

The spaces $H^k(\Omega) := W^{k,2}(\Omega)$ are Hilbert spaces, their norm is denoted $|| \cdot ||_{H^k(\Omega)} = || \cdot ||_{k,\Omega}$.

For s nonnegative and not an integer, we write $s = \lfloor s \rfloor + \sigma$ with $\sigma \in (0, 1)$, and $u \in W^{s,p}(\Omega)$ if and only if $u \in W^{\lfloor s \rfloor,p}(\Omega)$ and, for $|\alpha| = \lfloor s \rfloor$, $[u]_{\sigma,p,\alpha,\Omega} < \infty$, where

$$[u]^p_{\sigma,p,\alpha,\Omega} := \int_\Omega \int_\Omega \frac{|D^\alpha u(x) - D^\alpha u(y)|^p}{|x - y|^{n+\sigma p}} dxdy$$

$$[u]_{\sigma,\infty,\alpha,\Omega} := \text{ess sup}_{x,y\in\Omega,x\neq y} \frac{|D^\alpha u(x) - D^\alpha u(y)|}{|x - y|^\sigma}$$

The norm in $W^{s,p}(\Omega)$ is then defined as

$$||u||_{s,p,\Omega} = \left(||u||^p_{\lfloor s \rfloor,p,\Omega} + \sum_{|\alpha|=\lfloor s \rfloor} [u]^p_{\sigma,p,\alpha,\Omega} \right)^{1/p}$$

$$||u||_{s,\infty,\Omega} = ||u||_{\lfloor s \rfloor,\infty,\Omega} + \max_{|\alpha|=\lfloor s \rfloor} [u]_{\sigma,\infty,\alpha,\Omega}$$

The spaces $H^s(\Omega) := W^{s,2}(\Omega)$ are Hilbert spaces for $s \geq 0$. There are instrinsic definitions of the scalar product $(u,v)_s$, or it can also be defined by polarization of $||u||_s$.

For k integer, one defines $W^{k,p}(\Omega)$ semi-norms as follows:

$$|u|^p_{k,p,\Omega} = \sum_{|\alpha|=k} ||D^\alpha u||^p_{0,p,\Omega}$$

$$|u|_{k,\infty,\Omega} = \sum_{|\alpha|=k} ||D^\alpha u||_{0,\infty,\Omega}$$

The semi-norms on the spaces H^k are denoted $|u|_k$.

For the solution of essential boundary value problems we also need spaces of functions with imposed boundary conditions. In the standard way, one defines $W^{s,p}_0(\Omega)$ as the closure of the space of infinitely differentiable functions with compact support in Ω with respect to the $||\cdot||_{s,p,\Omega}$-norm. As before, $H^s_0(\Omega) := W^{s,2}_0(\Omega)$.

The spaces with negative s are defined by duality. Since the dual space of $H^s(\Omega)$ would not be a space of distributions, H^s_0 is chosen. That means

$$H^{-s}(\Omega) = (H^s_0(\Omega))'$$

$$W^{-s,p}(\Omega) = (W_0^{s,\frac{p}{p-1}}(\Omega))'$$

with the standard definition of the dual operator norm.

There are Sobolev imbedding theorems stating inclusion relationships between different $W^{s,p}$ spaces. We will present one of the versions in the following.

Theorem 2.1 (Sobolev imbedding theorem) *Let $p \in [1,\infty]$, $s < t$. The following imbeddings hold algebraically and topologically*

$$W^{s,p}(\Omega) \subset \begin{cases} W^{t,q}(\Omega) & \text{if } \frac{1}{q} = \frac{1}{p} - \frac{s-t}{n} > 0 \\ W^{t,q}_{loc}(\Omega) & \forall q \in [1,\infty) \text{ if } \frac{1}{p} = \frac{s-t}{n} \\ C^{\lfloor t \rfloor}(\Omega) & \text{if } \frac{1}{p} < \frac{s-\lfloor t \rfloor}{n} \end{cases}$$

The first imbedding is compact for all $q \in [1, \frac{np}{n-(s-t)p})$, if $n > (s-t)p$.

See, for instance, Girault and Raviart [48, Theorem I.1.3].

To give an intrinsic characterization of $W^{s,p}_0(\Omega)$, and to discuss boundary behavior of functions in $W^{s,p}$, we need to introduce trace operators and the trace theorem. Denote by γ_0 the operator that maps a function in $C(\Omega)$ to its boundary values in $C(\partial\Omega)$.

Theorem 2.2 (Trace theorem) *Assume that Ω has a boundary of class $C^{k,1}$, that $p \geq 1$, $s \geq 0$ and $s \leq k+1$, $s - \frac{1}{p} = l + \sigma$ with l a non-negative integer and $\sigma \in (0,1)$. Then the mapping γ_0 has a continuous extension as an operator*

$$W^{s,p}(\Omega) \mapsto W^{s-\frac{1}{p},p}(\partial\Omega)$$

In the case that Ω has a piecewise $C^{k,1}$ boundary, with the $C^{k,1}$ pieces $\partial\Omega_i$, γ_0 can be extended to an operator

$$W^{s,p}(\Omega) \mapsto \prod_i W^{s-\frac{1}{p},p}(\partial\Omega_i)$$

Its range is a subspace of $\prod_i W^{s-\frac{1}{p},p}(\partial\Omega_i)$ characterized by additional conditions associated with the intersection between $\partial\Omega_i$ and $\partial\Omega_j$.

For the $W^{s,p}$-case, see, e.g., Grisvard [52, section 1.5.2], and for the H^s-case, see, e.g., Bernardi and Maday [17, section 1].

Until now we have assumed that Ω is a domain of size $O(1)$. For domains of diameter H_Ω, we will work with the standard scaled norms, for instance, for $p = 2$ and $s = 1$ we have

$$||u||^2_{1,\Omega} := |u|^2_{1,\Omega} + \frac{1}{H^2_\Omega}||u||^2_{0,\Omega}$$

In the Sobolev spaces, many useful inequalities are known. We refer, e.g., to the discussions in Nečas [73] and in Dautray and Lions [32, chapter IV,§7].

We will only give the Friedrichs' inequality, needed in the proof that the H^1-semi-norm is equivalent to the H^1-norm on H^1_0:

$$\forall u \in H^1_0(\Omega) : ||u||_{0,\Omega} \leq C H_\Omega |u|_{1,\Omega}$$

Spaces or norms without explicitly stated domain Ω are understood to be defined on the appropriate global domain Ω.

2.2 The space $H(\mathsf{div}, \Omega)$

The *divergence* of a vector field $\mathbf{u} \in \mathbb{R}^n$ is defined as

$$\mathsf{div}\, \mathbf{u} := \sum_{i=1}^{n} \partial_{x_i} u_i$$

$H(\mathsf{div},\Omega)$ is the graph space of div over L^2, i.e.,

$$H(\mathsf{div},\Omega) = \{\mathbf{u} \in (L^2(\Omega))^n | \, \mathsf{div}\, \mathbf{u} \in L^2(\Omega)\}.$$

It is a Hilbert space under the graph norm

$$(\mathbf{u},\mathbf{v})_{\mathsf{div},\Omega} = (\mathbf{u},\mathbf{v})_0 + (\mathsf{div}\,\mathbf{u}, \mathsf{div}\,\mathbf{v})_0 \qquad\qquad ||\mathbf{u}||^2_{\mathsf{div},\Omega} = (\mathbf{u},\mathbf{u})_{\mathsf{div},\Omega}$$

We also need a space with more regularity for the discussion of the approximation properties of the Nédélec interpolant in chapter 7:

$$H^s(\mathsf{div},\Omega) = \{\mathbf{u} \in (H^{s+1}(\Omega))^n | \, \mathsf{div}\, u \in H^s(\Omega)\}$$

The trace operator γ_n that maps a vector field to its normal component on the boundary, and is, e.g., well-defined on the restriction to Ω of the infinitely differentiable functions with compact support in $\mathbf{R}^n$, can be extended to a continuous (and surjective) operator[1]

$$\gamma_n : H(\mathsf{div},\Omega) \mapsto H^{-\frac{1}{2}}(\partial\Omega)$$

For $\mathbf{u} \in H(\mathsf{div},\Omega)$ and $q \in H^1(\Omega)$ we have the following Green's formula:

$$(\mathbf{u}, \mathbf{grad}\, q)_{0,\Omega} + (\mathsf{div}\,\mathbf{u}, q)_{0,\Omega} = \langle \gamma_n(\mathbf{u}), q \rangle_{\frac{1}{2},\partial\Omega}$$

where here and in the following $\langle \cdot,\cdot \rangle_{\frac{1}{2},\partial\Omega}$ denotes the duality pairing between $H^{-\frac{1}{2}}(\partial\Omega)$ and $H^{\frac{1}{2}}(\partial\Omega)$.

Finally, we also need the subspaces

$$H_0(\mathsf{div},\Omega) = \{\mathbf{u} \in H(\mathsf{div},\Omega) | \gamma_n(\mathbf{u}) = 0\}$$

$$H(\mathsf{div}_0,\Omega) = \{\mathbf{u} \in H(\mathsf{div},\Omega) | \, \mathsf{div}\,\mathbf{u} = 0\}$$

$$H_0(\mathsf{div}_0,\Omega) = \{\mathbf{u} \in H_0(\mathsf{div},\Omega) | \, \mathsf{div}\,\mathbf{u} = 0\}$$

2.3 The space $H(\mathrm{curl})$ in two dimensions

Given a two-dimensional vector field $\mathbf{u}$ and a scalar function q of two variables, the following two curl operators can be defined

$$\mathbf{curl}\, q = (\partial_{x_2} q, -\partial_{x_1} q)$$

[1] Recall that we use scaled norms for $H(\mathsf{div},\Omega)$ and $H^{-\frac{1}{2}}(\partial\Omega)$ if H_Ω is not $O(1)$.

$$\operatorname{curl} \mathbf{u} = \partial_{x_1} u_2 - \partial_{x_2} u_1$$

$H(\operatorname{curl}, \Omega)$ is defined as the graph space of curl over $L^2(\Omega)$, and is a Hilbert space:

$$H(\operatorname{curl}, \Omega) = \{\mathbf{u} \in (L^2(\Omega))^2 | \operatorname{curl} \mathbf{u} \in L^2(\Omega)\},$$

$$(\mathbf{u}, \mathbf{v})_{\operatorname{curl},\Omega} = (\mathbf{u}, \mathbf{v})_0 + (\operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v})_0 \qquad ||\mathbf{u}||^2_{\operatorname{curl},\Omega} = (\mathbf{u}, \mathbf{u})_{\operatorname{curl},\Omega}$$

We also need a space with more regularity for the discussion of the approximation properties of the Nédélec interpolant in chapter 7:

$$H^s(\operatorname{curl}, \Omega) = \{\mathbf{u} \in (H^{s+1}(\Omega))^2 | \operatorname{curl} u \in H^s(\Omega)\}$$

A vector $\mathbf{u} = (u_1, u_2)$ belongs to $H(\operatorname{curl}, \Omega)$ if and only if $\mathbf{v} = (-u_2, u_1)$ belongs to $H(\mathsf{div}, \Omega)$. Denoting the unit tangent vector on $\partial\Omega$ with $\mathbf{t}$, we have $\mathbf{v} \cdot \mathbf{n} = -\mathbf{u} \cdot \mathbf{t}$ and $\operatorname{curl} \mathbf{u} = \mathsf{div}\, \mathbf{v}$. Therefore we can use the results of the previous section to show that $\gamma_t(\mathbf{u}) = \mathbf{u} \cdot \mathbf{t}|_{\partial\Omega}$ can be extended to a continuous (and surjective) operator

$$\gamma_t : H(\operatorname{curl}, \Omega) \mapsto H^{-\frac{1}{2}}(\partial\Omega)$$

A Green's formula can be proven for $\mathbf{u} \in H(\operatorname{curl}, \Omega)$ and $q \in H^1(\Omega)$

$$(\operatorname{curl} \mathbf{u}, q)_{0,\Omega} + (\mathbf{u}, \mathbf{curl}\, q)_{0,\Omega} = \langle \gamma_t(\mathbf{u}), q \rangle_{\frac{1}{2},\partial\Omega}$$

To allow us to state some results concisely for the two-dimensional and three-dimensional case in the same formula, we denote $\underset{2}{\mathbf{curl}} := \operatorname{curl}$.

2.4 The space $H(\mathbf{curl})$ in three dimensions

The curl vector operator is defined for a three-dimensional vector field $\mathbf{u}$ as

$$\mathbf{curl}\, \mathbf{u} := (\partial_{x_2} u_3 - \partial_{x_3} u_2, \partial_{x_3} u_1 - \partial_{x_1} u_3, \partial_{x_1} u_2 - \partial_{x_2} u_1)^T$$

$H(\mathbf{curl})$ is a Hilbert space with the graph norm and inner product:

$$H(\mathbf{curl}, \Omega) = \{\mathbf{u} \in (L^2(\Omega))^3 | \mathbf{curl}\, \mathbf{u} \in (L^2(\Omega))^3\},$$

$$(\mathbf{u}, \mathbf{v})_{\mathbf{curl},\Omega} = (\mathbf{u}, \mathbf{v})_0 + (\mathbf{curl}\, \mathbf{u}, \mathbf{curl}\, \mathbf{v})_0 \qquad ||\mathbf{u}||^2_{\mathbf{curl},\Omega} = (\mathbf{u}, \mathbf{u})_{\mathbf{curl},\Omega}$$

We also need a space with more regularity for the discussion of the approximation properties of the Nédélec interpolant in chapter 7:

$$H^s(\mathbf{curl}, \Omega) = \{\mathbf{u} \in (H^{s+1}(\Omega))^3 | \mathbf{curl}\, u \in (H^s(\Omega))^3\}$$

In the treatment of Maxwell's equations on non-convex Lipschitz domains, the following space plays a role (see the comments in section 7.1.2 and at the end of chapter 7):

$$X^p(\Omega) := \{\mathbf{u} \in (L^p(\Omega))^3, \mathbf{curl}\,\mathbf{u} \in (L^p(\Omega))^3, \mathbf{u} \times \mathbf{n} \in (L^p(\partial\Omega))^3\} \quad (d = 3).$$

The tangential components on the boundary $\partial\Omega$ can be defined as

$$\gamma_t(\mathbf{u}) = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n})\mathbf{n} = (\mathbf{n} \times \mathbf{u}) \times \mathbf{n}$$

We will also sometimes call $\mathbf{n} \times \mathbf{u}$ the tangential components even though we have only $|\mathbf{n} \times \mathbf{u}| = |\gamma_t(\mathbf{u})|$.

By extending the tangential trace operator γ_t, which certainly is well-defined and continuous for smooth enough $\mathbf{u}$, we can find a continuous operator

$$\gamma_t : H(\mathbf{curl}, \Omega) \mapsto (H^{-\frac{1}{2}}(\partial\Omega))^3$$

which is not surjective. Its range has been fully characterized (see, for instance, Alonso and Valli [3]).

A Green's formula can be proven for $\mathbf{u}, \mathbf{v} \in H(\mathbf{curl}, \Omega)$:

$$(\mathbf{curl}\,\mathbf{u}, \mathbf{v})_{0,\Omega} + (\mathbf{u}, \mathbf{curl}\,\mathbf{v})_{0,\Omega} = \langle \gamma_t(\mathbf{u}), q \rangle_{\frac{1}{2},\partial\Omega}$$

To allow us to state some results concisely for the two-dimensional and three-dimensional case in the same formula, we denote $\underset{3}{\mathbf{curl}} := \mathbf{curl}$.

We define the subspaces

$$H_0(\underset{n}{\mathbf{curl}}, \Omega) = \{\mathbf{u} \in H(\underset{n}{\mathbf{curl}}, \Omega) | \gamma_t(\mathbf{u}) = 0\}$$

$$H(\underset{n}{\mathbf{curl}}_0, \Omega) = \{\mathbf{u} \in H(\underset{n}{\mathbf{curl}}, \Omega) |\, \underset{n}{\mathbf{curl}}\,\mathbf{u} = 0\}$$

$$H_0(\underset{n}{\mathbf{curl}}_0, \Omega) = \{\mathbf{u} \in H_0(\underset{n}{\mathbf{curl}}, \Omega) |\, \underset{n}{\mathbf{curl}}\,\mathbf{u} = 0\}$$

2.5 Helmholtz decompositions

Theorem 2.3 (Orthogonal decompositions in $(L^2(\Omega))^n$) *The space $(L^2(\Omega))^n$ allows the following orthogonal decompositions*

$$(L^2(\Omega))^n = H(\mathsf{div}_0, \Omega) \oplus \mathbf{grad}\, H_0^1(\Omega)$$

$$(L^2(\Omega))^n = H_0(\mathsf{div}_0, \Omega) \oplus \mathbf{grad}\, H^1(\Omega)$$

$$(L^2(\Omega))^n = H_0(\mathsf{div}_0, \Omega) \oplus \mathbf{grad}\, \mathcal{H}^1(\Omega) \oplus \mathbf{grad}\, H_0^1(\Omega)$$

where $\mathcal{H}^1(\Omega)$ is the space of harmonic functions in $H^1(\Omega)$.

Proof: Dautray and Lions [34, Proposition 1 on page 215]. ∎

This theorem implies similar theorems in $H(\mathsf{div},\Omega)$ and $H(\underset{n}{\mathbf{curl}},\Omega)$, which we give below:

Theorem 2.4 (Orthogonal decompositions in $H(\mathsf{div},\Omega)$) *The* div *graph spaces allow the following orthogonal decompositions*

$$H(\mathsf{div},\Omega) = H(\mathsf{div}_0,\Omega) \oplus H^{\perp}(\mathsf{div},\Omega)$$

$$H_0(\mathsf{div},\Omega) = H_0(\mathsf{div}_0,\Omega) \oplus H_0^{\perp}(\mathsf{div},\Omega)$$

with

$$H^{\perp}(\mathsf{div},\Omega) = H(\mathsf{div},\Omega) \cap \mathbf{grad}\, H_0^1(\Omega)$$

$$H_0^{\perp}(\mathsf{div},\Omega) = H_0(\mathsf{div},\Omega) \cap \mathbf{grad}\, H^1(\Omega)$$

The two decompositions are orthogonal in both $(\cdot,\cdot)_{0,\Omega}$ and $(\cdot,\cdot)_{\mathsf{div},\Omega}$, and this implies

$$\forall H^{\perp}(\mathsf{div},\Omega) \cup H_0^{\perp}(\mathsf{div},\Omega) : ||\mathbf{u}||_{0,\Omega} \leq C H_{\Omega} ||\,\mathsf{div}\,\mathbf{u}||_{0,\Omega}$$

Theorem 2.5 (Orthogonal decompositions in $H(\underset{n}{\mathbf{curl}},\Omega)$) *$H(\underset{n}{\mathbf{curl}},\Omega)$ allows the following orthogonal decompositions*

$$H(\underset{n}{\mathbf{curl}},\Omega) = \mathbf{grad}\, H_0^1(\Omega) \oplus H^{\perp}(\underset{n}{\mathbf{curl}},\Omega)$$

$$H_0(\underset{n}{\mathbf{curl}},\Omega) = \mathbf{grad}\, H^1(\Omega) \oplus H_0^{\perp}(\underset{n}{\mathbf{curl}},\Omega)$$

with

$$H^{\perp}(\underset{n}{\mathbf{curl}},\Omega) = H(\mathsf{div},\Omega) \cap H_0(\underset{n}{\mathbf{curl}},\Omega)$$

$$H_0^{\perp}(\underset{n}{\mathbf{curl}},\Omega) = H_0(\mathsf{div},\Omega) \cap H(\underset{n}{\mathbf{curl}},\Omega)$$

The two decompositions are orthogonal in both $(\cdot,\cdot)_{0,\Omega}$ and $(\cdot,\cdot)_{\mathbf{curl},\Omega}$.

For simply connected Ω we have

$$H(\underset{n}{\mathbf{curl}}_0,\Omega) = \mathbf{grad}\, H^1(\Omega)$$

For multiply connected Ω this hold with a $\subset$, and the complement has been characterized (see, e.g., Dautray and Lions [34]).

We end this section with

Theorem 2.6 (Friedrichs' inequality for $H^{\perp}(\mathbf{curl}, \Omega)$) *If Ω is simply connected, then the following inequality holds:*

$$\forall \mathbf{u} \in H^{\perp}(\underset{n}{\mathbf{curl}}, \Omega) : ||\mathbf{u}||_{0,\Omega} \leq C H_{\Omega} || \underset{n}{\mathbf{curl}}\, \mathbf{u}||_{0,\Omega}$$

If, in addition, $\partial\Omega$ is connected, the same inequality holds for the space with vanishing tangential components:

$$\forall \mathbf{u} \in H_0^{\perp}(\underset{n}{\mathbf{curl}}, \Omega) : ||\mathbf{u}||_{0,\Omega} \leq C H_{\Omega} || \underset{n}{\mathbf{curl}}\, \mathbf{u}||_{0,\Omega}$$

For some extensions to multiply connected domains and to domains with boundaries consisting of several connected components, see Dautray and Lions [34] and Amrouche, Bernardi, Dauge, and Girault [5].

2.6 Regularity of the Laplace operator

In this section we will present several regularity results for the Dirichlet and Neuman problem for the Laplace operator. We will need them in the proof of the regularity of the **curl** potentials in the last section.

We will first discuss the case of Dirichlet boundary conditions. The Dirichlet problem is

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \\ u &= g && \text{on } \partial\Omega \end{aligned} \tag{2.1}$$

The following regularity results are well-known:

Theorem 2.7 *Assume that Ω is a bounded, open subset of $\mathbb{R}^n$ with a $C^{k+1,1}$ boundary. Then for $p \in (1, \infty)$, the solution operator $(f, g) \rightarrow u$ for (2.1) is continuous on*

$$W^{k,p}(\Omega) \times W^{k+2-\frac{1}{p},p}(\Omega) \mapsto W^{k+2,p}(\Omega)$$

Proof: See Girault and Raviart [48, theorem I.1.8,1)]. ∎

Theorem 2.8 *Assume that Ω is a two-dimensional, bounded polygon with no reentrant corner. Then there exists a real p_{Ω} depending on the greatest inner angle of $\partial\Omega$ such that $u \in W^{2,p}(\Omega)$ for $p \in (1, p_{\Omega})$ whenever $f \in L^p(\Omega)$ and $g \in \gamma_0(W^{2,p}(\Omega))$. The result is still true for $g = 0$ for any convex bounded polyhedron in the three-dimensional case.*

Proof: See Girault and Raviart [48, theorem I.1.8,2) and 3)]. ∎

For the three-dimensional homogenous case ($g = 0$) and H^s spaces we have more information about the exact regularity, see Dauge [31, Corollary 18.18]:

Theorem 2.9 *Suppose Ω is a three-dimensional, bounded Lipschitz polyhedron with ω being the largest angle between its faces. For $s \neq -\frac{1}{2}$ and*

$$s < \min\left\{\frac{3}{2}, \frac{\pi}{\omega} - 1\right\},$$

the Laplace operator is an isomorphism:

$$\Delta : H^{2+s}(\Omega) \cap H_0^1(\Omega) \leftrightarrow H^s(\Omega)$$

The Neumann problem is

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \\ \partial_{\mathbf{n}} u &= g && \text{on } \partial\Omega \end{aligned} \tag{2.2}$$

The data f and g have to satisfy the compatibility condition

$$\int_\Omega f + \int_{\partial\Omega} g = 0$$

if (2.2) is to have a solution.

Since (2.2) only involves derivatives of u, it will never be uniquely solvable. Therefore we work in the quotient spaces over $\mathbb{R}$ using the standard quotient norm . We remark that the quotient norm in $H^1/\mathbb{R}$ is equivalent to the H^1-seminorm.

We state the analogues of the first two theorems above, and note that exact regularity results for the Neumann problem are known. We skip their statement for reasons of space and refer to the literature.

Theorem 2.10 *Assume that Ω is a bounded, open subset of $\mathbb{R}^n$ with a $C^{k+1,1}$ boundary. Then the solution operator $(f, g) \to u$ for (2.2) is continuous on*

$$W^{k,p}(\Omega) \times W^{k+1-\frac{1}{p},p}(\Omega) \mapsto W^{k+2,p}(\Omega)/\mathbb{R}$$

Proof: See Girault and Raviart [48, theorem I.1.10,1)]. ∎

Theorem 2.11 *Assume that Ω is a two-dimensional, bounded polygon with no reentrant corner. Then there exists a real p_Ω depending on the greatest inner angle of $\partial\Omega$ such that $u \in W^{2,p}(\Omega)/\mathbb{R}$ for $p \in (1, p_\Omega)$ whenever $f \in L^p(\Omega)$ and $g \in \gamma_0(W^{1,p}(\Omega))$. The result is still true for $g = 0$ for any convex bounded polyhedron in the three-dimensional case.*

Proof: See Girault and Raviart [48, theorem I.1.10,2) and 3)]. ∎

2.7 Imbedding theorems

In this section, we will discuss under which circumstances we can infer that $\mathbf{u} \in H^s(\Omega)$ from $\mathbf{u} \in H(\underset{n}{\mathbf{curl}}, \Omega)$ and $\mathbf{u} \in H(\mathsf{div}, \Omega)$. In some sense, div and $\mathbf{curl}$ already cover all the directional derivatives, and so, locally, we should obtain the result with $s = 1$. We will see that this suspicion is correct and also that s will depend strongly on the imposed boundary conditions and on the convexity (or the measure of the reentrant corners) of Ω. All the results stated in this section are proven or referenced in Amrouche, Bernardi, Dauge, and Girault [5], and we refer to this article for the proofs and further comments. We state all the results for the three-dimensional case. The two-dimensional case is discussed in Girault and Raviart [48, section 3.2] and complete results could be obtained combining their methods and regularity results for non-convex polygons.

First, we need to introduce some spaces:

$$H(\Omega) = H(\mathbf{curl}, \Omega) \cap H(\mathsf{div}, \Omega)$$

with the norm

$$||\mathbf{u}||^2_{H(\Omega)} := ||\mathbf{u}||^2_{0,\Omega} + ||\,\mathbf{curl}\,\mathbf{u}||^2_{0,\Omega} + ||\,\mathsf{div}\,\mathbf{u}||^2_{0,\Omega}$$

and the following spaces with different boundary behavior:

$$H_T(\Omega) = \{\mathbf{u} \in H(\Omega) | \gamma_t(\mathbf{u}) = 0\}$$

$$H_N(\Omega) = \{\mathbf{u} \in H(\Omega) | \gamma_n(\mathbf{u}) = 0\}$$

$$H_0(\Omega) = H_T(\Omega) \cap H_N(\Omega)$$

The following results are known:

Theorem 2.12 (Vanishing boundary components) *$H_0(\Omega)$ coincides with $(H^1_0(\Omega))^3$.*

Theorem 2.13 (Imbeddings in $(L^2(\Omega))^n$) *The imbedding of $H(\Omega)$ into $(L^2(\Omega))^n$ is not compact. The imbeddings of $H_T(\Omega)$ and $H_N(\Omega)$ into $(L^2(\Omega))^n$ are compact.*

Theorem 2.14 (Smooth boundaries or convex domains) *If $\partial\Omega$ is of class $C^{1,1}$ or if Ω is convex, then $H_T(\Omega)$ and $H_N(\Omega)$ are continuously imbedded in $(H^1_0(\Omega))^3$. This also holds for inhomogenous boundary components in $(H^{\frac{1}{2}}(\partial\Omega))^3$ or $H^{\frac{1}{2}}(\partial\Omega)$, respectively, and in the two-dimensional case.*

Counterexamples to this theorem are known for Lipschitz domains.

Theorem 2.15 (Lipschitz boundaries) *If $\partial\Omega$ is Lipschitz, then there exists a real number $s > \frac{1}{2}$ such that $H_T(\Omega)$ and $H_N(\Omega)$ are continuously imbedded in $(H^s(\Omega))^3$. This also holds for inhomgenous boundary components in $(L^2(\Omega))^3$ or $L^2(\Omega)$, respectively.*

2.8 Regularity of curl potentials

In this section we will give some regularity results for the div-**curl** problem and for **curl** potentials.

First we discuss the div-**curl** problem in three dimensions. Improved and exact results could be obtained for the two-dimensional case using the Laplace problems for the div- and **curl**-potential and applying the known regularity results on polygons.

Theorem 2.16 (div-**curl in** H^s**)** *Assume that* Ω *is a bounded convex polyhedron. Then there exists a* $s_\Omega \in (0, \frac{1}{2})$ *such that for all* $s \in [0, s_\Omega)$ *and* $\mathbf{v} \in H_N(\Omega)$ *with*

$$\mathbf{curl}\,\mathbf{v} \in (H^s(\Omega))^3 \qquad \mathrm{div}\,\mathbf{v} \in H^s(\Omega)$$

we have the added regularity $\mathbf{v} \in (H^{s+1}(\Omega))^3$.

Proof: See Toselli [96, Theorem 2.1.1]. s_Ω is the maximal s for which the Dirichlet problem is regular in $H^s \to H^{2+s}$, see (2.1) and theorem 2.9. ∎

Theorem 2.17 (div-**curl in** L^p**)** *Assume that* Ω *is a bounded convex polyhedron. Then there exist a* $p_\Omega > 2$ *such that for all* $p \in [2, p_\Omega)$ *and* $\mathbf{v} \in H_N(\Omega)$ *or* $\mathbf{v} \in H_T(\Omega)$ *with*

$$\mathbf{curl}\,\mathbf{v} \in (L^p(\Omega))^3 \qquad \mathrm{div}\,\mathbf{v} \in L^p(\Omega)$$

we have the added regularity $\mathbf{v} \in (W^{1,p}(\Omega))^3$.

Proof: See Amrouche, Bernardi, Dauge, and Girault [5, Remark 2.19]. p_Ω is the maximal p for which the Dirichlet respective Neumann problem is regular in L^p, see (2.1) and theorem 2.8; or (2.2) and theorem 2.11, respectively. ∎

We give nowfour regularity results for the $\underset{n}{\mathbf{curl}}$-potential. We assume for simplicity that Ω is simply connected, and, in the three-dimensional case, that $\partial\Omega$ is connected. For extensions of the results to more general cases, see Girault and Raviart [48] and Amrouche, Bernardi, Dauge, and Girault [5].

Theorem 2.18 (H^s**, two dimensions)** *If* $\mathbf{u} \in (H^s(\Omega))^2$ *for* $s > 0$ *and* $\mathrm{div}\,\mathbf{u} = 0$, *then* $\mathbf{u} = \mathbf{curl}\, v$ *with* $v \in H^{s+1}(\Omega)$.

Proof: The result is given for s integer on Girault and Raviart [48, page 39]. We use (Hilbert space) interpolation between $\lfloor s \rfloor$ and $\lceil s \rceil$ to extend it to the case of general s. ∎

Theorem 2.19 (L^p, two dimensions) *If* $\mathbf{u} \in (L^p(\Omega))^2$ *for* $p \geq 2$ *and* $\operatorname{div} \mathbf{u} = 0$, *then* $\mathbf{u} = \mathbf{curl}\, v$ *with* $v \in W^{1,p}(\Omega)$.

Proof: See Girault and Raviart [48, page 39]. ∎

Theorem 2.20 (H^s, three dimensions) *If* $\mathbf{u} \in (H^s(\Omega))^3$ *for* $s \in [0,1]$ *and* $\operatorname{div} \mathbf{u} = 0$, *then* $\mathbf{u} = \mathbf{curl}\, \mathbf{v}$ *with* $\mathbf{v} \in (H^{s+1}(\Omega))^3$.

Proof: See Girault and Raviart [48, Remark I.3.12]. ∎

Theorem 2.21 (L^p, three dimensions) *Assume* Ω *is a bounded, convex polyhedron. If* $\mathbf{u} \in (L^p(\Omega))^3$ *for some* $p > 2$, $\operatorname{div} \mathbf{u} = 0$ *and* $\gamma_n(\mathbf{u}) = 0$, *then* $\mathbf{u} = \mathbf{curl}\, \mathbf{v}$ *with* $\mathbf{v} \in (W^{1,r}(\Omega))^3$ *for* $r \in (2, p]$.

Proof: See Girault and Raviart [48, Remark I.3.14]. ∎

Chapter 3

The model problem

In this chapter we will explain how one can obtain a problem of the form

$$?\mathbf{u} \in V : \forall \mathbf{v} \in V : (\alpha \mathbf{u}, \mathbf{v}) + (\beta\, \mathbf{curl}\, \mathbf{u}, \mathbf{curl}\, \mathbf{v}) + b.t. = \mathbf{f}(\mathbf{v}) \qquad (MP)$$

in the solution of electromagnetic problems. ($b.t.$ stands for boundary terms)

3.1 Maxwell's equations, reformulations

Maxwell's equations model the behavior of electromangetic waves and their interaction with matter.[1] They read in a linear material with dielectric permittivity ϵ, magnetic permeability μ and electric conductivity σ:

$$\begin{aligned}
\mathbf{curl}\, \mathbf{H} &= \mathbf{j} + \partial_t \mathbf{D} && (3.1)\\
\mathbf{curl}\, \mathbf{E} &= -\partial_t \mathbf{B} && (3.2)\\
\mathsf{div}\, \mathbf{D} &= \rho && (3.3)\\
\mathsf{div}\, \mathbf{B} &= 0 && (3.4)\\
\mathbf{D} &= \epsilon \mathbf{E} && (3.5)\\
\mathbf{B} &= \mu \mathbf{H} && (3.6)\\
\mathbf{j} &= \sigma \mathbf{E} + \mathbf{j}_i && (3.7)
\end{aligned}$$

[1] For a mathematical treatment of several problems connected with Maxwell's equation see Dautray and Lions [33, 32, 34, 35, 36, 37]

where $\mathbf{E}(\mathbf{x}, t)$ is the electric field, $\mathbf{B}(\mathbf{x}, t)$ is the magnetic induction, $\mathbf{D}(\mathbf{x}, t)$ is the electric flux density, $\mathbf{H}(\mathbf{x}, t)$ is the magnetic field, $\mathbf{j}(\mathbf{x}, t)$ is the electric current, $\mathbf{j}_i(\mathbf{x}, t)$ is an intrinsic current, and $\rho(\mathbf{x}, t)$ is the space charge density.

Equations (3.1) is the Maxwell-Ampère law, equation (3.2) is Faraday's law, equation (3.3) is Gauss' electrical law, and equation (3.4) is Gauss' magnetic law. (3.5), (3.6) and (3.7) are material laws, also known as constitutive relations. In the general case those material laws could be nonlinear, where nonlinear magnetic effects are more common and occur under normal circumstances, while nonlinear electric effects often occur in the very high energy case, for instance in second-harmonic generation with lasers in nonlinear optics. Even with linear material laws, the propagation of electromagnetic waves could be different in different directions, as it happens in a crystal, where the material properties $\epsilon(\mathbf{x})$, $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are tensors. ϵ, μ and σ are nonnegative because of their physical interpretation. In our work we will only work with isotropic materials, so that ϵ, μ and σ are only scalar functions of the spatial variable $\mathbf{x}$. We will restrict ourselves mostly to homogeneous materials and to nonhomogeneous materials with piecewise constant or piecewise separable material properties, but we could model in a similar way piecewise smooth material properties. The electromagnetic properties of the material can change with time, like the conductivity in microwave heating, and we would need some other model, maybe in form of a partial differential equation to take those effects into account. For instance in microwave heating we would have to couple the Maxwell system with a nonlinear heat equation and a model how the temperature affects the conductivity (see Yin [105]), but we will not treat such models here.

In general we can assume that the permitivity and the permeability are positive functions, bounded from below away from zero with uniform constant lower bounds ϵ_0 and μ_0, respectively; but in general we can only assume $\sigma \geq 0$ almost everywhere.

Usually the Maxwell system is written in two fields; we will write it in $\mathbf{B}$ and $\mathbf{E}$. In certain circumstances, like nonlinear or more complicated material laws, it may be preferable to formulate the Maxwell system in all four fields.

We can pose an initial value problem or an initial and boundary value problem. Then we will enforce initial conditions at time $t = 0$ and possibly some boundary conditions. It is also possible to pose stationary (i.e., problems that do not contain time derivatives) problems in frequency space or to look for solutions with certain behavior at infinity.

If both Gauss' laws (3.3) and (3.4) hold at the initial time, they continue to hold at any later time. Therefore those two equations are not needed for the evolution in the continuous formulation. If it is necessary that the numerical solutions obey Gauss' laws exactly, we have to make sure that the numerical schemes satisfy the divergence constraints exactly, so that in that case the laws may be not redundant. Assuming the solution procedures conserve

the divergences, we have to integrate

$$-\epsilon\partial_t \mathbf{E} - \sigma\mathbf{E} + \mathbf{curl}\,\frac{1}{\mu}\mathbf{B} = \mathbf{j}_i \tag{3.8}$$

$$\partial_t \mathbf{B} + \mathbf{curl}\,\mathbf{E} = \mathbf{0} \tag{3.9}$$

for instance under the initial conditions

$$\mathbf{B}|_{t=0} = \mathbf{B}_0 \tag{3.10}$$

$$\mathbf{E}|_{t=0} = \mathbf{E}_0 \tag{3.11}$$

The conditions on the divergences could be explicitly enforced by the introduction of Lagrangian multipliers. Since this requires the introduction of more fields, and leads to saddle point problems with several fields, and not to a problem of type (MP), we will not consider such methods here.

For $\epsilon = \mu = 1$, $\sigma = 0$, Maxwell's equations are a first order symmetric hyperbolic system (see for instance Garabedian [46, pg. 100], Feng [45], and Dautray and Lions [33, pg. 95-96]). For constant ϵ and μ and assuming $\sigma = 0$ we can introduce a change of variables to obtain the system for $\epsilon = \mu = 1$ (see for instance Feng [45]). In general Maxwell's equations will show both hyperbolic and parabolic properties, we can obtain both hyperbolic and parabolic equations for certain choices for ϵ, μ and σ, possibly interpreted as limit cases.

A priori the Maxwell equations are posed over the entire space $\mathbb{R}^3$, but we can impose exact or approximative boundary conditions on interfaces with perfect conductors or other bodies with idealized electromagnetic properties; radiation, absorbing, symmetry or reflecting boundary conditions to truncate the a priori infinite domains; or boundary conditions to enforce known physical conditions, like ambient illumination, light being guided into a coupler between optical fibers, or laser light pumped into the system.

An alternative to the use of boundary conditions to truncate the domain would be to discretize the infinite exterior domain by infinite elements or by an integral equation or boundary elements on the boundary, but we will not consider such approaches in this work.

At an interface inside the domain we have that the tangential components of $\mathbf{E}$ and $\mathbf{H}$ and the normal components of $\mathbf{D}$ and $\mathbf{B}$ are continuous, if we assume that there are no surface electric current densities and surface charge densities present on the interface. Should there be a surface electric current density $\mathbf{j}_s$ and a surface charge density ρ_s, we will have a jump of size $\mathbf{j}_s$ in the tangential components of $\mathbf{H}$ and a jump of size ρ_s in the normal component of $\mathbf{D}$, while the tangential components of $\mathbf{E}$, and the normal component of $\mathbf{B}$ are still continuous.

A perfect conductor is an idealized material that cannot sustain an electric field, i.e., electrical charges move instantaneous so that they are always in equilibrium with a zero electric field inside the material, so that we obtain the following two boundary conditions on the boundary Γ of the perfect conductor,

$$\mathbf{B} \cdot \mathbf{n}|_\Gamma = 0 \tag{3.12}$$

$$\mathbf{E} \times \mathbf{n}|_\Gamma = \mathbf{0} \tag{3.13}$$

enforcing a zero normal component of the induction $\mathbf{B}$ and zero tangential components of the electric field $\mathbf{E}$.

Similar to Gauss' laws above, the interface conditions or perfect conductor boundary conditions continue to hold if they hold at an initial time.

If we know the electric and magnetic field inside one of the domains, the interface conditions give boundary conditions for the other domain, enforcing a nonhomogeneous normal component on the induction and nonhomogeneous tangential components on the electric field. A special case is if we use that to model the reaction of the system to an "incoming" electric field from infinity by enforcing it on a boundary "far" from the origin.

In a very similar way we can also treat a boundary condition where we know the induction in a non-tangential direction or the electric field in two directions.

If we formulate the system as a second order system in only one of the two fields, boundary conditions on the other field turn into boundary conditions on the curl of the considered field, taking the appropriate time derivatives, and using (3.1) and (3.2).

There are also (approximate) radiation boundary conditions that force fields to be either outgoing or incoming (or absorbing boundary conditions that absorb outgoing radiation).

One example is the Silver-Müller boundary condition

$$\left(\mathbf{E} - \frac{1}{\sqrt{\mu\epsilon}}\mathbf{B} \times \mathbf{n}\right) \times \mathbf{n} = \mathbf{0} \tag{3.14}$$

If we want to compute a scattering problem we could use this condition to enforce that the part of the electromagnetic fields that is not the incoming wave is outgoing on the piece of the boundary where Silver-Müller conditions are enforced, by substituting $\mathbf{E} - \mathbf{E}_{inc}$ for $\mathbf{E}$, or equivalently changing the right hand side of the condition from $\mathbf{0}$ to $\mathbf{E}_{inc} \times \mathbf{n}$. Some implementations also allow for $\mathbf{E}$ to be multiplied by some real or complex constant, but we will not do that for the sake of simplicity. The extension to that case is straightforward.

In certain circumstances one of the two fields $\mathbf{E}$ or $\mathbf{B}$ is of more importance than the other field, and it is possible to reformulate Maxwell's system to obtain a second order evolution equation only in one of the fields.

If we are only interested in $\mathbf{E}$, we obtain:

$$\epsilon\partial_t^2\mathbf{E} + \sigma\partial_t\mathbf{E} + \mathbf{curl}\,\frac{1}{\mu}\,\mathbf{curl}\,\mathbf{E} = \partial_t\mathbf{j}_i \tag{3.15}$$

with the initial conditions

$$\mathbf{E}|_{t=0} = \mathbf{E}_0 \tag{3.16}$$

$$\partial_t\mathbf{E}|_{t=0} = \frac{1}{\epsilon}\,\mathbf{curl}\,\frac{1}{\mu}\mathbf{B}_0 - \sigma\mathbf{E}_0 - \mathbf{j}_i|_{t=0} \tag{3.17}$$

To recover $\mathbf{B}$, we have to integrate

$$\partial_t\mathbf{B} = -\,\mathbf{curl}\,\mathbf{E} \tag{3.18}$$

$$\mathbf{B}|_{t=0} = \mathbf{B}_0 \tag{3.19}$$

Perfect conductor boundary conditions turn into conditions on the tangential components of $\mathbf{E}$ or alternatively into conditions on the normal component of $\mathbf{curl}\,\mathbf{E}$ if we assume that the initial data satisfies the boundary condition, i.e., on a piece of the boundary with a perfect conductor Γ_c we obtain

$$\mathbf{E}\times\mathbf{n}|_{\Gamma_c} = \mathbf{0} \tag{3.20}$$

$$\mathbf{curl}\,\mathbf{E}\cdot\mathbf{n}|_{\Gamma_c} = 0 \tag{3.21}$$

If we know the electromagnetic fields in the material on the other side of the boundary to be $\mathbf{E}_i$ and $\mathbf{B}_i$, those conditions change into

$$\mathbf{E}\times\mathbf{n}|_{\Gamma_i} = \mathbf{E}_i\times\mathbf{n}|_{\Gamma_i} \tag{3.22}$$

$$\mathbf{curl}\,\mathbf{E}\cdot\mathbf{n}|_{\Gamma_i} = -\partial_t\mathbf{B}_i\cdot\mathbf{n}|_{\Gamma_i} \tag{3.23}$$

The Silver-Müller conditions are equivalent (if they hold at the initial time) to

$$\left(\partial_t\mathbf{E} + \frac{1}{\sqrt{\mu\epsilon}}\,\mathbf{curl}\,\mathbf{E}\times\mathbf{n}\right)\times\mathbf{n} = \mathbf{0} \tag{3.24}$$

If we are only interested in $\mathbf{B}$, we obtain similarly a second order wave equation in $\mathbf{B}$ for two special cases. First, if we assume constant ϵ and σ, we get:[2]

[2] Dautray and Lions [33, pg. 85] give

$$\partial_t\mathbf{B} + \mathbf{curl}\,\frac{1}{\sigma}\,\mathbf{curl}\,\frac{1}{\mu}\mathbf{B} = 0 \qquad \mathsf{div}\,\mathbf{B} = 0$$

as a model of magnetic induction in a plasma. This equation can be treated similarly to the equations we treat here.

$$\epsilon\partial_t^2\mathbf{B} + \sigma\partial_t\mathbf{B} + \mathbf{curl}\,\mathbf{curl}\,\frac{1}{\mu}\mathbf{B} = \mathbf{curl}\,\mathbf{j}_i \tag{3.25}$$

In the other case, assuming $\sigma = 0$ and a general, time-independent ϵ, we obtain:

$$\partial_t^2\mathbf{B} + \mathbf{curl}\,\frac{1}{\epsilon}\,\mathbf{curl}\,\frac{1}{\mu}\mathbf{B} = \mathbf{curl}\,\frac{1}{\epsilon}\mathbf{j}_i \tag{3.26}$$

Both of these formulations have to be solved with the initial conditions

$$\mathbf{B}|_{t=0} = \mathbf{E}_0 \tag{3.27}$$
$$\partial_t\mathbf{B}|_{t=0} = -\,\mathbf{curl}\,\mathbf{E}_0 \tag{3.28}$$

To recover $\mathbf{E}$ we have to integrate

$$\partial_t\mathbf{E} + \frac{\sigma}{\epsilon}\mathbf{E} = \frac{1}{\epsilon}\,\mathbf{curl}\,\frac{1}{\mu}\mathbf{B} - \frac{1}{\epsilon}\mathbf{j}_i \tag{3.29}$$
$$\mathbf{E}|_{t=0} = \mathbf{E}_0 \tag{3.30}$$

Perfect conductor boundary conditions give a condition on the normal component of $\mathbf{B}$, or alternatively, a condition on the tangential components of $\mathbf{curl}\,\frac{1}{\mu}\mathbf{B}$, if the condition is satisfied for the initial data:

$$\mathbf{B}\cdot\mathbf{n}|_{\Gamma_c} = 0 \tag{3.31}$$
$$\mathbf{curl}\,\frac{1}{\mu}\mathbf{B}\times\mathbf{n}|_{\Gamma_c} = \mathbf{0} \tag{3.32}$$

If we know $\mathbf{B}$ and $\mathbf{E}$ on the other side of the boundary to be $\mathbf{E}_i$ and $\mathbf{B}_i$, the boundary conditions read

$$\mathbf{B}\cdot\mathbf{n}|_{\Gamma_i} = \mathbf{B}_i\cdot\mathbf{n}|_{\Gamma_i} \tag{3.33}$$
$$\mathbf{curl}\,\frac{1}{\mu}\mathbf{B}\times\mathbf{n}|_{\Gamma_i} = \mathbf{curl}\,\frac{1}{\mu}\mathbf{B}_i\times\mathbf{n}|_{\Gamma_i} \tag{3.34}$$

The Silver-Müller boundary conditions correspond to the following boundary conditions on $\mathbf{B}$

$$\left(\mathbf{curl}\,\frac{1}{\mu}\mathbf{B} - \frac{1}{\sqrt{\mu\epsilon}}\left(\epsilon\partial_t\mathbf{B} - \sigma\mathbf{B}\right)\times\mathbf{n}\right)\times\mathbf{n} = \left(\mathbf{j}_i\times\mathbf{n}\right) \tag{3.35}$$

Another large area of applications for Maxwell's equations is in the design of devices with certain frequency-dependent behaviors, like waveguides or cavities. For those applications

it makes sense to consider the Maxwell system in the frequency domain on one mode of angular frequency ω, the so-called time-harmonic Maxwell equations.

Substituting a sinusoidal time dependence

$$\mathbf{E}(\mathbf{x}, t) = \hat{\mathbf{E}}(\mathbf{x}) \exp(i\omega t) \tag{3.36}$$

$$\mathbf{B}(\mathbf{x}, r) = \hat{\mathbf{B}}(\mathbf{x}) \exp(i\omega t) \tag{3.37}$$

we obtain the time-harmonic Maxwell system:

$$-(\epsilon i\omega + \sigma)\,\hat{\mathbf{E}} + \mathbf{curl}\,\frac{1}{\mu}\hat{\mathbf{B}} = \hat{\mathbf{j}}_i \tag{3.38}$$

$$i\omega\hat{\mathbf{B}} + \mathbf{curl}\,\hat{\mathbf{E}} = \mathbf{0} \tag{3.39}$$

We can also formulate the time-harmonic version of the second order system for $\mathbf{E}$:

$$\left(\sigma i\omega - \epsilon\omega^2\right)\hat{\mathbf{E}} + \mathbf{curl}\,\frac{1}{\mu}\,\mathbf{curl}\,\hat{\mathbf{E}} = \hat{f} \tag{3.40}$$

Similarly we obtain the time-harmonic equation in $\mathbf{B}$:

$$\left(\sigma i\omega - \epsilon\omega^2\right)\hat{\mathbf{B}} + \mathbf{curl}\,\mathbf{curl}\,\frac{1}{\mu}\hat{\mathbf{B}} = \hat{f} \tag{3.41}$$

In both cases $\hat{f}$ represents a forcing term derived from the right hand side of the original equation. The coefficient of $\hat{\mathbf{B}}$ and $\hat{\mathbf{E}}$ is also sometimes written

$$\left(\sigma i\omega - \epsilon\omega^2\right) = -\omega^2\epsilon' \qquad \text{with} \qquad \epsilon' = \epsilon - \frac{\sigma i}{\omega} \tag{3.42}$$

On those systems we can impose the boundary conditions of the original system, only adapted to the frequency domain. There are also more and other absorbing and radiation boundary conditions for the time-harmonic case.

Under certain circumstances we can assume that we have loss-less materials, i.e., $\sigma = 0$ which introduces several simplifications in the above equations, and concentrates on the wave propagation part.

If we have to model the behavior of some electromagnetic system under the influence of slowly varying fields and considerable dissipation (i.e., $\sigma > C > 0$), it is a sensible approximation to omit the wave propagation part, i.e., to omit the ∂_t^2 term. We obtain a parabolic equation.

It could also be of interest to compute stationary solutions of Maxwell's equations. Also the integration of the time-harmonic equations can be interpreted as the computation of a periodic solution that the system will tend to under certain circumstances. (See, for instance, Dautray and Lions [36, XII.§4 Remark 8].)

Sometimes it is also interesting to consider problems in two dimensions. Two ways to obtain such problems are to consider axisymmetric domains or infinite cylinders homogeneous in one direction, the later as possible models for waveguides. If we do the later, inserting the special form of the fields gives us with the two two-dimensional curl operators equations like (3.15) and (3.25), but with the operator $\mathbf{curl}\,\mathrm{curl}$ ($\mathbf{curl}$ here operating on a scalar function) instead of the operator $\mathbf{curl}\,\mathbf{curl}$.

3.2 Discretization

Now we have derived several time-dependent partial differential equations from Maxwell's equations. To solve them numerically, we have to discretize the partial differential equations both in time and space. Since we work with finite element/spectral element discretizations in space, we need a variational formulation of our equations first. We will obtain such formulations in the first subsection.

There are two conceptual approaches to integrating such time-dependent partial differential equations. The first approach consists in discretizing in space first, obtaining a large system of ordinary differential equations, which then will be integrated by a general purpose ordinary differential equation solver. (This is also called "method of lines".) If we choose an explicit method, we will have to contend with severe restrictions of the time step. If we choose an implicit method, we have to solve a large system of equations that a priori does not correspond to any discretized partial differential equation, but we can use larger timesteps. We will not use such an approach in this work.

The second approach consists in discretizing in time first. If we use an implicit scheme we obtain time-independent partial differential equations which can then be solved by any spatial approximation, in our case by spectral element methods. To derive such partial differential equations, we consider the time-dependent partial differential equation as an ordinary differential equation in a function space which we solve approximately with numerical methods for ordinary differential equations. This we will do in the second subsection.

The second approach can also be used with adaptive solvers and with adaptive time-stepping, following Bornemann who advocated this approach for parabolic equations and the wave equation (See for instance [20, 88]). We would have to solve two time-step equations for two time-discretizations (like Backward-Euler and Crank-Nicholson) and could

use error estimators to choose the appropriate time-step and accuracy in a virtual ordinary differential equation integrator for the function space ordinary differential equation.

The time-harmonic equations are already in a time-independent form. We will derive a variational formulation in the next section and identify it as an instance of (MP).

Of course, to obtain a completely discrete scheme, we will have to specify the discretization in space. Spectral element discretizations for the model problem will be presented in the next chapter.

3.2.1 Variational formulations

We have two second order time-harmonic problems (3.40),(3.41); three time-dependent second order problems (3.15), (3.25) and (3.26) and two first order systems (3.8), (3.9) and (3.38), (3.39).

We will first consider variational formulations for the first order system (3.8), (3.9). The system (3.38), (3.39) can be treated in the same way.

We multiply both equations with test functions:

$$(\epsilon\partial_t\mathbf{E} + \sigma\mathbf{E}, \psi) - (\mathbf{curl}\,\mu^{-1}\mathbf{B}, \psi) = -(\mathbf{j}_i, \psi) \tag{3.43}$$

$$(\partial_t\mathbf{B}, \phi) + (\mathbf{curl}\,\mathbf{E}, \phi) = 0 \tag{3.44}$$

and use the appropriate Green's formulae (see sections 2.3 and 2.4):

$$(\mathbf{curl}\,\mu^{-1}\mathbf{B}, \psi) = (\mu^{-1}\mathbf{B}, \mathbf{curl}\,\psi) + \langle\gamma_t(\mu^{-1}\mathbf{B}), \psi\rangle_{\frac{1}{2},\partial\Omega} \tag{3.45}$$

$$(\mathbf{curl}\,\mathbf{E}, \phi) = (\mathbf{E}, \mathbf{curl}\,\phi) + \langle\gamma_t(\mathbf{E}), \phi\rangle_{\frac{1}{2},\partial\Omega} \tag{3.46}$$

To obtain a symmetric formulation (with identical test and trial spaces) we have to choose in which of the two equations we will use Green's formula.

The first equation is chosen for instance in Monk [70] (the constrained space $H_0(\mathbf{curl})$ enforces the boundary conditions and makes the boundary terms vanish)

$$?\mathbf{E}(\mathbf{t}), \mathbf{H}(\mathbf{t}) : \forall\psi \in H_0(\mathbf{curl}), \forall\phi \in (L^2(\Omega))^3$$

$$(\epsilon\partial_t\mathbf{E} + \sigma\mathbf{E}, \psi) - (\mu^{-1}\mathbf{B}, \mathbf{curl}\,\psi) = -(\mathbf{j}_i, \psi) \tag{3.47}$$

$$(\partial_t\mathbf{B}, \phi) + (\mathbf{curl}\,\mathbf{E}, \phi) = 0 \tag{3.48}$$

The second equation is chosen for instance in Lin and Yan [66]:

$$?\mathbf{E}(\mathbf{t}), \mathbf{H}(\mathbf{t}) : \forall\psi \in (L^2(\Omega))^3, \forall\phi \in H_0(\mathbf{curl})$$

$$(\epsilon\partial_t\mathbf{E} + \sigma\mathbf{E}, \psi) - (\mathbf{curl}\,\mu^{-1}\mathbf{B}, \psi) = -(\mathbf{j}_i, \psi) \tag{3.49}$$

$$(\partial_t\mathbf{B}, \phi) + (\mathbf{E}, \mathbf{curl}\,\phi) = 0 \tag{3.50}$$

The second order equations are all of the form:

$$P(\partial_t)\mathbf{u} + \mathbf{curl}(\gamma\,\mathbf{curl}(\delta\mathbf{u})) = f \tag{3.51}$$

with the following substitutions:

Equation	$P(x)$	γ	δ	f
(3.15)	$\epsilon x^2 + \sigma x$	$\frac{1}{\mu}$	1	$\partial_t\mathbf{j}_i$
(3.25)	$\epsilon x^2 + \sigma x$	1	$\frac{1}{\mu}$	$\mathbf{curl}\,\mathbf{j}_i$
(3.26)	x^2	$\frac{1}{\epsilon}$	$\frac{1}{\mu}$	$\mathbf{curl}(\frac{1}{\epsilon}\mathbf{j}_i)$
(3.40)	$\sigma i\omega - \epsilon\omega^2$	$\frac{1}{\mu}$	1	$\hat{f}$
(3.41)	$\sigma i\omega - \epsilon\omega^2$	1	$\frac{1}{\mu}$	$\hat{f}$

Multiplying the last equation by a test function and using Green's formula we obtain:

$$(P(\partial_t)\,\mathbf{u}, \mathbf{v}) + (\gamma\,\mathbf{curl}(\delta\mathbf{u}), \mathbf{curl}\,\mathbf{v}) + \langle\gamma_t(\gamma\,\mathbf{curl}(\delta\mathbf{u})), \mathbf{v}\rangle_{\frac{1}{2},\partial\Omega} = (\mathbf{f}, \mathbf{v}) \tag{3.52}$$

We change variables $\delta\mathbf{u} \mapsto \mathbf{u}$ to match (MP) (and for the equations with $\delta \neq 1$, $\delta\mathbf{u}$ will have continuous tangential components across interfaces, so that it makes sense to discretize $\delta\mathbf{u}$):

$$\left(\frac{1}{\delta}P(\partial_t)\,\mathbf{u}, \mathbf{v}\right) + (\gamma\,\mathbf{curl}(\mathbf{u}), \mathbf{curl}\,\mathbf{v}) + \langle\gamma_t(\gamma\,\mathbf{curl}\,\mathbf{u}), \mathbf{v}\rangle_{\frac{1}{2},\partial\Omega} = (\mathbf{f}, \mathbf{v}) \tag{3.53}$$

For the time-harmonic examples this is already of the form (MP) with: $\alpha = \frac{\sigma i\omega - \epsilon\omega^2}{\delta}$ and $\beta = \gamma$, i.e., $\alpha = \sigma i\omega - \epsilon\omega^2$ and $\beta = \frac{1}{\mu}$ for (3.40), and $\alpha = \mu\omega(\sigma i - \epsilon\omega)$ and $\beta = 1$ for (3.41).

For the time-dependent problems we put the time-derivative parts on the left side and the rest on the right hand side:

$$\left(\frac{1}{\delta}P(\partial_t)\,\mathbf{u}, \mathbf{v}\right) = (\mathbf{f}, \mathbf{v}) - (\gamma\,\mathbf{curl}(\mathbf{u}), \mathbf{curl}\,\mathbf{v}) - b.t. \tag{3.54}$$

This will be the form to which we will apply time-stepping.

Some theory for such variational second order time-dependent problems is developed in Dautray and Lions [36, Chapter XVIII, pg. 467-679] and Showalter [90, Chapter 5].

3.2.2 Time-stepping schemes

We will consider the functions $\mathbf{u}$ and $\mathbf{f}$ at discrete times $t_n = n\Delta t$ and denote the functions so obtained $\mathbf{u}^n$ and $\mathbf{f}^n$.

The standard way (at least the textbook way) is to transform the second order equation into a first-order system, and then to use approaches for first-order equations. One advantage of that point of view is that one can see one step of the evolution as the approximation of an exponential function, as in semi-group theory, and that there are many different approaches to solve such partial differential equations. In addition, if one is willing to store the function and its derivatives, there are only few second-order methods for specific $P(\partial t)$ that seem to be advantageous (see Hairer, Nørsett and Wanner [55]). In our case we do not want to solve a first-order system, and we are only interested in either the electric or magnetic field at each timestep, so that we will use methods for second-order equations. (For such a method, but in a fully discrete approach, see Ciarlet Jr. and Zhou [26].)

We will only demonstrate the time discretization with linear multistep methods. Similarly, we could obtain problems of type (MP) for other implicit methods like implicit Runge-Kutta methods. (There are special methods for $P(x) = cx^2$, so-called Nystrom and Störmer methods, see Hairer, Nørsett and Wanner [55], the implicit version of which will lead to similar problems, but we will not treat them here. For such methods it would also be necessary to either have $\sigma = 0$ or transform the equation so that the first order time-derivatives disappear.)

We will approximate $\partial_t^2 \mathbf{u}$ and $\partial_t \mathbf{u}$ by linear combinations of $\mathbf{u}^{n-i}$:

$$(\partial_t^2 \mathbf{u})^n \approx \sum_{i=0}^{k} w_i^{(2)} \mathbf{u}^{n-i} \qquad (\partial_t \mathbf{u})^n \approx \sum_{i=0}^{l} w_i^{(1)} \mathbf{u}^{n-i}$$

There are many approximations for second and first derivatives, we will list some possibilities for $(\partial_t^2 \mathbf{u})^n$:

$$\begin{aligned}
\text{Central} \quad & k = 2 \quad w_0^{(2)} = \tfrac{1}{\Delta t^2},\ w_1^{(2)} = -\tfrac{2}{\Delta t^2},\ w_2^{(2)} = \tfrac{1}{\Delta t^2} \\
\text{Second order} \quad & k = 3 \quad w_0^{(2)} = \tfrac{2}{\Delta t^2},\ w_1^{(2)} = -\tfrac{5}{\Delta t^2},\ w_2^{(2)} = \tfrac{4}{\Delta t^2},\ w_3^{(2)} = -\tfrac{1}{\Delta t^2}
\end{aligned}$$

and for $(\partial_t u)^n$:

$$\begin{aligned}
\text{First order, backward} \quad & l = 1 \quad w_0^{(1)} = \tfrac{1}{\Delta t},\ w_1^{(1)} = -\tfrac{1}{\Delta t} \\
\text{First order, leap-frog} \quad & l = 2 \quad w_0^{(1)} = \tfrac{1}{2\Delta t},\ w_1^{(1)} = 0,\ w_2^{(1)} = -\tfrac{1}{2\Delta t} \\
\text{Second order} \quad & l = 2 \quad w_0^{(1)} = \tfrac{3}{2\Delta t},\ w_1^{(1)} = -\tfrac{2}{\Delta t},\ w_2^{(1)} = \tfrac{1}{2\Delta t}
\end{aligned}$$

The right hand side $r(\mathbf{u},\mathbf{f}) = (\mathbf{f},\mathbf{v}) - (\gamma\,\mathbf{curl}\,\mathbf{u},\mathbf{curl}\,\mathbf{v})$[3] has also to be evaluated and taken into account. We use a linear scheme here as well:

$$(r(\mathbf{u},\mathbf{f}))^n \approx \sum_{i=0}^{m} w_i^r r(\mathbf{u}^{n-i},\mathbf{f}^{n-i})$$

Some examples of such schemes are:[4]

Backward Euler	$m=0$	$w_0^r = 1$
Crank-Nicholson	$m=1$	$w_0^r = \frac{1}{2}$, $w_1^r = \frac{1}{2}$
Damped Crank-Nicholson	$m=1$	$w_0^r = \frac{1}{2} + \eta\Delta t$, $w_1^r = \frac{1}{2} - \eta\Delta t$
θ-method	$m=1$	$w_0^r = \theta$, $w_1^r = 1-\theta$
Centered Scheme	$m=2$	$w_0^r = \theta$, $w_1^r = 1-2\theta$, $w_2^r = \theta$
Third-order Adams-Moulton	$m=2$	$w_0^r = \frac{5}{12}$, $w_1^r = \frac{2}{3}$, $w_2^r = -\frac{1}{12}$

Substituting the form of $r(\mathbf{u},\mathbf{f})$, and splitting the expression into a part containing $\mathbf{u}^n$ and the rest (known from the data or from previous time-steps), we obtain:

$$\begin{aligned}\sum_{i=0}^{m} w_i^r r(\mathbf{u}^{n-i},\mathbf{f}^{n-i}) &= -w_0^r(\mathbf{curl}\,\mathbf{u}^n,\mathbf{curl}\,\mathbf{v}) + \sum_{i=0}^{m} w_i^r(\mathbf{f}^{n-i},\mathbf{v}) \\ &\quad - \sum_{i=1}^{m} w_i^r(\mathbf{curl}\,\mathbf{u}^{n-i},\mathbf{curl}\,\mathbf{v}) \end{aligned} \tag{3.55}$$

With $\frac{1}{\delta}P(\partial_t) = a\partial_t^2 + b\partial_t$ and substituting all the discretizations into the equation, we obtain as time-discretized system:

$$\begin{aligned}&\left(a\sum_{i=0}^{k} w_i^{(2)}\mathbf{u}^{n-i} + b\sum_{i=0}^{l} w_i^{(1)}\mathbf{u}^{n-i},\mathbf{v}\right) \\ &= w_0^r(\mathbf{curl}\,\mathbf{u}^n,\mathbf{curl}\,\mathbf{v}) + \sum_{i=0}^{m} w_i^r(\mathbf{f}^{n-i},\mathbf{v}) - \sum_{i=1}^{m} w_i^r(\mathbf{curl}\,\mathbf{u}^{n-i},\mathbf{curl}\,\mathbf{v})\end{aligned} \tag{3.56}$$

Collecting terms with $\mathbf{u}^n$ on the left hand side gives

$$\left(aw_0^{(2)}\mathbf{u}^n + bw_0^{(1)}\mathbf{u}^n,\mathbf{v}\right) + (w_0^r\gamma\,\mathbf{curl}\,\mathbf{u}^n,\mathbf{curl}\,\mathbf{v}) = (\mathbf{f}_{(1)},\mathbf{v}) + (\mathbf{f}_{(2)},\mathbf{curl}\,\mathbf{v}) \tag{3.57}$$

[3] We will ignore the boundary terms here, i.e., treat the problem with natural boundary conditions, or in case of $H_0(\mathbf{curl})$ with homogeneous essential boundary conditions.

[4] η is a small positive constant

using the following expressions on the right hand side

$$\mathbf{f}_{(1)} = \sum_{i=0}^{m} w_i^r \mathbf{f}^{n-i} - a \sum_{i=1}^{k} w_i^{(2)} \mathbf{u}^{n-i} - b \sum_{i=1}^{l} w_i^{(1)} \mathbf{u}^{n-i} \tag{3.58}$$

$$\mathbf{f}_{(2)} = \sum_{i=1}^{m} w_i^r \, \mathbf{curl}\, \mathbf{u}^{n-i} \tag{3.59}$$

With

$$\alpha = a w_0^{(2)} + b w_0^{(1)} \qquad \beta = w_0^r \gamma$$

this is of the form (MP) with $\mathbf{f}(\mathbf{v}) = (\mathbf{f}_{(1)}, \mathbf{v}) + (\mathbf{f}_{(2)}, \mathbf{curl}\, \mathbf{v})$ being a linear form on $H(\mathbf{curl})$.

As an example we list the discretization that we obtain for the second order equation for the electric field if we choose to discretize the first derivative with the leap-frog scheme, the second derivative with the first-order central difference, and the right hand side with Backward Euler ($\mathbf{f}_{(2)} = 0$)

$$\alpha = \frac{\epsilon}{\Delta t^2} + \frac{\sigma}{2\Delta t} \qquad \beta = \frac{1}{\mu} \qquad \mathbf{f}_{(1)} = \mathbf{f}^n + \frac{2\epsilon}{\Delta t^2}\mathbf{u}^{n-1} - \frac{\epsilon}{\Delta t^2}\mathbf{u}^{n-2} + \frac{\sigma}{2\Delta t}\mathbf{u}^{n-2}$$

Since we will work with small time-steps, we should rescale (by multiplying with Δt^2) the system so that we do not divide by Δt, that is, we will work with the following coefficients:

$$\alpha = \epsilon + \frac{\sigma}{2}\Delta t \qquad \beta = \frac{1}{\mu}\Delta t^2 \qquad \mathbf{f}_{(1)} = \Delta t^2 \mathbf{f}^n + 2\epsilon \mathbf{u}^{n-1} + \left(\frac{\sigma}{2}\Delta t - \epsilon\right)\mathbf{u}^{n-2} \tag{3.60}$$

3.2.3 Boundary conditions

If we pose the model problem without boundary terms, in $H_0(\mathbf{curl})$, we enforce zero tangential components of the solution.

If we pose the model problem without boundary terms, in $H(\mathbf{curl})$, we obtain the natural boundary conditions, which are that the tangential components of the $\gamma\, \mathbf{curl}$ of the solution are zero on the boundary. (See the Green's formula for the second-order equation.)

If we pose Silver-Müller boundary conditions on Γ_A in the model problem, we have to add a boundary term of the form

$$b.t. = (\rho \mathbf{u} \times \mathbf{n}, \mathbf{v} \times \mathbf{n})_{(L^2(\Gamma_A))^3}$$

to the time-step problem. (This term comes from the boundary term in Green's formula, which gives us a $\gamma\,\mathbf{curl}$ term on Γ_A, which we can transform into a boundary term of the stated form by using the Silver-Müller boundary conditions. The time-derivative occurring in the continuous version is discretized as above.)

For nonhomogeneous versions of the essential problem, we can find a lifting of the boundary values and subtract it off, to obtain a homogeneous problem. For the natural boundary condition, we have to add a boundary integral. Inhomogeneous Silver-Müller conditions also give an additional boundary integral on Γ_A.

Chapter 4

Polynomial approximation, quadrature and differentation

In this chapter, we will present the approximation of functions and of operations on them in spaces of polynomials. We will use polynomial spaces associated with tensor product meshes made out of one-dimensional Gauss-Legendre or Gauss-Lobatto-Legendre meshes. First we define the polynomial spaces, and, in the first section, we discuss some of the properties of the Legendre polynomials and their derivatives.

To discretize partial differential equations, we need to be able to interpolate, differentiate and integrate functions. We present algorithms and estimates for these operations on polynomials in the second and third section. For the theoretical analysis of the discretizations, we need approximation results and inverse inequalities, which we treat in the next two sections. In the last section we indicate how to extend the methods and results from the one-dimensional case to tensorized domains in an arbitrary number of dimensions.

As general references for this chapter we refer to Bernardi and Maday [17] and Canuto, Hussaini, Quarteroni, and Zang [24].

We work with $\mathbb{Q}$-type polynomial spaces. We denote by $\mathbb{P}_N(S) = \mathbb{Q}_N(S)$ the polynomials of degree N of one variable defined on the one-dimensional set $S \subset \mathbb{R}$. In our applications, S is an interval, most often the reference interval, $\Lambda =]-1, 1[$. We define $\mathbb{Q}_{M,N}(S)$ as the space of polynomials on the two-dimensional set S that have maximal degree M in x_1 and N in x_2. For $M = N$, we also write $\mathbb{Q}_M(S)$. $\mathbb{Q}_{L,M,N}(S)$ is the analogue for three dimensions, having maximal degree L in x_1, M in x_2 and N in x_3. For an arbitrary number of dimensions we denote by $\mathbb{Q}_{\{m_i\}}(\times_i [a_i, b_i])$ the space of polynomials on the Cartesian product of intervals $[a_i, b_i]$ that have maximal degree m_i in x_i.

4.1 Legendre polynomials

The sequence of Legendre polynomials $\{L_n\}$ is the family of orthogonal polynomials on Λ in $L^2(\Lambda, dx)$ chosen so that $L_n(1) = 1$ and that L_n is a polynomial of degree n.

L_n satifies the differential equation

$$\partial_x((1-x^2)L_n'(x)) + n(n+1)L_n = 0.$$

It follows immediately that

$$\int_{-1}^{1} L_n'(x)L_k'(x)(1-x^2)dx = n(n+1)\int_{-1}^{1} L_n(x)L_k(x)dx,$$

implying that $\{L_n'\}$ is a family of orthogonal polynomials in $L^2(\Lambda, (1-x^2)dx)$.

The L_n can be computed by the induction formula

$$L_0(x) = 1 \qquad L_1(x) = x$$
$$L_{n+1}(x) = \frac{2n+1}{n+1}xL_n(x) - \frac{n}{n+1}L_{n-1}(x) \quad \text{for} \quad n \geq 1$$

A similar induction formula holds for L_n':

$$L_0'(x) = 0 \qquad L_1'(x) = 1$$
$$L_{n+1}'(x) = \frac{2n+1}{n}xL_n'(x) - \frac{n+1}{n}L_{n-1}'(x) \quad \text{for} \quad n \geq 1$$

We denote the zeros of L_N by ζ_i^N. The set of all zeros of L_N will be denoted GL_N. They can be found by either an appropriately tuned root finding algorithm, or as eigenvalues of a special tridiagonal matrix [1]

$$Z_G = \mathsf{diag}((\beta_i)_{i=1}^{N-1}, -1) + \mathsf{diag}((\beta_i)_{i=1}^{N-1}, 1)$$

with

$$\beta_i = \frac{i}{\sqrt{4i^2-1}}$$

ξ_i^N are the zeros of $(1-x^2)L_N'$. The set of all these zeros is denoted GLL_N. They can be computed also either by a tuned root finding algorithm, or as the eigenvalues of a simple symmetric tridiagonal matrix Z_{GLL} together with $\xi_0^N = -1$ and $\xi_N^N = 1$

$$Z_{GLL} = \mathsf{diag}((\gamma_i)_{i=1}^{N-2}, -1) + \mathsf{diag}((\gamma_i)_{i=1}^{N-2}, 1)$$

[1] We use MATLAB notation for the matrices, $\mathsf{diag}(v, \pm 1)$ is the matrix that has the vector v above respective below the diagonal, and $\mathsf{diag}(v)$ is the diagonal matrix with diagonal v.

with

$$\gamma_i = \sqrt{\frac{i(i+2)}{\left(i+\frac{1}{2}\right)\left(i+\frac{3}{2}\right)}}$$

Legendre polynomial have many useful properties, see, for instance, Szegö [94], or for a short introduction in the context of spectral methods, Bernardi and Maday [17, section 3]. They are special cases of Jacobi polynomials [17, section 19] which are orthogonal polynomials for more general weights. These are used, e.g., in some discretizations for axisymmetric domains.

4.2 Gauss-Lobatto-Legendre interpolation and differentiation

We can associate to the Gauss-Legendre or Gauss-Lobatto-Legendre points Legendre nodal basis functions, i.e., functions that are one at one point and zero at all others. The basis functions for the Gauss-Legendre case are

$$\psi_i^N(x) = \frac{1}{L_N'(\zeta_i)} \frac{L_N(x)}{x-\zeta_i} \text{ for } i = 1, \cdots, N$$

In the Gauss-Lobatto basis the basis functions associated to the endpoints have a slightly simpler form

$$\phi_0^N = (-1)^{N-1} \frac{(1-x)L_N'(x)}{N(N+1)} \qquad \phi_N^N = -\frac{(x+1)L_N'(x)}{N(N+1)}$$

than the basis functions for the interior of the interval

$$\phi_i^N = -\frac{1}{N(N+1)L_N(\xi_i)} \frac{(1-x^2)L_N'(x)}{x-\xi_i}$$

but the last formula also holds for $i = 0$ and $i = N$.

We use a fast implementation of the evaluation of these basis functions with matrix operations to compute the interpolation matrices I_N^M, which take the values of a function on GLL_N and give as the result of the matrix-vector multiplication the values of the interpolant on GLL_M. The entries of the matrix are

$$I_N^M(i,j) = \phi_i^N(\xi_j^M)$$

It is important in the analysis of spectral methods to estimate the interpolation error. Here, we only present the results for the interpolation on the Gauss-Lobatto-Legendre points.

Similar, but worse and not optimal, results hold for the Gauss-Legendre nodes; see Bernardi and Maday [17, section 13].

Define $i_N u$ for any function $u \in C(\overline{\Lambda})$ as the only function in $\mathbb{P}_N(\Lambda)$ that interpolates u on GLL_N.

We have the following three theorems (for proofs and discussion, see [17, section 13]):

Theorem 4.1 (Interpolation in H^s) *For any real r and s with $s > \frac{1+r}{2}$ and $r \in [0,1]$, there exists a C only depending on s such that*

$$\forall u \in H^S(\Lambda) : ||u - i_N u||_{r,\Lambda} \leq C N^{r-s} ||u||_{s,\Lambda}$$

Theorem 4.2 (Stability in H^1) *For all $u \in H^1(\Lambda)$ we have the stability estimate*

$$||i_N u||_{1,\Lambda} \leq C ||u||_{1,\Lambda}$$

Theorem 4.3 (Interpolation between polynomial spaces) *The L^2-norm of i_N as a mapping from $\mathbb{P}_M$ to $\mathbb{P}_N$ is bounded linearly in $\frac{M}{N}$, i.e.,*

$$\forall u_M \in \mathbb{P}_M : ||i_N u_M||_{0,\Lambda} \leq C \left(1 + \frac{M}{N}\right) ||u_M||_{0,\Lambda}$$

On $\mathbb{P}_N$ differentiation can be computed exactly; the following theorem gives the differentiation matrix in the GLL nodal basis.

Theorem 4.4 (Spectral differentiation matrix) *The differentiation matrix D_N on $\mathbb{P}_N$ has the following entries*

$$D_N(k,j) = \begin{cases} \frac{L_N(\xi_k)}{(\xi_k - \xi_j) L_N(\xi_j)} & k \neq j \\ \frac{N(N+1)}{4} & k = j = 0 \\ \frac{-N(N+1)}{4} & k = j = N \\ 0 & \textit{else} \end{cases}$$

and satisfies

$$\forall u_N \in \mathbb{P}_N : D_N u_N = \partial_x u_N$$

The explicit form of D_N on the Gauss-Legendre basis is known. Estimates for $|\partial_x u - D_N u|$ for $u \in H^s$ or $u \in L^p$ are also known, we refer for all that to the literature, see, for instance, Canuto, Hussaini, Quarteroni, and Zang [24].

4.3 Gauss- and Gauss-Lobatto quadrature

Zeros and extrema of orthogonal polynomials can be used to define very accurate quadrature formulae, see, e.g., Davis and Rabinowitz [38] or Szegö [94].

The standard Gauss-Legendre formula is exact for $\mathbb{P}_{2N-1}(\Lambda)$ and reads

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{N} f(\zeta_i)\omega_i$$

The ω_i can be computed as soon as the ζ_i are found, by the formula:

$$\omega_i = \frac{2}{(1-\zeta_i^2)L_N'^2(\zeta)}$$

The Gauss-Lobatto-Legendre formula is also exact for $\mathbb{P}_{2N-1}(\Lambda)$ (but note that it uses one more quadrature point) and reads

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=0}^{N} f(\xi_i)\rho_i$$

The integration weights ρ_i are

$$\rho_i = \frac{2}{N(N+1)L_N^2(\xi_i)}$$

We can discretize the L^2 inner product on $L^2(\Lambda)$ using Gauss-Lobatto-Legendre quadrature:

$$(u,v)_{0,\Lambda} \approx (u,v)_N := \sum_{i=0}^{N} u(\xi_i)v(\xi_i)\rho_i = \underline{v}^T M_N \underline{u}$$

with $M_N = \mathsf{diag}(\rho_i)$ and $\underline{v}$ and $\underline{u}$ being the vectors of the values of v and u on GLL_N.

If an integration of a different order is needed, we interpolate $\underline{u}$ and $\underline{v}$ to a different GLL_M and use Gauss-Lobatto-Legendre quadrature there:

$$(u,v)_{0,\Lambda} \approx (u,v)_{N;M} := \underline{v}^T I_N^{M,T} M_M I_N^M \underline{u} = \underline{v}^T M_N^M \underline{u}$$

with $M_N^M := I_N^{M,T} M_M I_N^M$. ($I_N^M$ has been defined in the previous section.)

The error in the integration can be estimated, see, e.g., Canuto, Hussaini, Quarteroni, and Zang [24] or Bernardi and Maday [17].

4.4 Approximation results

We will list some approximation results for L^2, H^s, H_0^s, and $H(\mathbf{curl})$. We refer to chapter 7 for the definition of the polynomial spaces ND_N^I and ND_N^{II}.

To prove the approximation results, we exhibit one element in the polynomial space that satisfies the estimates. That element is usually defined as an orthogonal projection of the function that is to be approximated. Therefore we will start with the definition of several projections.

Let π_N be the orthogonal projection from $L^2(\Omega)$ onto $\mathbb{P}_N$, i.e.,

$$\forall v_N \in \mathbb{P}_N : (u - \pi_N u, v_N)_{0,\Lambda} = 0$$

For positive k and $N \leq 2k-1$ we define $P_N^{k,0}(\Omega) : \mathbb{P}_N(\Lambda) \cap H_0^k(\Lambda)$ and define the projection $\pi_N^{k,0} u$ by

$$\forall v_N \in \mathbb{P}_N^{k,0} : (\partial_x^k u - \partial_x^k \, \pi_N^{k,0} u, \partial_x^k v_N)_{0,\Lambda} = 0$$

Define π_N^k as the orthogonal projection from $H^k(\Lambda)$ onto $\mathbb{P}_N$. Let $\pi_N^{c,II}$ be the orthogonal projection from $H(\mathbf{curl}, \Omega)$ onto ND_N^{II}.

Then the following estimates hold (see Bernardi and Maday [17, section 6]) :

$$\text{If } s \geq 0\text{: } \forall u \in H^s(\Lambda) : ||u - \pi_N u||_{0,\Lambda} \leq C N^{-s} ||u||_{s,\Lambda}$$

$$\text{If } 0 \leq r \leq k \leq s\text{: } \forall u \in H^s(\Lambda) \cap H_0^k(\Lambda) : ||u - \pi_N^{k,0} u||_{0,\Lambda} \leq C N^{r-s} ||u||_{s,\Lambda}$$

$$\text{If } 0 \leq r \leq k \leq s\text{: } \forall u \in H^s(\Lambda) : ||u - \pi_N^{k} u||_{0,\Lambda} \leq C N^{r-s} ||u||_{s,\Lambda}$$

These are the best possible approximation results with respect to their exponents. There are versions of $\pi_N^{k,0}$ that preserve some or all the values of the function and its derivatives at the end points, see [17, section 6].

For the approximation in $H(\mathbf{curl}, \Lambda^3)$, the following estimates can be proven (see Ben Belgacem and Bernardi [15]):

$$\text{If } s > 0\text{: } \forall u \in H^s(\mathbf{curl}, \Lambda^3) : \inf_{u_N \in ND_N^I} ||\,\mathbf{curl}\, u - \mathbf{curl}\, u_N||_0 \leq C N^{-s} ||\,\mathbf{curl}\, u||_s$$

$$\text{If } s \geq 0\text{: } \forall u \in H^s(\mathbf{curl}, \Lambda^3) : ||u - \pi_N^{c,II} u||_{\mathbf{curl}} \leq C N^{-s} \sqrt{||u||_{s+1}^2 + ||\,\mathbf{curl}\, u||_s^2}$$

4.5 Inverse inequalities

We will give three inverse inequalities, one for H^s, one for L^p, and one that allows us to estimate the maximum norm over the entire interval given only the maximum over GLL_N.

Theorem 4.5 (Inverse inequality in H^s) *Let m be integer and r be real with $0 \leq m \leq r$. Then, for any polynomial $u_N \in \mathbb{P}_N(\Lambda)$ we have (with optimal exponent)*

$$||u_N||_{r,\Lambda} \leq CN^{2(r-m)}||u_N||_{m,\Lambda}$$

Proof: See Bernardi and Maday [17, Theorem 5.2]. The extension to real m is also discussed there. ∎

Theorem 4.6 (Inverse inequality in L^p) *For any real p and q with $1 \leq p \leq q \leq \infty$ there exists a positive constant C such that for any polynomial $u_N \in \mathbb{P}_N(\Lambda)$*

$$||u_N||_{0,q,\Lambda} \leq CN^{\frac{2}{p}-\frac{2}{q}}||u_N||_{0,p,\Lambda}$$

Proof: See Timan [95, page 236]. For a discussion of such inequalities, see also Canuto, Hussaini, Quarteroni, and Zang [24, chapter 9]. ∎

Theorem 4.7 (Inverse inequality for L^∞) *The following norm equivalence holds with $\delta_N \sim \log N$ for all $u_N \in \mathbb{P}_N(\Lambda)$*

$$\frac{1}{\delta_N}||u_N||_\infty \leq \max_{i=0,\cdots,N}|u_N(\xi_i)| \leq ||u_N||_\infty$$

Proof: See Quateroni and Valli [85, Remark 4.4.1 on page 119]. ∎

4.6 Extension to tensorized domains

We use rectangular elements with tensor basis functions, i.e., $\phi_{j_1,j_2,\cdots,j_n}(x_1,x_2,\cdots,x_n) = \prod_{i=1}^n \phi_{j_i}(x_i)$. The interpolation on such a tensor basis can be built from the one-dimensional interpolations on the one-dimensional meshes that span the tensor product mesh.

The elements of the mass matrices are integrals of products of basis functions over the domain. Since the basis functions have tensor form and we work on rectangular elements, the integral can be factored into n one-dimensional integrals, and the mass matrix for the

domain is therefore the tensor product of the one-dimensional mass matrices on the one-dimensional meshes. (See chapter 9 for an introduction to tensor product matrices.)

Partial differentiation only acts along one direction. Therefore it can be written as the tensor product of the differentiation matrix in the differentiated direction and identity matrices in the other directions.

The projection operators that we discussed in the fourth section can also be defined for the multidimensional case, and it turns out that both the L^2- and H^k-projections are constructed as a tensor product of one-dimensional L^2- and H^k-projections, respectively.

Finally, the estimates for one-dimensional projection and interpolation operators generalize to the case of arbitrary many dimensions, see Bernardi and Maday [17, sections 7 and 14].

Chapter 5

Domain decomposition and iterative methods

5.1 Domain decomposition methods

The fundamental idea of domain decomposition methods is to reduce the solution of a problem

$$\begin{cases} Lu &= f & \text{in } \Omega \\ u &= g & \text{on } \partial\Omega \end{cases}$$

to the solution of problems on parts of the domain (or easier problems on the entire domain) of a similar form:

$$\begin{cases} L_i u_i &= f_i & \text{in } \Omega_i' \\ u_i &= g_i & \text{on } \partial\Omega_i' \end{cases}$$

The first such method was proposed by Hermann Amandus Schwarz in 1869 as a theoretical device to deduce the existence and uniqueness of the boundary value problem for Poisson's equation for domains with a general boundary from the same result on simple domains. (For a presentation of the method in this context, see, e.g., Courant and Hilbert [30, Kapitel 4,§4,2] or Dautray and Lions [33, chapter II,§7,2].) This method is known as the *alternating Schwarz method*. For two subregions it can be described as follows: given two overlapping subregions Ω_1' and Ω_2' ($\Omega = \Omega_1' \cup \Omega_2'$), and an initial guess u^0 that assumes the correct boundary values on $\partial\Omega$, approximations u^{n+1} are constructed from u^n in two sequential steps:

$$\begin{cases} Lu^{n+\frac{1}{2}} &= f & \text{in } \Omega_1' \\ u^{n+\frac{1}{2}} &= u^n & \text{on } \partial\Omega_1' \end{cases}$$

$$\left\{ \begin{array}{rcll} Lu^{n+1} & = & f & \text{in } \Omega_2' \\ u^{n+1} & = & u^{n+\frac{1}{2}} & \text{on } \partial\Omega_2' \end{array} \right.$$

The convergence of the method has first been proven by Schwarz using a maximum principle, but it can also be proven by Hilbert space methods. The latter way is the most successful since much of the work relies on the classical calculus of variations and finite or spectral elements.

We can write the method in a variational form, using the bilinear form $a(\cdot,\cdot)$ associated with the operator L, i.e., in the case of Poisson's equation $L = -\Delta$, $a(u,v) = \int_\Omega \nabla u \cdot \nabla v$, as follows:

Solve

$$?\delta u^{n+\frac{1}{2}} \in V_1 := H_0^1(\Omega_1') : \forall v \in V_1 : a(\delta u^{n+\frac{1}{2}}, v) = a(u - u^n, v)$$

Set $u^{n+\frac{1}{2}} = u^n + \delta u^{n+\frac{1}{2}}$.

Solve

$$?\delta u^{n+1} \in V_2 := H_0^1(\Omega_2') : \forall v \in V_1 : a(\delta u^{n+1}, v) = a(u - u^{n+\frac{1}{2}}, v)$$

Set $u^{n+1} = u^{n+\frac{1}{2}} + \delta u^{n+1}$.

If we define the projections $P_i : V \to V_i$ by

$$?P_i u \in V_i : \forall v \in V_i : a(P_i u, v) = a(u, v)$$

and denote by e^n the error in step n, i.e., $e^n = u - u^n$, then we obtain

$$e^{n+1} = (I - P_2)(I - P_1)e^n$$

Domain decomposition methods are extensions of this algorithm in several ways. First one can solve the subproblems with different types of boundary conditions, yielding Neumann-Neumann methods or Robin-Robin methods, among others. Second, instead of sequential updates, one could solve several problems in parallel and update the solution in parallel, leading to additive methods. Very often, domain decomposition methods are used as preconditioners for the original system. In this way, it is no longer important to construct a domain decomposition method that converges when used on its own. Instead one looks for a good spectral approximation of the problem, and the convergence and robustness of the method is improved by accelerators such as Krylov subspace methods.

The fundamental idea of multigrid methods has been discovered several times, and it first found wide acceptance in the early 1980s. Later it was noted that one could unify the

theory of both ”standard” domain decomposition methods and multigrid methods in one framework, if the different spaces V_i in the variational formulation do not only correspond to different subdomains, but also to discretizations on the same domain, but at different resolutions. For a development of this framework, but for the analysis of multigrid methods, see Bramble [21].

In the next section we will describe a general framework for Schwarz methods and their analysis, and some results that we will use in the last chapter. For an introduction to Schwarz methods we refer to Smith, Bjørstad, and Gropp [91]; Widlund [102]; Dryja and Widlund [44]; Dryja, Smith, and Widlund [43]; and references therein.

5.2 The Schwarz framework

We will restrict ourselves to the symmetric coercive case. There are more general settings for Schwarz methods, such as for nonsymmetric and indefinite problems, and mixed problems, for which we refer to the literature.

Let V be a finite dimensional space and let $a(\cdot,\cdot)$ be a symmetric coercive bilinear form on V. The following problem is considered:

$$?u \in V : \forall v \in V : a(u,v) = f(v)$$

with $f \in V'$, V' denoting the dual of V.

Assume that a decomposition $V = \sum_{i=0}^{J} V_i$ and $J+1$ symmetric positive-definite bilinear forms $a_i(u,v)$ (for $u,v \in V_i$) are given. Then we can define approximate projections $T_i : V \rightarrow V_i \subset V$ by

$$?T_i u : \forall v \in V_i : a_i(T_i u, v) = a(u,v)$$

Using these approximate projections, we can define many domain decomposition operators, among them the *additive Schwarz method*

$$T_{ASM} = \sum_{i=0}^{J} T_i,$$

the *multiplicative Schwarz method*

$$T_{MSM} = I - \prod_{i=0}^{J} (I - T_i)$$

with its error propagation operator

$$E_{MSM} = I - T_{MSM} = \prod_{i=0}^{J}(I - T_i),$$

its symmetrized version

$$T_{SMSM} = I - \prod_{i=0}^{J}(I - T_i)\prod_{i=0}^{J}(I - T_{J-i}),$$

or hybrid methods such as

$$T_{HY1} = T_0 + I - \prod_{i=1}^{J}(I - T_i).$$

These methods can be used as preconditioners B within the preconditioned conjugate gradient method or GMRES with the building blocks $B_i = R_i^T(R_iAR_i^T)^{-1}R_i$ with A being the stiffness matrix corresponding to $a(\cdot,\cdot)$ and R_i being the restriction from Ω to Ω_i'. (See, e.g., Smith, Bjørstad, and Gropp [91, pages 151–152].)

One can also use these methods to write the original problem in the form $Tu = g$, where T is a polynomial $P(T_0, T_1, \ldots, T_J)$ in the operators T_i satisfying $P(0, 0, \ldots, 0) = 0$. (All methods listed above have this property.) Then g can be computed without knowing the exact solution by solving problems on the subspaces, see [91, page 150]. The operator equation $Tu = g$ can then be solved without further preconditioning by the conjugate gradient method with inner product $a(\cdot,\cdot)$ (for symmetric positive definite operators T) or by GMRES.

In the analysis of these methods, the following assumptions are common ($||u||_a^2 := a(u,u)$):

Assumption 1 (Stable decomposition): There is a minimal constant C_0 such that for all $u \in V$ there exists a representation $u = \sum_i u_i$ with $u_i \in V_i$ satisfying

$$\sum_i a_i(u_i, u_i) \leq C_0^2 a(u,u)$$

Assumption 2 (Strengthened Cauchy-Schwarz inequalities): Define $\epsilon_{ij} \in [0,1]$ as the smallest constant such that (for $i, j \geq 1$)

$$\forall u_i \in V_i : \forall u_j \in V_j : a(u_i, u_j) \leq \epsilon_{ij}||u_i||_a||u_j||_a$$

Denote the spectral radius of the matrix $\epsilon = (\epsilon_{ij})_{i,j=1}^J$ by $\rho(\epsilon)$.

Note that the coarse space is excluded in Assumption 2.

Assumption 3 (Local solvers): Assume that $\omega \in [1, 2)$ is the smallest constant such that

$$\forall u \in V_i, i = 0, \cdots, J : a(u, u) \leq \omega a_i(u, u)$$

A bound for $\rho(\epsilon)$ in assumption 2 can be obtained from a

Assumption 4 (Coloring assumption): The overlapping subregions Ω_i', $i = 1, \cdots, J$ can be colored with N_C colors so that subregions with the same color do not intersect.

The coloring assumption implies that $\rho(\epsilon) \leq N_C$ in assumption 2.

Under the above assumptions the following statements can be proven:

Lemma 5.1 (Lower bound for T_{ASM}) *C_0^{-2} with C_0 from assumption 1 is a lower bound on the spectrum of T_{ASM}.*

Proof: See Smith, Bjørstad, and Gropp [91, Lemma 1 on page 154]. ∎

Lemma 5.2 (Upper bound for T_{ASM}) *If assumptions 2 and 3 hold, $\omega(1+\rho(\epsilon))$ is an upper bound for the largest eigenvalue of T_{ASM}.*

Proof: See the proof of Lemma 3 on page 157 in [91]. ∎

Theorem 5.3 (Bound on $\kappa(T_{ASM})$) *Given assumptions 1, 2, and 3, the following bound on the condition number of the additive Schwarz method holds:*

$$\kappa(T_{ASM}) \leq \omega(1 + \rho(\epsilon))C_0^2$$

Proof: Combine lemmata 5.1 and 5.2. ∎

Theorem 5.4 (Bound on $\kappa(T_{SMSM})$) *Given assumptions 1, 2, and 3, the condition number of the symmetrized multiplicative Schwarz method allows the following bound:*

$$\kappa(T_{SMSM}) \leq \frac{(1 + 2\omega^2\rho(\epsilon)^2)C_0^2}{2 - \omega}$$

Proof: Lemma 4 on page 158 of [91]. ∎

Theorem 5.5 (Bound on $||E_{MSM}||$) *Given assumptions 1, 2, and 3, the norm of the error propagation operator for the multiplicative Schwarz method is bounded by:*

$$||E_{MSM}||_a \leq \sqrt{1 - \frac{2-\omega}{(2\omega^2\rho(\epsilon)^2 + 1)C_0^2}}$$

Proof: See Theorem 2 in Dryja and Widlund [44] and references therein. ∎

There are some results on hybrid methods, comparison theorems between methods (see, e.g., Mandel [69]), and sharper results for the multiplicative versions (see, for instance, Griebel and Oswald [51]). We refer to the literature for these results and extensions.

5.3 Iterative methods

After the discretization, the numerical solution of the (linear) partial differential equation is reduced to the solution of a large linear system of equations (A being a $M \times M$ matrix)

$$Ax = b. \tag{5.1}$$

Special structure of the partial differential equation and of the discretization usually leads to special properties of A. For instance, in the spectral element discretization of Poisson's equation, A is symmetric, positive definite and relatively (block-wise) sparse; on a rectangular domain with a rectangular mesh of elements, it is a sum of tensor products.

Equation (5.1) can be solved directly or iteratively. Direct methods have certain advantages: they deliver exact solutions (up to rounding errors in the computation) at a predictable cost and this cost only depends on the algebraic structure of the problem. In most of the algorithms most of the computations are spent on computing some kind of factorization of A which then is used to solve the problem for a given b. In this way, the computation of solutions of equation (5.1) with different b but the same A is much cheaper. On the other hand, the solution time of Gaussian elimination, for dense A without any special structure, grows like $O(M^3)$ and the storage grows like $O(M^2)$. Also, the parallelization of direct methods is not an easy undertaking and requires new algorithmic ideas. Still, in the case of special structure, like separable partial differential equations on separable domains, direct solvers are competitive even in time and storage. For some examples, see chapters 6 and 9.

Iterative methods have advantages in that they are usually much lower in storage, optimal methods might need only $O(M)$ time for a given accuracy, and that an important class of them does not need the elements of the matrix A, but only the result of Av for a given vector v. This can result in faster algorithms using less storage if the action of A is much easier to compute than its matrix representation. Some of the handicaps for an iterative method are

that the solution is only approximate, that the performance may depend on the numerical values and the geometry, and that usually the entire process has to be repeated if the right hand side has changed.

We will describe three methods out of the many that could be used. For more methods and discussions about implementation and when to choose which, see, for instance, Barrett et al [12].

The simplest iterative method is Richardson's method. If B is the matrix form of the preconditioner, the iteration is

$$u^{n+1} = u^n + \tau B(f - Au^n)$$

The optimal choice for τ is

$$\tau_{opt} = \frac{2}{\lambda_{max}(BA) + \lambda_{min}(BA)},$$

and with this choice the following error estimate holds (with $||\cdot||_2$ being the l_2-norm of a vector):

$$||e^n||_2 \leq \left(\frac{\kappa(BA) - 1}{\kappa(BA) + 1}\right)^n ||e^0||_2$$

This method is almost never used in practice since other methods usually perform better and do not need a well-chosen parameter that depends on a priori knowledge of the spectrum of BA.

The second method – the method of choice for symmetric, positive definite problems – is the (preconditioned) conjugate gradient method. It does not need a parameter and only stores five vectors of length M in the iteration. If BA has a low condition number or has clustered eigenvalues, the conjugate gradient method performs very well, the convergence often even improves with the number of iterations.

The method is given in figure 5.1, with system matrix A, preconditioner B and inner product $\langle \cdot, \cdot \rangle$.

For the preconditioned conjugate gradient method, the following error estimate can be proven (with $||v||_A^2 = v^T Av$):

$$||e^n||_A \leq 2\left(\frac{\sqrt{\kappa(BA)} - 1}{\sqrt{\kappa(BA)} + 1}\right)^n ||e^0||_A$$

The conjugate gradient method is closely related to the Lanczos process, one of the methods to compute eigenvalues. Because of this connection, the extremal eigenvalues and therefore

Given: x^0.
$r^0 := b - Ax^0$
$n = 0$
Until *stopping criterion satisfied* do
    $z^n = Br^n$
    $\rho^n = \langle r^n, z^n \rangle$
    if $n > 1$ then
        $\beta^n = \rho^n / \rho^{n-1}$
        $p^n = z^n + \beta^n p^n$
    else
        $p^n = z^n$
    $q^n = Ap^n$
    $\alpha^n = \rho^n / \langle p^n, q^n \rangle$
    $x^{n+1} = x^n + \alpha^n p^n$
    $r^{n+1} = r^n - \alpha^n q^n$
    $n = n + 1$
x^n contains the approximate solution

Figure 5.1: The preconditioned conjugate gradient method

the condition number of BA can be estimated from the values of the α^n and β^n : the extremal eigenvalues of BA are approximated by the extremal eigenvalues of the tridiagonal matrix (see, for instance, Golub and Van Loan [49, section 10.2.5 on page 528] and O'Leary and Widlund [77]):

$$\begin{pmatrix} 1/\alpha^0 & -\sqrt{\beta^1/\alpha^0} & & \\ -\sqrt{\beta^1/\alpha^0} & 1/\alpha^1 + \beta^1/\alpha^0 & -\sqrt{\beta^2/\alpha^1} & \\ & -\sqrt{\beta^2/\alpha^1} & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}$$

The third and last method is the GMRES method, which is the standard method for non-symmetric systems. We will not give its form or discuss its derivation or implementation; we refer to the literature (see, e.g., Barrett et al [12, especially section 2.3.4], and Saad and Schultz [87]). Both the preconditioned conjugate gradient method and GMRES are implemented in several packages of Krylov subspace methods, such as the KSP component of PETSc (Balay, Gropp, McInnes, and Smith [11] and Balay et al [10]). Reference implementations are also available [12].

The convergence of GMRES can be characterized by the two quantities

$$c_A = \inf_{x \neq 0} \frac{(x, Ax)}{(x, x)} \qquad\qquad C_A = \sup_{x \neq 0} \frac{||Ax||}{||x||}$$

with the estimate of the norm of the residual

$$||r_n|| \leq \left(1 - \frac{c_A^2}{C_A^2}\right)^{n/2} ||r_0||$$

One of the disadvantages of the GMRES iteration is that all the iterates have to be stored, and that in iteration step k both the time and the storage needed are of order $O(kM)$. In practice, therefore, one usually works with the restarted version GMRES(k), in which k steps of GMRES are executed and the iteration is then restarted.

There are sharper and different error bounds possible for the different iterative methods, some explicating the dependence of the convergence on the distribution of the eigenvalues or the pseudo-spectrum. We will not discuss such estimates here. We note that all the error estimates (and, in the case of Richardson's method and other iterative methods, the optimal parameters) depend on the condition number or extremal eigenvalues of the operator, or of parts of it. Therefore bounds of the extremal eigenvalues proved for domain decomposition methods imply error estimates for the various iterative methods presented.

For more information and a deeper introduction to iterative methods see the many books available, for instance, Hackbusch [53, 54], Greenbaum [50], Saad [86], or Barrett et al [12].

Chapter 6

Spectral elements for Poisson and Helmholtz equations

6.1 The discretization

We discretize the Poisson and Helmholtz equation to obtain some insight and experience towards the discetization of the model problem. Poisson and Helmholtz equations also need to be solved in solvers or preconditioners that take computational advantage of the Helmholtz decomposition, i.e., treat curl-free and divergence-free part separately. They are also interesting as model problems in their own right.

Here we discretize (with $\mathbf{n}$ denoting the outward normal)

$$-\Delta u + \alpha u = f \qquad \partial_{\mathbf{n}} u = 0$$

in d dimensions on rectangular elements (i.e., on Cartesian products of intervals). This corresponds, for $\alpha = 0$, to the Laplace equation, for $\alpha > 0$, to a positive definite problem, and for $\alpha \leq 0$, to the Helmholtz equation.

We use the variational formulation:

$$?u \in H^1(\Omega) : \forall v \in H^1(\Omega) : a(u,v) = f(v) \tag{6.1}$$

with

$$a(u,v) = (\mathbf{grad}\, u, \mathbf{grad}\, v)_0 + \alpha (u,v)_0$$
$$f(v) = (f,v)_0$$

or written out

$$a(u,v) = \sum_{i=1}^{d} (\partial_{x_i} u, \partial_{x_i} v)_0 + \alpha(u,v)_0$$

First we will discretize this formulation on one element, and afterwards, we will discuss how to subassemble the system for a rectangular arrangements of elements.

On an element $K = \times_{i=1}^{d} [a_i, b_i]$ we use polynomials with maximal degree m_i in variable x_i. Thus the local space, denoted $Q_{\{m_i\}}$, is:

$$\mathbb{Q}_{m_1,m_2,\cdots,m_d}(K) =: \mathbb{Q}_{\{m_i\}_{i=1}^{d}}(K)$$

We can choose the nodal values on a grid of size $(m_i + 1)_{i=1}^{d}$ as degrees of freedom.

The gradient of a function from the local space lies in the Cartesian product

$$\mathbf{grad}\, u \in \times_{i=1}^{d} \mathbb{Q}_{\{m_j - \delta_{ij}\}_{j=1}^{d}}$$

In the ith term of the sum constituting the bilinear form both of the factors have the same degree, so that the integrand is in $\mathbb{Q}_{\{2m_j - 2\delta_{ij}\}_{j=1}^{d}}$. To compute the integrals exactly, we will have to use Gauss-Lobatto-Legendre quadrature with degrees M_{ij} with $M_{ij} \geq m_j - \delta_{ij} + 1$. (In the case of Gauss-Legendre quadrature we obtain the same result without the $+1$.)

Therefore, in the direction of differentiation we integrate exactly on the original grid. In the directions in which we do not differentiate we need one grid point more to integrate exactly. Using the quadrature associated with the original grid we obtain diagonal matrices in the tensor product in these directions.

The integrand of the last term is in $\mathbb{Q}_{\{2m_j\}_{j=1}^{d}}$. We will use Gauss-Legendre-Lobatto quadrature of degree M_j here. We also assume that the function f on the right hand side of the partial differential equation is given or approximated on the same grid as u and v and that it is therefore also given as a function in $\mathbb{Q}_{\{m_j\}_{j=1}^{d}}$. In this case the term $(f,v)_0$ has the same form as the term $(u,v)_0$, and it will be integrated exactly in the same way.

In what follows, we use the notation $\underline{u}$ for the point values of u on the Gauss-Legendre-Lobatto mesh. Whenever we will use it, we will assume that u is regular enough so that we can define point values. We use $\overline{u}$ as a shorthand for the standard nodal interpolant of the array of point values u on the Gauss-Legendre-Lobatto mesh. We also refer to chapter 4 where the following matrices are defined: I_n^m is the matrix that interpolates from a Gauss-Legendre-Lobatto mesh of size n to one of size m, M_n^m is the mass matrix of size n that we obtain by interpolating to a grid of size m and using Gauss-Legendre-Lobatto quadrature there. D_m is the one-dimensional spectral differentiation matrix on a Gauss-Legendre-Lobatto mesh of size m. Those matrices are derived from the appropriate matrices $\hat{I}$, $\hat{M}$

and $\hat{D}$ on $[-1,1]$ by scaling, to wit, in direction x_i we obtain $I_\cdot = \hat{I}_\cdot$, $D_\cdot = \frac{2}{b_i-a_i}\hat{D}_\cdot$ and $M_\cdot = \frac{b_i-a_i}{2}\hat{M}_\cdot$.

A general term in the expression for $(\mathbf{grad}\,\cdot, \mathbf{grad}\,\cdot)$ is

$$\begin{aligned}
(\partial_{x_i}u, \partial_{x_i}v)_0 &\approx (\partial_{x_i}u, \partial_{x_i}v)_{\{M_{ij}\}_{j=1}^d} \\
&= \Big(\overline{((\otimes_{j=1}^{i-1} I_{m_j}) \otimes D_{m_i} \otimes (\otimes_{j=i+1}^{d} I_{m_j}))\underline{u}} \\
&\qquad\qquad , \overline{((\otimes_{j=1}^{i-1} I_{m_j}) \otimes D_{m_i} \otimes (\otimes_{j=i+1}^{d} I_{m_j})\underline{v}}\Big)_{\{M_{ij}\}_{j=1}^d} \\
&= \underline{v}^T((\otimes_{j=1}^{i-1} M_{m_j}^{M_{ij}}) \otimes (D_{m_i}^T M_{m_i}^{M_{ii}} D_{m_i}) \otimes (\otimes_{j=i+1}^{d} M_{m_j}^{M_{ij}}))\underline{u} \\
&= \underline{v}^T((\otimes_{j=1}^{i-1} M_{m_j}^{M_{ij}}) \otimes K_{m_i}^{M_{ii}} \otimes (\otimes_{j=i+1}^{d} M_{m_j}^{M_{ij}}))\underline{u}
\end{aligned}$$

there, $K_{m_i}^{M_{ii}}$ is the one-dimensional Laplace operator.

Combining all the terms, we obtain:

$$\underline{v}^T \left(\sum_{i=1}^{d} ((\otimes_{j=1}^{i-1} M_{m_j}^{M_{ij}}) \otimes K_{m_i}^{M_{ii}} \otimes (\otimes_{j=i+1}^{d} M_{m_j}^{M_{ij}})) + \alpha(\otimes_{j=1}^{d} M_{m_j}^{M_{ij}}) \right) \underline{u} = \underline{v}^T(\otimes_{j=1}^{d} M_{m_j}^{M_j})\underline{f}$$

Since this has to hold for all vectors $\underline{v}$, we obtain the same equation without the $\underline{v}^T$:

$$\left(\sum_{i=1}^{d} ((\otimes_{j=1}^{i-1} M_{m_j}^{M_{ij}}) \otimes K_{m_i}^{M_{ii}} \otimes (\otimes_{j=i+1}^{d} M_{m_j}^{M_{ij}})) + \alpha(\otimes_{j=1}^{d} M_{m_j}^{M_{ij}}) \right) \underline{u} = (\otimes_{j=1}^{d} M_{m_j}^{M_j})\underline{f} \tag{6.2}$$

There is a lot of freedom choosing the degrees of quadrature. We will choose degrees for groups of directions as follows: directions in which we do not differentiate are integrated with a degree of M_j (if that degree is m_j, we obtain diagonal mass matrices, with degree $m_j + 1$ we integrate exactly; these are the two main choices we will consider), directions in which we differentiate are integrated with degree S_i (which is usually chosen to be m_i corresponding to exact integration).[1] In brief

$$M_{ij} = M_j \text{ for } i \neq j \qquad M_{ii} = S_i \text{ (stiffness matrix)} \qquad M_{ii} = M_i \text{ (mass matrix)} \tag{6.3}$$

[1] Over-integration of the stiffness matrix may make sense for $(\alpha u, v) + (\beta\,\mathbf{grad}\,u, \mathbf{grad}\,v)$ with variable α and β, in our case it will not give any different result than exact integration. Under-integration is not advised, since we would work with a less exact and worse behaved stiffness matrix of the same size and structure.

Since $M_{m_j}^{M_j}$ is non-singular, being a mass matrix, we can multiply by the tensor product matrix

$$(\otimes_{j=1}^{d}(M_{m_j}^{M_j})^{-1})$$

and obtain

$$\sum_{i=1}^{d}((\otimes_{j=1}^{i-1}I_{m_j})\otimes((M_{m_i}^{M_i})^{-1}K_{m_i}^{S_i})\otimes(\otimes_{j=i+1}^{d}I_{m_j}))+\alpha(\otimes_{j=1}^{d}I_{m_j})=\underline{f} \qquad (6.4)$$

This is a form amenable to the fast diagonalization method of section 9.2.

Next, we consider how to subassemble elements in a (hyper-)rectangular arrangement. Let us assume that each direction of the global arrangment is split into N_i parts, let R_j^i denote the restriction in direction i to the jth part, $1 \le j \le N_i$. Assume that the parts have degree m_{ij} and that they are covered with a Gauss-Lobatto-Legendre mesh. The solution on the arrangement is represented by a d-dimensional array of size $(1+\sum_{j=1}^{N_i} m_{ij})_{i=1}^{d}$. We enforce continuity between the local spaces $Q_{\{m_{ij}\}}$ and obtain a piecewise continuous polynomial global space which we will denote by $V_{\{m_{ij}\}}$. The element corresponding to the position $(j_i)_{i=1}^{d}$ in the arrangement is obtained by applying the tensor product matrix $\otimes_{i=1}^{d}R_{j_i}^i$ to the vector form of the array and the extension from the element to its corresponding place in the array is $\otimes_{i=1}^{d}R_{j_i}^{i,T}$.

Assuming $m_{ij}=m_i$, i.e., equal degrees in the parts in each direction, to avoid more indices, and using the quadrature degrees M_j and S_i as above, we have contributions as in equation (6.2) on each of the elements:

$$A_{(j_i)_{i=1}^{d}} := \left(\sum_{i=1}^{d}((\otimes_{j=1}^{i-1}M_{m_j}^{M_j})\otimes K_{m_i}^{S_i}\otimes(\otimes_{j=i+1}^{d}M_{m_j}^{M_j}))+\alpha(\otimes_{j=1}^{d}M_{m_j}^{M_j})\right)$$

If we allow varying degrees m_{ij}, then these contributions will depend on their index $(j_i)_{i=1}^{d}$. In the uniform case treated here they are actually all the same and independent of the index. The following manipulations do not exploit this fact, so that they will also hold for the nonuniform case, giving the result stated below.

To subassemble the system, we have to add up the contributions from all the elements:

$$\sum_{j_k,k=1,\ldots,d}(\otimes_{i=1}^{d}R_{j_k}^{i,T})\quad A_{(j_i)_{i=1}^{d}}(\otimes_{i=1}^{d}R_{j_k}^{i})\quad \underline{u}=$$

$$\sum\nolimits_{j_k,k=1,\ldots,d}\qquad(\otimes_{i=1}^{d}R_{j_k}^{i,T})(\otimes_{j=1}^{d}M_{m_j}^{M_j})(\otimes_{i=1}^{d}R_{j_k}^{i})\underline{f}$$

After some algebraic manipulations one obtains the following form:

$$\left(\sum_{i=1}^{d}((\otimes_{j=1}^{i-1}\tilde{M}_j)\otimes\tilde{K}_i\otimes(\otimes_{j=i+1}^{d}\tilde{M}_j))+\alpha(\otimes_{j=1}^{d}\tilde{M}_j)\right)\underline{u}=(\otimes_{j=1}^{d}\tilde{M}_j)\underline{f} \tag{6.5}$$

where $\tilde{M}_.$ and $\tilde{K}_.$ are defined as:

$$\tilde{M}_j=\sum_{j_j=1}^{N_j}R_{j_j}^{j,T}M_{m_j}^{M_j}R_{j_j}^{j}\qquad \tilde{K}_i=\sum_{j_i=1}^{N_i}R_{j_i}^{i,T}K_{m_i}^{S_i}R_{j_i}^{i} \tag{6.6}$$

If we premultiply the system with the tensor product of the inverses of the assembled mass matrices, $\tilde{M}_j^{-1}$ we obtain:

$$\sum_{i=1}^{d}((\otimes_{j=1}^{i-1}I_{m_j})\otimes((\tilde{M}_i)^{-1}\tilde{K}_i)\otimes(\otimes_{j=i+1}^{d}I_{m_j}))+\alpha(\otimes_{j=1}^{d}I_{m_j})=\underline{f} \tag{6.7}$$

This is again of the form required for the fast diagonalization method of section 9.2.

For varying degrees m_{ij} we just have to define $\tilde{M}_.$ and $\tilde{K}_.$ as

$$\tilde{M}_j=\sum_{k=1}^{N_j}R_{k}^{j,T}M_{m_{jk}}^{M_{jk}}R_{k}^{j}\qquad \tilde{K}_i=\sum_{k=1}^{N_i}R_{k}^{i,T}K_{m_{ik}}^{S_{ik}}R_{k}^{i} \tag{6.8}$$

and the systems are still of the form (6.5) and (6.7).

The derivation so far has been for homogenous Neumann boundary conditions, so that the solution is only determined up to a constant. In our implementation, we force the component corresponding to the eigenvector with eigenvalue zero to be zero, if no exact solution is known. If we know the exact solution, we force the numerical solution to have the same (approximate) integral as the known exact solution.

For nonhomogenous Neumann boundary conditions we obtain a boundary term on the right hand side, which also can be discretized by spectral methods. Assuming $\partial_{\mathbf{n}}u=g$, $f(v)$ in (6.1) is now $(f,v)_0+\int_{\partial\Omega}gv$. To discretize the additional boundary term

$$\int_{\partial\Omega}gv\approx\underline{v}^TG$$

we have to restrict v to $\partial\Omega$ and take the inner product in $L^2(\partial\Omega)$ (corresponding to a mass matrix $M_{\partial\Omega}$ in the discrete system), i.e.,

$$\int_{\partial\Omega}gv=(g,R_{\partial\Omega}v)_{0,\partial\Omega}\approx\underline{v}^TR_{\partial\Omega}^TM_{\partial\Omega}\underline{g}$$

The boundary $\partial\Omega$ is split into 2^d components and on each of the components the (composite) Gaussian quadrature associated to the given Gauss-Lobatto-Legendre mesh on that part of the boundary is used to compute the appropriate part of $M_{\partial\Omega}$. As an example, in the two-dimensional case with mass matrices $\tilde{M}_1$ in the x-direction and $\tilde{M}_2$ in the y-direction, with $g_d = g|_{\{y=-1\}}$, $g_u = g|_{\{y=1\}}$, $g_l = g|_{\{x=-1\}}$ and $g_r = g|_{\{x=1\}}$, and $I \otimes R_d$, $I \otimes R_u$, $R_l \otimes I$ and $R_r \otimes I$ being the restrictions to the lines $\{y = -1\}$, $\{y = 1\}$, $\{x = -1\}$ and $\{x = 1\}$, the boundary term is

$$R_{\partial\Omega}^T M_{\partial\Omega} \underline{g} = (I \otimes R_d^T)(\tilde{M}_1 \otimes I)\underline{g}_d + (I \otimes R_u^T)(\tilde{M}_1 \otimes I)\underline{g}_u + (R_l^T \otimes I)(I \otimes \tilde{M}_2)\underline{g}_l + (R_r^T \otimes I)(I \otimes \tilde{M}_2)\underline{g}_r$$

We will not give explicit forms for general d for $R_{\partial\Omega}$ and $M_{\partial\Omega}$ as sums of tensor product matrices, since they are not central to our discussion and they require a lot of notation to define concisely.

$R_{\partial\Omega}^T M_{\partial\Omega} \underline{g}$ is added to the right hand side of (6.5). In fast diagonalization methods we work with (6.7) and therefore need to multiply the above vector by the inverse of the (d-dimensional) mass matrix.

For homogenous Dirichlet boundary conditions we take the submatrix for the interior of the domain (which still has the same tensor product structure) and invert it. The solution technique is the same, except that we work with a principal minor of the discretization matrices. This corresponds to a choice of $H_0^1(\Omega)$ in (6.1) instead of $H^1(\Omega)$.

For nonhomogenous Dirichlet boundary conditions we first compute a lifting of the boundary values, correct the right hand side, and solve the resulting problem with homogenous Dirichlet boundary conditions.

6.2 Theoretical analysis

We will only work out the coercive case, i.e., $\alpha \geq 0$. We assume Dirichlet boundary conditions, for simplicity. Most of the results that we will cite assume uniform degree, that is, $m_i = N$.

Denote the approximation of $a(u, v)$ constructed by quadrature in the last section by $a_q(u, v)$ and also the approximation of $f(v)$ by $f_q(v)$. The discrete variational problem derived in the last section is then written

$$?u \in V_{\{m_{ij}\}} : \forall v \in V_{\{m_{ij}\}} : a_q(u, v) = f_q(v) \tag{6.9}$$

In the case of uniform degree N of the elements, and uniform quadrature degree N in all the integrations, we write a_N and f_N for a_q and f_q.

For $\alpha \geq 0$ we have the following version of Strang's Lemma:

Lemma 6.1 *Assume that a_q is elliptic with ellipticity constant γ_q on $V_{\{m_{ij}\}}$:*

$$\forall v \in V_{\{m_{ij}\}} : a_q(v,v) \geq \gamma_q ||v||_1$$

and assume that $a(u,v)$ has norm C_a over $H_0^1 \times H_0^1$. Then the following error estimate holds for solutions u_q of (6.9) and u of (6.1) (see [17, inequality (15.14)])

$$||u - u_q||_1 \leq C\left(1 + \frac{C_a}{\gamma_q}\right)\left(\inf_{v \in V_{\{m_{ij}\}}}\left(||u - v||_1 + \sup_{w \in V_{\{m_{ij}\}}} \frac{a(v,w) - a_q(v,w)}{||w||_1}\right) + \sup_{w \in V_{\{m_{ij}\}}} \frac{f(w) - f_q(w)}{||w||_1}\right)$$

To use this for the analysis of our methods, we need to be able to estimate the interpolation and the consistency errors. The interpolation error is bounded by just exhibiting a proper v which is chosen to be an appropriate projection of u. For the analysis of such projections see [17, sections 6 and 7]. To approximate the data of the problem, we need polynomial approximation estimates which can be found in [17, sections 13 and 14]. (For a small sampling of such results see also Chapter 4.) The consistency errors are bounded analyzing the quadrature, and the ellipticity and boundedness of a_q follow in a straighforward way from its form.

This allows us to prove, for the Laplace equation and therefore for all $\alpha \geq 0$, first a H^1-error estimate and then, by duality, the following L^2-error estimate [17, Theorem 15.14]:

Theorem 6.2 *Assume that $f \in H^\sigma$ for some $\sigma > d/2$ and $u \in H^s$ for some $s \geq 1$. Then the following error estimate holds for the solution u_N of $a_N(u,v) = f_N(v)$ with respect to the solution u of (6.1):*

$$||u - u_N||_0 \leq c(N^{-s}||u||_{H^s} + N^{-\sigma}||f||_{H^\sigma})$$

$h - N$ versions can also be treated, by combining the techniques used in proving Theorem 6.2 with the $h - p$ estimates of Babuška and coworkers (see, e.g., Babuška and Guo [9]).

For $\alpha < 0$ the bilinear form is not coercive, there is only a Gårding inequality satisfied. We could apply the standard arguments for non-coercive problems (see, e.g., Brenner and Scott [22, sections 5.6-5.8]) to obtain error estimates for fine enough grids respectively high enough degrees that depend on the magnitude of α. We will not venture into this subject in this thesis, we will contend ourselves with some numerical tests in the next section.

6.3 Numerical experiments

We will first run some experiments with the one-dimensional version, studying how the quality of the numerical solutions depends on the degree S of the integration of the stiffness matrix and the degree M of the integration of the mass matrix.

The exactly integrated stiffness matrices have the eigenvalues shown in figure 6.1 for the solution of Neumann and Dirichlet problems. The eigenvalues below 10^{-10} for the Neumann problem correspond to the zero eigenvalue of the continous problem. That the numerical eigenvalues are not exactly zero is caused both by the eigenvalue computation algorithm and the fact that spectral differentiation matrices differentiate constants not to the zero-polynomial, but to a polynomial having nodal values that are multiples of the machine accuracy. In our algorithms that use eigendecompositions, we set all eigenvalues below a threshold (usually 10^{-10}) to zero and treat the associated eigenvectors as zero eigenvectors.

We first solved a nonhomogenous Dirichlet problem on $[-1, 1]$ with the exact solution $u(x) = e^{\sin(x)}$ with different degrees of integration, testing both overintegration and slight underintegration for the stiffness matrix and also strong underintegration for the mass matrix. See figure 6.2 for the results.

We see that an underintegration of the stiffness matrix (with $S = m - 2$) brings devastating consequences, instead of exponential convergence we only obtain (even relatively slow) algebraic convergence (of an order of $2.86 \approx 3$).

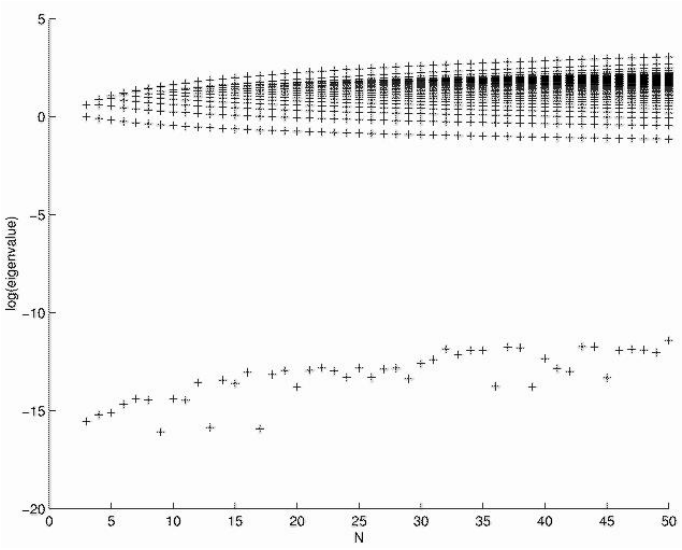
The other cases all look more or less alike and harbor the signs of exponential convergence, it only seems that decreasing the degree of integration of the mass matrix delays the convergence. To verify this impression, we first look closer at the case of exact integration of the stiffness matrix, see figure 6.3.

We see that there is no difference between exact and slightly underintegrated mass matrices, and the other two choices are worse, corresponding to non-optimal exponents in the exponential convergence. Estimating the loss of convergence by eye, the one with $M = m - 1$ seems to correspond to a difference in the exponent of about 1, and the one with $M = m-5$ seems to lag by about 10.

We would obtain the same results, if we would look at the case of the other integrations of the stiffness matrix, only that all the graphs are indistinguishable in the case where we integrate the stiffness matrix with $S = m - 2$.

If we look closer at the dependence on the integration of the stiffness matrix with a given integration of the mass matrix, we observe, both for the exact integration and the slight underintegration of the mass matrix, the same behavior. We show the later case in figure 6.4. The first two choices yield the same result, the third one is slightly less accurate. For underintegrated mass matrices, we can not distinguish between the results. For strong

Eigenvalues: Neumann boundary conditions



Eigenvalues of the interior part: Dirichlet boundary conditions

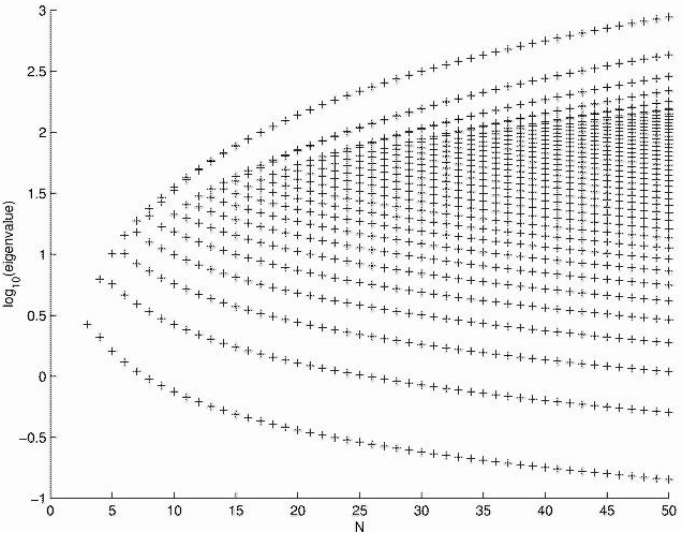


Figure 6.1: One-dimensional Poisson problem, Eigenvalues of the stiffness matrix for the Neumann and the Dirichlet problem

underintegration of the mass matrix, integration with $S = m - 1$ outperforms the others by a very small margin. (We chose not to show the case $S = m - 2$ which is already seen to be far worse in figure 6.2.)

In the next figure, figure 6.5, we show the case of Neumann boundary conditions. The result

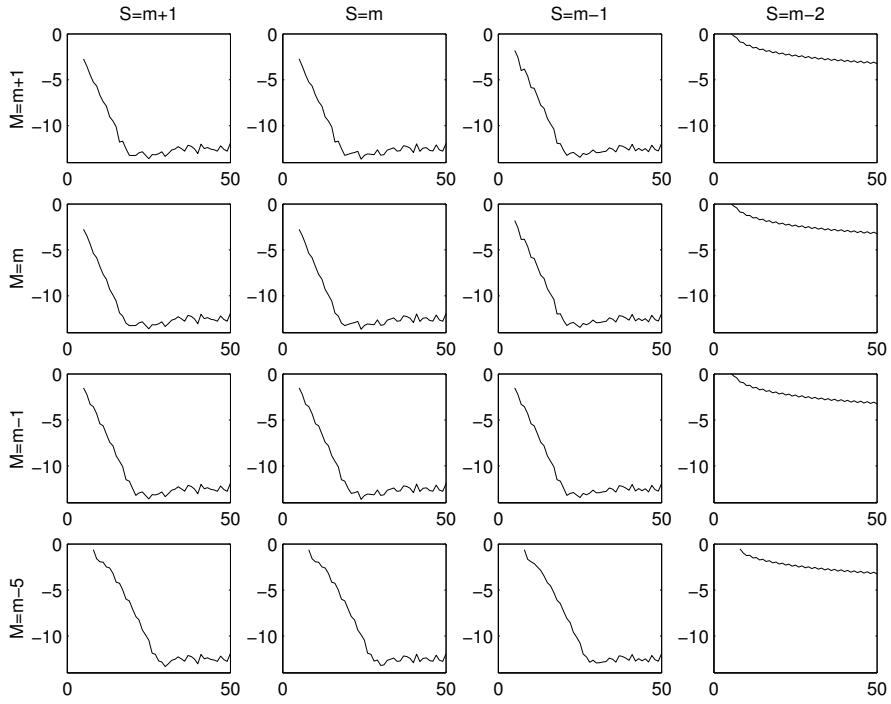


Figure 6.2: One-dimensional Poisson problem with Dirichlet boundary conditions: differing degrees of integration, S for the stiffness matrix and M for the mass matrix.

looks very similar to the results in 6.2, and we can make exactly the same observations as for Dirichlet boundary conditions. Figures 6.6 and 6.7 show this fact and correspond to figures 6.3 and 6.4 in the Dirichlet case.

The results agree with our theoretical expectations. Maday and Rønquist [68] and Bernardi and Maday [17] argue that for the Laplace equation overintegration of the stiffness matrix does not improve the results. They also show that for some problems with variable coefficients or in distorted geometries, overintegration is needed for optimal convergence.

In the case of underintegration, we can use Strang's lemma, Lemma 6.1, which bounds the error of the solution as a constant times the sum of the approximation error and the consistency error of the bilinear form (i.e., the stiffness matrix) and the error of integration on the right hand side (i.e., the mass matrix). Underintegration of the stiffness matrix leads to a very bad approximation of the bilinear form, and this error term dominates the error estimate in this case. If we approximate the stiffness matrix well, the dominating term is the approximation of the mass matrix. If we use lower-order integration (i.e., less than $M = m$) we loose some constant in the exponent, but still obtain exponential convergence.

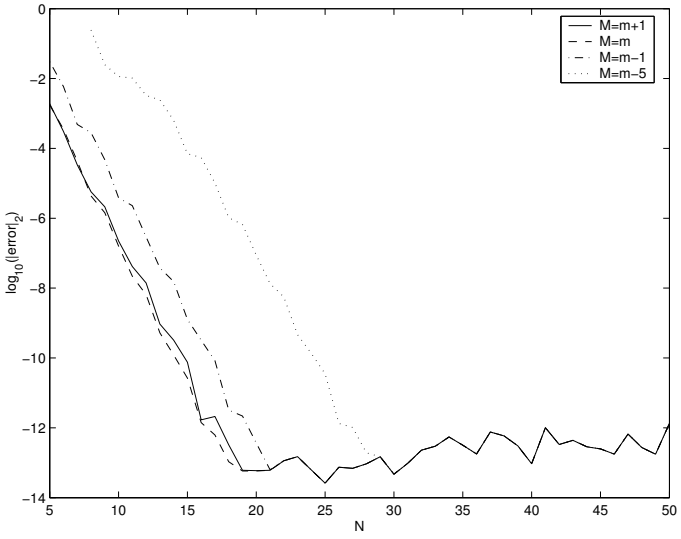


Figure 6.3: One-dimensional Poisson problem, Dirichlet boundary conditions, exact integration of the stiffness matrix: Influence of the integration of the mass matrix

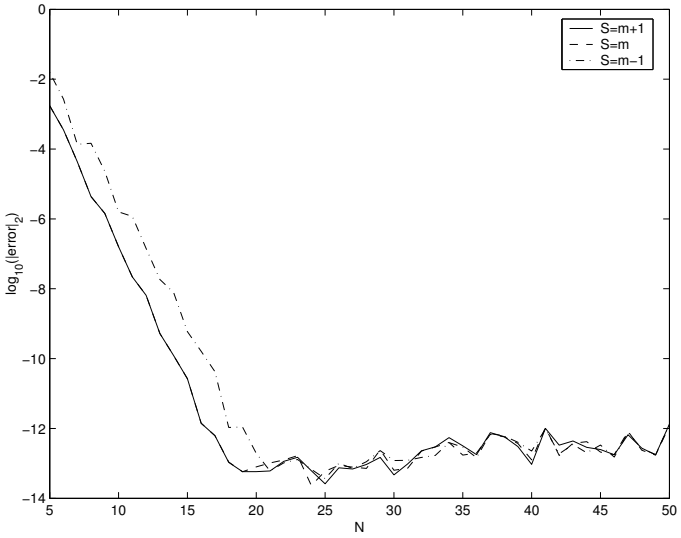


Figure 6.4: One-dimensional Poisson problem, Dirichlet boundary conditions, mass matrix slightly underintegrated: Influence of the integration of the stiffness matrix

In conclusion, reasonable orders of integration are $S = m$ and $M = m + 1$ and $M = m$. $S = m + 1$ gives similar results (without numerical errors they should be exactly the same)

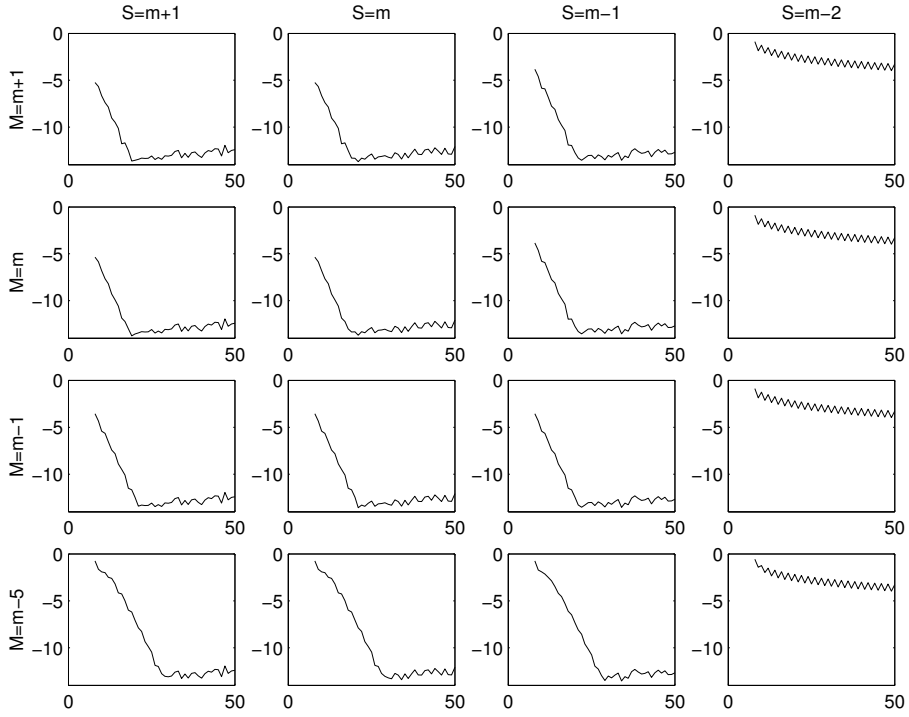


Figure 6.5: One-dimensional Poisson problem with Neumann boundary conditions: differing degrees of integration, S for the stiffness matrix and M for the mass matrix.

at higher costs. $S = m-1$ still gives quite good results, but they are worse than for $S = m$, and the choice $S = m - 1$ does not lead to lower computational effort compared with $S = m$.

$M = m$ leads to diagonal mass matrices, which makes the inversion of the mass matrices lower in computational cost, so that its use could be advised if we need to invert mass matrices repeatedly.

Next we test the one-dimensional Helmholtz operator with the four choices $S = m, m - 1$ and $M = m + 1$,$M = m$. We perform the tests only for Dirichlet boundary conditions and expect the same results for the Neumann boundary conditions. We show the results for $\alpha = 1$, $\alpha = 100$ and $\alpha = -100$ in figure 6.8. We use the same exact solution as for the Poisson equation.

The cases with $S = m$ outperform the ones with $S = m - 1$ by a small margin. The results for $M = m + 1$ and $M = m$ are very close, and for $S = m$ and $M = m + 1$ the results actually seems to be slightly more accurate in the exponential convergence.

So large positive or negative α do not seem to change the behavior of the solution process

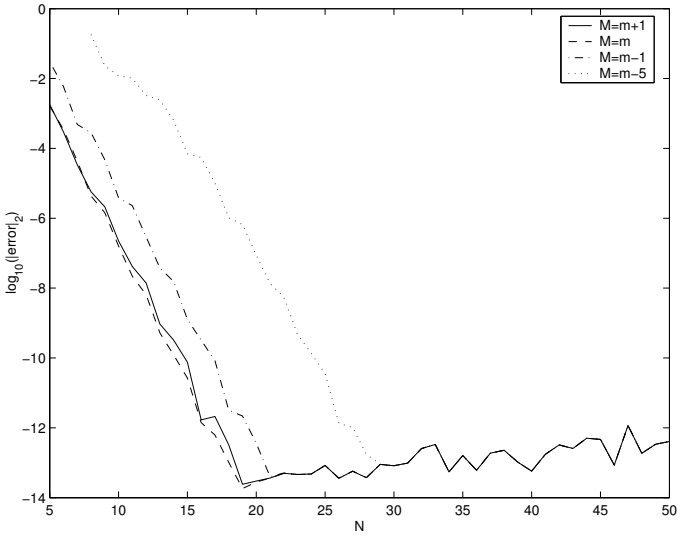


Figure 6.6: One-dimensional Poisson problem, Neumann boundary conditions, exact integration of the stiffness matrix: Influence of the integration of the mass matrix

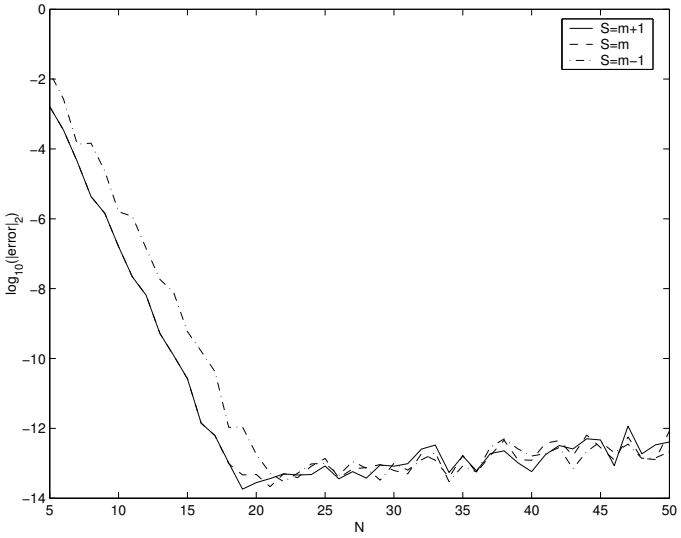
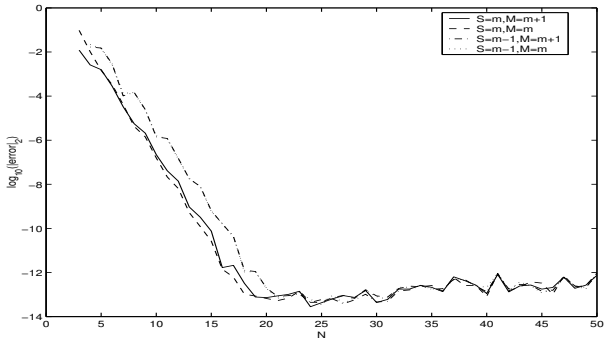


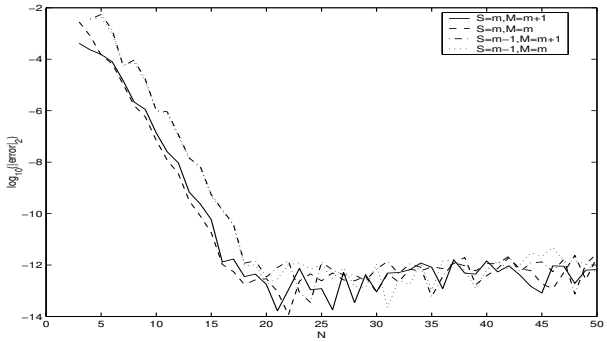
Figure 6.7: One-dimensional Poisson problem, Neumann boundary conditions, mass matrix slightly underintegrated: Influence of the integration of the stiffness matrix

by much.

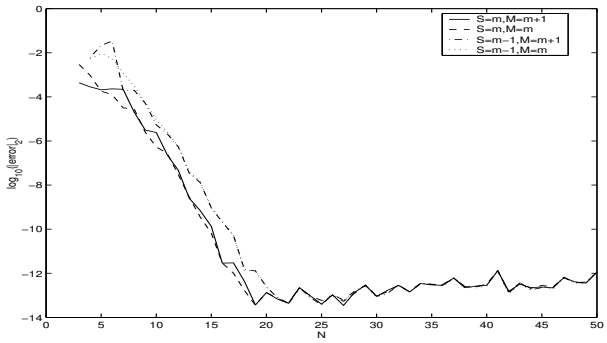
If we run the same test, but with a solution that depends on α and is increasingly oscillatory



$\alpha = 1$



$\alpha = 100$

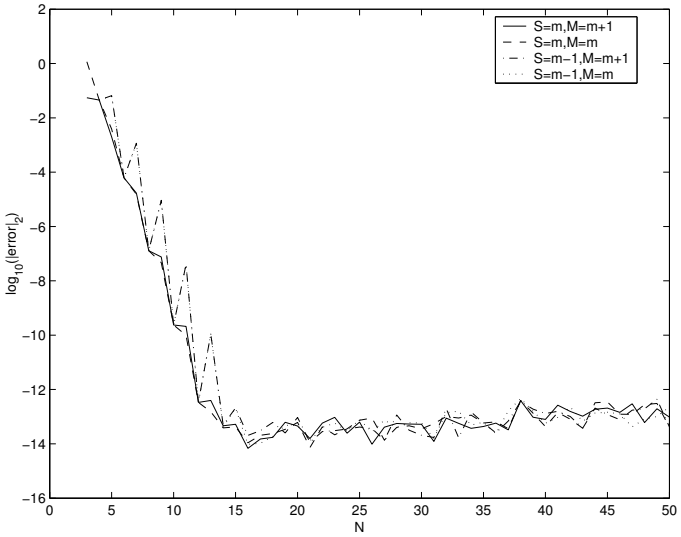


$\alpha = -100$

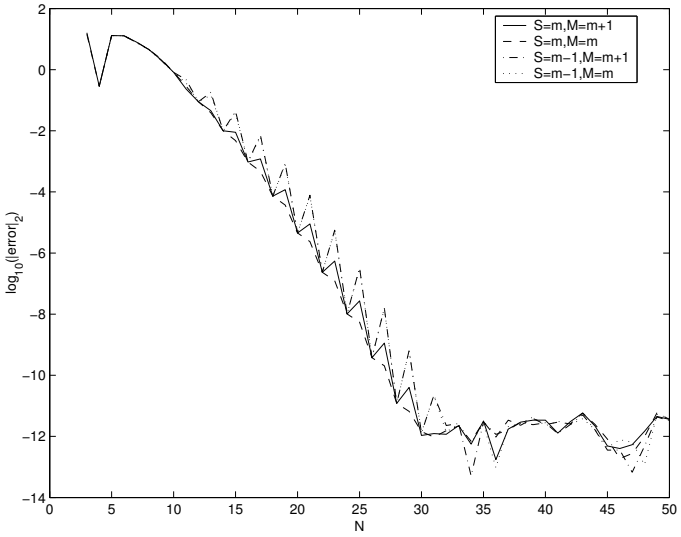
Figure 6.8: Solving one-dimensional Helmholtz problems with Dirichlet boundary conditions: tests for $\alpha = 1$, $\alpha = 100$ and $\alpha = -100$

for large negative α, we catch a more typical solution of the Helmholtz equation. We chose the solution $(x^2 - 1)\cos(\alpha x)$, and tested for $\alpha = -1$ and $\alpha = -10$. Besides solving the problem accurately, the grid has to be fine enough, or the degree has to be large enough to resolve a solution of high frequency. That explains the delay in reaching the optimal error

in figure 6.9. We also observe an odd-even effect which seems to be more pronounced for the cases with underintegrated stiffness matrices.



$\alpha = -1$



$\alpha = -10$

Figure 6.9: Solving one-dimensional Helmholtz problems with Dirichlet boundary conditions: tests for $\alpha = -1$ and $\alpha = -10$ with an oscillatory exact solution

We also performed some experiments for the two-dimensional case, again for $\alpha = 0$, $\alpha =$

-1 and $\alpha = -10$. We impose nonhomogenous Neumann boundary conditions. The exact solutions used were: for $\alpha = 0$, $u(x, y) = \sin(x + y)\cos(x - y)$, for $\alpha < 0$, $u(x, y) = \cos(\alpha\, x)\sin(2\alpha\,(x + y))$.

In figure 6.10 we show the results for $\alpha = 0$. The two versions with exact stiffness matrices perform very much alike, and we see a slight odd-even effect. The two versions with inexact stiffness matrices are also very close together, but show a stronger odd-even effect. For odd N, they are as accurate as the first two, for even N they are worse.

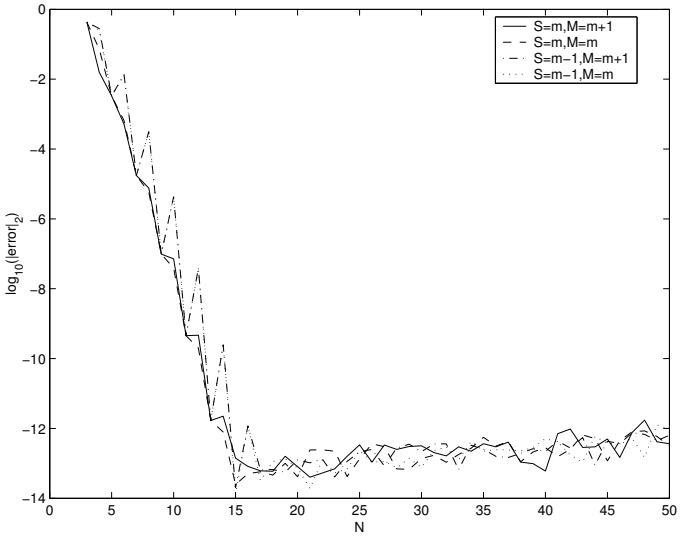


Figure 6.10: Solving a two-dimensional Poisson problem with Neumann boundary conditions

In figure 6.11, the case $\alpha = -1$ is shown. Here there seems to be no odd-even effect; the version with exact stiffness and diagonal mass matrix performs slightly better than the version with the exact mass matrix, and the two versions with inexact mass matrices perform alike and worse.

The results for $\alpha = -10$ with an oscillatory solution are presented in figure 6.12. Note that we had to use a wider range of N to capture the different stages of the behavior of the error. First, till about $N = 30$, all versions perform alike. All of them do not resolve the solution yet, and there is no convergence. Then, between $N = 30$ and $N = 40$, the solution starts to converge, first slowly, and then, after $N = 40$, attaining spectral convergence. Around $N = 55$, we reach the best accuracy of the implementation of the method, and the error does not improve any longer.

Finally, we show two examples for solutions of subassembled problems. We pose problems with Neumann boundary conditions with given exact solutions.

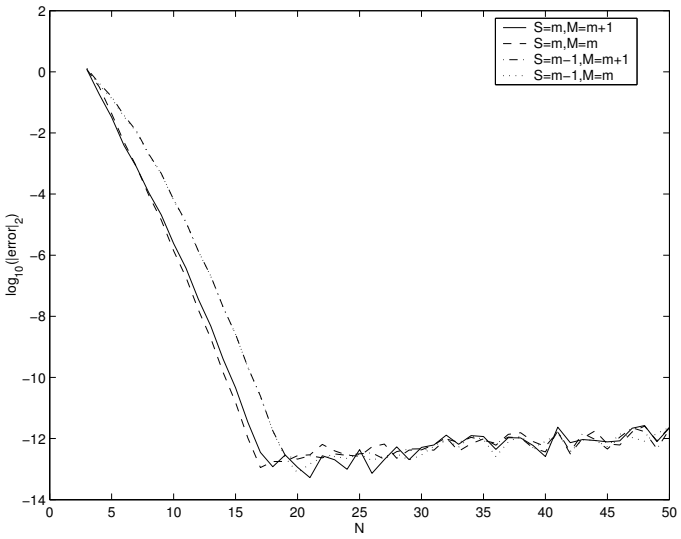


Figure 6.11: Solving a two-dimensional Helmholtz problem ($\alpha = -1$) with Neumann boundary conditions

In figure 6.13 we show an one-dimensional example on 10 elements of equal size in $[-1, 1]$ with the exact solution $u(x) = e^{\sin(x)}$.

In figure 6.14 a two-dimensional example is shown, with exact solution $\cos(\pi x)\cos(\pi y)e^{\frac{x+y}{4}}$, using 10x10 spectral elements of equal size in $[-1, 1]^2$.

We observe that the results are very simar to the single element case. We find spectral convergence in the first half of the graph, and after reaching a minimal error there is no further reduction as we reach the best accuracy numerically possible. We start with a smaller error and reach the maximal accuracy around $N = 10$ because we already start with an approximation resolving most of the features of the solution on the initial grid of 10 or 10x10 spectral elements.

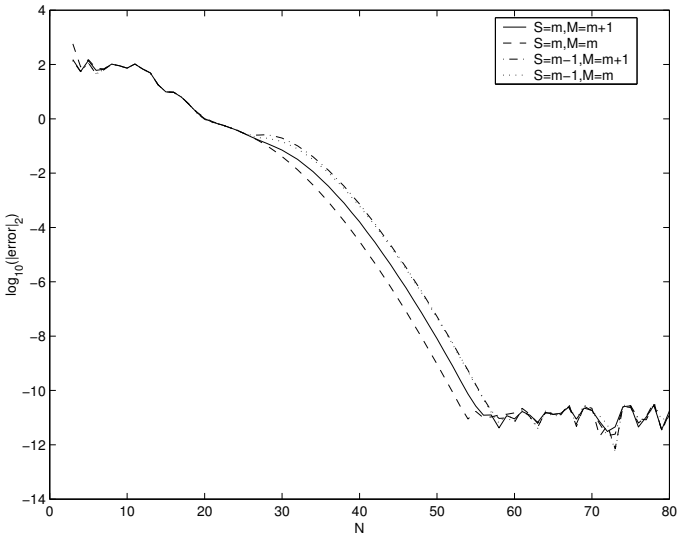


Figure 6.12: Solving a two-dimensional Helmholtz problem ($\alpha = -10$) with Neumann boundary conditions

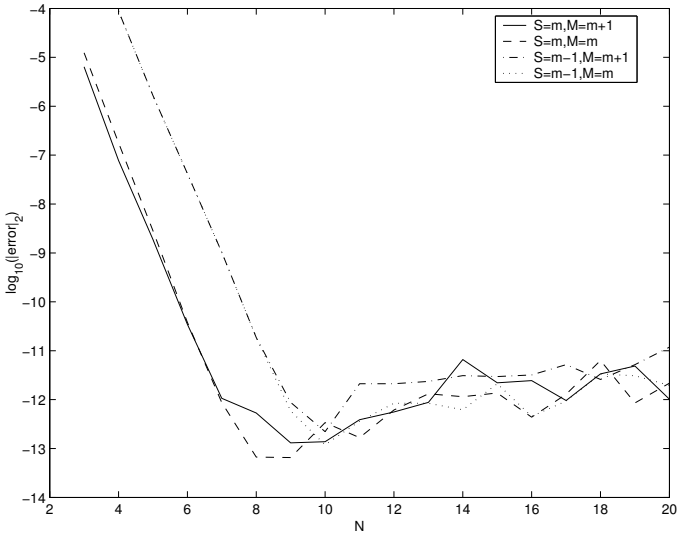


Figure 6.13: Solving an one-dimensional Poisson problem on 10 spectral elements with Neumann boundary conditions

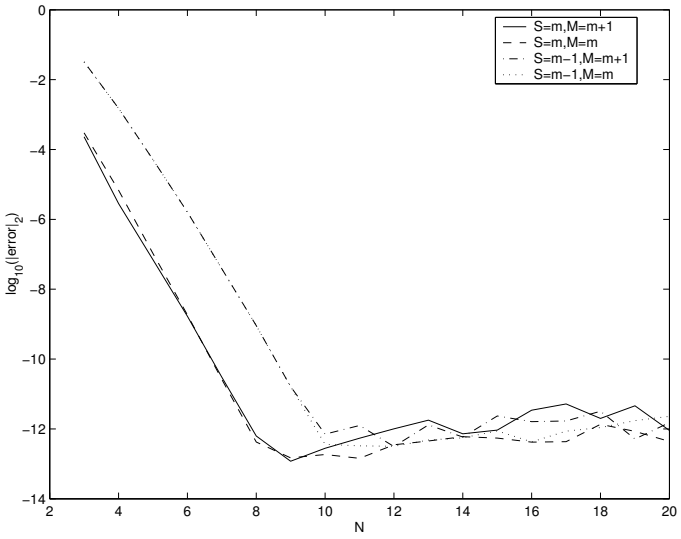


Figure 6.14: Solving a two-dimensional Poisson problem on 10x10 spectral elements with Neumann boundary conditions

Chapter 7

Spectral element spaces for vector field problems

In this chapter we will construct and analyze spectral element spaces for vector field problems posed in the graph spaces $H(\mathbf{curl})$ and $H(\mathsf{div})$. Unlike H^1 conforming elements, they require only partial continuity across the element interfaces. To derive optimal results, carefully constructed interpolation operators are needed. Unfortunately, the known interpolation operators require more regularity than their H^1 counterparts; in particular, they are not defined for the whole space ($H(\mathbf{curl})$ or $H(\mathsf{div})$, respectively). In the $H(\mathbf{curl})$ case such elements have been first proposed by Nédélec in a h-version [74]; we will introduce generalized Nédélec spectral elements in the first section.

In two dimensions, just like in the continuous setting, the $H(\mathrm{curl})$ case can be obtained by a rotation of the $H(\mathsf{div})$ case. In the analysis of the approximation properties of the $\mathbf{curl}$ of the interpolant in three dimensions we use the commuting diagram property and the appropriate $H(\mathsf{div})$ conforming spaces. Therefore, in the second section, we present these $H(\mathsf{div})$ conforming spaces called Raviart-Thomas-Nédélec spaces. In the next section we introduce some other spectral element spaces and state the commuting diagram properties for the appropriate interpolation operators. We also report on the discrete analogue of the Helmholtz decomposition and the kernel of $\mathbf{curl}$ (compare section 2.5 for the continuous case). The approximation properties of the Raviart-Thomas-Nédélec spaces are known, even in the hN-version, and are needed for the proof of some of the approximation properties of the Nédélec elements, and we will state them in the next section. The approximation and interpolation results for the Nédélec elements follow. We need to study Nédélec type interpolants between Nédélec elements of different order and between Raviart-Thomas-Nédélec elements of different order, for later use as restriction and extension operators in multi-level algorithms and in the analysis of domain decomposition preconditioners. We derive the form of the interpolants in the sixth section, and numerically study their be-

havior. Finally, in the analysis of the model problem and of the domain decomposition preconditioners, we will need a discrete Friedrichs' inequality and an approximation result, which we formulate and prove in the last section.

7.1 Generalized Nédélec elements in $H(\mathrm{curl})$

In this section we will construct $H(\mathrm{curl})$ conforming elements. Only tangential components have to match across interfaces in $V(\mathrm{curl})$ to guarantee conformity in $H(\mathrm{curl})$. If we work with polynomials of equal degree on the two sides of the interface, this matching leads to the continuity of the tangential components across the interface.

Remark: To be precise, the equality of tangential components is only enforced in the sense of $H_{00}^{-1/2}$ on the interface (see, e.g., Hiptmair [57, Corollary 2.6, pg.9]). By enforcing continuity of the tangential components we certainly satisfy equality in that sense. Weaker conditions bring with them technical difficulties and matching operators that are harder to treat analytically, algorithmically and numerically. The weak continuity conditions used in Mortar element methods are posed in a similar space $(H^{1/2})'$ (see, e.g., Wohlmuth [103, following (2.2)]) so that one could adapt such methods to find more general $H(\mathrm{curl})$ conforming elements. Those elements will a priori not have a local characterization. We will not pursue these ideas any further in this thesis.

7.1.1 Local spaces

We use high order polynomial local spaces. Since we want to use tensorial bases, we have to choose $\mathbb{Q}_{m,n}$-like spaces for each component. To construct $H(\mathrm{curl})$ conforming elements, the tangential components of the vector field have to match on the interface between elements. Therefore the tangential components (and their degree) of the local spaces have to agree across an interface. In the easiest case, a rectangular arrangement of elements, and standard $\mathbb{Q}_{m,n}$ spaces, this forces the degree of the tangential components to be the same across the domain. If we wish to have different local (tangential) degrees in the elements, we could choose to implement only weak continuity conditions across the interface, leading to mortar elements[1]. We could also use variable order elements (see, e.g., Demkowicz [41] and Ainsworth and Coyle [2]) or local uniform refinement and domain decomposition methods constructed for such situations (for similar methods for the Poisson equation see Pavarino [82, Chapter 4]). In most variable order element approaches the degrees of

[1] For an introduction to mortar elements see Bernardi, Maday, and Patera [18, 19], for a more modern version see Wohlmuth [103], and for mortar elements for Maxwell's equation see the work by Ben Belgacem and coworkers, for instance in Ben Belgacem, Buffa, and Maday [16].

freedom are associated with geometrical objects, such as interiors of elements or groups of elements, edges and vertices. In that way the matching of degrees of freedom across element interfaces is automatic.

In this thesis, we will concentrate on standard $\mathbb{Q}_{m,n}$ spaces, since we are interested in solvers that use their tensorized structure. It may be possible to generalize some of the methods to additional cases, but we will not strive for utmost generality in this respect.

Consider a rectangular element K. The most general local space for two dimensions is then $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$.

Nédélec elements of the first kind (Nédélec [74]) of order k on rectangles use the following local spaces:

$$ND_k^I(K) = \mathbb{Q}_{k-1,k}(K) \times \mathbb{Q}_{k,k-1}(K) \supset \mathbf{grad}\, \mathbb{Q}_{k,k}$$

while Nédélec elements of the second kind (Nédélec [75]) of order k on rectangles use

$$ND_k^{II}(K) = \mathbb{Q}_{k,k}(K) \times \mathbb{Q}_{k,k}(K)$$

We could also choose different degrees in different directions to generalize these spaces, i.e.,

$$ND_{m,n}^I(K) = \mathbb{Q}_{m-1,n}(K) \times \mathbb{Q}_{m,n-1}(K) \supset \mathbf{grad}\, \mathbb{Q}_{m,n}$$
$$ND_{m,n}^{II}(K) = \mathbb{Q}_{m,n}(K) \times \mathbb{Q}_{m,n}(K)$$

The global spaces corresponding to these local spaces will be denoted by the same symbol, but set in a blackboard style, i.e., $\mathbb{ND}_k^I(\Omega)$, $\mathbb{ND}_k^{II}(\Omega)$, $\mathbb{ND}_{m,n}^I(\Omega)$ and $\mathbb{ND}_{m,n}^{II}(\Omega)$. $\mathbb{ND}_k^{I,0}(\Omega)$, $\mathbb{ND}_k^{II,0}(\Omega)$, $\mathbb{ND}_{m,n}^{I,0}(\Omega)$ and $\mathbb{ND}_{m,n}^{II,0}(\Omega)$ stand for the discrete spaces with zero tangential components at the boundary.

We can find potentials (in the Helmholtz decomposition) in local spectral element spaces, which makes certain operations, such as curl-free corrections, numerically more accessible. (See, for instance, Hiptmair [59].)

If we compute the curl of the local Nédélec in the two-dimensional spaces, we obtain:

$$\mathrm{curl}\, ND_{m,n}^I(K) \subset \mathbb{Q}_{m-1,n-1}(K)$$

$$\mathrm{curl}\, ND_{m,n}^{II}(K) \subset \mathbb{Q}_{m,n-1}(K) + \mathbb{Q}_{m-1,n}(K)$$

There is no continuity between the curl of the local Nédélec spaces across the interfaces in $\mathrm{curl}\, \mathbb{ND}(\Omega)$.

If we rotate the spaces of the Nédélec elements of the first kind by ninety degrees, we obtain:

$$\text{Rotation by } 90^\circ \text{ of} \quad ND_k^I(K) \quad \text{is} \quad \mathbb{Q}_{k,k-1}(K) \times \mathbb{Q}_{k-1,k}(K)$$

$$\text{Rotation by } 90^\circ \text{ of } \quad ND^I_{m,n}(K) \quad \text{ is } \quad \mathbb{Q}_{m,n-1}(K) \times \mathbb{Q}_{m-1,n}(K)$$

These spaces will turn out to be the $H(\mathsf{div})$ conforming Raviart-Thomas-Nédélec spaces RT_k and $RT_{m,n}$ in two dimensions that will be introduced in the next section.

In three dimensions, the local spaces for the (generalized) Nédélec elements are defined as:

$$ND^I_k(K) = \mathbb{Q}_{k-1,k,k}(K) \times \mathbb{Q}_{k,k-1,k}(K) \times \mathbb{Q}_{k,k,k-1}(K) \supset \mathbf{grad}\, \mathbb{Q}_{k,k,k}(K)$$

$$ND^I_{l,m,n}(K) = \mathbb{Q}_{l-1,m,n}(K) \times \mathbb{Q}_{l,m-1,n}(K) \times Q_{l,m,n-1}(K) \supset \mathbf{grad}\, \mathbb{Q}_{l,m,n}(K)$$

$$ND^{II}_k(K) = (\mathbb{Q}_{k,k,k}(K))^3$$

$$ND^{II}_{l,m,n}(K) = (\mathbb{Q}_{l,m,n}(K))^3$$

Computing the $\mathbf{curl}$ of the Nédélec elements of the first kind gives us

$$\mathbf{curl}\, ND^I_k(K) \subset \mathbb{Q}_{k,k-1,k-1}(K) \times \mathbb{Q}_{k-1,k,k-1}(K) \times \mathbb{Q}_{k-1,k-1,k}(K)$$

$$\mathbf{curl}\, ND^I_{l,m,n}(K) \subset \mathbb{Q}_{l,m-1,n-1}(K) \times \mathbb{Q}_{l-1,m,n-1}(K) \times \mathbb{Q}_{l-1,m-1,n}(K)$$

Here, normal components in the global space $\mathbf{curl}\, \mathbb{ND}(\Omega)$ will match across interfaces.

The supersets will turn out to be the $H(\mathsf{div})$ conforming Raviart-Thomas-Nédélec spaces RT_k and $RT_{l,m,n}$ in three dimensions that will be introduced in the next section.

7.1.2 Degrees of freedom and interpolants

We want to construct spectral element type discretizations. Therefore we will use tensorized nodal basis functions built from interpolants on a Gauss-Legendre (GL) or Gauss-Lobatto-Legendre (GLL) mesh.

On rectangles, there are continuity conditions on the edges, as shown in figure 7.1.

To have tangential degrees of freedom match across the interfaces, we should use GLL meshes in y for u_1 and in x for u_2. If we use GLL meshes in x for u_1 and in y for u_2, degrees of freedom at the same position on different sides of the interface should not be identified, since normal components do not have to match.

At corners, either all components or no components have to match. In the general case, point values are not defined, and it does not make sense to match undefined objects. Considering polynomials and endpoints as limits we would be enticed to match all components. In several numerical experiments enforcing or not enforcing corner continuity did not seem to lead to vastly different results.

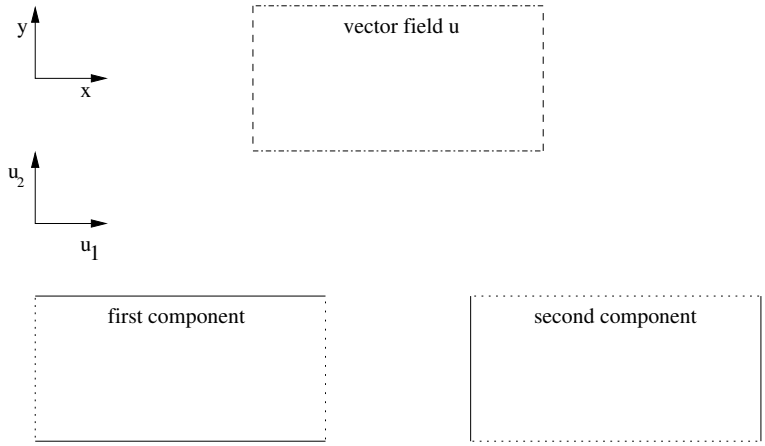


Figure 7.1: Continuity conditions for $H(\mathrm{curl})$-conforming elements in 2D. Dash-dotted line: first component continuous. Dashed line: second component continuous. Solid line: component is continuous. Dotted line: no continuity enforced.

For certain error indicators (see, e.g., Monk [72], and Beck, Hiptmair, Hoppe, and Wohlmuth [14]) and other computations it is useful to have jumps in the normal components available, which would favor GLL meshes. If we use GL meshes in x for u_1 and in y for u_2, we do not have any degrees of freedom in the normal components located on the interface, and we can use the slightly more accurate GL quadrature.

Recall from section 4.1, that GLL_m and GL_m stand for the Gauss-Lobatto-Legendre and Gauss-Legendre mesh with m points, and that the nodal values on it determine a polynomial of degree $m-1$ uniquely. (We assume that the meshes are appropriately scaled and translated so that they cover the sides of K.) We will give the spectral element type degrees of freedom for the general local spaces $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$ and $\mathbb{Q}_{l_1,m_1,n_1}(K) \times \mathbb{Q}_{l_2,m_2,n_2}(K) \times \mathbb{Q}_{l_3,m_3,n_3}(K)$ in two and three dimensions, respectively.

The degrees of freedom for the GLL-only method are the nodal values at

$$(\mathrm{GLL}_{m_1+1} \otimes \mathrm{GLL}_{n_1+1}) \times (\mathrm{GLL}_{m_2+1} \otimes \mathrm{GLL}_{n_2+1}) \tag{7.1}$$

where the normal components on the boundary are defined as the appropriate one-sided limit from the inside of the element.

The degrees of freedom for the GLL-GL method are the nodal values at

$$(\mathrm{GLL}_{m_1+1} \otimes \mathrm{GL}_{n_1+1}) \times (\mathrm{GL}_{m_2+1} \otimes \mathrm{GLL}_{n_2+1}) \tag{7.2}$$

Similarly, in three dimensions, the GLL-GL method uses the nodal values on

$$(\mathrm{GL}_{l_1+1} \otimes \mathrm{GLL}_{m_1+1} \otimes \mathrm{GLL}_{n_1+1}) \times (\mathrm{GLL}_{l_2+1} \otimes \mathrm{GL}_{m_2+1} \otimes \mathrm{GLL}_{n_2+1}) \\ \times (\mathrm{GLL}_{l_3+1} \otimes \mathrm{GLL}_{m_3+1} \otimes \mathrm{GL}_{n_3+1}) \tag{7.3}$$

as degrees of freedom, while the GLL-only method uses as degrees of freedom the values on the GLL mesh in all components and directions:

$$(\mathrm{GLL}_{l_1+1} \otimes \mathrm{GLL}_{m_1+1} \otimes \mathrm{GLL}_{n_1+1}) \times (\mathrm{GLL}_{l_2+1} \otimes \mathrm{GLL}_{m_2+1} \otimes \mathrm{GLL}_{n_2+1}) \\ \times (\mathrm{GLL}_{l_3+1} \otimes \mathrm{GLL}_{m_3+1} \otimes \mathrm{GLL}_{n_3+1}). \tag{7.4}$$

In our numerical experiments, we use the GLL-only method. We will give the derivation of the one element system for the Maxwell model problem in chapter 8 for this case only, but it is straightforward to extend it to the GLL-GL method.

Denote the standard nodal interpolation operator on one element for the GLL-only method by $\mathcal{I}^{GLL}_{m_1,n_1;m_2,n_2}$, $\mathcal{I}^{GLL}_{m,n,I}$ (for the $ND^I_{m,n}$ space) etc., and the one for the GLL-GL method, similarly, only with a superscript GLL-GL instead of GLL. The global interpolation operator is defined element by element using $\mathcal{I}_\cdot$, and is denoted by $\mathbf{I}_\cdot$. We need to be able to define point values to define these interpolants. To decrease the required regularity (so that we can approximate also solutions of lower regularity), there are several ways, for instance using averages (Clement [28]), a dual basis (Scott and Zhang [89], Brenner and Scott [22, section 4.8]), or quasi-interpolants (Oswald [78, section 2.1.1]). In the analysis of our methods, we would prefer to have several properties: the interpolation should be defined locally, it should respect boundary values, and the appropriate interpolant of the $\mathbf{curl}$ of the function should be equal to the $\mathbf{curl}$ of the interpolant of the function (one of the commuting diagram properties). It is possible to define nodal interpolation operators on the GLL mesh with the first two properties, but it is not clear how to enforce the third condition.

The degrees of freedom introduced by Nédélec for elements of the first kind of order k are ($\mathbf{t}_e$ is the direction vector of the edge e):

$$\int_e \mathbf{u} \cdot \mathbf{t}_e p \qquad p \in \mathbb{Q}_{k-1}(e) \qquad \text{for all edges } e \text{ of } K. \tag{7.5}$$

$$\int_K \mathbf{u} \cdot \mathbf{p} \qquad \mathbf{p} \in \mathbb{Q}_{k-1,k-2}(K) \times \mathbb{Q}_{k-2,k-1}(K) \qquad \text{for } k > 1. \tag{7.6}$$

For the Nédélec elements of the second kind we could choose:

$$\int_e \mathbf{u} \cdot \mathbf{t}_e p \qquad p \in \mathbb{Q}_k(e) \qquad \text{for all edges } e \text{ of } K. \tag{7.7}$$

$$\int_K \mathbf{u}\cdot\mathbf{p} \qquad \mathbf{p}\in \mathbb{Q}_{k-2,k}(K)\times\mathbb{Q}_{k,k-2}(K) \qquad \text{for } k>1. \tag{7.8}$$

Similarly, we can define such degrees of freedom for the anisotropic case for elements of both kinds.

In three dimensions the degrees of freedom are the appropriate interior moments and edge moments, as above, and there are also face moments as degrees of freedom. For instance, Nédélec elements of the first kind in three dimensions have the following degrees of freedom:

$$\int_e \mathbf{u}\cdot\mathbf{t}_e p \qquad p\in\mathbb{Q}_{k-1}(e) \qquad \text{for all edges } e \text{ of } K. \tag{7.9}$$

$$\int_F (\mathbf{u}\times\mathbf{n})\cdot\mathbf{p} \qquad \mathbf{p}\in\mathbb{Q}_{k-2,k-1}(F)\times\mathbb{Q}_{k-1,k-2}(F) \qquad \text{for all faces } F \text{ of } K. \tag{7.10}$$

$$\int_K \mathbf{u}\cdot\mathbf{p} \qquad \mathbf{p}\in\mathbb{Q}_{k-1,k-2,k-2}(K)\times\mathbb{Q}_{k-2,k-1,k-2}(K)\times\mathbb{Q}_{k-2,k-2,k-1}(K) \quad \text{for } k>1. \tag{7.11}$$

The degrees of freedom of Nédélec elements of the second kind, and of the anisotropic versions, have the same form, only that the spaces for p and $\mathbf{p}$ will have different degrees.

Associated to these degrees of freedom is an interpolation operator, which we will denote by $\boldsymbol{\Pi}^{ND}_{m_1,n_1;m_2,n_2}$, $\boldsymbol{\Pi}^{ND,I}_{m,n}$, and $\boldsymbol{\Pi}^{ND,I}_{k}$, which is defined element by element using the element versions $\Pi^{ND}_{m_1,n_1;m_2,n_2}$, $\Pi^{ND,I}_{m,n}$, and $\Pi^{ND,I}_{k}$.We also introduce the analogous notations for the elements of the second kind and for the three-dimensional case. This interpolation operator is local, respects tangential boundary conditions and satisfies the commuting diagram property[2], but it is not defined for all vector fields in $H(\operatorname{curl})$. To wit, the interior degrees of freedom are defined for all of $H(\operatorname{curl})$, but the edge moments (and the face moments in three dimensions) need more regularity. There are different spaces used in the literature on which the moments are defined; the spaces used most often are $(H^{1+\epsilon}(\Omega))^d$, $(W^{1,s}(\Omega))^d$ and

$$X^p(\Omega) := \{\mathbf{u}\in(L^p(\Omega))^2, \operatorname{curl}\mathbf{u}\in L^p(\Omega), \mathbf{u}\cdot\mathbf{t}\in L^p(\partial\Omega)\} \qquad (d=2)$$

$$X^p(\Omega) := \{\mathbf{u}\in(L^p(\Omega))^3, \mathbf{curl}\,\mathbf{u}\in(L^p(\Omega))^3, \mathbf{u}\times\mathbf{n}\in(L^p(\partial\Omega))^3\} \quad (d=3).$$

(See for instance in Girault and Raviart [48], Arnold, Falk, and Winther [8], and Amrouche, Bernardi, Dauge, and Girault [5]). If we opt for modified degrees of freedom on the edges (and possibly on the faces in three dimensions), we either need to invoke a nontrivial matching or a non-local definition.

[2] In two dimensions, the curl diagram commutes if one uses the L_2 projection on $\operatorname{curl}\mathbb{ND}_k$; in three dimensions, the **curl** diagram commutes if one uses the Raviart-Thomas-Nédélec interpolant on $\mathbb{RT}_k \supset \mathbf{curl}\,\mathbb{ND}_k$. See lemma 7.1.

There is no interpolation operator known that satisfies the commuting diagram property and that is also defined on all of $H(\mathbf{curl})$, works on quadrilateral meshes, and in both the two-dimensional and the three-dimensional case. Very recently there has been some progress on an interpolation operator in two dimensions on triangular elements satisfying the commuting diagram property defined on a less regular space, $(H^{\epsilon}(\Omega))^2 \cap H(\mathrm{curl})$, see Demkowicz and Babuška [42], which is optimal in N (except for an arbitrarily small δ, on which the bound depends) with respect to the $H(\mathbf{curl})$ norm.

To avoid the added technical difficulties in this approach – since, to the best of our knowledge, all known convergence proofs use the commuting diagram property – we will use more regular spaces. This also makes sense considering the main subject of the thesis, since we are ultimately interested in spectral approximations of smooth parts of the fields, where we will have to assume higher regularity in the proofs a priori.

These interpolation operators can also be used to restrict functions that are locally of high degree to global low order spaces, as needed when defining coarse spaces in multi-level methods or domain decomposition algorithms. The different definitions of the degrees of freedom and the interpolants so constructed will lead to different operators with different properties. The implementation and analysis of such restriction operators is discussed in section 7.6. They are used in section 10.2, to implement a two-level method, and in section 11.2, to derive some required estimates for the condition number bounds.

7.2 Raviart-Thomas-Nédélec elements in $H(\mathsf{div})$

In this section, we will construct $H(\mathsf{div})$ conforming elements. Therefore we will have to enforce continuity of the normal components across the interface. (The remark made above in the last section about the precise conditions for $H(\mathbf{curl})$ conforming elements applies to the $H(\mathsf{div})$ case with the appropriate changes, but we will not discuss it for the $H(\mathsf{div})$ case.)

The local spaces for (generalized) Raviart-Thomas-Nédélec elements in two dimensions are

$$RT_k(K) := \mathbb{Q}_{k,k-1}(K) \times \mathbb{Q}_{k-1,k}(K)$$

$$RT_{m,n}(K) := \mathbb{Q}_{m,n-1}(K) \times \mathbb{Q}_{m-1,n}(K)$$

and in three dimensions

$$RT_k(K) := \mathbb{Q}_{k,k-1,k-1}(K) \times \mathbb{Q}_{k-1,k,k-1}(K) \times \mathbb{Q}_{k-1,k-1,k})(K)$$

$$RT_{l,m,n}(K) := \mathbb{Q}_{l,m-1,n-1}(K) \times \mathbb{Q}_{l-1,m,n-1}(K) \times \mathbb{Q}_{l-1,m-1,n}(K)$$

Applying div to the Raviart-Thomas-Nédélec spaces, we obtain:

$$\mathsf{div}\, RT_k(K) = \mathbb{Q}_{k-1,k-1}(K)$$

$$\mathsf{div}\, RT_{m,n}(K) = \mathbb{Q}_{m-1,n-1}(K)$$

and in three dimensions

$$\mathsf{div}\, RT_k(K) = \mathbb{Q}_{k-1,k-1,k-1}(K)$$

$$\mathsf{div}\, RT_{l,m,n}(K) = \mathbb{Q}_{l-1,m-1,n-1}(K)$$

There is no continuity between the div of the local spaces in the global space $\mathsf{div}\, \mathbb{RT}(\Omega)$.

As in section 7.1, in spectral element methods we will usually work with degrees of freedom that correspond to point values of polynomial interpolants on Cartesian products of GL or GLL meshes. The following assumes a local space $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$ and $\mathbb{Q}_{l_1,m_1,n_1}(K) \times \mathbb{Q}_{l_2,m_2,n_2}(K) \times \mathbb{Q}_{l_3,m_3,n_3}(K)$ for the two-dimensional and three-dimensional case, respectively.

For the GLL-only method we use exactly the same mesh as in the previous section in (7.1) and (7.4).

For the $H(\mathsf{div})$ variant of the GLL-GL method we use the meshes

$$(\mathrm{GLL}_{m_1+1} \otimes \mathrm{GL}_{n_1+1}) \times (\mathrm{GL}_{m_2+1} \otimes \mathrm{GLL}_{n_2+1}) \tag{7.12}$$

$$\begin{aligned}(\mathrm{GLL}_{l_1+1} \otimes \mathrm{GL}_{m_1+1} \otimes \mathrm{GL}_{n_1+1}) \times (\mathrm{GL}_{l_2+1} \otimes \mathrm{GLL}_{m_2+1} \otimes \mathrm{GL}_{n_2+1}) \\ \times (\mathrm{GL}_{l_3+1} \otimes \mathrm{GL}_{m_3+1} \otimes \mathrm{GLL}_{n_3+1})\end{aligned} \tag{7.13}$$

The standard nodal interpolation operator can be defined as soon as point values are defined. As in the $H(\mathrm{curl})$ case, we can extend its domain to include functions of lower regularity. We can easily make it respect boundary and interface values, but it does not have the commuting diagram property (for the **curl** part of the diagram), i.e., there is no nodal interpolation operator on the Nédélec spaces known, so that the **curl** of it is equal to the nodal interpolation operator on the Raviart-Thomas-Nédélec spaces of the **curl** of the interpolated function.

To obtain an interpolation operator that makes the diagram commute, we define alternative degrees of freedom as we did for Nédélec elements.

We define edge moments

$$\int_e \mathbf{u} \cdot \mathbf{n} p \qquad p \in \mathbb{Q}_{k-1}(e) \qquad \text{for all edges } e \text{ of } K.$$

and interior moments

$$\int_K \mathbf{u}\cdot\mathbf{p} \qquad \mathbf{p}\in\mathbb{Q}_{k-2,k-1}(K)\times\mathbb{Q}_{k-1,k-2}(K) \qquad \text{for } k>1.$$

These two sets of moments uniquely determine a function $\mathbf{u}\in RT_k(K)$.

In three dimensions face moments are used instead of edge moments:

$$\int_F \mathbf{u}\cdot\mathbf{n}p \qquad p\in\mathbb{Q}_{k-1,k-1}(F) \qquad \text{for all faces } F \text{ of } K.$$

and the interior moments are defined with the appropriate space:

$$\int_K \mathbf{u}\cdot\mathbf{p} \quad \mathbf{p}\in\mathbb{Q}_{k-2,k-1,k-1}(K)\times\mathbb{Q}_{k-1,k-2,k-1}(K)\times\mathbb{Q}_{k-1,k-1,k-2}(K) \quad \text{for } k>1.$$

The extension of these degrees of freedom to the case of different degrees in different direction, as in $RT_{m,n}$ and $RT_{l,m,n}$ is straightforward.

Associated to these degrees of freedom on an element is an interpolation operator which will be denoted by Π_k^{RT}, $\Pi_{m,n}^{RT}$ and $\Pi_{l,m,n}^{RT}$, and which is used element by element to define the global interpolation operator $\mathbf{\Pi}_k^{RT}$, $\mathbf{\Pi}_{m,n}^{RT}$ and $\mathbf{\Pi}_{l,m,n}^{RT}$. These interpolation operators are not defined for all of $H(\mathsf{div})$, since the edge moments (in two dimensions) or the face moments (in three dimensions) are not defined for general functions $\mathbf{u}\in H(\mathsf{div})$. They are certainly well-defined when the normal trace of $\mathbf{u}$ is sufficiently regular; $\mathbf{u}\in(H^r(\Omega))^d$ with $r>\frac{1}{2}$ is enough. We can rotate the interpolation operator of Demkowicz and Babuška [42] on triangles to obtain an interpolation operator on $H(\mathsf{div})$ in two dimensions that is defined on $(H^\epsilon(\Omega))^2\cap H(\mathsf{div})$ and bounded, and arbitrarily close to optimal in N. There is no interpolation operator known that is defined in all of $H(\mathsf{div})$ and satisfies the commuting diagram property with some interpolation operator in $H(\mathrm{curl})$.

There is a interpolation operator on $\mathsf{div}\,\mathbb{RT}_k$, which makes the div diagram commute with the Raviart-Thomas-Nédélec interpolant, and it turns out to be the L^2-projection (Suri [93, equation (2.28) and Theorem 2.2]).

7.3 Commuting diagram properties and discrete Helmholtz decomposition

In this section we assume that the domain Ω is a simply connected polygon or polyhedron, with a connected boundary. In the statement of the commuting diagram properties, and in the analysis of the spaces $\mathbb{ND}$ and $\mathbb{RT}$, we need the standard scalar piecewise polynomial spaces.

The H^1-conforming space with continuity across the interfaces is:

$$\mathbb{S}_N(\Omega) := \{q \in H^1(\Omega) | q|_K \in \mathbb{Q}_N(K) \quad \forall K\}$$

We can also define a space with enforced zero boundary values $\mathbb{S}_N^0(\Omega) \subset H_0^1(\Omega)$ and anisotropic versions $\mathbb{S}_{m,n}(\Omega)$ and $\mathbb{S}_{l,m,n}(\Omega)$ with the local spaces $\mathbb{Q}_{m,n}(K)$ and $\mathbb{Q}_{l,m,n}(K)$, respectively.

We denote the standard nodal interpolation operator onto $\mathbb{S}_N(\Omega)$ by $\mathbf{\Pi}_N^S$.

The L^2-conforming space, in which no continuity is required across the interfaces, is defined analogously:

$$\mathbb{W}_N(\Omega) := \{q \in L^2(\Omega) | q|_K \in \mathbb{Q}_N(K) \quad \forall K\}$$

Here the appropriate restricted space is $\mathbb{W}_N^0(\Omega)$ which is a subset of $L_0^2(\Omega)$, the subspace of functions in $L^2(\Omega)$ having zero mean. The versions of $\mathbb{W}_N(\Omega)$ and $\mathbb{W}_N^0(\Omega)$ with different degree in different directions will be denoted $\mathbb{W}_{m,n}(\Omega)$,$\mathbb{W}_{l,m,n}(\Omega)$, $\mathbb{W}_{m,n}^0(\Omega)$ and $\mathbb{W}_{l,m,n}^0(\Omega)$.

The interpolation operators $\mathbf{\Pi}_N^W$ and $\mathbf{\Pi}_N^{W,0}$ are the L^2-projections onto $\mathbb{W}_N(\Omega)$ and $\mathbb{W}_N^0(\Omega)$, respectively.

Lemma 7.1 (Commuting diagram properties) *Assume that q, $\mathbf{u}$ and $\mathbf{v}$ are sufficiently regular. Then the following identities hold*

$$\begin{aligned}
\mathbf{grad}\left(\mathbf{\Pi}_N^S q\right) &= \mathbf{\Pi}_N^{ND,I}\left(\mathbf{grad}\, q\right), \\
\mathrm{curl}\left(\mathbf{\Pi}_N^{ND,I}\mathbf{u}\right) &= \mathbf{\Pi}_N^W\left(\mathrm{curl}\,\mathbf{u}\right), \\
\mathbf{curl}\left(\mathbf{\Pi}_N^S q\right) &= \mathbf{\Pi}_N^{RT}\left(\mathbf{curl}\, q\right), \\
\mathbf{curl}\left(\mathbf{\Pi}_N^{ND,I}\mathbf{u}\right) &= \mathbf{\Pi}_N^{RT}\left(\mathbf{curl}\,\mathbf{u}\right), \\
\mathsf{div}\left(\mathbf{\Pi}_N^{RT}\mathbf{v}\right) &= \mathbf{\Pi}_N^W\left(\mathsf{div}\,\mathbf{v}\right).
\end{aligned}$$

Proof: See Hiptmair [57, Theorem 2.30]. ∎

This lemma also holds for the anisotropic case with the obvious changes.

Lemma 7.2 (Kernel of $\mathbf{curl}$) *If Ω is simply connected, with a connected boundary, the kernels of the curl operator defined in $\mathbb{ND}_N^I(\Omega)$ and $\mathbb{ND}_N^{I,0}(\Omega)$ are $\mathbf{grad}\,\mathbb{S}_N(\Omega)$ and $\mathbf{grad}\,\mathbb{S}_N^0(\Omega)$, respectively.*

We can now state the following discrete analogue of the Helmholtz decomposition (compare section 2.5 for the continuous case) for the Nédélec spaces into a **curl**-free part and a div-free part:

$$\mathbb{ND}_N^I(\Omega) = \mathbf{grad}\,\mathbb{S}_N(\Omega) \oplus \mathbb{ND}_N^{I,+}(\Omega)$$
$$\mathbb{ND}_N^{I,0}(\Omega) = \mathbf{grad}\,\mathbb{S}_N^0(\Omega) \oplus \mathbb{ND}_N^{I,0,+}(\Omega)$$

with the orthogonal complements

$$\mathbb{ND}_N^{I,+}(\Omega) := \{\mathbf{u} \in \mathbb{ND}_N^I(\Omega) | (\mathbf{u}, \mathbf{grad}\, p_N)_0 = 0 \quad \forall p_N \in \mathbb{S}_N(\Omega)\} \tag{7.14}$$

$$\mathbb{ND}_N^{I,0,+}(\Omega) := \{\mathbf{u} \in \mathbb{ND}_N^{I,0}(\Omega) | (\mathbf{u}, \mathbf{grad}\, p_N)_0 = 0 \quad \forall p_N \in \mathbb{S}_N^0(\Omega)\} \tag{7.15}$$

In general, the spaces $\mathbb{ND}_N^{I,+}(\Omega)$ and $\mathbb{ND}_N^{I,0,+}(\Omega)$ are not included in $H^\perp(\mathbf{curl};\Omega)$ and $H_0^\perp(\mathbf{curl};\Omega)$, the analogous spaces in the continuous Helmholtz decomposition.

The discrete Friedrichs' inequality proven in the last section of this chapter gives a L^2-bound for **curl** on $\mathbb{ND}_N^{I,+}(\Omega)$, the orthogonal complement of its kernel.

7.4 Approximation properties of Raviart-Thomas-Nédélec elements

We will write in this and the following section $\hat{\Pi}$ for interpolation operators on the reference element.

The approximation properties of the Raviart-Thomas-Nédélec elements in the two-dimensional case are treated for the hN-version in Suri [93].

A N-version estimate is proven for the reference element in [93, Lemma 3.1].

Lemma 7.3 *Assume $\mathbf{u} \in (H^r)^2$ for some $r > 1$. Then there exists a constant C independent of N and $\mathbf{u}$ such that*

$$||\mathbf{u} - \hat{\Pi}_N^{RT}\mathbf{u}||_0 \leq CN^{-(r-\frac{1}{2})}||\mathbf{u}||_r.$$

The transformation of $\mathbf{u}$ between the reference element and any other element is given in [93, equation (2.18)], and for this mapping, in two dimensions, we have a lemma [93, Lemma 3.2], which allows us to prove the hN-version of the above estimate [93, Theorem 3.1]:

Lemma 7.4 *Assume $\mathbf{u} \in (H^r)^2$ for some $r > 1$, and let h be the size of the elements. Then there exists a constant C independent of h, N, and $\mathbf{u}$ such that*

$$||\mathbf{u} - \mathbf{\Pi}_N^{RT}\mathbf{u}||_0 \leq Ch^{\min(N,r)}N^{-(r-\frac{1}{2})}||\mathbf{u}||_r.$$

In the original paper it is also proven that $(I - \mathbf{\Pi}^{RT})$ is bounded by $Ch^{\min(N,r)}N^{-(r-\frac{1}{2})}$ as a map from $H^r(\mathsf{div})$ to $H(\mathsf{div})$. (We note parenthetically that Suri's definition of S_k^1 corresponds to our definition for RT_{k+1} which explains the difference in exponents between his presentation and ours.)

The hN approximation properties of the Raviart-Thomas-Nédélec elements in three dimensions are derived in Monk [71].

On the reference element the N-version estimate is as follows [71, Theorem 3.5]:

Lemma 7.5 *Assume* $\mathbf{u} \in (H^r)^2$ *for some* $r > \frac{1}{2}$. *Then there exists a constant* C *independent of* N *and* $\mathbf{u}$ *such that*

$$||\mathbf{u} - \hat{\Pi}_N^{RT}\mathbf{u}||_0 \leq CN^{-(r-\frac{1}{2})}||\mathbf{u}||_r.$$

The mapping between the reference element and any given element in three dimensions is treated in [71, equation (69) and Lemma 3.6], which allows us to prove the hN-version of the estimate [71, Theorem 3.7]:

Lemma 7.6 *Assume* $\mathbf{u} \in (H^r)^2$ *for some* $r > \frac{1}{2}$, *and let* h *be the size of the elements. Then there exists a constant* C *independent of* h, N, *and* $\mathbf{u}$ *such that*

$$||\mathbf{u} - \mathbf{\Pi}_N^{RT}\mathbf{u}||_0 \leq Ch^{\min(N,r)}N^{-(r-\frac{1}{2})}||\mathbf{u}||_r.$$

The techniques used in the proofs of the previous two lemmata in [71] should allow the extension of the lemmata 7.3 and 7.4 to the case $r > \frac{1}{2}$.

All the proofs in this section work by expanding both the vector field and its interpolant in Legendre polynomials, comparing coefficients and bounding the interpolation error as the difference of these two expansions.

These results can most probably be improved for regular enough $\mathbf{u}$ by adapting Ben Belgacem's and Bernardi's strategy in [15] to the $H(\mathsf{div})$ case. More explicitly, in their work they identified the interpolation operator for the Nédélec space as a tensor product of one-dimensional L^2- and modified H^1-projections examining the expansions, and derived optimal estimates (which would correspond to estimates on $\mathbf{\Pi}_N^{RT}$ without the $\frac{1}{2}$ in the exponent) by tensorizing known estimates of the one-dimensional projections. The derivations and numerical experiments in section 7.6 seem to encourage such an approach as well.

We will discuss the behavior of $\mathbf{\Pi}_N^{RT}$ on Raviart-Thomas-Nédélec elements of higher index in section 7.6. Such estimates and bounds will be needed later for the analysis of the domain decomposition preconditioners.

7.5 Approximation properties of Nédélec elements

The hN-version of the edge element approximation, i.e., enforcing only tangential continuity and using the Nédélec definitions for the degrees of freedom (7.5) and (7.6); is treated in Monk [71] for Nédélec elements of the first kind. He proves a N-version estimate on the reference element [71, Theorem 3.1]:

Lemma 7.7 *Assume* $\mathbf{u} \in (H^r)^3$ *for some* $r > 1$*, and let the Nédélec interpolant* $\Pi_N^{ND,I}$ *be defined by the edge, face, and interior moments. Then, there exists a constant* C *independent of* $\mathbf{u}$ *and* N *such that*

$$||\mathbf{u} - \hat{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq CN^{-(r-1)}||\mathbf{u}||_r.$$

The proof consists of writing $\mathbf{u}$ in its expansion in Legendre polynomials, and identifying the Nédélec interpolant in terms of this expansion [71, pages 123–125]. The remainder terms of the expansion, corresponding to the interpolation error, are then bounded by some hard and tedious algebra, see [71, pages 125-130].

Using a scaling argument and a mapping from the reference element (the Piola transform, see [71, equation (19)]), with the appropriate bounds for that transform, allow us to prove the hN-version of the above lemma [71, Theorem 3.3]:

Lemma 7.8 *Assume* $\mathbf{u} \in (H^r)^3$ *for some* $r > 1$ *and let the Nédélec interpolant* $\Pi_N^{ND,I}$ *be defined elementwise. Let* h *be the size of the elements. Then*

$$||\mathbf{u} - \mathbf{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq Ch^{\min(N,r)}N^{-(r-1)}||\mathbf{u}||_r.$$

Additionaly, we have the following stability estimates for $\mathbf{u} \in (W^{1,s})^3$ *for some* $s > 2$ *or for* $\mathbf{u} \in (H^{1+\epsilon})^3$ *for some* $\epsilon > 0$*:*

$$||\mathbf{u} - \mathbf{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq C(hN^{-1}||\mathbf{u}||_{W^{1,s}} + h||\mathbf{u}||_1),$$

$$||\mathbf{u} - \mathbf{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq C(hN^{-1}||\mathbf{u}||_{1+\epsilon} + h||\mathbf{u}||_1).$$

It is necessary to estimate how well the $\mathbf{curl}$ of the Nédélec interpolant approximates the $\mathbf{curl}$ of the function. If the function $\mathbf{u}$ is sufficiently regular ($\mathbf{u} \in (H^r)^3$ for $r > \frac{3}{2}$ is enough) we can use the commuting diagram property for the Nédélec elements to reduce the approximation of the $\mathbf{curl}$ to the interpolation error of the corresponding interpolation operator in the N-version of the Raviart-Thomas-Nédélec spaces (which we presented in lemma 7.5 in the previous section) . This gives us as N-version estimate on a reference element (compare [71, Theorem 3.5]):

Lemma 7.9 *Assume* $\mathbf{u} \in H^r(\mathbf{curl})$ *for* $r > \frac{1}{2}$. *Then there is a constant* C *depending on* r *but not on* N *nor* $\mathbf{u}$ *such that*

$$||\,\mathbf{curl}\,\mathbf{u} - \mathbf{curl}\,\hat{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq CN^{-(r-\frac{1}{2})}||\,\mathbf{curl}\,\mathbf{u}||_r.$$

With an analogous argument as that that was leading to lemma 7.8, we obtain the hN-version [71, Theorem 3.4]:

Lemma 7.10 *Suppose* $\mathbf{u} \in H^r(\mathbf{curl})$ *for* $r > \frac{1}{2}$. *Then there is a constant* C *independent of* $\mathbf{u}$, h, *and* N *such that*

$$||\,\mathbf{curl}\,\mathbf{u} - \mathbf{curl}\,\mathbf{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq Ch^{\min(N,r)}N^{-(r-\frac{1}{2})}||\,\mathbf{curl}\,\mathbf{u}||_r.$$

Ben Belgacem and Bernardi [15] prove an optimal N-version estimate assuming more regularity of $\mathbf{u}$ in section 4 of their paper for the N-extension of Nédélec elements of the first kind. The technique of proof requires $\mathbf{u} \in (H^r)^3$ for some $r > 2$ and $\mathbf{curl}\,\mathbf{u} \in (H^s)^3$ for some $s > \frac{3}{2}$. The idea is similar to that of Monk's paper discussed above. One starts of with an expansion of $\mathbf{u}$ having vanishing Nédélec degrees of freedom on the boundary (i.e., face moments and edge moments) in Legendre polynomials L_N and in the polynomials $(1-x^2)L'_N$ spanning $P_N \cap H_0^1$ (on the unit cube which serves as reference element). The projection to a subset of the latter can be identified as an one-dimensional modified H^1-projection, if $\mathbf{u}$ is regular enough. The entire interpolation operator is identified as being a collection of tensor products of projections and its analysis is standard using the techniques and results of Bernardi and Maday [17, sections 6 and 7]. For non-zero boundary degrees of freedom one identifies the interpolation operator for the face moments and edge moments to be the appropriate modified H^1- or H_0^1-projection, again assuming enough regularity of the traces on the faces and edges. Adding the three parts of the interpolation operator together, one obtains again a representation by tensor products of projections which leads to the final estimate. It would be very useful to have a similar estimate, or a slightly degraded estimate for $\mathbf{u} \in (H^r)^3$ with $r = 2 - \epsilon$ since that would simplify several arguments, for instance the proof of a Friedrichs-like inequality later on.

Ben Belgacem's and Bernardi's results are given in [15, Theorem 4.9] for the cube as reference element:

Lemma 7.11 *For any real number* $r \geq 2$ *there exist a positive constant* C *such that for all functions* $\mathbf{u} \in (H^r)^3$ *the following estimate holds*

$$||\mathbf{u} - \hat{\Pi}_N^{ND,I}\mathbf{u}||_0 \leq CN^{-r}||\mathbf{u}||_r.$$

Lemma 7.12 *For any real number $s \geq \frac{3}{2}$ there exist a positive constant C such that for all functions $\mathbf{u} \in H^s(\mathbf{curl})$ the following estimates holds*

$$|| \mathbf{curl}\, \mathbf{u} - \mathbf{curl}\, \hat{\Pi}_N^{ND,I} \mathbf{u}||_0 \leq CN^{-s} || \mathbf{curl}\, \mathbf{u}||_s.$$

Using the same techniques as in Monk [71], we easily derive the hN-version estimates:

Lemma 7.13 *For any real number $r \geq 2$ there exist a positive constant C such that for all functions $\mathbf{u} \in (H^r)^3$ the following estimate holds*

$$||\mathbf{u} - \mathbf{\Pi}_N^{ND,I} \mathbf{u}||_0 \leq Ch^{\min(N,r)} N^{-r} ||\mathbf{u}||_r.$$

Lemma 7.14 *For any real number $s \geq \frac{3}{2}$ there exist a positive constant C such that for all functions $\mathbf{u} \in H^s(\mathbf{curl})$ the following estimates holds*

$$|| \mathbf{curl}\, \mathbf{u} - \mathbf{curl}\, \mathbf{\Pi}_N^{ND,I} \mathbf{u}||_0 \leq Ch^{\min(N,s)} N^{-s} || \mathbf{curl}\, \mathbf{u}||_s.$$

Ben Belgacem and Bernardi also present an estimate for the approximation of the tangential components on the boundary needed for the analysis of problems with Silver-Müller boundary conditions; see [15, Theorem 4.10].

In two dimension, Ben Belgacem's and Bernardi's estimate should extend in the same form with less regularity; only $r > \frac{3}{2}$ should be needed in lemmata 7.11 and 7.13 and $s > 1$ in lemmata 7.12 and 7.14.

In two dimensions We can also use that the $H(\mathrm{curl})$ case is a rotation of the $H(\mathsf{div})$ case by ninety degrees. We herefore have the following two lemmata corresponding to the lemmata 7.3 and 7.4. The first lemma is valid on a reference element, and the second one is valid for an arbitrary element in a quasi-uniform conforming mesh. (As noted above, the proof should extend to the case $r > \frac{1}{2}$.)

Lemma 7.15 *Assume $\mathbf{u} \in (H^r)^2$ for some $r > 1$. Then there exists a constant C independent of N and $\mathbf{u}$ such that*

$$||\mathbf{u} - \hat{\Pi}_N^{ND} \mathbf{u}||_0 \leq CN^{-(r-\frac{1}{2})} ||\mathbf{u}||_r.$$

Lemma 7.16 *Assume $\mathbf{u} \in (H^r)^2$ for some $r > 1$, and let h be the size of the elements. Then there exists a constant C independent of h, N, and $\mathbf{u}$ such that*

$$||\mathbf{u} - \mathbf{\Pi}_N^{ND} \mathbf{u}||_0 \leq Ch^{\min(N,r)} N^{-(r-\frac{1}{2})} ||\mathbf{u}||_r$$

Also, $(I - \mathbf{\Pi}^{ND})$ is bounded by $Ch^{\min(N,r)} N^{-(r-\frac{1}{2})}$ as a map from $H^r(\mathrm{curl})$ to $H(\mathrm{curl})$.

For the analysis of the domain decomposition methods in chapters 10 and 11, we will need to study the properties of the Nédélec interpolant between Nédélec spaces of different degrees. We will do so in the next section.

7.6 Nédélec type interpolants on vector field spectral elements

In this section we will first derive the explicit form of the Nédélec type interpolants on $\mathbb{RT}$ and $\mathbb{ND}$ from local spaces of the form

$$\mathbb{Q}_{p_1,q_1,r_1}(K) \times \mathbb{Q}_{p_2,q_2,r_2}(K) \times \mathbb{Q}_{p_3,q_3,r_3}(K)$$

to local spaces

$$\mathbb{Q}_{l_1,m_1,n_1}(K) \times \mathbb{Q}_{l_2,m_2,n_2}(K) \times \mathbb{Q}_{l_3,m_3,n_3}(K)$$

in the three-dimensional case, and from

$$\mathbb{Q}_{p_1,q_1}(K) \times \mathbb{Q}_{p_2,q_2}(K)$$

to

$$\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$$

in the two-dimensional case.

We will realize that all the interpolants in the two-dimensional and three-dimensional case can be written as tensor products of two types of terms, one corresponding to a L^2-projection, while the other is of a similar form, but includes boundary terms.

Second, we will numerically compute the norm of these interpolants using the L^2-norm on the spaces. We do that by reformulating the problem as a generalized eigenvalue problem. Since both matrices in these generalized eigenvalue problems are tensor products, we can reduce the generalized eigenvalue problems to the easier generalized eigenvalue problems on the factors of the tensor product. We numerically study the bounds on the second type of term; the first type has a trivial bound. We show that the Nédélec interpolants are uniformly bounded independently of N for a constant difference in degrees, such as from $\mathbb{ND}_{N+C}$ to $\mathbb{ND}_N$, but it has an approximate $\sqrt{N}$ bound for $\mathbb{ND}_{2N}$ to $\mathbb{ND}_N$. Besides serving as basic estimates in our analysis of the domain decomposition preconditioner in chapter 11, these experiments show that multiplication with some lower-order terms can be stable when using Nédélec type degrees of freedom and interpolants; but that nonlinear equations with terms like $(u_i)^c$ with $c > 1$ may suffer under worse approximation properties than linear ones.

An analytic derivation of these results seems to be possible. On one hand, one could follow the expansion arguments of Suri, or Monk, or Ben Belgacem and Bernardi, specializing them to the case with few specific non-zero coefficients, and find bounds using similar techniques as in their papers. On the other hand, one could analyze the form of the second type of term, by either some linear algebra (using that the term is a low-order pertubation of a known projection) or by identifying the one-dimensional continuous projection operator that has the term as discretization, and analyzing this projection.

Unfortunately, we lack both the time and space to attempt such a derivation within the scope of this thesis, but we will do so in future work.

7.6.1 Nédélec interpolants between Nédélec spaces

First we will discuss the two-dimensional case. We can restrict our derivation to the first component of the interpolant, the form of the second component follows by symmetry considerations. By the standard rotation argument, we can derive the form of the interpolant for the Raviart-Thomas-Nédélec spaces in two dimensions.

We will always derive the interpolation operator from the GLL-only spectral element degrees of freedom on $\mathbb{Q}_{p_1,q_1}(K)$ to the Nédélec type degrees of freedom on $\mathbb{Q}_{m_1,n_1}(K)$. We will also assume $p_i \geq m_i$, $q_i \geq n_i$. The case $p_i = m_i$ and $q_i = n_i$ gives us the mapping between the GLL degrees of freedom and the Nédélec degrees of freedom on $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$, and taking the inverse and applying it to the above result we obtain the Nédélec interpolation operator as an operator on the spectral element degrees of freedom. (The case $p_i \leq m_i$, $q_i \leq n_i$ can be treated by lifting $\mathbb{Q}_{p_1,q_1}(K) \times \mathbb{Q}_{p_2,q_2}(K)$ to $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$ by the standard polynomial interpolation $(I^{m_1}_{p_1} \otimes I^{n_1}_{q_1}) \times (I^{m_2}_{p_2} \otimes I^{n_2}_{q_2})$ and using the result for the $p_i = m_i$ and $q_i = n_i$ case.)

As discussed in section 7.1, the Nédélec degrees of freedom are

$$\int_e \mathbf{u} \cdot \mathbf{t}_e p \qquad p \in \mathbb{Q}_\cdot(e) \qquad \text{for all edges } e \text{ of } K.$$

$$\int_K \mathbf{u} \cdot \mathbf{p} \qquad \mathbf{p} \in \mathbb{Q}_{\cdot,\cdot}(K) \times \mathbb{Q}_{\cdot,\cdot}(K) \qquad \text{for degrees} > 1.$$

We can organize these degrees of freedom according to components, for u_1 there are

$$\int_K u_1 \cdot p_1^I \qquad p_1^I \in \mathbb{Q}_{m_1,n_1-2}$$

$$\int_{\{y=-1\}} u_1 p_1^A \qquad p_1^A \in \mathbb{Q}_{m_1}$$

$$\int_{\{y=1\}} u_1 p_1^B \qquad p_1^B \in \mathbb{Q}_{m_1}$$

and for u_2

$$\int_K u_2 \cdot p_2^I \qquad p_2^I \in \mathbb{Q}_{m_2-2,n_2}$$

$$\int_{\{x=-1\}} u_2 p_2^A \qquad p_2^A \in \mathbb{Q}_{n_2}$$

$$\int_{\{x=1\}} u_2 p_2^B \qquad p_2^B \in \mathbb{Q}_{n_2}$$

To derive tensor product forms mapping to Nédélec degrees of freedom, we have to arrange them in two two-dimensional arrays p_1 and p_2:

$$p_1(i,1) = p_1^A(i) \qquad p_1(i,j) = p_1^I(i,j-1) \qquad p_1(i,n_1) = p_1^B(i)$$

$$p_2(1,j) = p_2^A(j) \qquad p_2(i,j) = p_1^I(i-1,j) \qquad p_2(m_2,j) = p_1^B(j)$$

or with self-explanatory notation

$$p_1^A = (I_{m_1} \otimes e_1^{n_1}) p_1 \qquad p_1^B = (I_{m_1} \otimes e_{n_1}^{n_1}) p_1 \qquad p_1^I = (I_{m_1} \otimes R_{I,n_1}) p_1$$

$$p_2^A = (e_1^{m_1} \otimes I_{n_1}) p_2 \qquad p_2^B = (e_{m_1}^{m_1} \otimes I_{n_1}) p_2 \qquad p_2^I = (R_{I,m_1} \otimes I_{n_1}) p_2$$

Now the degrees of freedom for u_1 can be discretized (using the one-dimensional mass and interpolation matrices from chapter 4):

$$\begin{aligned} \int_K u_1 \cdot p_1^I &= p_1^T((I_{m_1} \otimes R_{I,n_1}^T)(I_{m_1}^{p_1,T} \otimes I_{n_1-2}^{q_1,T})(M_{p_1}^{p_1+1} \otimes M_{q_1}^{q_1+1}) u_1 \\ &= p_1^T((I_{m_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (R_{I,n_1}^T I_{n_1-2}^{q_1,T} M_{q_1}^{q_1+1})) u_1 \end{aligned}$$

$$\begin{aligned} \int_{\{y=-1\}} u_1 p_1^A &= p_1^T (I_{m_1} \otimes e_1^{n_1,T})(I_{m_1}^{p_1,T} \otimes 1)(M_{p_1}^{p_1+1} \otimes 1)(I_{p_1} \otimes e_1^{p_1}) u_1 \\ &= p_1^T((I_{m_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_1^{n_1,T} e_1^{p_1})) u_1 \end{aligned}$$

$$\begin{aligned} \int_{\{y=1\}} u_1 p_1^B &= p_1^T (I_{m_1} \otimes e_{n_1}^{n_1,T})(I_{m_1}^{p_1,T} \otimes 1)(M_{p_1}^{p_1+1} \otimes 1)(I_{p_1} \otimes e_{p_1}^{p_1}) u_1 \\ &= p_1^T((I_{m_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_{n_1}^{n_1,T} e_{p_1}^{p_1})) u_1 \end{aligned}$$

Adding up these expressions, we obtain that the first component of the Nédélec interpolant from the spectral element degrees of freedom to the Nédélec degrees of freedom is:

$$((I_{m_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_1^{n_1,T} e_1^{p_1} + R_{I,n_1}^T I_{n_1-2}^{q_1,T} M_{q_1}^{q_1+1} + e_{n_1}^{n_1,T} e_{p_1}^{p_1}))$$

We will introduce the following notation for the two types of terms, since they will appear in all our interpolants:

$$L_{p_1}^{m_1} := I_{m_1}^{p_1,T} M_{p_1}^{p_1+1} \qquad H_{q_1}^{n_1} := e_1^{n_1,T} e_1^{p_1} + R_{I,n_1}^T I_{n_1-2}^{q_1,T} M_{q_1}^{q_1+1} + e_{n_1}^{n_1,T} e_{p_1}^{p_1}$$

To obtain the version of the Nédélec interpolant that maps between spectral element degrees of freedom, we multiply this by the inverse of the same mapping for the case $m_1 = p_1$ and $n_1 = q_1$:

$$\mathcal{L}_{p_1}^{m_1} := (L_{m_1}^{m_1})^{-1} L_{p_1}^{m_1} \qquad \mathcal{H}_{q_1}^{n_1} := (H_{n_1}^{n_1})^{-1} H_{q_1}^{n_1}$$

and finally obtain that $\Pi_{m_1,n_1;m_2,n_2}^{ND}\mathbf{u}$ on $\mathbb{Q}_{p_1,q_1}(K) \times \mathbb{Q}_{p_2,q_2}(K)$ has the form

$$\Pi_{m_1,n_1;m_2,n_2}^{ND}\mathbf{u} = ((\mathcal{L}_{p_1}^{m_1} \otimes \mathcal{H}_{q_1}^{n_1})u_1, (\mathcal{H}_{p_2}^{m_2} \otimes \mathcal{L}_{q_2}^{n_2})u_2) \tag{7.16}$$

Rotating this expression by ninety degrees, we obtain that $\Pi_{m_1,n_1;m_2,n_2}^{RT}\mathbf{u}$ on $\mathbb{Q}_{p_1,q_1}(K) \times \mathbb{Q}_{p_2,q_2}(K)$ has the form:

$$\Pi_{m_1,n_1;m_2,n_2}^{RT}\mathbf{u} = ((\mathcal{H}_{p_1}^{m_1} \otimes \mathcal{L}_{q_1}^{n_1})u_1, (\mathcal{L}_{p_2}^{m_2} \otimes \mathcal{H}_{q_2}^{n_2})u_2) \tag{7.17}$$

Now we will perform the analogous derivations in three dimensions. We have the following degrees of freedom for the three-dimensional case:

$$\int_e \mathbf{u} \cdot \mathbf{t}_e p \qquad p \in \mathbb{Q}_\cdot(e) \qquad \text{for all edges } e \text{ of } K.$$

$$\int_F (\mathbf{u} \times \mathbf{n}) \cdot \mathbf{p} \qquad \mathbf{p} \in \mathbb{Q}_{\cdot,\cdot}(F) \times \mathbb{Q}_{\cdot,\cdot}(F) \qquad \text{for all faces } F \text{ of } K.$$

$$\int_K \mathbf{u} \cdot \mathbf{p} \quad \mathbf{p} \in \mathbb{Q}_{\cdot,\cdot,\cdot}(K) \times \mathbb{Q}_{\cdot,\cdot,\cdot}(K) \times \mathbb{Q}_{\cdot,\cdot,\cdot}(K) \quad \text{for degrees} > 1.$$

The degrees of freedom connected to the first component u_1 are of the following nine types:

$$\int_{\{y=-1,z=-1\}} u_1 p_1^A \qquad p_1^A \in \mathbb{Q}_{l_1} \qquad\qquad \int_{\{y=-1,z=1\}} u_1 p_1^B \qquad p_1^B \in \mathbb{Q}_{l_1}$$

$$\int_{\{y=1,z=-1\}} u_1 p_1^C \qquad p_1^C \in \mathbb{Q}_{l_1} \qquad\qquad \int_{\{y=1,z=1\}} u_1 p_1^D \qquad p_1^D \in \mathbb{Q}_{l_1}$$

$$\int_{\{y=-1\}} u_1 p_1^E \qquad p_1^E \in \mathbb{Q}_{l_1,n_1-2} \qquad\qquad \int_{\{y=1\}} u_1 p_1^F \qquad p_1^F \in \mathbb{Q}_{l_1,n_1-2}$$

$$\int_{\{z=-1\}} u_1 p_1^G \qquad p_1^G \in \mathbb{Q}_{l_1,m_1-2} \qquad\qquad \int_{\{z=1\}} u_1 p_1^H \qquad p_1^H \in \mathbb{Q}_{l_1,m_1-2}$$

$$\int_K u_1 p_1^I \qquad p_1^I \in \mathbb{Q}_{l_1,m_1-2,n_1-2}$$

We arrange these degrees of freedom in a three-dimensional array p_1 as follows:

$$p_1(i,1,1) = p_1^A(i) \qquad p_1(i,1,n_1) = p_1^B(i) \qquad p_1(i,m_1,1) = p_1^C(i)$$
$$p_1(i,m_1,n_1) = p_1^D(i) \qquad p_1(i,1,j) = p_1^E(i,j-1) \qquad p_1(i,m_1,j) = p_1^F(i,j-1)$$
$$p_1(i,j,1) = p_1^G(i,j-1) \qquad p_1(i,j,n_1) = p_1^H(i,j-1)$$
$$p_1(i,j,k) = p_1^I(i,j-1,k-1)$$

and they can be computed by the following expressions from the array p_1:

$$p_1^A = (I_{l_1} \otimes e_1^{m_1} \otimes e_1^{n_1})p_1 \qquad p_1^B = (I_{l_1} \otimes e_1^{m_1} \otimes e_{n_1}^{n_1})p_1$$
$$p_1^C = (I_{l_1} \otimes e_{m_1}^{m_1} \otimes e_1^{n_1})p_1 \qquad p_1^D = (I_{l_1} \otimes e_{m_1}^{m_1} \otimes e_{n_1}^{n_1})p_1$$
$$p_1^E = (I_{l_1} \otimes e_1^{m_1} \otimes R_{I,n_1})p_1 \qquad p_1^E = (I_{l_1} \otimes e_{m_1}^{m_1} \otimes R_{I,n_1})p_1$$
$$p_1^G = (I_{l_1} \otimes R_{I,m_1} \otimes e_1^{n_1})p_1 \qquad p_1^H = (I_{l_1} \otimes R_{I,m_1} \otimes e_{n_1}^{n_1})p_1$$
$$p_1^I = (I_{l_1} \otimes R_{I,m_1} \otimes R_{I,n_1})p_1$$

Computing these degrees of freedom exactly on our polynomial space by Gaussian quadrature in $\mathbb{Q}_{p_1,q_1,r_1}$, we obtain the following:

$$\begin{aligned}
\int_{\{y=-1,z=-1\}} u_1 p_1^A &= p_1^T(I_{l_1} \otimes e_1^{m_1,T} \otimes e_1^{n_1,T})(I_{l_1}^{p_1,T} \otimes 1 \otimes 1) \\
&\qquad (M_{p_1}^{p_1+1} \otimes 1 \otimes 1)(I_{p_1} \otimes e_1^{q_1} \otimes e_1^{r_1})u_1 \\
&= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_1^{m_1,T} e_1^{q_1}) \otimes (e_1^{n_1,T} e_1^{r_1}))u_1 \\
\int_{\{y=-1,z=1\}} u_1 p_1^B &= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_1^{m_1,T} e_1^{q_1}) \otimes (e_{n_1}^{n_1,T} e_{r_1}^{r_1}))u_1 \\
\int_{\{y=1,z=-1\}} u_1 p_1^C &= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_{m_1}^{m_1,T} e_{q_1}^{q_1}) \otimes (e_1^{n_1,T} e_1^{r_1}))u_1 \\
\int_{\{y=1,z=1\}} u_1 p_1^D &= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_{m_1}^{m_1,T} e_{q_1}^{q_1}) \otimes (e_{n_1}^{n_1,T} e_{r_1}^{r_1}))u_1
\end{aligned}$$

$$\begin{aligned}
\int_{\{y=-1\}} u_1 p_1^E &= p_1^T(I_{l_1} \otimes e_1^{m_1,T} \otimes R_{I,n_1}^T)(I_{l_1}^{p_1,T} \otimes 1 \otimes I_{n_1-2}^{r_1,T}) \\
&\qquad (M_{p_1}^{p_1+1} \otimes 1 \otimes M_{r_1}^{r_1+1})(I_{p_1} \otimes e_1^{q_1} \otimes I_{r_1})u_1 \\
&= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_1^{m_1,T} e_1^{q_1}) \otimes (R_{I,n_1}^T I_{n_1-2}^{r_1,T} M_{r_1}^{r_1+1}))u_1 \\
\int_{\{y=1\}} u_1 p_1^F &= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_{m_1}^{m_1,T} e_{q_1}^{q_1}) \otimes (R_{I,n_1}^T I_{n_1-2}^{r_1,T} M_{r_1}^{r_1+1}))u_1 \\
\int_{\{z=-1\}} u_1 p_1^G &= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (R_{I,m_1}^T I_{m_1-2}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (e_1^{n_1,T} e_1^{r_1}))u_1 \\
\int_{\{z=1\}} u_1 p_1^H &= p_1^T((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (R_{I,m_1}^T I_{m_1-2}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (e_{n_1}^{n_1,T} e_{r_1}^{r_1}))u_1
\end{aligned}$$

$$\begin{aligned}\int_K u_1 p_1^I &= p_1^T (I_{l_1} \otimes R_{I,m_1}^T \otimes R_{I,n_1}^T)(I_{l_1}^{p_1,T} \otimes I_{m_1-2}^{q_1,T} \otimes I_{n_1-2}^{r_1,T}) \\ &\qquad (M_{p_1}^{p_1+1} \otimes M_{q_1}^{q_1+1} \otimes M_{r_1}^{r_1+1}) u_1 \\ &= p_1^T ((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (R_{I,m_1}^T I_{m_1-2}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (R_{I,n_1}^T I_{n_1-2}^{r_1,T} M_{r_1}^{r_1+1})) u_1\end{aligned}$$

We recognize that their sum is of the tensor product form:

$$\begin{aligned}&((I_{l_1}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (e_1^{m_1,T} e_1^{q_1} + R_{I,m_1}^T I_{m_1-2}^{q_1,T} M_{q_1}^{q_1+1} + e_{m_1}^{m_1,T} e_{q_1}^{q_1}) \\ &\qquad \otimes (e_1^{n_1,T} e_1^{r_1} + R_{I,n_1}^T I_{n_1-2}^{r_1,T} M_{r_1}^{r_1+1} + e_{n_1}^{n_1,T} e_{r_1}^{r_1})) = (L_{p_1}^{l_1} \otimes H_{q_1}^{m_1} \otimes H_{r_1}^{n_1})\end{aligned}$$

Multiplying by the inverse of the equal degree version, we obtain the mapping on spectral element degrees of freedom:

$$(((L_{l_1}^{l_1})^{-1} L_{p_1}^{l_1}) \otimes ((H_{m_1}^{m_1})^{-1} H_{q_1}^{m_1}) \otimes ((H_{n_1}^{n_1})^{-1} H_{r_1}^{n_1})) = (\mathcal{L}_{p_1}^{l_1} \otimes \mathcal{H}_{q_1}^{m_1} \otimes \mathcal{H}_{r_1}^{n_1})$$

After similar computations on the other components, we obtain $\Pi^{ND}_{l_1,m_1,n_1;l_2,m_2,n_2;l_3,m_3,n_3}\mathbf{u}$ on $\mathbb{Q}_{p_1,q_1,r_1}(K) \times \mathbb{Q}_{p_2,q_2,r_2}(K) \times \mathbb{Q}_{p_3,q_3,r_3}(K)$ as:

$$\begin{aligned}&\Pi^{ND}_{l_1,m_1,n_1;l_2,m_2,n_2;l_3,m_3,n_3}\mathbf{u} = \\ &((\mathcal{L}_{p_1}^{l_1} \otimes \mathcal{H}_{q_1}^{m_1} \otimes \mathcal{H}_{r_1}^{n_1}) u_1, (\mathcal{H}_{p_2}^{l_2} \otimes \mathcal{L}_{q_2}^{m_2} \otimes \mathcal{H}_{r_2}^{n_2}) u_2, (\mathcal{H}_{p_3}^{l_3} \otimes \mathcal{H}_{q_3}^{m_3} \otimes \mathcal{L}_{r_3}^{n_3}) u_3) \qquad (7.18)\end{aligned}$$

7.6.2 Nédélec interpolants between Raviart-Thomas-Nédélec spaces

We already obtained the form in the two-dimensional case in the last section by rotation, so we only are left with the three-dimensional case.

As discussed in section 7.2, Raviart-Thomas-Nédélec spaces have the following Nédélec type degrees of freedom in three dimensions:

$$\int_F \mathbf{u} \cdot \mathbf{n} p \qquad p \in \mathbb{Q}_{\cdot,\cdot}(F) \qquad \text{for all faces } F \text{ of } K.$$

$$\int_K \mathbf{u} \cdot \mathbf{p} \quad \mathbf{p} \in \mathbb{Q}_{\cdot,\cdot,\cdot}(K) \times \mathbb{Q}_{\cdot,\cdot,\cdot}(K) \times \mathbb{Q}_{\cdot,\cdot,\cdot}(K) \quad \text{for degrees} > 1.$$

The degrees of freedom associated with the first component are:

$$\int_{\{x=-1\}} u_1 p_1^A \qquad p_1^A \in \mathbb{Q}_{m_1,n_1} \qquad\qquad \int_{\{x=1\}} u_1 p_1^B \qquad p_1^B \in \mathbb{Q}_{m_1,n_1}$$

$$\int_K u_1 p_1^I \qquad p_1^I \in \mathbb{Q}_{l_1-2,m_1,n_1}$$

We have to arrange these three types of degrees in a three-dimensional array so that we can obtain tensor product forms of the mapping. We choose the following layout:

$$p_1(1,i,j) = p_1^A(i,j) \qquad p_1(l_1,i,j) = p_1^B(i,j) \qquad p_1(i,j,k) = p_1^I(i-1,j,k)$$

The expressions for p_1^A, p_1^B and p_1^I in terms of p_1 are:

$$p_1^A = (e_1^{l_1} \otimes I_{m_1} \otimes I_{n_1})p_1 \qquad p_1^B = (e_{l_1}^{l_1} \otimes I_{m_1} \otimes I_{n_1})p_1$$
$$p_1^I = (R_{I,l_1} \otimes I_{m_1} \otimes I_{n_1})p_1$$

We can compute the degrees of freedom exactly by Gaussian quadrature, since we are in a polynomial space:

$$\begin{aligned}
\int_{\{x=-1\}} u_1 p_1^A &= p_1^T (e_1^{l_1,T} \otimes I_{m_1} \otimes I_{n_1})(1 \otimes I_{m_1}^{q_1,T} \otimes I_{n_1}^{r_1,T}) \\
&\qquad (1 \otimes M_{q_1}^{q_1+1} \otimes M_{r_1}^{r_1+1})(e_1^{p_1} \otimes I_{q_1} \otimes I_{r_1})u_1 \\
&= p_1^T ((e_1^{l_1,T} e_1^{p_1}) \otimes (I_{m_1}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (I_{n_1}^{r_1,T} M_{r_1}^{r_1+1}))u_1 \\
\int_{\{x=1\}} u_1 p_1^B &= p_1^T ((e_{l_1}^{l_1,T} e_{p_1}^{p_1}) \otimes (I_{m_1}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (I_{n_1}^{r_1,T} M_{r_1}^{r_1+1}))u_1 \\
\int_K u_1 p_1^I &= p_1^T (R_{I,l_1}^T \otimes I_{m_1} \otimes I_{n_1})(I_{l_1-2}^{p_1,T} \otimes I_{m_1}^{q_1,T} \otimes I_{n_1}^{r_1,T}) \\
&\qquad (M_{p_1}^{p_1+1} \otimes M_{q_1}^{q_1+1} \otimes M_{r_1}^{r_1+1})u_1 \\
&= p_1^T ((R_{I,l_1}^T I_{l_1-2}^{p_1,T} M_{p_1}^{p_1+1}) \otimes (I_{m_1}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (I_{n_1}^{r_1,T} M_{r_1}^{r_1+1}))u_1
\end{aligned}$$

We recognize that their sum (which is also the mapping from spectral element degrees of freedom to Nédélec type degrees of freedom from $\mathbb{Q}_{p_1,q_1,r_1}$ to $\mathbb{Q}_{l_1,m_1,n_1}$) is of the tensor product form:

$$\begin{aligned}
((e_1^{l_1,T} e_1^{p_1} + R_{I,l_1}^T I_{l_1-2}^{p_1,T} M_{p_1}^{p_1+1} + e_{l_1}^{l_1,T} e_{p_1}^{p_1}) \otimes (I_{m_1}^{q_1,T} M_{q_1}^{q_1+1}) \otimes (I_{n_1}^{r_1,T} M_{r_1}^{r_1+1})) \\
= (H_{p_1}^{l_1} \otimes L_{q_1}^{m_1} \otimes L_{r_1}^{n_1})
\end{aligned}$$

Multiplying this result with the inverse of the case $p_1 = l_1$, $q_1 = m_1$, $r_1 = n_1$, we obtain the mapping between spectral element degrees of freedom:

$$(((H_{l_1}^{l_1})^{-1} H_{p_1}^{l_1}) \otimes ((L_{m_1}^{m_1})^{-1} L_{q_1}^{m_1}) \otimes ((L_{n_1}^{n_1})^{-1} L_{r_1}^{n_1})) = (\mathcal{H}_{p_1}^{l_1} \otimes \mathcal{L}_{q_1}^{m_1} \otimes \mathcal{L}_{r_1}^{n_1})$$

Similar derivations for the other components show that the complete form of the interpolation operator $\Pi_{l_1,m_1,n_1;l_2,m_2,n_2;l_3,m_3,n_3}^{RT}\mathbf{u}$ on $\mathbb{Q}_{p_1,q_1,r_1} \times \mathbb{Q}_{p_2,q_2,r_2} \times \mathbb{Q}_{p_3,q_3,r_3}(K)$ is:

$$\begin{aligned}
&\Pi_{l_1,m_1,n_1;l_2,m_2,n_2;l_3,m_3,n_3}^{RT}\mathbf{u} = \\
&\quad ((\mathcal{H}_{p_1}^{l_1} \otimes \mathcal{L}_{q_1}^{m_1} \otimes \mathcal{L}_{r_1}^{n_1})u_1, (\mathcal{L}_{p_2}^{l_2} \otimes \mathcal{H}_{q_2}^{m_2} \otimes \mathcal{L}_{r_2}^{n_2})u_2, (\mathcal{L}_{p_3}^{l_3} \otimes \mathcal{L}_{q_3}^{m_3} \otimes \mathcal{H}_{r_3}^{n_3})u_3)
\end{aligned} \tag{7.19}$$

7.6.3 L^2-bounds on the norm of the interpolant

We will explain the idea in a two-dimensional model case.

Assume $\Pi\mathbf{u} = ((P_1^x \otimes P_1^y)u_1, (P_2^x \otimes P_2^y)u_2)$ is an interpolation operator from $\mathbb{Q}_{p_1,q_1} \times \mathbb{Q}_{p_2,q_2}$ to $\mathbb{Q}_{m_1,n_1} \times \mathbb{Q}_{m_2,n_2}$, and we want to derive a L^2-bound:

$$||\Pi\mathbf{u}||_0 \leq C||\mathbf{u}||_0 \tag{7.20}$$

Such a bound follows from the L^2-bounds on the components,

$$||(P_1^x \otimes P_1^y)u_1||_0 \leq C_1||u_1||_0$$

$$||(P_2^x \otimes P_2^y)u_2||_0 \leq C_2||u_2||_0$$

imply $C \leq \sqrt{C_1^2 + C_2^2}$ in (7.20).

We can reformulate the problems on the components as generalized eigenvalue problems by considering the squares of the estimates:

$$\begin{aligned}||(P_1^x \otimes P_1^y)u_1||_0^2 &= u_1^T(P_1^{x,T} \otimes P_1^{y,T})(M_{m_1}^{m_1+1} \otimes M_{n_1}^{n_1+1})(P_1^x \otimes P_1^y)u_1 \\ &\leq C_1^2||u_1||_0^2 = C_1^2 u_1^T(M_{p_1}^{p_1+1} \otimes M_{q_1}^{q_1+1})u_1\end{aligned}$$

We see that the square of the component L^2-bound is the largest eigenvalue of the generalized eigenvalue problem:

$$((P_1^{x,T}M_{m_1}^{m_1+1}P_1^x) \otimes (P_1^{y,T}M_{n_1}^{n_1+1}P_1^y))x = \lambda((M_{p_1}^{p_1+1}) \otimes (M_{q_1}^{q_1+1}))x$$

Since the matrices on both sides are tensor product matrices, we can reduce the generalized eigenvalue problem to the generalized eigenvalue problems on the factors of the tensor product:

$$(P_1^{x,T}M_{m_1}^{m_1+1}P_1^x)x_1 = \lambda_1(M_{p_1}^{p_1+1})x_1$$

$$(P_1^{y,T}M_{n_1}^{n_1+1}P_1^y)x_2 = \lambda_2(M_{q_1}^{q_1+1})x_2$$

The upper bound for the tensor product problem is given by the product of the maximal eigenvalues of the two factor problems.

Since all the factors of the tensor product interpolation operators are of one of the two types $\mathcal{L}$ and $\mathcal{H}$, it is enough to consider the following two generalized eigenvalue problems:

$$(\mathcal{L}_p^m)^T M_m^{m+1}\mathcal{L}_p^m x = \lambda M_p^{p+1}x \tag{7.21}$$

$$(\mathcal{H}_p^m)^T M_m^{m+1}\mathcal{H}_p^m x = \lambda M_p^{p+1}x \tag{7.22}$$

$\mathcal{L}$ is the discrete form of the L^2-projection. As such, it has the trivial upper bound of 1. The maximal eigenvalue of (7.21) is also 1. We will present numerical verifications of this fact in the next subsection.

We will present some numerical results for $\mathcal{H}_p^m$ for some different $m = N$ and $p = p(N)$ in the next subsection.

7.6.4 Numerical results

In several experiments, we observed that for the case $p = N + c$, $m = N$ the maximal eigenvalue of (7.22) is bounded independently of N. We also see in all experiments that (7.21) has the maximal eigenvalue 1, up to some numerical inaccuracies. These are in the order of 10^{-11} even for a degree of 200.

In figure 7.2 we show the case $p = N + 1$. (In the analysis of the domain decomposition preconditioners, this case correspond to the multiplication of an elementwise linear partition of unity, i.e., overlaps are only made of complete elements.) The value of λ_{max} of (7.22) for $N = 200$ is 2.0101.

In figure 7.3 we show the case $p = N + 10$. The maximal eigenvalue of (7.22) is still bounded independently of N, the value of λ_{max} at $N = 200$ is 6.1511.

As a last examples for the case $p = N + c$, we show in figure 7.4 the case $c = 100$. The maximal eigenvalue of the problem (7.22) is bounded independently of N and decreasing for increasing N, as in the two cases above. The value of λ_{max} at $N = 200$ is 63.8463.

It is easy to see that a bound independent of N for $p = N + 1$ implies that the L^2-bound of the interpolation operator is independent of N for $p = N + c$ for any c. The reasoning is the following: we can write the interpolation operator for $N + c \to N$ as a product of the operators $N + c \to N + c - 1$, $N + c - 1 \to N + c - 2$, ..., $N + 1 \to N$, in total c factors. Each of the norms of the factors is bounded by a constant C that can be derived from the bound on the maximal eigenvalue for (7.22) for $p = N + 1$, and therefore the entire operator should be bounded by C^c. We see in the figures and in the reported bounds at $N = 200$ that this estimate is too pessimistic, the bounds are growing rather slowly with c.

For the discussion of the approximation for nonlinear equations, and also for one of the ways to treat overlap of less than an entire element, we need to study the eigenvalue problem for $p = cN$. The numerical results show that the maximal eigenvalue of (7.22) grows linearly (or slower) with N, which would correspond to a bound on the interpolation operator that grows with $\sqrt{N}$.

In figure 7.5 we show the results for the case $p = 2N$. This case is important in the discussion of the approximation of quadratic nonlinear terms, and in the discussion of partitions

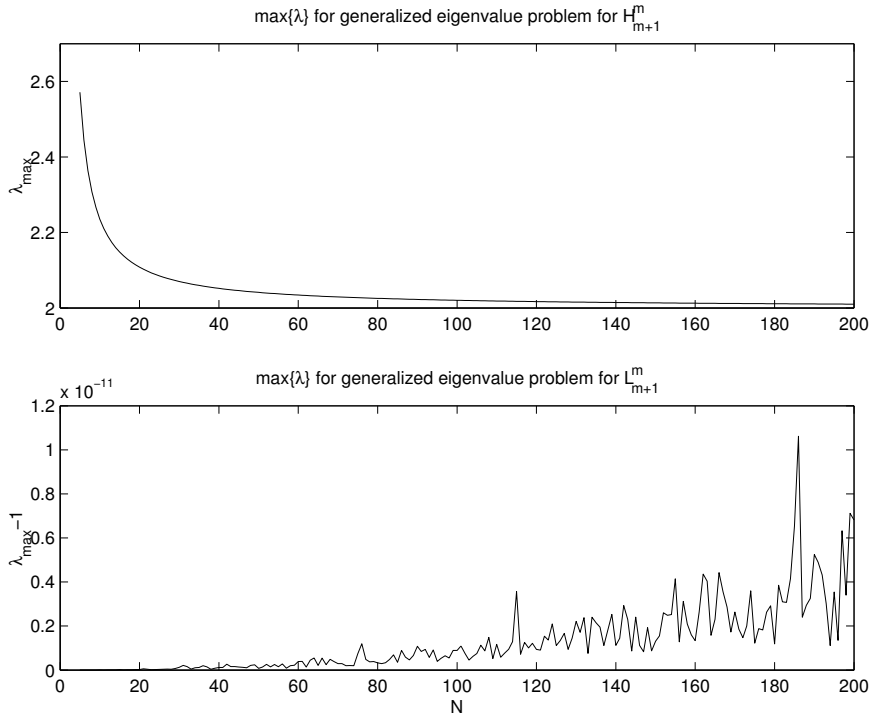


Figure 7.2: Maximal eigenvalues for the two generalized eigenvalue problems with $p = N + 1$, $m = N$. Top panel: problem (7.22). Bottom panel: problem (7.21). Note that in the latter, $\lambda_{max} - 1$ is shown.

of unity that are of the same degree as the spectral element functions. We show λ_{max}/N for (7.22) in the upper part, and it looks that asymptotically the growth is linear or slightly sublinear, the coefficient of N estimated from the values between $N = 180$ and $N = 200$ is 0.76.

In figures 7.6 and 7.7 we show the cases $p = \lfloor 1.5N \rfloor$ and $p = \lceil 1.1N \rceil$. In both of the cases we observe approximately linear growth of λ_{max}, estimated from the values between $N = 180$ and $N = 200$ we obtain a constant in front of the N of 0.32 and 0.06, approximately. We also performed experiments for other c in $p = cN$, which we do not show here, and we found in all of them approximately linear growth.

We also tested some other cases with $p = N + f(N)$ for $f(N)$ growing slower than N. We saw growth in N for $f(N) = \sqrt{N}$. The results for $f(N) = 30\log(N)$ are shown in the next figure 7.8 and show that for this case there seems to be a bound independent of N. (The value of λ_{max} for $N = 200$ is 113.6448.) One of the questions arising from these experiments is if there is a c such that for $f(N) = N^c$ we have a bound independent of N,

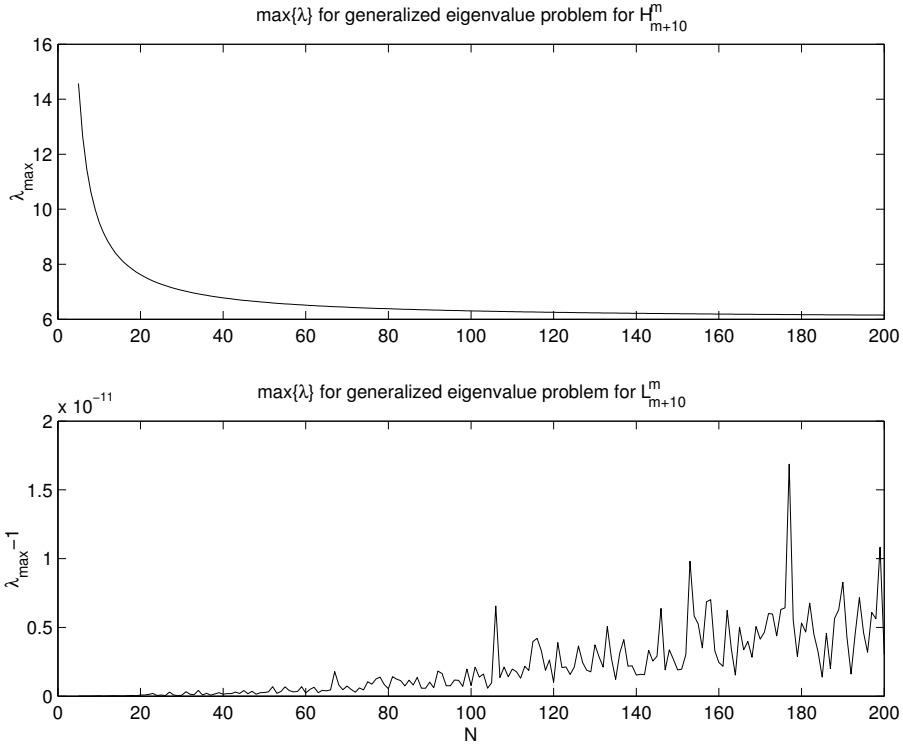


Figure 7.3: Maximal eigenvalues for the two generalized eigenvalue problems with $p = N + 10$, $m = N$. Top panel: problem (7.22). Bottom panel: problem (7.21). Note that in the latter, $\lambda_{max} - 1$ is shown.

or if the maximal eigenvalue will grow for any power in $f(N)$.

We formulate the result of these numerical experiments (assuming that we can generalize them and observe the same results for all c in cN) and the consequences obtained by tensorization arguments in the following observation (or numerically supported conjecture):

Observation 7.1: *The maximal eigenvalue of the generalized eigenvalue problem (7.22) is bounded independently of N for $m = N$, $p = N + c$ for all c, and allows a bound linear in N for $m = N$, $p = cN$. The interpolation operator for the Nédélec and Raviart-Thomas-Nédélec spaces from degree $N + c$ to N is bounded independently of N, and is bounded by $C\sqrt{N}$ with C independent of N from degree cN to N.*

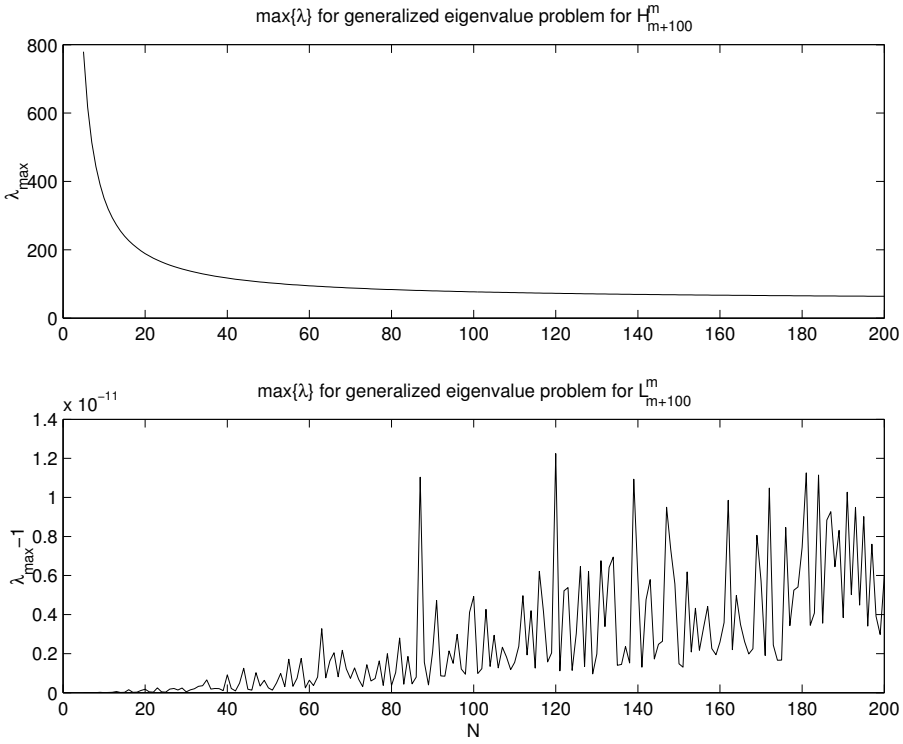


Figure 7.4: Maximal eigenvalues for the two generalized eigenvalue problems with $p = N + 100$, $m = N$. Top panel: problem (7.22). Bottom panel: problem (7.21). Note that in the latter, $\lambda_{max} - 1$ is shown.

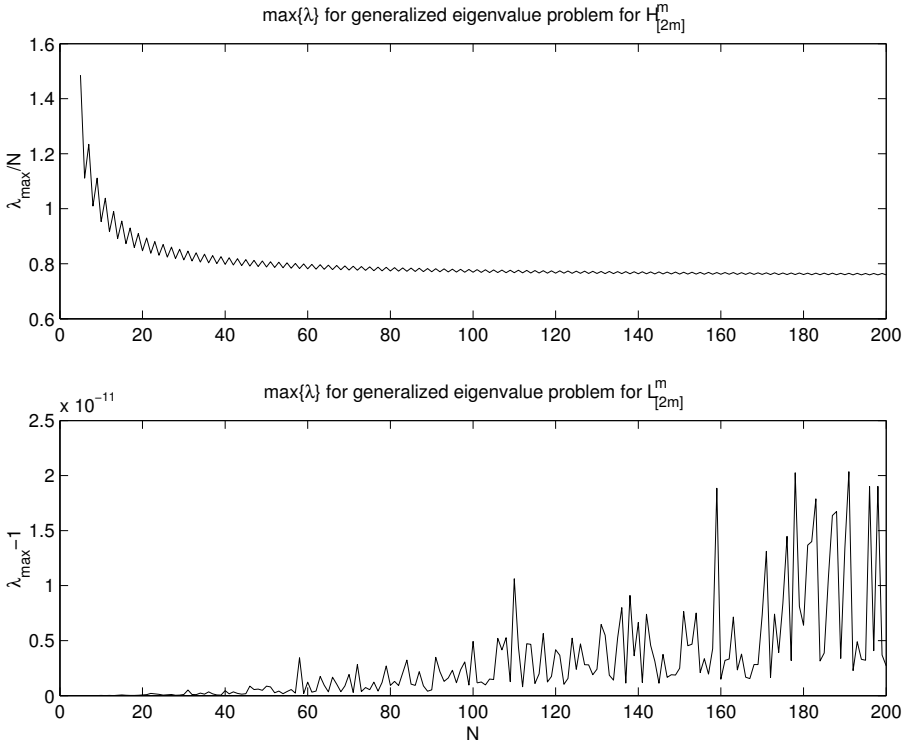


Figure 7.5: Maximal eigenvalues for the two generalized eigenvalue problems with $p = 2N$, $m = N$. Top panel: problem (7.22), plot of λ_{max}/N. Bottom panel: problem (7.21), plot of $\lambda_{max} - 1$.

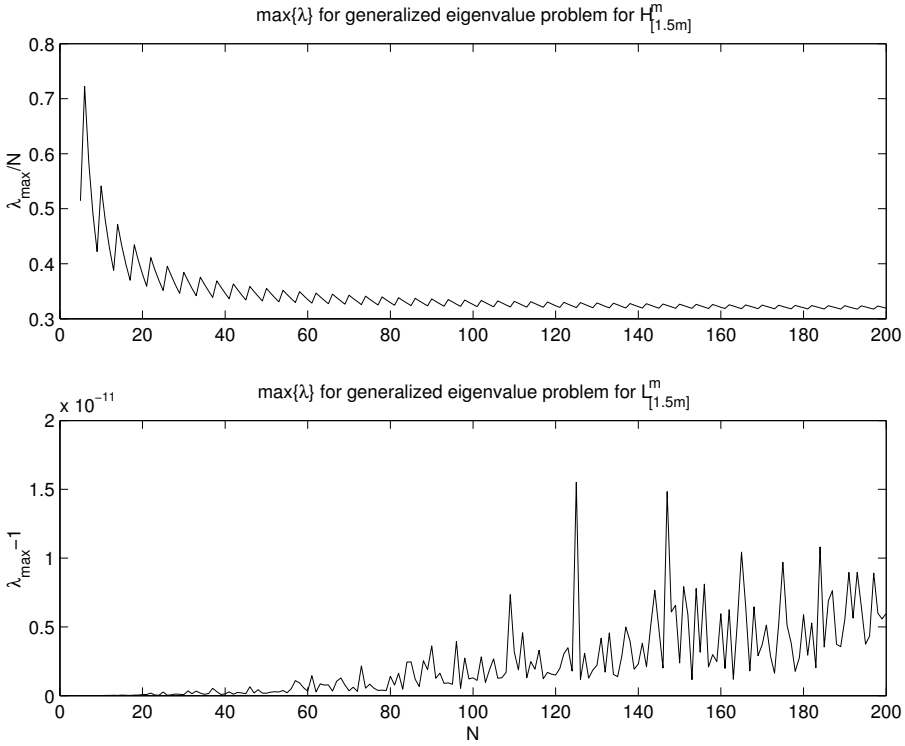


Figure 7.6: Maximal eigenvalues for the two generalized eigenvalue problems with $p = \lfloor 1.5N \rfloor$, $m = N$. Top panel: problem (7.22), plot of λ_{max}/N. Bottom panel: problem (7.21), plot of $\lambda_{max} - 1$.

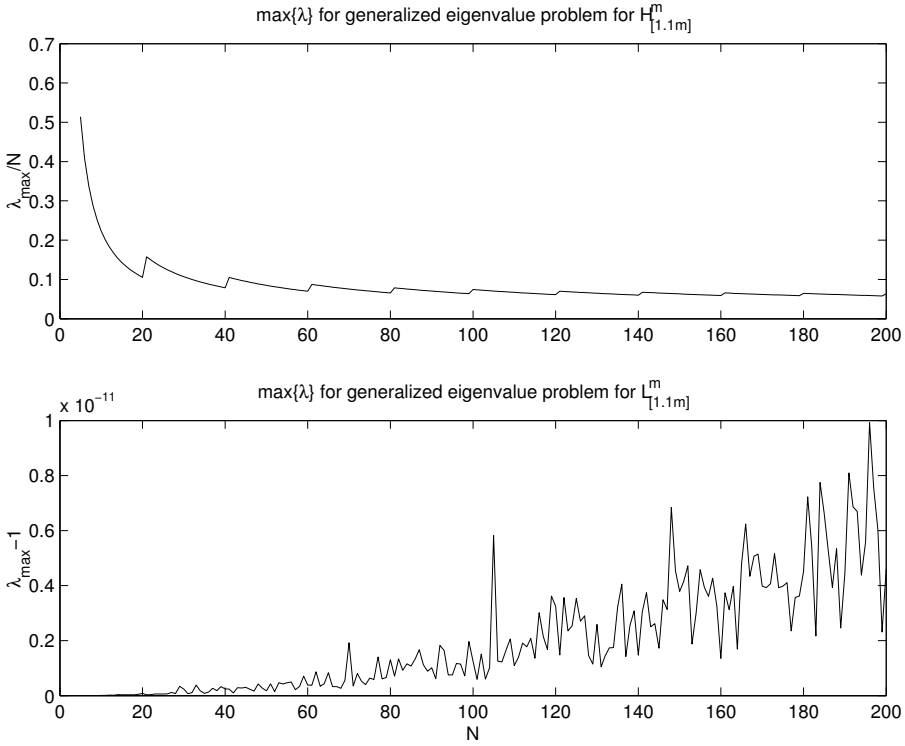


Figure 7.7: Maximal eigenvalues for the two generalized eigenvalue problems with $p = \lceil 1.1N \rceil$, $m = N$. Top panel: problem (7.22), plot of λ_{max}/N. Bottom panel: problem (7.21), plot of $\lambda_{max} - 1$.

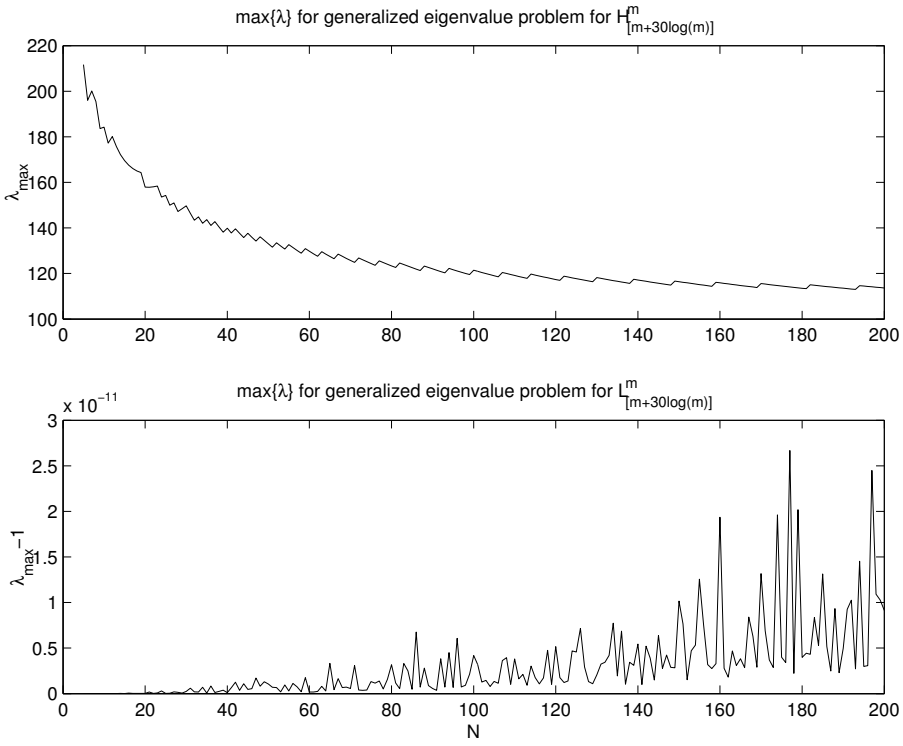


Figure 7.8: Maximal eigenvalues for the two generalized eigenvalue problems with $p = \lceil N + 30\log(N) \rceil$, $m = N$. Top panel: problem (7.22), plot of λ_{max}. Bottom panel: problem (7.21), plot of $\lambda_{max} - 1$.

7.7 Discrete Friedrichs' inequality

We saw in chapter 2 that on the complement of the kernel of $\mathbf{curl}$, the weakly divergence free functions (i.e. functions that are orthogonal to all the gradients of H_0^1), we have a Friedrichs' inequality (see theorem 2.6)

$$||\mathbf{u}||_{0,\Omega} \leq C H_\Omega ||\,\mathbf{curl}\,\mathbf{u}||_{0,\Omega}$$

In various situations, we need to ascertain the analogous inequality on a polynomial space that is orthogonal to a space of gradients of another polynomial space. The fundamental idea is to split the function on the constrained polynomial space into one which is continously weakly divergence free and estimate the rest. The complicating feature of the proof is that the needed interpolant, the Nédélec interpolant (again needed because of its commuting diagram property), is not defined on $(H^1)^d$, where we would prefer to work. Therefore we have to choose more regular spaces to work on. Girault and Raviart prove this Friedrichs' inequality for finite elements (the h-version) in [48, Proposition 5.1] using $W^{1,s}$ spaces and Monk [71, Theorem 4.1] proves it in the hN-version using $H^{1+\epsilon}$ spaces.

We give proofs only for the three-dimensional case. The two-dimensional result can be proven in a similar way, certain steps simplify and sharper results can be obtained. We will indicate some of these improvements.

In the proof of Friedrichs' inequality we need an approximation result which will be useful later in the analysis of our domain decomposition preconditioners:

Lemma 7.17 *Assume that the bounded and convex domain Ω with $H_\Omega = O(1)$ has a Lipschitz boundary and is covered with an uniformly regular mesh of elements of size h. Assume also that $\mathbf{w} \in H_0^\perp(\mathbf{curl})$ and that $\mathbf{curl}\,\mathbf{u} \in \mathbb{W}_N(\Omega)$. Then the Nédélec interpolant allows the following L^2-bounds:*

$$||\mathbf{w} - \mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 \leq Ch\left(1 + C_1\left(\frac{2}{1-\epsilon}\right)N^{-1+\epsilon}\right)||\,\mathbf{curl}\,\mathbf{w}||_0$$

$$||\mathbf{w} - \mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 \leq Ch\left(1 + C_2\left(\frac{\epsilon}{2}\right)N^{-1+\epsilon}\right)||\,\mathbf{curl}\,\mathbf{w}||_0$$

where $C_1(s)$ is the regularity constant of the $\mathbf{curl}$ potential problem from $\mathbf{c} = \mathbf{curl}\,\mathbf{w} \in (L^s)^3$ to $\mathbf{w} \in (W^{1,s})^3$ and $C_2(\epsilon)$ is the regularity constant of the same problem, but from $\mathbf{c} = \mathbf{curl}\,\mathbf{w} \in (H^\epsilon)^3$ to $\mathbf{w} \in (H^{1+\epsilon})^3$.

Proof: Both of the bounds are proven in a very similar way, starting from the stability estimates in lemma 7.8. If we would have an interpolation estimate such as in lemmata 7.11 and

7.13 for any $r < 2$, or a slightly degraded one, we would obtain by (Hilbert space) interpolation between $H^{1+\epsilon}$ and H^r an interpolation estimate that would allow a direct proof of the lemma, but with a better constant $CC_3(\epsilon)hN^{-1+f(\epsilon)}$. For two dimensions that is possible. For three dimensions we still need a proof of such an optimal H^r interpolation estimate. See figure 7.9 for a graphical representation. The solid line shows the upper bound below which we could prove this lemma using a H^s interpolation estimate with $s \in (1, \frac{3}{2})$, and the other lines show different interpolation estimates from the lemmata. In two dimensions, we should have a $H^{\frac{3}{2}+\epsilon}$ interpolation estimate and therefore a proof with the better constant.

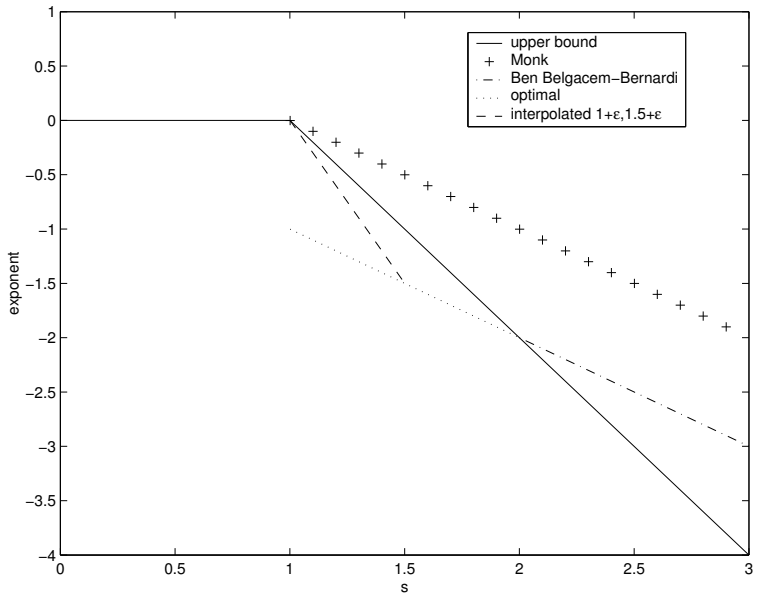


Figure 7.9: Exponents in the proof of Friedrichs' inequality, H^s case

Proof of the $W^{1,s}$ case: We start off with the stability estimate from lemma 7.8:

$$||\mathbf{w} - \mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 \leq Ch[N^{-1}||\mathbf{w}||_{1,s} + ||\mathbf{w}||_1]$$

Next comes the realization that $\mathbf{w}$ is a solution of the **curl** potential problem:

$$\mathbf{curl}\,\mathbf{w} \in (\mathbb{Q}_{N,N,N}(K))^3 \subset (L^p(\Omega))^3 \quad \forall p \qquad \mathrm{div}\,\mathbf{w} = 0 \qquad \mathbf{w} \times \mathbf{n}|_{\partial\Omega} = 0$$

Since this problem is regular from $(L^p)^3$ to $(W^{1,p})^3$ for $p \in (2, s_\Omega)$ for convex domains (see theorem 2.21 in section 2.8), we have

$$||\mathbf{w}||_{1,s} \leq C_1(s)||\,\mathbf{curl}\,\mathbf{w}||_{0,s} = C_1(s)||\,\mathbf{curl}\,\mathbf{w}||_{L^s} \tag{7.23}$$

Using again the fact that $\mathbf{curl}\,\mathbf{w} \in (\mathbb{Q}_{N,N,N}(K))^3$, and that therefore an inverse inequality holds (see section 4.5) we obtain

$$||\,\mathbf{curl}\,\mathbf{w}||_{L^s} \leq CN^{2(\frac{1}{2}-\frac{1}{s})}||\,\mathbf{curl}\,\mathbf{w}||_0 = CN^{1-\frac{2}{s}}||\,\mathbf{curl}\,\mathbf{w}||_0 \tag{7.24}$$

Using the result from section 2.7, theorem 2.14, that $H_N(\Omega)$ is imbedded continuously in H^1 for convex domains, we have

$$||\mathbf{w}||_1 \leq C(||\mathbf{w}||_0 + ||\,\mathbf{curl}\,\mathbf{w}||_0)$$

If we use the continuous Friedrichs' inequality (theorem 2.6) – recall that we work on a domain of diameter of order 1 – we obtain the following bound with a different C

$$||\mathbf{w}||_1 \leq C||\,\mathbf{curl}\,\mathbf{w}||_0 \tag{7.25}$$

Using (7.23), (7.24), and (7.25) in the stability estimate, we finally obtain

$$||\mathbf{w} - \mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 \leq Ch[N^{-1}||w||_{1,s} + ||w||_1] \leq Ch\left(1 + C_1\,(s)\,N^{-\frac{2}{s}}\right)||\,\mathbf{curl}\,\mathbf{w}||_0$$

The ϵ form of that estimate follows by an easy calculation.

Proof of the $H^{1+\epsilon}$ case: We start of with the stability estimate from lemma 7.8:

$$||\mathbf{w} - \mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 \leq C(hN^{-1}||\mathbf{w}||_{1+\epsilon} + h||\mathbf{w}||_1)$$

Next comes the realization that $\mathbf{w}$ is a solution of the $\mathbf{curl}$ potential problem:

$$\mathbf{curl}\,\mathbf{w} \in (\mathbb{Q}_{N,N,N}(K))^3 \subset (H^\epsilon(\Omega))^3 \quad \epsilon \in [0, \frac{1}{2}) \quad \mathsf{div}\,\mathbf{w} = 0 \quad \mathbf{w} \times \mathbf{n}|_{\partial\Omega} = 0$$

Since this problem is regular from $(H^\epsilon)^3$ to $(H^{1+\epsilon})^3$ for $\epsilon \in [0, \epsilon_0)$ for convex domains (see section 2.8, theorem 2.20), we have

$$||\mathbf{w}||_{1+\epsilon} \leq C_2(\epsilon)||\,\mathbf{curl}\,\mathbf{w}||_\epsilon$$

and since $\mathbf{curl}\,\mathbf{w}$ is a piecewise polynomial, we have an inverse estimate (see section 4.5)

$$||\,\mathbf{curl}\,\mathbf{w}||_\epsilon \leq CN^{2\epsilon}||\,\mathbf{curl}\,\mathbf{w}||_0$$

Using the last two inequalities and (7.25) in the stability estimate, we obtain

$$||\mathbf{w} - \Pi_N^{ND,I}\mathbf{w}||_0 \leq Ch\left(C_2(\epsilon)N^{-1+2\epsilon} + 1\right)||\,\mathbf{curl}\,\mathbf{w}||_0$$

The form stated in the lemma follows by substituting $\frac{\epsilon}{2}$ for ϵ. ∎

Theorem 7.18 (Discrete Friedrichs' inequality for the hN case) *Assume that the bounded and convex domain Ω with $H_\Omega = O(1)$ has a Lipschitz boundary and is covered with an uniformly regular mesh of elements of size h. Let $\mathbf{\Phi}_N$ be discretely divergence free of degree N, i.e., assume $\mathbf{\Phi}_N \in \mathbb{ND}_N^{I,+}$. Then there exist constants C and C' such that*

$$||\mathbf{\Phi}_N||_0 \leq C\left(1 + h\left(1 + C_{1,2}(\epsilon)N^{-1+\epsilon}\right)\right)||\,\mathbf{curl}\,\mathbf{\Phi}_N||_0 \leq C'||\,\mathbf{curl}\,\mathbf{\Phi}_N||_0$$

Proof: The second inequality follows trivially from the first, since the coefficient is a function that decreases with increasing N and decreasing h, so that $N = 1$ and $h = H_\Omega$ substituted into the coefficient give a trival bound for C'.

The first inequality is proven in several steps.

Define $p \in H_0^1(\Omega)$ as the solution of the generalized Neumann problem

$$\forall q \in H_0^1(\Omega): \quad (\mathbf{grad}\,p, \mathbf{grad}\,q) = (\mathbf{\Phi}_N, \mathbf{grad}\,q)$$

Then, $\mathbf{w} := \mathbf{\Phi}_N - \mathbf{grad}\,p$ satisfies

$$\mathbf{curl}\,\mathbf{w} = \mathbf{curl}\,\mathbf{\Phi}_N \qquad \mathsf{div}\,\mathbf{w} = 0 \qquad \mathbf{w} \times \mathbf{n}|_{\partial\Omega} = 0$$

Since Ω is convex, either the $W^{1,s}$ or the $H^{1+\epsilon}$ regularity used in the proof of the previous lemma guarantees that $\mathbf{w} \in (W^{1,s})^3$ or $\mathbf{w} \in (H^{1+\epsilon})^3$, and that therefore $\mathbf{\Pi}_N^{ND,I}\mathbf{w}$ is defined. Since $\mathbf{\Phi}_N$ is in the Nédélec space, its interpolant is defined, and therefore $\mathbf{\Pi}_N^{ND,I}(\mathbf{grad}\,p)$ is defined. The appropriate version of the commuting diagram property (see, e.g., Girault and Raviart [48, Lemma 5.10]) shows that there is a piecewise polynomial p_N such that

$$\mathbf{\Pi}_k^{ND,I}(\mathbf{grad}\,p) = \mathbf{grad}\,p_N$$

and therefore $\mathbf{\Phi}_N = \mathbf{\Pi}_N^{ND,I}\mathbf{w} + \mathbf{grad}\,p_N$. Now $(\mathbf{\Phi}_N, \mathbf{grad}\,q_N) = 0$ for all $q_N \in \mathbb{S}_N(\Omega)$, therefore also especially for $q_N = p_N$. This gives that $(\mathbf{\Phi}_N, \mathbf{\Phi}_N) = (\mathbf{\Pi}_N^{ND,I}\mathbf{w}, \mathbf{\Phi}_N) + (\mathbf{grad}\,p_N, \mathbf{\Phi}_N) = (\mathbf{\Pi}_N^{ND,I}\mathbf{w}, \mathbf{\Phi}_N)$ and an application of the Cauchy-Schwarz inequality gives

$$||\mathbf{\Phi}_N||_0 \leq ||\mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0$$

Next we use lemma 7.17 and the triangle inequality to show

$$\begin{aligned}||\mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 &\leq ||\mathbf{w}||_0 + ||\mathbf{w} - \mathbf{\Pi}_N^{ND,I}\mathbf{w}||_0 \\ &\leq C\left(1 + h\left(1 + C_{1,2}(\epsilon)N^{-1+\epsilon}\right)\right)||\,\mathbf{curl}\,\mathbf{w}||_0\end{aligned}$$

where $C_{1,2}(\epsilon)$ is one of the two $C_2(\frac{\epsilon}{2})$ and $C_1(\frac{2}{1-\epsilon})$.

The proof is completed by recalling that $\mathbf{curl}\,\mathbf{w} = \mathbf{curl}\,\mathbf{\Phi}_N$. ∎

If we can prove a spectral version of Lemma 4.7 in Amrouche, Bernardi, Dauge, and Girault [5, page 855], that is, if we can give a bound of the L^2-norm of $\mathbf{\Pi}_N^{ND,I}$ in terms of the X^p-norm with an explicit dependence on N and a coefficient that does not depend on N, then we can prove the hN-version of the discrete Friedrichs' inequality for non-convex domains following the proof of [5, Proposition 4.6] adapted for the case of the potential with tangential boundary values [5, Proposition 4.12]. A variant of lemma 7.17 can then be proven following the proof of a similar inequality in Arnold, Falk and Winther [8, (2.4)].

Chapter 8

Spectral Elements for the Maxwell model problem

In this chapter, we will discretize $\alpha Id + \beta\,\mathbf{curl}\,\mathrm{curl}$ in two dimensions on rectangular elements. We naturally work with the variational formulation

$$?\mathbf{u} \in H(\mathrm{curl}) : \forall \mathbf{v} \in H(\mathrm{curl}) : (\alpha\mathbf{u}, \mathbf{v})_0 + (\beta\,\mathrm{curl}\,\mathbf{u}, \mathrm{curl}\,\mathbf{v})_0 = \mathbf{f}(\mathbf{v}) \qquad (8.1)$$

and we will construct a discrete function space $V(\mathrm{curl})$ to approximate $H(\mathrm{curl})$.

We will provide details only in the two-dimensional case. Almost everything carries over into three dimensions, and we will discuss differences between the two-dimensional case and the three-dimensional case in remarks.

In the first section we describe how to discretize the problem on one element. In the second, short, section, we discuss the discretization on domains consisting out of more than one element. For the case where the domain is logically rectangular, we give subassembly procedures for the $H(\mathrm{curl})$, H^1, and $H(\mathsf{div})$ conforming case in the next section. In the last section we discuss how to enforce different types of boundary conditions.

There are only a few numerical experiments in this chapter. We will present numerical experiments that apply the discretizations and methods from this chapter in chapter 9, where we discuss fast direct solvers for them, and in chapter 10, where we will show their use in domain decomposition preconditioners.

8.1 Discretization on one element

We try to discretize

$$?\mathbf{u} \in H(\mathrm{curl}) : \forall \mathbf{v} \in H(\mathrm{curl}) : (\alpha\mathbf{u}, \mathbf{v})_0 + (\beta\,\mathrm{curl}\,\mathbf{u}, \mathrm{curl}\,\mathbf{v})_0 = \mathbf{f}(\mathbf{v})$$

with $\mathbf{f}(\mathbf{v}) = (\mathbf{f}_{(1)}, \mathbf{v})_0 + (\mathbf{f}_{(2)}, \operatorname{curl} \mathbf{v})_0$ on a rectangular element $[a, b] \times [c, d]$. We will perform the derivation on $[-1, 1]^2$ and then obtain the general result by scaling.

If we have more general mappings F from the reference element, we can discretize the equations similarly by considering $\mathbf{u} \circ F$ and $\mathbf{v} \circ F$ instead of $\mathbf{u}$ and $\mathbf{v}$, and multiplying all integrands by the determinant of the Jacobian of F. The special structure needed for our fastest solvers will not be available for general F, but a fast application of the stiffness matrix is still possible.

As indicated in the first section of the previous chapter, we choose $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$ as the local space $V(\operatorname{curl})$. For multi-element problems we will have to enforce tangential continuity on the product space of all local spaces.

In an exact Galerkin method we would compute all the integrals exactly. If α, β, $\mathbf{f}_{(1)}$, $\mathbf{f}_{(2)}$ are polynomials, then we can achieve this by using Gaussian quadrature of high enough order. For arbitrary α, β, $\mathbf{f}_{(1)}$,$\mathbf{f}_{(2)}$ we would have to be able to analytically compute all the integrals which is impossible in the general case. For arbitrary coefficients we will therefore use numerical integration, which gives an additional error term in the analysis of the method by the appropriate variant of Strang's lemma (see for instance Bernardi and Maday [17], or Ciarlet [27]).

Since we have Gaussian quadratures of many orders at our easy disposal (see section 4.3), we can easily study the influence of the accuracy of the quadrature.

Overintegration (of not exactly computable terms) and underintegration may make sense. Overintegration of critical terms may improve the overall accuracy, underintegration may result in an advantageous special form or properties of the matrices, without loosing too much accuracy and keeping the same order of convergence.

The exact analysis of underintegration for our discretization in its full generality is non-trivial, and would require error estimates for anisotropic polynomial spaces and analysis of the approximate bilinear forms obtained by general tensor product Gaussian quadrature rules. While that seems to be a feasible and interesting enterprise on its own (for some very recent work on error estimates for anisotropic discretizations in the context of the h-version for Nédélec elements see Nicaise [76]), we lack the space and the time to execute it in the context of this thesis. We will take a hint from the theory and our experiments in the chapter 6: we will integrate differentiated directions in the components exactly and we will integrate undifferentiated directions exactly or one order lower (which leads to diagonal mass matrices). A list with appropriate choices of degrees will be given in the statement of the discretization.

Let us define (see sections 4.3 and 4.6) that $(u, v)_{M,N}$ denotes the Gaussian integration on $\mathrm{GLL}_M \times \mathrm{GLL}_N$.

We assume constant α and β for simplicity.

Remark: If we have separable α and β, then we can obtain a discretization of a very similar form, only that the weighted inner products $(\alpha\cdot,\cdot)$ and $(\beta\cdot,\cdot)$ will give modified mass matrices that are tensor products of one dimensional standard mass matrices scaled by the appropriate parts of α and β.

We try to approximate for $\mathbf{u},\mathbf{v}\in V(\mathrm{curl})$

$$(\alpha\mathbf{u},\mathbf{v})_0+(\beta\,\mathrm{curl}\,\mathbf{u},\mathrm{curl}\,\mathbf{v})_0=(\mathbf{f}_{(1)},\mathbf{v})_0+(\mathbf{f}_{(2)},\mathrm{curl}\,\mathbf{v})_0 \tag{8.2}$$

Written more explicitly,

$$?u_1\in\mathbb{Q}_{m_1,n_1},?u_2\in\mathbb{Q}_{m_2,n_2}:\forall v_1\in\mathbb{Q}_{m_1,n_1}:\forall v_2\in\mathbb{Q}_{m_2,n_2}$$
$$\begin{aligned}\alpha(u_1,v_1)_0+\alpha(u_2,v_2)_0+\beta(\partial_x u_2-\partial_y u_1,\partial_x v_2-\partial_y v_1)_0\\ =(f_1,v_1)_0+(f_2,v_2)_0+(f_3,\partial_x v_2-\partial_y v_1)_0\end{aligned}$$

In the following we will discuss the degrees and the discretization of the different terms separately.

$(u_1,v_1)_0$ is the integral of $u_1\cdot v_1$ over the rectangular element. Therefore, $u_1\cdot v_1$ is contained in $\mathbb{Q}_{2m_1,2n_1}$. To integrate this exactly, we have to use $(\cdot,\cdot)_{M_1,N_1}$ with $M_1\geq m_1+1$ and $N_1\geq n_1+1$. Such exact integration leads to a non-diagonal mass matrix. Choosing $M_1=m_1$ and $N_1=n_1$ results in a diagonal mass matrix and is often used in spectral element methods, especially since no order of convergence is lost in standard examples such as the isotropic discretization of the Laplace equation.

Similarly, $u_2\cdot v_2$ is contained in $\mathbb{Q}_{2m_2,2n_2}$ and it is integrated exactly with $(\cdot,\cdot)_{M_2,N_2}$ under the condition $M_2\geq m_2+1$ and $N_2\geq n_2+1$. If we decrease both degrees by one, we obtain diagonal matrices.

We will treat the different parts of the $(\mathrm{curl}\,\cdot,\mathrm{curl}\,\cdot)$ term separately, since they have different degrees.

$$(\mathrm{curl}\,\mathbf{u},\mathrm{curl}\,\mathbf{v})=(\partial_y u_1,\partial_y v_1)+(\partial_x u_2,\partial_x v_2)-(\partial_y u_1,\partial_x v_2)-(\partial_x u_2,\partial_y v_1)$$

The first part, $(\partial_y u_1,\partial_y v_1)$ leads to an integration of a function in $\mathbb{Q}_{2m_1,2n_1-2}$ and is integrated exactly with $(\cdot,\cdot)_{M_3,N_3}$ given $M_3\geq m_1+1$, $N_3\geq n_1$. $(\partial_x u_2,\partial_x v_2)$ leads to an integration in $\mathbb{Q}_{2m_2-2,2n_2}$ which will be exact with $(\cdot,\cdot)_{M_4,N_4}$ under the condition $M_4\geq m_2$ and $N_4\geq n_2+1$. Both of the last two parts lead to the integration of a polynomial in $\mathbb{Q}_{m_1+m_2-1,n_1+n_2-1}$ and are integrated by $(\cdot,\cdot)_{M_5,N_5}$. The integration is exact under the conditions $M_5\geq\frac{m_1+m_2}{2}$, $N_5\geq\frac{n_1+n_2}{2}$.

Putting the parts back together, we approximate

$$(\alpha\mathbf{u},\mathbf{v})\approx(\alpha\mathbf{u},\mathbf{v})_S:=\alpha(u_1,v_1)_{M_1,N_1}+\alpha(u_2,v_2)_{M_2,N_2}$$

$$(\beta \operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v}) \approx (\beta \operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v})_S := \beta(\partial_y u_1, \partial_y v_1)_{M_3,N_3} + \\ \beta(\partial_x u_2, \partial_x v_2)_{M_4,N_4} - \beta(\partial_y u_1, \partial_x v_2)_{M_5,N_5} - \beta(\partial_x u_2, \partial_y v_1)_{M_5,N_5}$$

With the same approach we obtain a spectral approximation $\mathbf{f}_S(\mathbf{u})$ of $\mathbf{f}(\mathbf{u})$, and finally the discrete problem

$$?\mathbf{u} \in V(\operatorname{curl}) : \forall \mathbf{v} \in V(\operatorname{curl}) : (\alpha \mathbf{u}, \mathbf{v})_S + (\beta \operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v})_S = \mathbf{f}_S(\mathbf{u}) \qquad (8.3)$$

Next we will find a matrix representation of this problem and make explicit its structure as a system of linear equations. In the derivation of this representation we will realize an additional simplifying restriction on the degrees of the integration formulae.

We will make a distinction between an arbitrary function in $H(\operatorname{curl})$ resp. $V(\operatorname{curl})$ and its vector of nodal values[1]. For any function u we will denote the vector of nodal values by $\underline{u}$, for any vector of nodal values v, we will denote the corresponding function, obtained by straightforward interpolation using the Gauss-(Lobatto-)Legendre nodal basis, by $\overline{v}$.

To compute $(\alpha \mathbf{u}, \mathbf{v})_S$ in matrix form, we need to interpolate from $\mathbb{Q}_{m_i,n_i}$ to $\mathbb{Q}_{M_i,N_i}$ and then use Gaussian quadrature there (see chapter 4 for Gaussian quadratures and mass matrices, interpolation matrices and differentiation matrices for the one-dimensional case)

$$\begin{aligned}
(\alpha \mathbf{u}, \mathbf{v})_S &= \alpha(u_1, v_1)_{M_1,N_1} + \alpha(u_2, v_2)_{M_2,N_2} \\
&= \alpha \underline{v}_1^T (I_{m_1}^{M_1} \otimes I_{n_1}^{N_1})^T (M_{M_1} \otimes M_{N_1})(I_{m_1}^{M_1} \otimes I_{n_1}^{N_1}) \underline{u}_1 \\
&\quad + \alpha \underline{v}_2^T (I_{m_2}^{M_2} \otimes I_{n_2}^{N_2})^T (M_{M_2} \otimes M_{N_2})(I_{m_2}^{M_2} \otimes I_{n_2}^{N_2}) \underline{u}_2 \\
&= \alpha \underline{v}_1^T (M_{m_1}^{M_1} \otimes M_{n_1}^{N_1}) \underline{u}_1 + \alpha \underline{v}_2^T (M_{m_2}^{M_2} \otimes M_{n_2}^{N_2}) \underline{u}_2
\end{aligned}$$

$\operatorname{curl} \mathbf{u} = \partial_x u_2 - \partial_y u_1$. A priori, the two terms in the definition of curl are not of the same degree. If we desire an exact representation of $\operatorname{curl} \mathbf{u}$ on a GLL$(m_3) \times$ GLL(n_3) mesh, we would require $m_3 \geq \max\{m_1, m_2 - 1\}$ and $n_3 \geq \max\{n_1 - 1, n_2\}$ and we would obtain $\operatorname{curl} \mathbf{u}$ on that grid as

$$\begin{aligned}
\underline{\operatorname{curl} \mathbf{u}}_{m_3,n_3} &= (I_{m_2}^{m_3} \otimes I_{n_2}^{n_3})(D_{m_2} \otimes I_{n_2}) \underline{u}_2 - (I_{m_1}^{m_3} \otimes I_{n_1}^{n_3})(I_{m_1} \otimes D_{n_1}) \underline{u}_1 \\
&= ((I_{m_2}^{m_3} D_{m_2}) \otimes I_{n_2}^{n_3}) \underline{u}_2 - (I_{m_1}^{m_3} \otimes (I_{n_1}^{n_3} D_{n_1})) \underline{u}_1
\end{aligned}$$

We will discretize the parts separately and not enforce a common mesh for $\operatorname{curl} \mathbf{u}$ in the bilinear form.

The parts of $(\operatorname{curl} \cdot, \operatorname{curl} \cdot)_S$ transform into matrix form as follows:

$$\begin{aligned}
\beta(\partial_y u_1, \partial_y v_1)_{M_3,N_3} &= (\overline{(I_{m_1} \otimes D_{n_1}) \underline{u}_1}, \overline{(I_{m_1} \otimes D_{n_1}) \underline{v}_1})_{M_3,N_3} \\
&= \beta \underline{v}_1^T (I_{m_1} \otimes D_{n_1})^T (M_{m_1}^{M_3} \otimes M_{n_1}^{N_3}) (I_{m_1} \otimes D_{n_1}) \underline{u}_1 \\
&= \beta \underline{v}_1^T (M_{m_1}^{M_3} \otimes (D_{n_1}^T M_{n_1}^{N_3} D_{n_1})) \underline{u}_1
\end{aligned}$$

[1] We assume that the function is sufficiently regular so that these point values are defined.

$$\begin{aligned}\beta(\partial_x u_2, \partial_x v_2)_{M_4,N_4} &= (\overline{(D_{m_2}\otimes I_{n_2})\underline{u}_2}, \overline{(D_{m_2}\otimes I_{n_2})\underline{v}_2})_{M_4,N_4}\\ &= \beta \underline{v}_2^T (D_{m_2}\otimes I_{n_2})^T (M_{m_2}^{M_4}\otimes M_{n_2}^{N_4})(D_{m_2}\otimes I_{n_2})\underline{u}_2\\ &= \beta \underline{v}_2^T ((D_{m_2}^T M_{m_2}^{M_4} D_{m_2})\otimes M_{n_2}^{N_4})\underline{u}_2\end{aligned}$$

$$\begin{aligned}\beta(\partial_y u_1, \partial_x v_2)_{M_5,N_5} &= (\overline{(I_{m_1}\otimes D_{n_1})\underline{u}_1}, \overline{(D_{m_2}\otimes I_{n_2})\underline{v}_2})_{M_5,N_5}\\ &= \beta \underline{v}_2^T (D_{m_2}\otimes I_{n_2})^T (I_{m_2}^{M_5}\otimes I_{n_2}^{N_5})^T (M_{M_5}\otimes M_{N_5})\\ &\qquad (I_{m_1}^{M_5}\otimes I_{n_1}^{N_5})(I_{m_1}\otimes D_{n_1})\underline{u}_1\\ &= \beta \underline{v}_2^T ((D_{m_2}^T I_{m_2}^{M_5,T} M_{M_5} I_{m_1}^{M_5})\otimes (I_{n_2}^{N_5,T} M_{N_5} I_{n_1}^{N_5} D_{n_1}))\underline{u}_1\end{aligned}$$

$$\begin{aligned}\beta(\partial_x u_2, \partial_y v_1)_{M_5,N_5} &= (\overline{(D_{m_2}\otimes I_{n_2})\underline{u}_2}, \overline{(I_{m_1}\otimes D_{n_1})\underline{v}_1})_{M_5,N_5}\\ &= \beta \underline{v}_1^T (I_{m_1}\otimes D_{n_1})^T (I_{m_1}^{M_5}\otimes I_{n_1}^{N_5})^T (M_{M_5}\otimes M_{N_5})\\ &\qquad (I_{m_2}^{M_5}\otimes I_{n_2}^{N_5})(D_{m_2}\otimes I_{n_2})\underline{u}_2\\ &= \beta \underline{v}_1^T ((I_{m_1}^{M_5,T} M_{M_5} I_{m_2}^{M_5} D_{m_2})\otimes (D_{n_1}^T I_{n_1}^{N_5,T} M_{N_5} I_{n_2}^{N_5}))\underline{u}_2\end{aligned}$$

Assuming for simplicity that $f_1 \in \mathbb{Q}_{m_1,n_1}$, $f_2 \in \mathbb{Q}_{m_2,n_2}$ and $f_3 \in \mathbb{Q}_{m_3,n_3}$, and that we treat the terms (f_i, v_i) like the terms (u_i, v_i), the right hand side is approximated by:

$$\underline{v}_1^T (M_{m_1}^{M_1}\otimes M_{n_1}^{N_1})\underline{f}_1 + \underline{v}_2^T (M_{m_2}^{M_2}\otimes M_{n_2}^{N_2})\underline{f}_2$$

$$-\underline{v}_1^T ((I_{m_1}^{m_3,T} M_{m_3})\otimes (D_{n_1}^T I_{n_1}^{n_3,T} M_{n_3}))\underline{f}_3 + \underline{v}_2^T ((D_{m_2}^T I_{m_2}^{m_3,T} M_{m_3})\otimes (I_{n_2}^{n_3,T} M_{n_3}))\underline{f}_3$$

If we want to combine the $\alpha \underline{v}_1^T \ldots \underline{u}_1$ term and the $\beta \underline{v}_1^T \ldots \underline{u}_1$ term, we need to choose $M_3 = M_1$. Similarly, we need $N_4 = N_2$ to combine $\alpha \underline{v}_2^T \ldots \underline{u}_2$ and $\beta \underline{v}_2^T \ldots \underline{u}_2$.

Under these conditions, and collecting terms, we obtain an equation of the form:

$$\begin{aligned}&\underline{v}_1^T (M_1^x \otimes A^y)\underline{u}_1 + \underline{v}_1^T (B^x \otimes C^y)\underline{u}_2 + \underline{v}_2^T (C^x \otimes B^y)\underline{u}_1 + \underline{v}_2^T (A^x \otimes M_2^y)\underline{u}_2 \;=\\ &\underline{v}_1^T (M_1^x \otimes M_1^y)\underline{f}_1 + \underline{v}_2^T (M_2^x \otimes M_2^y)\underline{f}_2 - \underline{v}_1^T (F_1^x \otimes F_1^y)\underline{f}_3 + \underline{v}_2^T (F_2^x \otimes F_2^y)\underline{f}_3 \qquad (8.4)\end{aligned}$$

with, for instance,

$$M_1^x = M_{m_1}^{M_1} \qquad M_1^y = M_{n_1}^{N_1} \qquad M_2^x = M_{m_2}^{M_2} \qquad M_2^y = M_{n_2}^{N_2}$$

$$A^x = \alpha M_{m_2}^{M_2} + \beta D_{m_2}^T M_{m_2}^{M_4} D_{m_2} = \alpha M_{m_2}^{M_2} + \beta K_{m_2}^{M_4}$$

$$A^y = \alpha M_{n_1}^{N_1} + \beta D_{n_1}^T M_{n_1}^{N_3} D_{n_1} = \alpha M_{n_1}^{N_1} + \beta K_{n_1}^{N_3}$$

$$B^x = -\beta I_{m_1}^{M_5,T} M_{M_5} I_{m_2}^{M_5} D_{m_2}$$

$$B^y = I_{n_2}^{N_5,T} M_{N_5} I_{n_1}^{N_5} D_{n_1}$$

$$C^x = -\beta D_{m_2}^T I_{m_2}^{M_5,T} M_{M_5} I_{m_1}^{M_5} = B^{x,T}$$
$$C^y = D_{n_1}^T I_{n_1}^{N_5,T} M_{N_5} I_{n_2}^{N_5} = B^{y,T}$$
$$F_1^x = I_{m_1}^{m_3,T} M_{m_3} \qquad F_1^y = D_{n_1}^T I_{n_1}^{n_3,T} M_{n_3}$$
$$F_2^x = D_{m_2}^T I_{m_2}^{m_3,T} M_{m_3} \qquad F_2^y = I_{n_2}^{n_3,T} M_{n_3}$$

A^x and A^y are scaled discretizations of one-dimensional Helmholtz operators, and they contain K_n^N which is a spectral discretization of an one-dimensional Laplace operator. We have studied the properties of these operators in chapter 6.

Since (8.4) has to be valid for all possible vectors $\underline{v}_1$ and $\underline{v}_2$, we can especially choose test vectors that are zero in one component and arbitrary in the other, and therefore obtain, finally, the system of equation on one element as:

$$(M_1^x \otimes A^y)\underline{u}_1 + (B^x \otimes C^y)\underline{u}_2 = (M_1^x \otimes M_1^y)\underline{f}_1 - (F_1^x \otimes F_1^y)\underline{f}_3 \quad (8.5)$$
$$(C^x \otimes B^y)\underline{u}_1 + (A^x \otimes M_2^y)\underline{u}_2 = (M_2^x \otimes M_2^y)\underline{f}_2 + (F_2^x \otimes F_2^y)\underline{f}_3 \quad (8.6)$$

This is a symmetric system of equations.

We have to choose 8 degrees of integrations, namely $M_1 = M_3$, M_2, M_4, M_5, N_1, $N_2 = N_4$, N_3 and N_5. There is no differentiation in the directions associated to the quadrature degrees M_1, M_2, N_1, N_2. These directions have to be integrated with $m_i + 1$ resp. $n_i + 1$ for exact integration and with m_i resp. n_i for diagonal mass matrices.

We differentiate in the directions associated with the quadrature degrees N_3 and M_4. We use exact integration with degrees n_1 and m_2.

In the directions associated with the quadrature degrees M_5 and N_5, we have a product in which differentiated and not differentiated components are mixed. We will test both exact and slightly inexact integration using $M_5 = \frac{m_1+m_2}{2}, \frac{m_1+m_2}{2} - 1$ and $N_5 = \frac{n_1+n_2}{2}, \frac{n_1+n_2}{2} - 1$

To compute on $[a, b] \times [c, d]$, we have to multiply the mass matrices for x by $\frac{b-a}{2}$ and the ones for y by $\frac{d-c}{2}$. The differentiation matrices D_{m_i} and D_{n_i} will be replaced by $\frac{2}{b-a} D_{m_i}$ and $\frac{2}{d-c} D_{n_i}$.

If we use this method as a spectral method (i.e., no subdivision into elements, the entire rectangular domain is discretized with one spectral element), it may be advantageous to multiply the two equations of the system with the inverse of the mass matrices, i.e., $((M_i^x)^{-1} \otimes (M_i^y)^{-1})$. Then one obtains a system of the form

$$(I_1^x \otimes \mathcal{A}^y)\underline{u}_1 + (\mathcal{B}^x \otimes \mathcal{C}^y)\underline{u}_2 = \underline{f}_1 - (\mathcal{F}_1^x \otimes \mathcal{F}_1^y)\underline{f}_3 \quad (8.7)$$
$$(\mathcal{C}^x \otimes \mathcal{B}^y)\underline{u}_1 + (\mathcal{A}^x \otimes I_2^y)\underline{u}_2 = \underline{f}_2 + (\mathcal{F}_2^x \otimes \mathcal{F}_2^y)\underline{f}_3 \quad (8.8)$$

We will discuss how to solve such systems fast in chapter 9.

As an example, and to show the convergence of such elements, we present the results in figure 8.1, 8.2, 8.3 and 8.4.

In figures 8.1 and 8.2 we show the error solving a tangential boundary value problem with a spectral method with the Nédélec I space, i.e., $\mathbb{Q}_{k-1,k} \times Q_{k,k-1}$ and in figures 8.3 and 8.4 we show the analogous results for the Nédélec II case. All the figures correspond to a case with $\alpha = 1$, $\beta = 1$ and with the exact solution $\mathbf{u} = (\sin(\frac{\pi}{2}y)x, \sin(\frac{\pi}{2}x)y)$ on $[-1, 1]^2$.

We tested the following choices for quadrature degrees: exact integration and slight underintegration for purely differentiated terms, exact integration and slight underintegration for the mass matrices, and exact integration and slight underintegration for the mixed terms.

In figures 8.1 and 8.3 we tested the exact integration of the mixed terms, in 8.2 and 8.4 we underintegrated the mixed term by one degree. The results look in all cases very similar. (The spike in figure 8.4 corresponds to a badly conditioned eigensystem in the fast diagonalization method in the solution algorithm, and it could be avoided by a slightly different numerical algorithm.) For Nédélec I elements, underintegration of the mixed terms introduces an odd-even effect. We always observe exponential convergence. The versions with exact integration of the differentiated terms outperform the ones with slight underintegration by a small margin in the exponential convergence. For $N \geq 20$, it is even harder to compare the different choices. It seems that for underintegrated mixed terms the case with exactly integrated mass matrices and differentiated terms performs best, in the other cases there is no choice that performs always best.

In the numerical tests for the model problem in the rest of the thesis we use the versions with exactly integrated differentiated terms and mixed terms, and the two choices for the integration of the mass matrix.

In the three-dimensional case, we can analogously derive a system for $\underline{u}_1$, $\underline{u}_2$ and $\underline{u}_3$, in which all the blocks K_{ij} ($i = 1, 2, 3$, $j = 1, 2, 3$) of the stiffness matrix K are tensor product matrices. These blocks are also tensor products of discretizations of two-dimensional Helmholtz problems and mass matrices on the diagonal, and mixtures of mass, differentiation, and interpolation matrices on the off-diagonal, and the system is still symmetric. The fast solution of such systems will be explored in future work, and the extension of at least some of our algorithms to the three-dimensional case seems to be possible.

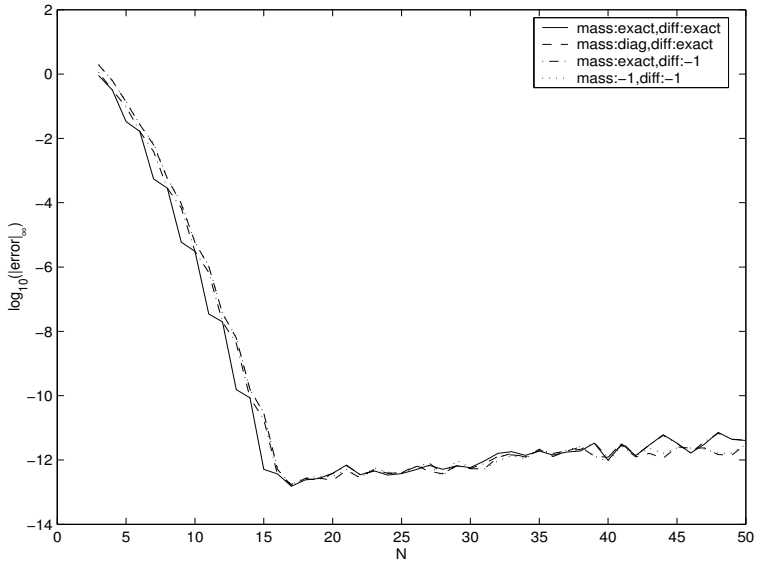


Figure 8.1: Two-dimensional $Id + \mathbf{curl}\,\mathbf{curl}$ problem, Nédélec I type elements, mixed terms integrated exactly: Results for different quadrature degrees.

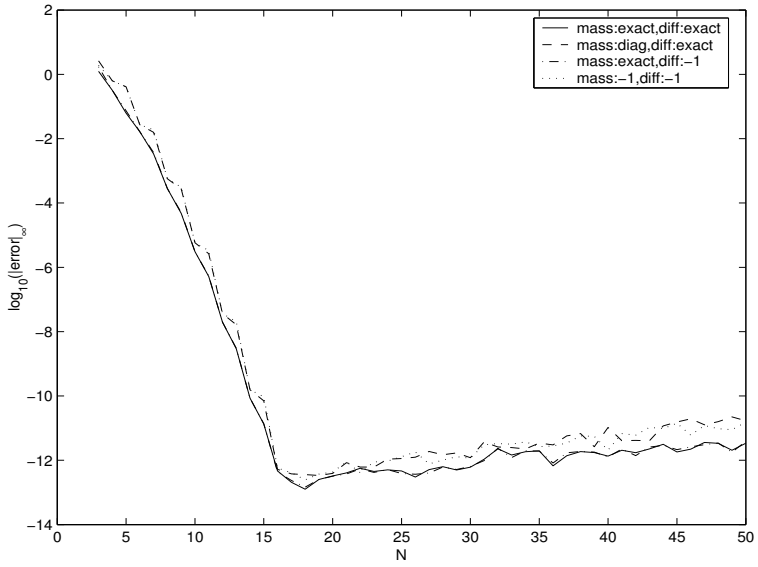


Figure 8.2: Two-dimensional $Id + \mathbf{curl}\,\mathbf{curl}$ problem, Nédélec I type elements, mixed terms slightly underintegrated: Results for different quadrature degrees.

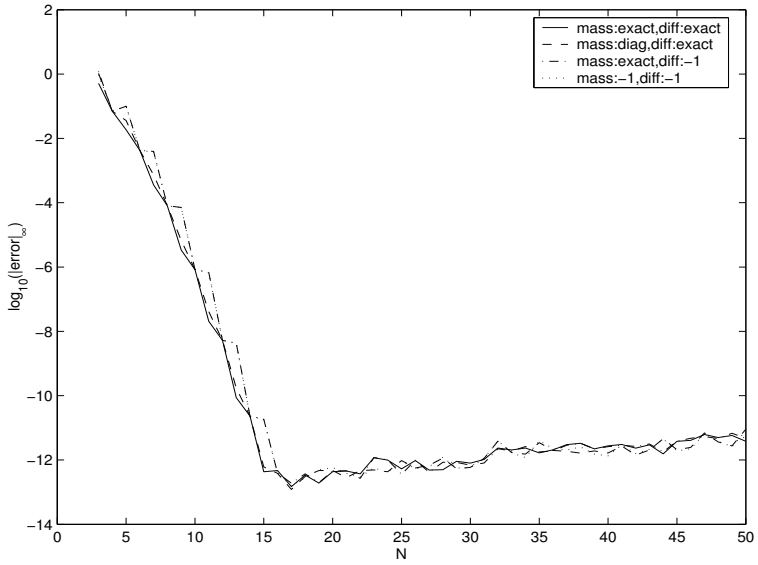


Figure 8.3: Two-dimensional $Id + \mathbf{curl}\,\mathbf{curl}$ problem, Nédélec II type elements, mixed terms integrated exactly: Results for different quadrature degrees.

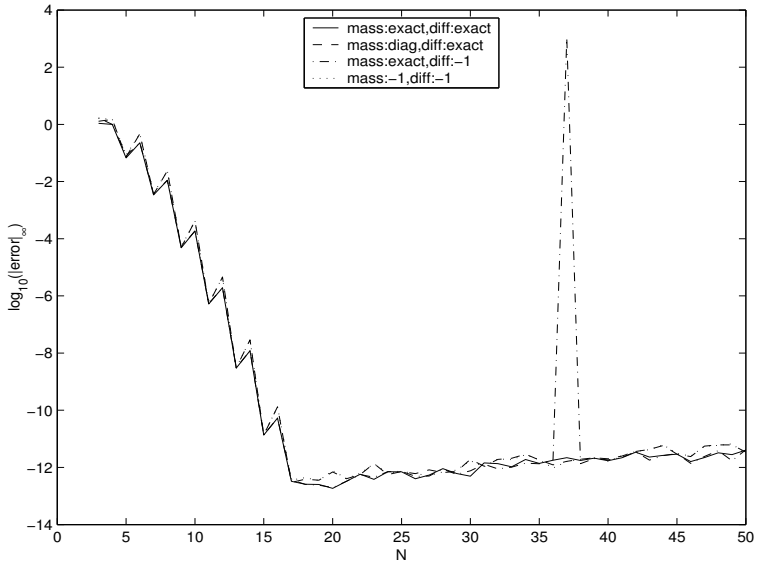


Figure 8.4: Two-dimensional $Id + \mathbf{curl}\,\mathbf{curl}$ problem, Nédélec II type elements, mixed terms slighly underintegrated: Results for different quadrature degrees.

8.2 Discretization on a collection of elements

Unfortunately, most domains occuring in practice are too large or too irregular to be mapped to a single rectangular element. Therefore we will have to work with collections of elements. We discretize the domain by a number of mapped elements and compute the element matrices by mapping back to the reference element and by using the discretization from the last section there. In the context of this thesis, we will only consider examples on rectangular meshes of elements, and therefore only need to scale the matrices derived on the reference element to get the correct element matrices. We plan to extend our methods to cases with general, well-behaved mappings f from the reference element in future work.

If we work on a domain decomposed into several elements, we will have to subassemble the discretization on the elements into that of the problem on the entire domain. In the case when the entire domain is logically rectangular and split into a rectangular mesh of elements of matching degrees, the global discretization has the same structure as on the elements, and we give the subassembly procedure in the next section. In the general case a standard subassembly procedure for unstructured finite elements can be used, except that we have to treat the nontangential components on the interfaces like interior components. On a block structured mesh, we will first use the methods from the next section to subassemble the rectangular blocks, and then subassemble the blocks with a subassembly procedure for unstructured finite elements.

We have to choose the element mesh and the degrees of the elements. We could consider them as given and leave the burden of choice to the designer of the discretization for a particular problem. We could choose them heuristically, to resolve features of the right hand side and expected features of the solution (for instance by using a geometric grading of the mesh close to a corner to resolve singularities, or by using points-per-wavelength rules for the choice of degrees). Lastly, we could design error estimators and refinement schemes to develop adaptive algorithms which automatically choose those degrees starting from a given initial discretization. Here, for the sake of brevity and simplicity, we will consider the mesh and the degrees as given.

8.3 Subassembling vector field spectral elements

In this section we will explain how to subassemble vector field spectral elements under different continuity conditions. Even though we apply it here to (8.5) and (8.6), the derivation applies to any system of that form, for instance, also to the subassembly of discretizations of $\alpha Id + \beta\, \mathbf{grad}\, \mathsf{div}$ in $H(\mathsf{div})$.

The subassembly procedures given in this section can be easily generalized to the case of

three or more dimensions with different continuity conditions on different components.

8.3.1 Enforcing continuity in tangential components

We assume $\underline{f}_3 \equiv 0$ in the following computations. They can be easily extended to include that term.

Assume that there is a rectangular domain Ω split into $N_1 \times N_2$ rectangular elements. On each of the elements Ω_{ij} ($i = 1, \ldots, N_1$, $j = 1, \ldots, N_2$) we use a local space $\mathbb{Q}_{m^1_{ij}, n^1_{ij}} \times \mathbb{Q}_{m^2_{ij}, n^2_{ij}}$. Since tangential components have to match, the first component has to match in the x-direction and the second component has to match in the y-direction with adjacent subdomains. That implies $m^1_{ij} = m^1_{ik} =: m^1_i$ and $n^2_{ij} = n^2_{kj} =: n^2_j$. A priori the choice of n^1_{ij} and m^2_{ij} is not restricted by the matching conditions, and they should be chosen such that the local discretization is accurate enough but not too expensive.

If we want to obtain a linear system of equations with a tensor product structure like (8.5) and (8.6), we have to chose the n^1_{ij} and m^2_{ij} so that they also match across the domain, i.e., $n^1_{ij} =: n^1_j$ and $m^2_{ij} =: m^2_i$. In the following we will work with such a choice, and Ω_{ij} therefore has as local space

$$\mathbb{Q}_{m^1_i, n^1_j} \times \mathbb{Q}_{m^2_i, n^2_j}$$

On each of the elements Ω_{ij} we have contributions like (8.5) and (8.6):

$$(M^x_{1,i} \otimes A^y_j)\underline{u}_1 + (B^x_i \otimes C^y_j)\underline{u}_2 = (M^x_{1,i} \otimes M^y_{1,j})\underline{f}_1 \tag{8.9}$$

$$(C^x_i \otimes B^y_j)\underline{u}_1 + (A^x_i \otimes M^y_{2,j})\underline{u}_2 = (M^x_{2,i} \otimes M^y_{2,j})\underline{f}_2 \tag{8.10}$$

Now the solution on the rectangular domain is given as two two-dimensional arrays

$$u_1 \in \mathbb{R}^{(\sum_{k=1}^{N_1}(m^1_k+1))\times((\sum_{l=1}^{N_2} n^1_l)+1)} \quad \text{and} \quad u_2 \in \mathbb{R}^{((\sum_{k=1}^{N_1} m^2_k)+1)\times(\sum_{l=1}^{N_2}(n^2_l+1))}$$

First we need to define two different types of one-dimensional restriction operators, S for the directions without enforced continuity and R for those with continuity,

$$v^i = S^x_i v \qquad v^i_l = v_{\sum_{k=1}^{i-1}(m^1_k+1)+l} \qquad \text{for } l = 1, \ldots, m^1_i + 1 \tag{8.11}$$

$$v^j = R^y_j v \qquad v^j_l = v_{(\sum_{k=1}^{j-1} n^1_k)+l} \qquad \text{for } l = 1, \ldots, n^1_j + 1 \tag{8.12}$$

$$v^i = R^x_i v \qquad v^i_l = v_{(\sum_{k=1}^{i-1} m^2_k)+l} \qquad \text{for } l = 1, \ldots, m^2_i + 1 \tag{8.13}$$

$$v^j = S^y_j v \qquad v^j_l = v_{\sum_{k=1}^{j-1}(n^2_k+1)+l} \qquad \text{for } l = 1, \ldots, n^2_j + 1 \tag{8.14}$$

Using the restriction operators just defined, we obtain the values on the element Ω_{ij} from the global arrays u_1 and u_2 as follows:

$$u^{ij}_1 = (S^x_i \otimes R^y_j)u_1 \qquad u^{ij}_2 = (R^x_i \otimes S^y_j)u_2$$

To add a contribution $\mathbf{v}^{ij} = (v_1^{ij}, v_2^{ij})$ to the global array $\mathbf{v} = (v_1, v_2)$, we have to compute

$$v_1' = v_1 + (S_i^{x,T} \otimes R_j^{y,T}) v_1^{ij} \qquad\qquad v_2' = v_2 + (R_i^{x,T} \otimes S_j^{y,T}) v_2^{ij}$$

Now, subassembling by adding all the contributions of the form (8.9), we obtain for the first equation:

$$\sum_{ij} (S_i^{x,T} \otimes R_j^{y,T}) \left\{ (M_{1,i}^x \otimes A_j^y)(S_i^x \otimes R_j^y) u_1 + (B_i^x \otimes C_j^y)(R_i^x \otimes S_j^y) u_2 \right\}$$

$$= \sum_{ij} (S_i^{x,T} \otimes R_j^{y,T})(M_{1,i}^x \otimes M_{1,j}^y)(S_i^x \otimes R_j^y) f_1$$

After some algebraic manipulations we obtain

$$\left(\left(\sum_i S_i^{x,T} M_{1,i}^x S_i^x \right) \otimes \left(\sum_j R_j^{y,T} A_j^y R_j^y \right) \right) u_1$$

$$+ \left(\left(\sum_i S_i^{x,T} B_i^x R_i^x \right) \otimes \left(\sum_j R_j^{y,T} C_j^y S_j^y \right) \right) u_2$$

$$= \left(\left(\sum_i S_i^{x,T} M_{1,i}^x S_i^x \right) \otimes \left(\sum_j R_j^{y,T} M_{1,j}^y R_j^y \right) \right) f_1$$

and realize that this is still of the form (8.9) resp. (8.5)

$$(\tilde{M}_1^x \otimes \tilde{A}^y) u_1 + (\tilde{B}^x \otimes \tilde{C}^y) u_2 = (\tilde{M}_1^x \otimes \tilde{M}_1^y) f_1 \tag{8.15}$$

if we set

$$\tilde{M}_1^x := \sum_i S_i^{x,T} M_{1,i}^x S_i^x \qquad\qquad \tilde{M}_1^y := \sum_j R_j^{y,T} M_{1,j}^y R_j^y \tag{8.16}$$

$$\tilde{A}^y := \sum_j R_j^{y,T} A_j^y R_j^y \qquad\qquad \tilde{B}^x := \sum_i S_i^{x,T} B_i^x R_i^x \tag{8.17}$$

$$\tilde{C}^y := \sum_j R_j^{y,T} C_j^y S_j^y \tag{8.18}$$

$\tilde{M}_1^x$ is a block-diagonal matrix in which the blocks are the mass matrices from the elements. Both $\tilde{M}_1^y$ and $\tilde{A}^y$ are subassembled one-dimensional mass matrices; and subassembled stiffness matrices for the Helmholtz type operator, respectively.

Similarly the second component is subassembled

$$\sum_{ij}(R_i^{x,T}\otimes S_j^{y,T})\left\{(C_i^x\otimes B_j^y)(S_i^x\otimes R_j^y)u_1+(A_i^x\otimes M_{2,j}^y)(R_i^x\otimes S_j^y)u_2\right\}$$

$$=\sum_{ij}(R_i^{x,T}\otimes S_j^{y,T})(M_{2,i}^x\otimes M_{2,j}^y)(R_i^x\otimes S_j^y)f_2$$

to give

$$(\tilde{C}^x\otimes\tilde{B}^y)u_1+(\tilde{A}^x\otimes\tilde{M}_2^y)u_2=(\tilde{M}_2^x\otimes\tilde{M}_2^y)f_2 \tag{8.19}$$

with

$$\tilde{M}_2^x:=\sum_i R_i^{x,T}M_{2,i}^xR_i^x \qquad \tilde{M}_2^y:=\sum_j S_j^{y,T}M_{2,j}^yS_j^y \tag{8.20}$$

$$\tilde{A}^x:=\sum_i R_i^{x,T}A_i^xR_i^x \qquad \tilde{C}^x:=\sum_i R_i^{x,T}C_i^xS_i^x \tag{8.21}$$

$$\tilde{B}^y:=\sum_j S_j^{y,T}B_j^yR_j^y \tag{8.22}$$

The equations (8.15) and (8.19) are still a symmetric system of equations, and have the same tensor product structure as (8.9) and (8.10).

8.3.2 Enforcing continuity in all components

There may be circumstances where we want to enforce continuity of all components across element interfaces. We could have a H^1 conforming formulation of a H^1 conforming problem, for instance if we try to approximate vector Laplace or Helmholtz problems, especially with some additional coupling between components. We could use it also to show that H^1 conforming approaches perform worse for $H(\mathrm{curl})$ formulations than $H(\mathrm{curl})$ conforming ones do. Finally, if we try to construct preconditioners for higher-order spectral element discretizations using lower order discretizations defined on the Gauss-Lobatto-Legendre mesh associated to the higher-order spectral element (so-called Deville-Mund preconditioners), it would make sense to enforce total continuity for the lower-order discretization inside the higher-order elements and impose tangential continuity only across interfaces of the higher-order elements, since that would correspond to the continuity conditions in the higher-order spectral element spaces, and also simplify the mapping of degrees of freedom between higher-order and lower-order space.

To derive the form of the system that we obtain when we subassemble (8.9) and (8.10), and to enforce the continuity of all components across interfaces, we follow the above

derivation, but change all the restriction operators S into R to obtain

$$(\overline{M}_1^x \otimes \overline{A}^y)u_1 + (\overline{B}^x \otimes \overline{C}^y)u_2 = (\overline{M}_1^x \otimes \overline{M}_1^y)f_1 \quad (8.23)$$
$$(\overline{C}^x \otimes \overline{B}^y)u_1 + (\overline{A}^x \otimes \overline{M}_2^y)u_2 = (\overline{M}_2^x \otimes \overline{M}_2^y)f_2 \quad (8.24)$$

with

$$\overline{M}_1^x := \sum_i R_i^{x,T} M_{1,i}^x R_i^x \qquad \overline{M}_1^y := \sum_j R_j^{y,T} M_{1,j}^y R_j^y \quad (8.25)$$

$$\overline{A}^y := \sum_j R_j^{y,T} A_j^y R_j^y \qquad \overline{B}^x := \sum_i R_i^{x,T} B_i^x R_i^x \quad (8.26)$$

$$\overline{C}^y := \sum_j R_j^{y,T} C_j^y R_j^y \quad (8.27)$$

$$\overline{M}_2^x := \sum_i R_i^{x,T} M_{2,i}^x R_i^x \qquad \overline{M}_2^y := \sum_j R_j^{y,T} M_{2,j}^y R_j^y \quad (8.28)$$

$$\overline{A}^x := \sum_i R_i^{x,T} A_i^x R_i^x \qquad \overline{C}^x := \sum_i R_i^{x,T} C_i^x R_i^x \quad (8.29)$$

$$\overline{B}^y := \sum_j R_j^{y,T} B_j^y R_j^y \quad (8.30)$$

8.3.3 Enforcing continuity in normal components

In the case that the element discretization (8.9) and (8.10) corresponds to a $H(\mathsf{div})$ conforming discretization of a problem in $H(\mathsf{div})$, we have to subassemble the contributions from the elements enforcing continuity of the normal component across the interfaces. We obtain the subassembled system by following the derivation of the first subsection and exchanging all S and R:

$$(\check{M}_1^x \otimes \check{A}^y)u_1 + (\check{B}^x \otimes \check{C}^y)u_2 = (\check{M}_1^x \otimes \check{M}_1^y)f_1 \quad (8.31)$$
$$(\check{C}^x \otimes \check{B}^y)u_1 + (\check{A}^x \otimes \check{M}_2^y)u_2 = (\check{M}_2^x \otimes \check{M}_2^y)f_2 \quad (8.32)$$

with

$$\check{M}_1^x := \sum_i R_i^{x,T} M_{1,i}^x R_i^x \qquad \check{M}_1^y := \sum_j S_j^{y,T} M_{1,j}^y S_j^y \quad (8.33)$$

$$\check{A}^y := \sum_j S_j^{y,T} A_j^y S_j^y \qquad \check{B}^x := \sum_i R_i^{x,T} B_i^x S_i^x \quad (8.34)$$

$$\check{C}^y := \sum_j S_j^{y,T} C_j^y R_j^y \quad (8.35)$$

$$\check{M}_2^x := \sum_i S_i^{x,T} M_{2,i}^x S_i^x \qquad \check{M}_2^y := \sum_j R_j^{y,T} M_{2,j}^y R_j^y \tag{8.36}$$

$$\check{A}^x := \sum_i S_i^{x,T} A_i^x S_i^x \qquad \check{C}^x := \sum_i S_i^{x,T} C_i^x R_i^x \tag{8.37}$$

$$\check{B}^y := \sum_j R_j^{y,T} B_j^y S_j^y \tag{8.38}$$

8.4 Enforcing boundary conditions

If we solve the systems (8.5) and (8.6) or (8.15) and (8.19) as they are, we will solve a problem with natural boundary conditions. The natural boundary conditions for the model problem are that the tangential components of $\beta\,\mathbf{curl}$ of the solution are equal to zero. (See chapter 3, especially section 3.2.3.) If β is a scalar function this is equivalent to the vanishing of the tangential components of $\mathbf{curl}\,\mathbf{u}$ on the boundary. In the two-dimensional case, $\mathrm{curl}\,\mathbf{u}$ is a scalar, so that $\mathrm{curl}\,\mathbf{u} = 0$ on the boundary.

If we have inhomogenous natural boundary conditions on a part of the boundary, say Γ_{NBC}, then we have to subtract a boundary integral from the right hand side of the variational formulation. If

$$\gamma_t(\beta\,\mathbf{curl}\,\mathbf{u})|_{\Gamma_{NBC}} = g_{NBC}$$

then the additional boundary term is

$$-\int_{\Gamma_{NBC}} g_{NBC}\gamma_t(\mathbf{v}) \tag{8.39}$$

On a rectangular geometry aligned with the coordinate axes, $\gamma_t(\mathbf{v})$ is always one of the components (in the two-dimensional case) or two of the components (in the three-dimensional case). So the boundary integral turns into an integral of one of components with g_{NBC} or of the inner product of two of the components with the vector function g_{NBC}. In the two-dimensional case we can discretize it exactly like the boundary integral for inhomogenous Neumann boundary conditions in section 6.1. In the three-dimensional case we obtain a discretization by Gaussian quadrature on the boundary in a similar way.

If we solve the essential boundary value problem, i.e., pose (8.1) in $H_0(\mathbf{curl})$, then we have to force the tangential degrees of freedom on the boundary to be zero. Algorithmically, we pass to the non-tangential part of the system (8.5) and (8.6) or (8.15) and (8.19) and solve it exactly as we solved it in the case for natural boundary conditions. Taking the non-tangential part corresponds to a restriction

$$(u_{NT}^1, u_{NT}^2) = ((I^x \otimes R_I^y)u_1, (R_I^x \otimes I^y)u_2)$$

with I denoting the identity in the appropriate direction and R_I the restriction to the interior part (everything except the first and the last component of the vector). The system for the non-tangential part has again the form of (8.5) and (8.6), only that the matrices A^y and A^x are replaced by their principal minor involving only the interior, and that the matrices B^x, B^y, C^x, and C^y are replaced by submatrices missing the first and last row or column.

Nonhomogenous tangential boundary conditions are treated similarly to Dirichlet boundary conditions for the Helmholtz type problem from chapter 6. We first perform a lifting of the tangential boundary conditions – we usually take the nodal interpolant of the boundary conditions which seems to be working satisfactorily, but we could as well use one with smaller maximal gradient – and then use the lifting to correct the right hand side of the discrete problem, and reduce it to a problem with zero tangential boundary conditions which we solve as described in the last paragraph.

All these boundary conditions still preserve the tensor product structure of the system, since they only change the right hand side or correspond to taking submatrices.

The Silver-Müller boundary condition, imposed on a part Γ_A of the boundary , corresponds to the addition of a term of the form

$$\int_{\Gamma_A} \rho \gamma_t(\mathbf{u}) \gamma_t(\mathbf{v}) \tag{8.40}$$

to the bilinear form on the left hand side. We will only be able to write the system in the form (8.5) and (8.6) for special forms of ρ, such as for constant ρ.

To give an indication of how such a problem with constant ρ is solved, we will explain the idea in a special case without working out all the details in the general setting.

Assume that $\Omega = [-1,1]^2$ is discretized by one spectral element, and therefore we have a system (8.5) and (8.6) with the matrices given after (8.4). Let Γ_A be $[-1,1] \times -1$. On Γ_A, $\gamma_t(\mathbf{u})$ and $\gamma_t(\mathbf{v})$ are u_1 and v_1, respectively. We discretize (8.40) and add it to (8.5), after removing the $\underline{v}_1^T$. (e_1^y denotes the vector $(1,0,\ldots,0)$ of length n_1.)

$$\begin{aligned} \int_{\Gamma_A} \rho \gamma_t(\mathbf{u}) \gamma_t(\mathbf{v}) &= \int_{-1}^{1} \rho u_1|_{y=-1} v_1|_{y=-1} dx \approx \rho \underline{v_1|_{y=-1}}^T M_1^x \underline{u_1|_{y=-1}} \\ &= [(I_1^x \otimes e_1^y)\underline{v}_1]^T [M_1^x \otimes \rho][(I_1^x \otimes e_1^y)\underline{u}_1] = \underline{v}_1^T (M_1^x \otimes \rho e_1^{y,T} e_1^y)\underline{u}_1 \end{aligned} \tag{8.41}$$

to obtain a system (8.5') and (8.6), where the only change is that A^y has been replaced by $A^{y\prime} = A^y + \rho e_1^{y,T} e_1^y$. The system (8.5') and (8.6) can be solved exactly like (8.5) and (8.6).

In the case of arbitrary ρ, we can split the variables u of the system (8.15) and (8.19) which we will denote $K\mathbf{u} = M\mathbf{f}$ into two vectors u_A and u_I corresponding to the tangential components of $\mathbf{u}$ on Γ_A and the rest, respectively, and obtain the system:

$$\begin{pmatrix} K_{AA} & K_{AI} \\ K_{IA} & K_{II} \end{pmatrix} \begin{pmatrix} u_A \\ u_I \end{pmatrix} = \begin{pmatrix} M_{AA} & M_{AI} \\ M_{IA} & M_{II} \end{pmatrix} \begin{pmatrix} f_A \\ f_I \end{pmatrix} =: \begin{pmatrix} \tilde{f}_A \\ \tilde{f}_I \end{pmatrix} \tag{8.42}$$

We can reduce the solution of this system to the solution of the Schur complement system:

$$\left(K_{AA} - K_{AI}K_{II}^{-1}K_{IA}\right) u_A = \tilde{f}_A - K_{AI}K_{II}^{-1}\tilde{f}_I \tag{8.43}$$

followed by the solution of the tangential boundary value problem:

$$K_{II}u_I = \tilde{f}_I - K_{IA}u_A \tag{8.44}$$

$K_{II}^{-1}v_I$ can be computed fast by our direct solvers for the tangential boundary value problem.

$S_I = -K_{AI}K_{II}^{-1}K_{IA}$ can be constructed by as many tangential boundary value problem solves as there are mesh points (n_A) on Γ_A.

In this way we can construct $S_A = K_{II} + S_I$ and $f_S = \tilde{f}_A - K_{AI}K_{II}^{-1}\tilde{f}_I$ by $n_A + 1$ solves of a tangential boundary value problem.

$S_A u_A = f_S$ can then be solved by a direct solver. If we have uniform degree $m_1 = n_1 = m_2 = n_2 = N$, this system is of size $cN \times cN$ instead of $2N^2 \times 2N^2$, since u_A discretizes the solution on a manifold of lower dimension than u_I. u_I is then computed by one more tangential boundary value problem solve.

We will consider both the tensor product solvers for problems with Silver-Müller boundary conditions for constant ρ and the Schur complement approach for arbitrary ρ in future work.

Chapter 9

Fast direct solvers for tensor product systems

In this chapter, we will present fast solvers that take advantage of the tensor product structure of the discretizations. Discretizing the scalar Poisson or Helmholtz problem (or any separable problem for that matter) on a rectangular domain, as in chapter 6, yields a sum of d tensor products matrices for a d-dimensional problem. Discretizing the Maxwell model problem (and similar problems) in $H(\mathbf{curl})$ on a rectangular geometry, as in chapter 8, leads to a block tensor product matrix.

First we present a short introduction into tensor product matrices, operations on them, and efficient implementations of such operations in the first section. The second section presents sum of tensor product discretizations and their solution. We give the general form of discretizations to which the method can be applied, and discuss some ways to actually implement the solution algorithm. In the third section we discuss the block tensor product matrix case, which is of use in the solution of vector field problems, and is here applied to the solution of the Maxwell model problem on a rectangular domain. In direct substructuring and iterative substructuring methods we solve a Schur complement system involving only the shared, tangential, components on the interface. In the fourth section we discuss how to apply the local Schur complement, its inverse, and the global Schur complement to a vector. We also describe the subassembly and direct solution of the Schur complement system. We close the chapter with a section presenting some numerical examples for some of the methods introduced in this chapter.

9.1 Tensor product matrices

We denote the tensor product of d matrices A_i of size $n_i \times n_i$ as follows:

$$T = (\otimes_{i=1}^{d} A_i)$$

It is the matrix T with the entries

$$T(j(k_1, \ldots, k_d), j(m_1, \ldots, m_d)) = \prod_{l=1}^{d} A_l(k_l, m_l)$$

where $j(\cdot)$ is the mapping from the index in the d-dimensional grid of size $(n_1, n_2, \ldots n_d)$, containing in total $N = \prod_{l=1}^{d} n_l$ grid points, to the index in the vector.

We define the mappings j and J between vectors of dimension $\prod_{l=1}^{d} n_l$ and d dimensional arrays as

$$J[U](j(k_1, \ldots, k_d)) = U(k_1, \ldots, k_d)$$

and

$$j[u](k_1, \ldots, k_d) = u(j(k_1, \ldots, k_d))$$

A matrix-vector multiplication of a tensor product matrix $\otimes_{i=1}^{d} A_i$ with a vector u representing a function on a regular d dimensional grid of size $(n_1, n_2, \ldots, n_d)$ can be implemented by representing the vector u as a d dimensional array $j[u]$ and multiplying the array along dimension i with the $n_i \times n_i$ matrix A_i. In restriction and prolongation operators the matrices A_i can also be rectangular matrices instead of square matrices.

For instance, in the case of two dimensions, we can write with $U = j[u]$:

$$(A \otimes B)u = AUB^T$$

In the way just explained, multiplication with a tensor product matrix can be implemented in $O((\sum_{l=1}^{d} n_i) \prod_{l=1}^{d} n_i) = O((\sum_{l=1}^{d} n_i)N)$ with a standard matrix-matrix multiplication, instead of the $O(N^2)$ needed for a general matrix of the same size. We can reduce the operation count further by using a fast matrix-matrix multiplication.[1]

Assuming that a matrix-matrix multiplication of two $n \times n$ matrices needs $O(n^\alpha)$ time[2], and that the multiplication of a $n \times n$ with a $n \times m$ matrix takes $O(mn^\beta)$ time[3], the

[1] See Golub and Van Loan [49] for an introduction to matrix computations; Strassen [92] or Coppersmith and Winograd [29] for original algorithms for square matrices; Knight [63] or Huang and Pan [61] for algorithms for rectangular matrices. See also [65, 81, 64, 80, 79].

[2] α is smaller or equal 2.376, see Coppersmith and Winograd [29], $\alpha = 2$ or $\alpha > 2$ is the subject of a bet between Trefethen and Alfeld, see `http://www.math.utah.edu/~alfeld/bet.html`.

[3] $\beta = \alpha - 1$ is given for the special case $m = n^r$ with r a rational number in Huang and Pan [61], among other results. $\beta = \alpha - 1$ is especially true for $m = n$, for instance in the case of an uniform number of grid points in all directions, $m = n = n_i$. In this case all statements in the following involving $\beta - 1$ should be read $\alpha - 2$.

multiplication by a tensor product matrix utilizing fast matrix-matrix multiplication takes $O(\sum_{k=1}^{d} n_k^{\beta-1} N)$ time. In the best possible case, if α should turn out to be 2 (and $\beta = 1$), this would reduce to $O(dN)$. If the factors A_i in the tensor product have additional structure, the complexity can be even further reduced. (See, e.g., Buis and Dyksen [23], and references therein.)

The inverse of a tensor product matrix is the tensor product of the inverses of the tensor product factors. If the inverses of the factors are available, or can be computed easily, the inverse can be applied as a tensor product. Even when the explicit computation of the inverse is more time-intensive, it is likely, especially in higher-dimensional cases, that its computation will be of lower complexity than the other steps in the algorithms.

If the inverses are not available, or it would be too expensive to form them, we can use the idea from de Boor [40, 39] to implement the inverse of the tensor product matrix using solvers (with multiple right hand sides) for the problems $A_i x_i = b_i$.

For further discussions about implementation and use of tensor product matrices, see [23, 40, 39, 83].

9.2 Sums of tensor product matrices: solving scalar problems

Many finite difference discretizations of partial differential equations of the form

$$Lu = \sum_{i=1}^{d} P_i(\frac{\partial}{\partial x_i})u = f \tag{9.1}$$

can be written in a form $L_h u_h = f_h$ with

$$L_h = \sum_{i=1}^{d} (\otimes_{j=1}^{i-1} I_j) \otimes L_i \otimes (\otimes_{j=i+1}^{d} I_j). \tag{9.2}$$

Here, L_i is a $n_i \times n_i$ matrix related to the discretization of $P_i(\frac{\partial}{\partial x_i})$, and I_i is the identity matrix for the ith coordinate direction.

Also many finite element or spectral element discretizations of (9.1) for rectangular meshes can be written in a similar form (if tensor product basis functions and tensor product numerical integration are used):

$$L_h = \sum_{i=1}^{d} (\otimes_{j=1}^{i-1} M_j) \otimes K_i \otimes (\otimes_{j=i+1}^{d} M_j). \tag{9.3}$$

Here, L_i is a $n_i \times n_i$ matrix related to the discretization of an one-dimensional variational problem involving $P_i(\frac{\partial}{\partial x_i})$ – an one-dimensional stiffness matrix – and M_i is an approximation for the integration operator – an one-dimensional mass matrix.

(9.3) can be transformed into (9.2) by multiplying both sides of $L_h u_h = f_h$ with the tensor product of the inverses of the one-dimensional mass matrices. We will give references for methods for the solution of (9.2) and for the solution of (9.3). Many of the algorithms are given for the two-dimensional or three-dimensional case in the literature. The two-dimensional case corresponds to the matrix equations (with $U = j[u]$):

$$A_T U + U B_T^T = F_T \qquad \text{respective} \qquad A_T U B_T^T + C_T U D_T^T = F_T.$$

These equations are also known as *Sylvester matrix equations*.

Several fast solvers for this system are transform methods, i.e., they multiply the system $L_h u = f_h$ with a judiciously chosen tensor product matrix such that the resulting system is of a special form and can be solved very efficiently. For an introduction to some of such methods, see, e.g., Canuto, Hussaini, Quarteroni, and Zang [24, section 5.1], or Gardiner, Laub, Amato, and Moler [47].

The one we chose to implement is the algorithm of Lynch, Rice, and Thomas [67] (or its generalization), also called the *fast diagonalization method*. It consists of diagonalizing all the non-identity factors in the form (9.2). It has the advantage of being easy to implement, and it generalizes to an arbitrary number of dimensions.

Using this algorithm, the matrix L_h from (9.2) can be inverted in the following way

$$L_h^{-1} = (\otimes_{i=1}^d P_{x_i}) \left(\sum_{i=1}^d (\otimes_{j=1}^{i-1} I_{x_j}) \otimes \Lambda_{x_i} \otimes (\otimes_{j=i+1}^d I_{x_j}) \right)^{-1} (\otimes_{i=1}^d P_{x_i}^{-1})$$

where

$$L_{x_i} P_{x_i} = P_{x_i} \Lambda_{x_i}$$

is the diagonalization of L_{x_i} to Λ_{x_i}, i.e., its spectral decomposition[4].

The inverse of the middle factor of the product in L_h^{-1} corresponds to a diagonal scaling in each direction. Multiplication with the middle factor corresponds on the level of d-dimensional arrays to a component-wise multiplication of $U = j[u]$ by another array of the same size. (We can also apply other matrix operators besides the inverse defined by a functional calculus on the eigenvalues of the matrix in this way.) The complexity of this step is $O(\prod_{k=1}^d n_k) = O(N)$ since it requires exactly one multiplication per variable.

[4]Using the QR algorithm from Golub and Van Loan [49, section 7.5.6], we need $O(\sum_i n_i^3)$ to compute that, which in the case of equal size in all directions simplifies to $O(dn^3)$. This is a lower order term for $d \geq 3$; for $d = 2$ we need to do at least $O(n)$ solves to amortize this set-up cost.

The first and the third factor are tensor products. If we use the fast matrix-matrix multiplication method to compute the product of the factors and a vector, we need $O(\sum_{k=1}^{d} n_k^{\beta-1} N)$ time to apply the tensor products. This is the dominant factor in the complexity for $\beta > 1$, and it is of the same magnitude for $\beta = 1$ as the component wise multiplication.

Therefore, if $\alpha = 2$ and $\beta = 1$, the fast diagonalization method is of quasi-optimal complexity $O(dN)$ if it uses that fast matrix multiplication method. For uniform number of points and straightforward matrix multiplication we loose a power in n and obtain an algorithm of complexity $O(dn^{d+1}) = O(dnN)$. This algorithm is attractive especially for higher-dimensional problems.

Instead of diagonalizing all the matrices L_i in the sum of tensor product discretization, we could diagonalize all except one and solve the remaining decoupled one-dimensional problems. In some cases, especially in two dimensions, that results in lower computational cost, see Canuto, Hussaini, Quarteroni, and Zang [24, pages 135–136] and Zang and Haidvogel [106].

Explicit transformation to the eigenbasis may be unstable, if the eigensystem is ill-conditioned. In this case, a transformation to Schur forms is more stable, see Bartels and Stewart [13]. Since we did not observe instabilities or reductions in accuracy in our tests, we used the fast diagonalization method.

We solve the spectral element system by transforming it first into the form (9.2) by multiplying by the inverse of the mass matrix, and using the fast diagonalization method. Alternative, possibly more stable algorithms and implementations for the two-dimensional case (i.e., the Sylvester matrix equations) are described in Gardiner, Laub, Amato, and Moler [47] and Kågström and Poromaa [62].

For a fuller discussions of the issues involved in the implementation and the choice between the different algorithms, see the papers cited above and references therein.

We could describe here also for the scalar case the fast application, construction, and subassembly of Schur complement systems with respect to the interfaces that we present in the fourth section for the vector field case. Since the implementation is straightforward and very similar to (and easier than) the case discussed in the fourth section, we will not do so for conciseness.

9.3 Block tensor product matrices: Solving vector field problems

In this section we describe fast direct solvers for the systems (8.5), (8.6); (8.7), (8.8); and (8.15), (8.19). (8.5) and (8.6) are (with the matrices defined after (8.4)):

$$\begin{aligned}(M_1^x \otimes A^y)\underline{u}_1 + (B^x \otimes C^y)\underline{u}_2 &= (M_1^x \otimes M_1^y)\underline{f}_1 - (F_1^x \otimes F_1^y)\underline{f}_3 \qquad (8.5)\\ (C^x \otimes B^y)\underline{u}_1 + (A^x \otimes M_2^y)\underline{u}_2 &= (M_2^x \otimes M_2^y)\underline{f}_2 + (F_2^x \otimes F_2^y)\underline{f}_3 \qquad (8.6)\end{aligned}$$

(8.15) and (8.19) are of the same form, only with different matrices as defined in (8.16)-(8.18) and (8.20)-(8.22).

(8.7) and (8.8) are

$$\begin{aligned}(I_1^x \otimes \mathcal{A}^y)\underline{u}_1 + (\mathcal{B}^x \otimes \mathcal{C}^y)\underline{u}_2 &= \underline{f}_1 - (\mathcal{F}_1^x \otimes \mathcal{F}_1^y)\underline{f}_3 \qquad (8.7)\\ (\mathcal{C}^x \otimes \mathcal{B}^y)\underline{u}_1 + (\mathcal{A}^x \otimes I_2^y)\underline{u}_2 &= \underline{f}_2 + (\mathcal{F}_2^x \otimes \mathcal{F}_2^y)\underline{f}_3 \qquad (8.8)\end{aligned}$$

We will only discuss the solution of (8.5) and (8.6) in the following form

$$\begin{aligned}(M_1^x \otimes A^y)u_1 + (B^x \otimes C^y)u_2 &= g_1 \qquad (9.4)\\ (C^x \otimes B^y)u_1 + (A^x \otimes M_2^y)u_2 &= g_2 \qquad (9.5)\end{aligned}$$

since (8.7) and (8.8) correspond to a specific choice of M_1^x and M_2^y in (8.5) and (8.6), and the exact form of the right hand side does not matter in the proposed algorithm.

We reduce (9.4) and (9.5) to a system in u_1, by solving the second equation (9.5) for u_2, and substituting the result into (9.4):

$$u_2 = ((A^x)^{-1} \otimes (M_2^y)^{-1})(g_2 - (C^x \otimes B^y)u_1) \qquad (9.6)$$

$$(M_1^x \otimes A^y - (B^x(A^x)^{-1}C^x) \otimes (C^y(M_2^y)^{-1}B^y))u_1 = \tilde{g}_1 \qquad (9.7)$$

$$\text{with:} \qquad \tilde{g}_1 = g_1 - ((B^x(A^x)^{-1}) \otimes (C^y(M_2^y)^{-1}))g_2 \qquad (9.8)$$

We could use the methods mentioned in the last section for Sylvester matrix equations to solve the system

$$(A_T \otimes B_T + C_T \otimes D_T)u_1 = \tilde{g}_1 \qquad (9.9)$$

with

$$A_T = M_1^x \qquad B_T = A^y \qquad C_T = -(B^x(A^x)^{-1}C^x) \qquad D_T = (C^y(M_2^y)^{-1}B^y) \qquad (9.10)$$

We chose to reduce (9.9) to the form (9.2) and then use the fast diagonalization method. This reduction is effected by premultiplying (9.9) with[5] $(A_T^{-1} \otimes D_T^{-1})$:

$$(I \otimes (D_T^{-1} B_T) + (A_T^{-1} C_T) \otimes I) u_1 = (A_T^{-1} \otimes D_T^{-1}) \tilde{g}_1 \tag{9.11}$$

For the case of the subassembled system (8.15) and (8.19) with (8.16)-(8.18) and (8.20)-(8.22), we can find a simplified, subassembled form of D_T:

$$\begin{aligned} D_T &= C^y (M_2^y)^{-1} B^y \\ &= \left(\sum_j R_j^{y,T} C_j^y S_j^y \right) \left(\sum_j S_j^{y,T} M_{2,j}^y S_j^y \right)^{-1} \left(\sum_j S_j^{y,T} B_j^y R_j^y \right) \\ &= \left(\sum_j R_j^{y,T} C_j^y (M_{2,j}^y)^{-1} B_j^y R_j^y \right) \end{aligned}$$

since different S_j^y have non-overlapping support. There are no such simplifications in the other matrices in (9.11).

Now (9.11) can be solved with the fast diagonalization method from the last section to obtain u_1. We can then use (9.6) to obtain u_2 at the cost of two more tensor product matrix vector applications, if we store $(A^x)^{-1}$ and $(M_2^y)^{-1}$. The second is a block diagonal matrix, and is therefore easily inverted, the first one is a discretized Helmholtz operator for which we already have computed the spectral decomposition in the setup of the fast diagonalization method, so that we can easily form its inverse.

We have implemented this method and will present some examples and timings in the last section.

In a few numerical tests, we observed almost singular matrices of eigenvectors in our chosen set-up for the fast diagonalization method. For those cases, an alternative reduction to the form (9.2) seems to lead to a stable solution algorithm. We intend to implement a generalized Sylvester equation solver in future work as a slower, but more stable alternative.

9.4 Direct and iterative substructuring methods

In certain circumstances it is preferable to work on the system of the interface variables (i.e., variables that are shared across element interfaces). For instance, in a direct solution

[5] $(C_T^{-1} \otimes B_T^{-1})$ results in the same form with the inverse of the matrices in the sum of tensor product form (9.11). In the case of the reduced system (8.7) and (8.8), $A_T = A_T^{-1} = I_1^x$ and therefore we prefer the choice made in the text.

of the model problem on non-rectangular domains, it is advantageous, both in terms of computing time and needed memomory, to reduce the large global system to the system on the interface. This is standard practice for p-version finite element methods and is called in that context *static condensation*.

For large systems, a recursive application of this idea is possible and has been widely used in the engineering community, especially in structural analysis, under the name of *(direct) substructing*. In it one introduces several levels of super-elements, and the large domain is split into a few (tens to hundreds) highest-level substructures. The interior variables of elements, and then, recursively, the interior variables of the super-elements are eliminated. The (small) global system on the interface of the highest-level substructures is solved directly, and the local solution is found by backsolving in the super-elements and local solves on the element level (see, e.g., Smith, Bjørstad, and Gropp [91, section 4.1] or Przemieniecki [84]).

We will present numerical results for a direct solver for the Schur complement system on the element interfaces in the next section. We could use the fast direct tensor product solvers from the last section on rectangular superelements, or higher levels of substructuring by easy extensions of our algorithms and implementations. We will explain later in this section how to subassemble the Schur complement system for the interfaces.

There are also iterative substructuring methods that try to solve the Schur complement system by iterative methods such as Krylov subspace methods. Even though the condition number of the Schur complement is usually much smaller than the condition number of the entire system, the increased computational expense of handling the Schur complement tends to diminish potential savings in iterative methods without preconditioners. Therefore efficient preconditioner for the Schur complement need to be constructed. We will not present such preconditioners in this thesis, we will just discuss the implementation of some of the modules that need to be implemented in such preconditioners and iterative methods.

A system of the form (8.5) and (8.6) can be seen as a system

$$Ku = f_M (:= Mf)$$

with K being a block tensor product matrix, and u being a concatenation of u_1 and u_2. We split u into a vector u_T containing the tangential components on the element interfaces, and into a vector u_I containing the other, interior, variables.

$$\begin{pmatrix} K_{II} & K_{IT} \\ K_{TI} & K_{TT} \end{pmatrix} \begin{pmatrix} u_I \\ u_T \end{pmatrix} = \begin{pmatrix} f_I \\ f_T \end{pmatrix}$$

We will reduce this system to the Schur complement system

$$S_T u_T = f_S \tag{9.12}$$

We need to subassemble S_T from local contributions $S^{(i)}$ and also decide on a layout of u_T in terms of its local contributions $u_T^{(i)}$ (we assume that we have N elements):

$$u_T = \sum_{i=1}^{N} R_i^T u_T^{(i)}$$

The local contributions $S^{(i)}$ come from the element matrices $K^{(i)}$ and $M^{(i)}$ and the load vectors $f_M^{(i)}$. We split these into tangential and interior part:

$$K^{(i)} = \begin{pmatrix} K_{II}^{(i)} & K_{IT}^{(i)} \\ K_{TI}^{(i)} & K_{TT}^{(i)} \end{pmatrix} \qquad f_M^{(i)} = \begin{pmatrix} f_I^{(i)} \\ f_T^{(i)} \end{pmatrix} \qquad u^{(i)} = \begin{pmatrix} u_I^{(i)} \\ u_T^{(i)} \end{pmatrix}$$

We can locally eliminate the variables $u_I^{(i)}$ to obtain as Schur complement of the element matrix:

$$S^{(i)} = K_{TT}^{(i)} - K_{TI}^{(i)}(K_{II}^{(i)})^{-1}K_{TT}^{(i)}$$

Subassembling these local Schur complements, we obtain the global Schur complement matrix S_T:

$$S_T = \sum_{i=1}^{N} R_i^T S^{(i)} R_i$$

Likewise, the right hand side f_S can be subassembled from local contributions:

$$f_S = f_T - \sum_{i=1}^{N} R_i^T K_{TI}^{(i)}(K_{II}^{(i)})^{-1} f_I^{(i)}$$

In iterative substructuring methods we do not need to form S_T or $S^{(i)}$ explicitly, we just need a routine to apply to S_T to a vector u. That can be done in parallel by first computing the local parts of u, $u^{(i)} = R_i u$, appying $S^{(i)}$ on each element $s^{(i)} = S^{(i)} u_i$ and then assembling the results $S_T u = s = \sum_{i=1}^{N} R_i^T s^{(i)}$.

Applying $S^{(i)}$ to a vector $u^{(i)}$ on the element corresponds to three sparse matrix-vector multiplication and the evaluation of $(K_{II}^{(i)})^{-1}v$, which corresponds to the solution of a tangential boundary value problem in the interior of the element.

In some domain decomposition preconditioners such as Neumann-Neumann methods we also need a way to apply $(S^{(i)})^{-1}$ fast. As explained in Smith, Bjørstad, and Gropp [91, section 4.2.1], an inverse of $S^{(i)}$ can be found be factoring $K^{(i)}$ and restricting the result, i.e.,

$$(S^{(i)})^{-1}u = \begin{pmatrix} 0 & I \end{pmatrix}(K^{(i)})^{-1} \begin{pmatrix} 0 \\ I \end{pmatrix} u$$

This corresponds to a local solve involving the entire element matrix $K^{(i)}$. As we saw in chapter 8, this corresponds to the solution of a natural boundary value problem, which can be obtained fast with fast diagonalization methods as explained in the third section of this chapter.

If we intend to solve the Schur complement system for the interface variables directly, we need to explicitly form $S^{(i)}$ and S.

One of the ways to explicitly form $S^{(i)}$ without computations on $n^2 \times n^2$ matrices is to compute $S^{(i)}e_k$ for all the unit vectors on the tangential components on the interface of the element. $S^{(i)}e_k$ could be computed as above with $v = e_k$. We implemented an optimized version that takes advantage of the special form of the right hand side and of the special matrix that we multiply the solution with (that is, $K_{TI}^{(i)}$) to avoid unneeded computations.

To form the global Schur complement system, we first have to obtain a mapping from the local (tangential) variables to the global vector of tangential components, and then we can use a standard subassembly procedure with that index information.

We show the tangential variables for one of the elements schematically in figure 9.1.

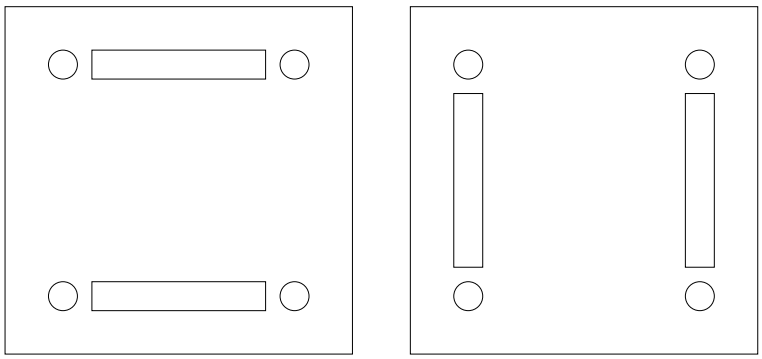


Figure 9.1: Tangential degrees of freedom for one of the elements to be subassembled.

To find the local-to-global mapping, we have to make sure that all the local tangential components are indexed, and in such a way that variables to be identified have the same index. One way to do so is to (arbitrarily; usually geometrically) order the elements, then to iterate over the elements and give indices (in increasing order, or in increasing order for different types of variables, such as corner or interior of edge) to variables that are not indexed yet. If such a variable is shared with other elements, the given index is also entered into the other elements' variable index.

The corner variables are drawn separate from the interior edge in figure 9.1. As discussed in section 7.1.2 on page 60, it is not a priori clear which components should be matched

at corners to obtain the best results. There are three choices: no matching (local element always indexes variable and does not copy index to anywhere else), matching together with the tangential edges they are endpoints of (that means that we treat them as a part of the edge in the indexing, and we only have to keep track if the edge has been indexed or not) or matching all components (treat corner and edges separately, keep track of indexing state of both corners and edges).

Using the index information computed as explained above, represented mathematically as R_i^T, the subassembly of S_T corresponds to adding the element Schur complement matrix $S^{(i)}$ to the appropriate submatrix of S_T. The right hand side f_S can be also be obtained by adding the local (and locally computed) contributions to the appropriate subvector of f_S.

The subassembly procedure has been implemented for the general case. The topological information has to be given locally, element by element. The program will be extended to derive this information from a global geometric description at a later point.

The system (9.12) is then solved directly by Gaussian elimination.

We will show some numerical examples for the Schur complement direct solver introduced above in the next section.

9.5 Numerical experiments

In this section we test both the vector field tensor product solver introduced in section 10.3, and the interface Schur system solver introduced in the last section.

We solve a zero tangential boundary value problem in $[-1,1]^2$ with the exact solution $\mathbf{u} = (\sin(\frac{\pi}{2}y)x, \sin(\frac{\pi}{2}x)y)$ and $\alpha = \beta = 1$.

The domain $[-1,1]^2$ is covered by a uniform mesh of $M \times M$ identical spectral Nédélec elements of degree $N \times N$. We use the discretization given in chapter 8. We chose the integration degrees high enough so that all terms are integrated exactly, unless otherwise noted. We treat corners in the subassembly like the edges they are endpoints of. (The Schur solvers can be easily extended to any of the discussed continuity conditions at the corners. Initial tests did not show any significant differences for different corner conditions.)

We implemented the methods in MATLAB version 6 in a straightforward, modular manner; without attempting to optimize the code. The codes were run on a Ultra 10 workstation with 512 Mb main memory with an UltraSPARC 1 processor running at 440Mhz. The CPU times reported later were obtained by the cputime function provided by matlab. We monitored the running matlab jobs with top, and stopped them when swapping and iowait took more than 90% of the time for extended periods of time. CPU time measurements for jobs dominated by swapping, paging or waiting for IO are extremely unreliant, and the wall

clock time essentially measures the performance of the paging algorithm and of the swap disk.

In figure 9.2[6], we present a comparison of the accuracy of the interface Schur solver and of the vector field tensor product solver, on 5×5 spectral elements, and we vary the degree of the spectral elements. We test both exactly integrated and slighly underintegrated (diagonal) mass matrices. The two solvers perform very similar, the Schur solvers having a slighly higher accuracy for large N. Comparing the results in these figures with the results from chapter 6, especially figures 6.13 and 6.14, we realize that the convergence for the Maxwell model problem is very similar to the convergence of spectral element methods for the Poisson problem.

In figure 9.3, we show the accuracy of the two solvers for local degrees $N \times N$ with $N = 5, 10, 15$, when the number of spectral elements is varied. Because of the higher memory requirements and CPU times dominated by swapping, we report the results for the Schur solver for the 10×10 and 15×15 case only for 1×1 to 10×10 spectral elements. The gap for the Blocktensor 15x15 case at 7 spectral elements stems from a badly conditioned eigensystem in the fast diagonalization solver. A method that performs well also for this case can be obtained by a slight change in the implementation, choosing the setup for the fast diagonalization solver that gives the better conditioned eigensystems, or opting for a solver for the generalized Sylvester matrix equation as described in Gardiner, Laub, Amato, and Moler [47].

The two methods perform very much alike with respect to accuracy. The use of an increasing number of subdomains corresponds to a h-extension and therefore we do not expect exponential convergence. For both the 5×5 and 10×10 case, we observe algebraic convergence, in the latter followed by stagnation after the maximal accuracy of the method is reached. The 15×15 case performs already best for 1 spectral element and has its maximal accuracy there. Since we are already at the maximal accuracy of the method, there is no hope of improved accuracy for larger number of elements, unless other steps to improve accuracy are taken, such as quadruple precision or iterative refinement.

For the higher-degree examples ($N = 10, 15$), we see that the Schur solver yields slightly more accurate solutions. As reported above and below it needs more memory and time for large M and is therefore not competetive for regular decompositions into many spectral elements.

In the next four figures, figures 9.4–9.7, we report some timings of the two solvers. We use exact stiffness, mass, and mixed term matrices.

Figures 9.4 and 9.5 correspond to the cases "Schur" and "Blocktensor" in figure 9.2. Figure 9.4 shows the results for the vector field tensor product solver. We see that most of the time

[6]The figures for this section are given at the end of the chapter.

is spent on subassembling, the time of the actual solve is growing slowly with N, being less than two seconds for degree 50×50. The element matrices are computed in the element-wise setup, which takes an almost neglible time.

In figure 9.5 we show the timings for the interface Schur solver. Here, the most time is spent on the Schur and local solves.[7] The element-wise setup time, which includes the computation of the elementwise Schur complement with respect to the tangential components on the interface, grows rather slowly with N, being less than four seconds for degree 50×50. Subassembling the Schur complement matrix and setting up the Schur complement solver takes the least percentage of the time.

Comparing figures 9.4 and 9.5 we see that for the problem considered, the block tensor product solve is slightly less than two times as fast. Comparing the times excluding the setup times – for instance in local solvers in domain decomposition methods we will only perform the setup once and use only the prepared solver in the iterations – we see that the solve in the block tensor product case is much faster than the forming of the right hand side, the Schur solve and the local solves in the interface Schur solver, at degree 50 we need < 2 seconds for the first, compared with > 14 seconds for the latter.

In figures 9.6 and 9.7 we give the CPU times for the solvers for varying numbers of spectral elements of degree 10×10. We observe very similar behavior in figure 9.6 compared to figure 9.4. Preparing the tensor product solve is the most time-consuming step, the actual solve takes less than one second for 20×20 spectral elements. In figure 9.7, the corresponding figure for the interface Schur solver, the solution of the interface Schur sytem and of the local problems is still the most time-consuming part. We observe that for larger numbers of subdomains the subassembly takes longer than the element-wise setup. This is to be expected, since the elements are all of the same fixed degree, while the Schur complement system grows in size. Comparing again the performance of the block tensor product solver and the interface Schur solver, at $M = 10$, we see that the Schur solver together with set-up is more than twenty times slower than the tensor product solve, the solve step itself is more than a hundred times slower, > 10 seconds against < 0.1 second.

[7] The former could be parallelized using a standard parallel dense linear solver. The local solves are embarrassingly parallel: after the interface values are known, the local solves are completely independent of each other.

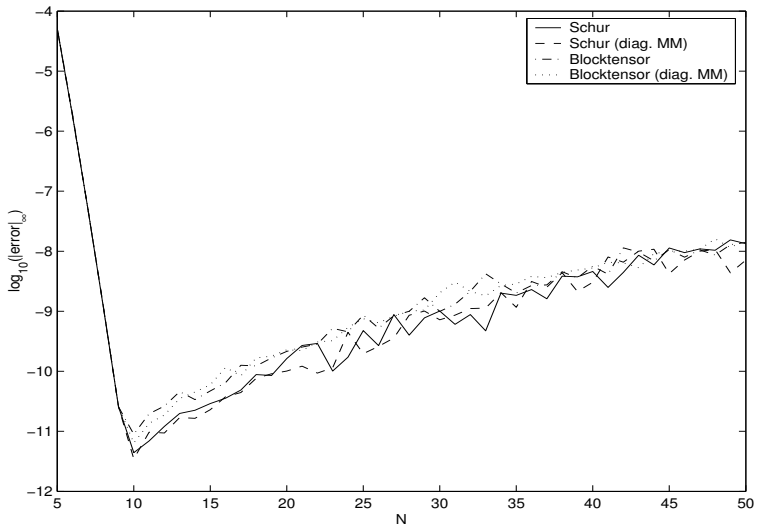


Figure 9.2: Direct solution of $Id + \mathbf{curl}\,\mathbf{curl}$ problems: Comparison between interface Schur solvers and vector field tensor product solvers, 5×5 spectral elements of degree N (Nédélec II).

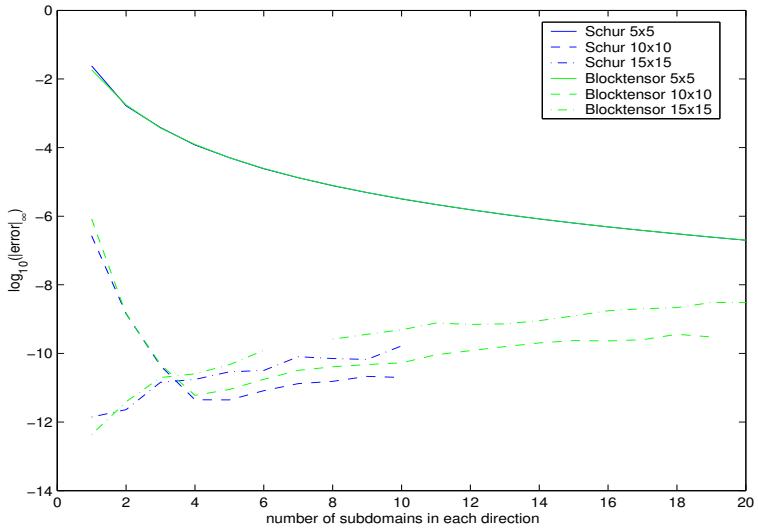


Figure 9.3: Direct solution of $Id + \mathbf{curl}\,\mathbf{curl}$ problems: Comparison between interface Schur solvers and vector field tensor product solvers, varying numbers of spectral elements from 1×1 to 20×20 (Nédélec II).

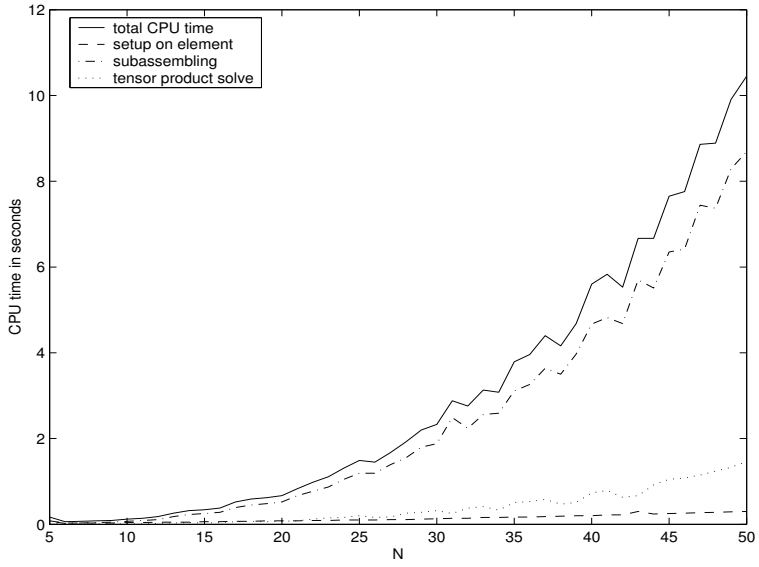


Figure 9.4: Direct solution of $Id + \mathbf{curl}\,\mathbf{curl}$ problems: CPU times for the vector field tensor product solver, 5×5 spectral elements of degree N (Nédélec II).

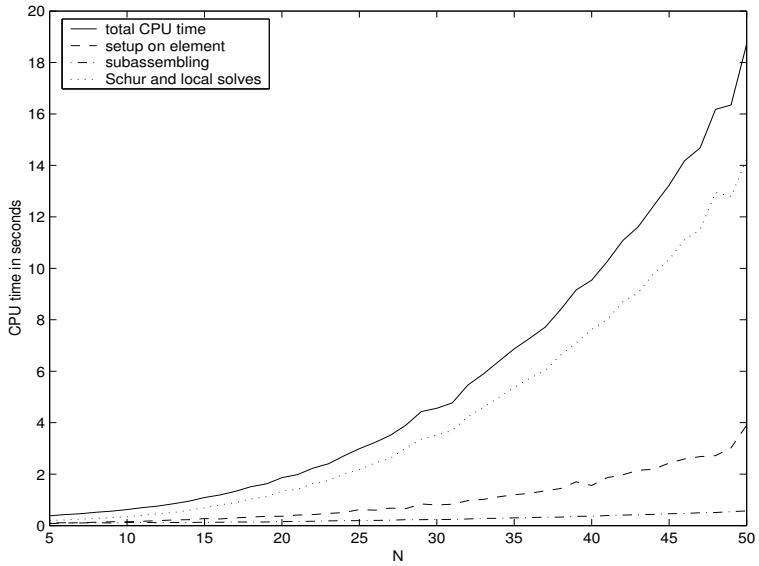


Figure 9.5: Direct solution of $Id + \mathbf{curl}\,\mathbf{curl}$ problems: CPU times for the interface Schur solver, 5×5 spectral elements of degree N (Nédélec II).

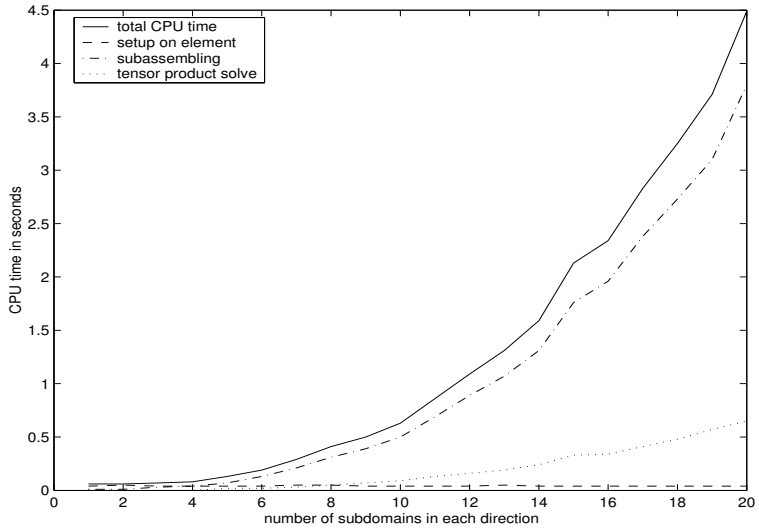


Figure 9.6: Direct solution of $Id + \mathbf{curl}\,\mathbf{curl}$ problems: CPU times for the vector field tensor product solver, Nédélec II elements of degree 10×10, varying numbers of spectral elements, from 1×1 to 20×20.

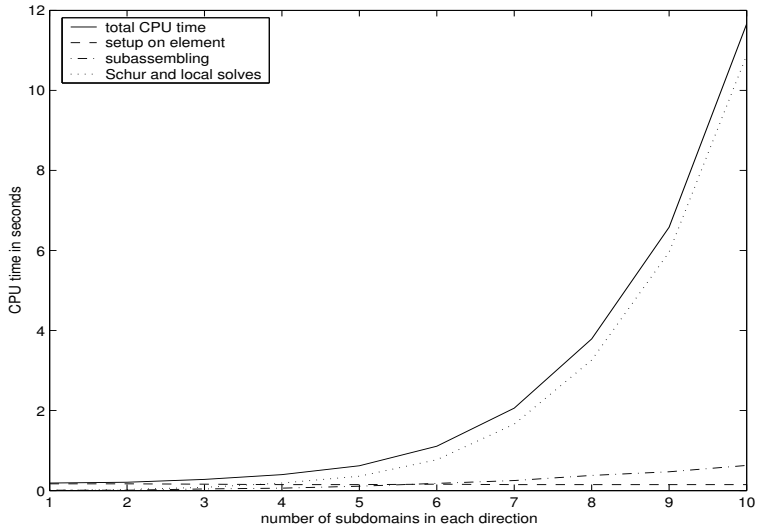


Figure 9.7: Direct solution of $Id + \mathbf{curl}\,\mathbf{curl}$ problems: CPU times for the interface Schur solver, Nédélec II elements of degree 10×10, varying numbers of spectral elements, from 1×1 to 10×10.

Chapter 10

Overlapping Schwarz methods: Implementation and results in two dimensions

In this chapter, we will present implementations of one- and two-level overlapping Schwarz preconditioners for the model problem

$$\eta_1(\mathbf{u},\mathbf{v})_0 + \eta_2(\mathbf{curl}\,\mathbf{u},\mathbf{curl}\,\mathbf{v})_0 = (\mathbf{f},\mathbf{u})_0$$

in two dimensions.

To demonstrate why good preconditioners are needed, we present the results in tables 10.1 and 10.2. They show the behavior of a conjugate gradient method without preconditioner for $\eta_1 = \eta_2 = 1$. In table 10.1 we show how increasing the number of spectral elements $M \times M$ influences the results, and in table 10.2 we show the effect of increasing the local degrees $N \times N$ of the spectral elements (Nédélec II). We report the number of iterations that the conjugate gradient method needed to reduce the l_2-norm of the residual by a factor of $TOL = 10^{-3}$, a condition number estimate obtained from the conjugate gradient parameters α^n and β^n, the maximum error of the last iterate on the associated Gauss-Lobatto-Legendre mesh, and the CPU time.

We see that the number of iterations and the condition number grow very fast with the number of spectral elements and the degree, the condition number reaching about 10^6 for 10×10 spectral elements of degree 10×10.

Obviously, there is much room for improvement. To show that very efficient preconditioners can be constructed, we present table 10.3 with results from one- and two-level preconditioners that we will implement in this chapter. We see that for 10×10 spectral elements,

M	iter	$\kappa_{est}(K)$	$\|\|\mathbf{u}_{iter} - \mathbf{u}^*\|\|_\infty$	t_{CPU} in s
1	53	1.99e+03	6.57e-02	0.2
2	200	3.10e+04	9.23e-02	1.1
3	323	7.17e+04	7.46e-02	2.8
4	424	1.29e+05	8.22e-02	5.8
5	655	2.23e+05	3.03e-02	15.8
6	774	3.23e+05	2.83e-02	26.1
7	916	4.44e+05	2.55e-02	43.8
8	1147	6.14e+05	1.19e-02	82.6
9	1301	7.80e+05	1.07e-02	123.7
10	1441	9.65e+05	1.01e-02	171.1

Table 10.1: Results for cg without preconditioner: $M \times M$ spectral elements of degree 10×10, $TOL = 10^{-3}$.

N	iter	$\kappa_{est}(K)$	$\|\|\mathbf{u}_{iter} - \mathbf{u}^*\|\|_\infty$	t_{CPU} in s
2	11	1.02e+02	9.21e-02	< 0.1
3	45	1.60e+03	1.56e-02	0.2
4	92	9.32e+03	8.28e-03	0.7
5	155	2.10e+04	1.18e-02	1.8
6	249	7.32e+04	1.21e-02	4.1
7	401	1.16e+05	1.23e-02	10.1
8	776	3.17e+05	7.22e-03	35.8
9	965	4.28e+05	1.09e-02	65.6
10	1441	9.65e+05	1.01e-02	169.5

Table 10.2: Results for cg without preconditioner, 10×10 spectral elements of degree $N \times N$, $TOL = 10^{-3.}$

and for a reduction of the l_2-norm of the residual[1] by $TOL = 10^{-6}$, already the one-level method improves the number of iterations from around 3600 to 31, and decreases the CPU time from around 450 seconds to less than 8 seconds; the two-level methods decrease the iteration count further to 15 and the time to less than 4 seconds.

The addition of a second level is paramount to maintaining the performance for large numbers of spectral elements, see table 10.4 in section 10.3. We cite only one pair of examples

[1] We use $TOL = 10^{-6}$ for this and all the following runs in this chapter.

Method	iter	$\kappa_{est}(K)$	$\|\|\mathbf{u}_{iter} - \mathbf{u}^*\|\|_\infty$	t_{CPU} in s
No preconditioner	3580	1.44e+06	5.73e-05	448.6
one-level	31	38.2	3.21e-06	7.6
two-level($N_0 = 2$)	15	4.93	3.78e-06	3.8
two-level($N_0 = 3$)	15	4.52	9.95e-07	3.8
two-level($N_0 = 4$)	15	4.51	9.48e-07	3.9
two-level($N_0 = 5$)	14	4.49	1.88e-06	3.8

Table 10.3: Comparison of different methods for $\eta_1 = \eta_2 = 1$, $M = N = 10$.

from that table: for 30×30 spectral elements of degree 10×10, the one-level method needs 85 iterations to reach $TOL = 10^{-6}$ and 251 CPU seconds, the two-level method with $N_0 = 2$ needs only 15 iterations and 47.2 CPU seconds to do the same.

All the examples that we will show in this chapter are for $\eta_1 = \eta_2 = 1$ and the standard exact solution $\mathbf{u} = (\sin(\frac{\pi}{2}y)x, \sin(\frac{\pi}{2}x)y)$ on the square $[-1, 1]^2$. Unfortunately we do not have enough time and space to fully explore the performance of our methods for varying η_1 or η_2, or for highly oscillatory or singular exact solutions; we just could test how they perform for this standard case. We are emboldened by their excellent performance to test our methods in future work for all these scenarios. We mention that Toselli [96, section 3.6] presented some numerical evidence that the performance of lower order Nédélec elements does not detoriate too much for η_1 or η_2 very small or very large, in fact, for fixed overlap, the empirical condition numbers and iteration counts are bounded from above by a constant.

This chapter is organized as follows: in the first section we state the problem and the preconditioners and discuss their implementation. The second section presents a numerical exploration of the one-level method with overlapping subregions made out of 2×2 spectral elements, and we show the dependence of the performance on the degree and the number of spectral elements. The third section presents two-level methods, their dependence on the degree and on the number of spectral elements. We also explore the dependence on the degree of the coarse space. We end the section and the chapter with two examples with overlaps smaller than a complete spectral element.

10.1 Implementation of Schwarz preconditioners

We solve the variational problem:

$$?\mathbf{u} \in V : \forall \mathbf{v} \in V : a(\mathbf{u}, \mathbf{v}) := \eta_1(\mathbf{u}, \mathbf{v})_0 + \eta_2(\mathbf{curl}\,\mathbf{u}, \mathbf{curl}\,\mathbf{v})_0 = (\mathbf{f}, \mathbf{u})_0$$

Our implementation uses as computational subspace V the general $\mathbb{ND}^0_{m_1,n_1;m_2,n_2}$ (see section 7.1), i.e., the space with tangential continuity across the element interfaces with the local space $\mathbb{Q}_{m_1,n_1}(K) \times \mathbb{Q}_{m_2,n_2}(K)$ and zero tangential components on $\partial\Omega$. Usually, we use the Nédélec II spaces of degree N, setting $m_1 = m_2 = n_1 = n_2 = N$. The domain is covered by an uniform mesh of $M \times M$ spectral elements, implying $h = \frac{2}{M}$. The coarse space is the Nédélec II space of degree N_0, and we use the block tensor fast direct solver developed in section 9.3 to solve the coarse problem exactly. In the Schwarz framework, this corresponds to the exact projection into $V_0 = \mathbb{ND}^{II}_{N_0}(\Omega, T_h)$:

$$?T_0 u : \forall \mathbf{v} \in V_0 : a(T_0\mathbf{u}, \mathbf{v}) = a(\mathbf{u}, \mathbf{v})$$

For element-wise overlap, $\delta = h$, we choose the four spectral elements touching each interior vertex in the spectral element mesh as overlapping subregion Ω'_i (therefore $H = 2h$) and we solve a zero tangential boundary value problem in each of the subregions, using the block tensor fast direct solver to solve the local problems exactly. This corresponds to the exact projections into $V_i = \mathbb{ND}^0_{m_1,n_1;m_2,n_2}(\Omega'_i)$, $i = 1, \cdots, J := (M-1)^2$:

$$?T_i u : \forall \mathbf{v} \in V_i : a(T_i\mathbf{u}, \mathbf{v}) = a(\mathbf{u}, \mathbf{v})$$

See figure 10.1 for a picture of the four overlapping subregions that share one spectral element. (The number of colors in the coloring assumption is therefore $N_C = 4$.) We call this case the 2×2 vertex centered case.

For overlap less than one element, we use a rectangular overlapping subregion $\Omega'_{i,\delta}$ extending $\frac{h+\delta}{2}$ in each direction from the central vertex. (See the middle subregion in figure 10.3, and the four overlapping subregions sharing the center of one spectral element in figure 10.2.) On the boundary, several choices for overlapping subregions are conceivable. We chose to extend the subregions belonging to interior vertices next to the boundary up to the boundary, see figure 10.3. As local solvers we use the inversion of the submatrix of the discretization associated to the Gauss-Lobatto-Legendre points inside the subregion, and we call the set of all basis functions associated to those points $V_{i,\delta}$. The local solve does not correspond to a standard zero tangential boundary value solve inside $\Omega'_{i,\delta}$, since the spectral element approximation of the local correction is only zero at the Gauss-Lobatto-Legendre mesh in $\Omega'_i \setminus \Omega'_{i,\delta}$, but it will not be zero everywhere in $\Omega'_i \setminus \Omega'_{i,\delta}$. Written in another way, supp$V_{i,\delta} = \Omega'_i$, and not $\Omega'_{i,\delta}$. The local solve induces a projection on $V_{i,\delta}$:

$$?T_{i,\delta} u : \forall \mathbf{v} \in V_{i,\delta} : a(T_{i,\delta}\mathbf{u}, \mathbf{v}) = a(\mathbf{u}, \mathbf{v})$$

These local solvers are not standard solvers, and we are still in the process of analyzing and testing them. We use these local solvers by analogy to domain decomposition methods for

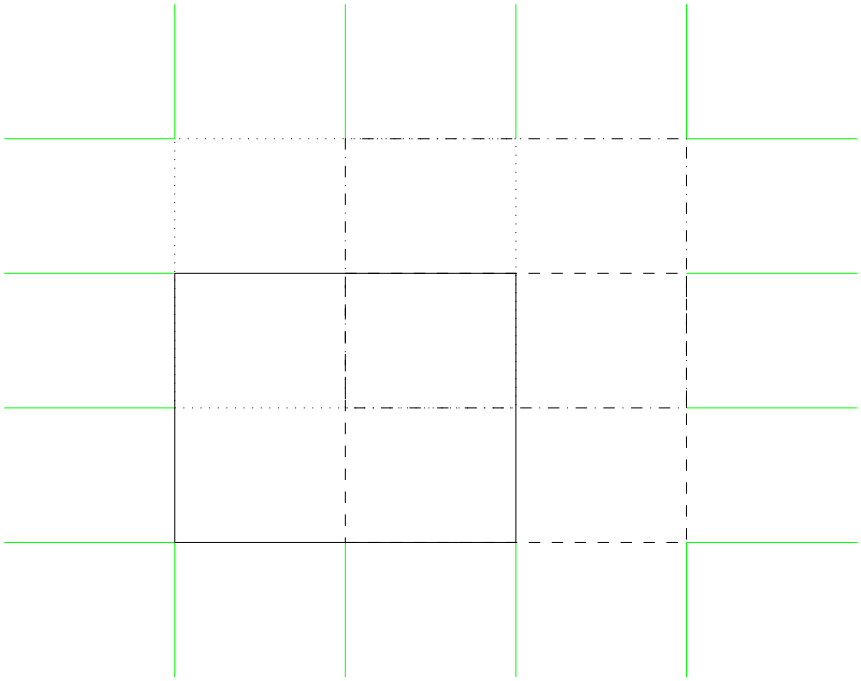


Figure 10.1: Four overlapping subregions in the 2x2 vertex centered case: elementwise overlap.

general matrices (see, e.g., Hackbusch [53, Kapitel 11]) and the preconditioner proposed and analyzed by Casarin [25, Theorem 3.5.2] for Poisson's equation.

The one-level methods tested in the next section use the preconditioner

$$T_{as1} = \sum_{i=1}^{J} T_i.$$

The two-level methods tested in section 10.3 are of the two types

$$T_{as2} = T_0 + \sum_{i=1}^{J} T_i \qquad\qquad T_{as2,\delta} = T_0 + \sum_{i=1}^{J} T_{i,\delta}.$$

We implemented a modified version of the preconditioned conjugate gradient method from Barrett et al [12] in MATLAB. Instead of using vectors for x^n, r^n, p^n, q^n, and z^n (see the conjugate gradient algorithm in figure 5.1), we use two two-dimensional arrays for each of them to represent the vector fields on the rectangular region Ω. The application of the stiffness matrix, the preconditioners, and the inner product are implemented by matrix-matrix

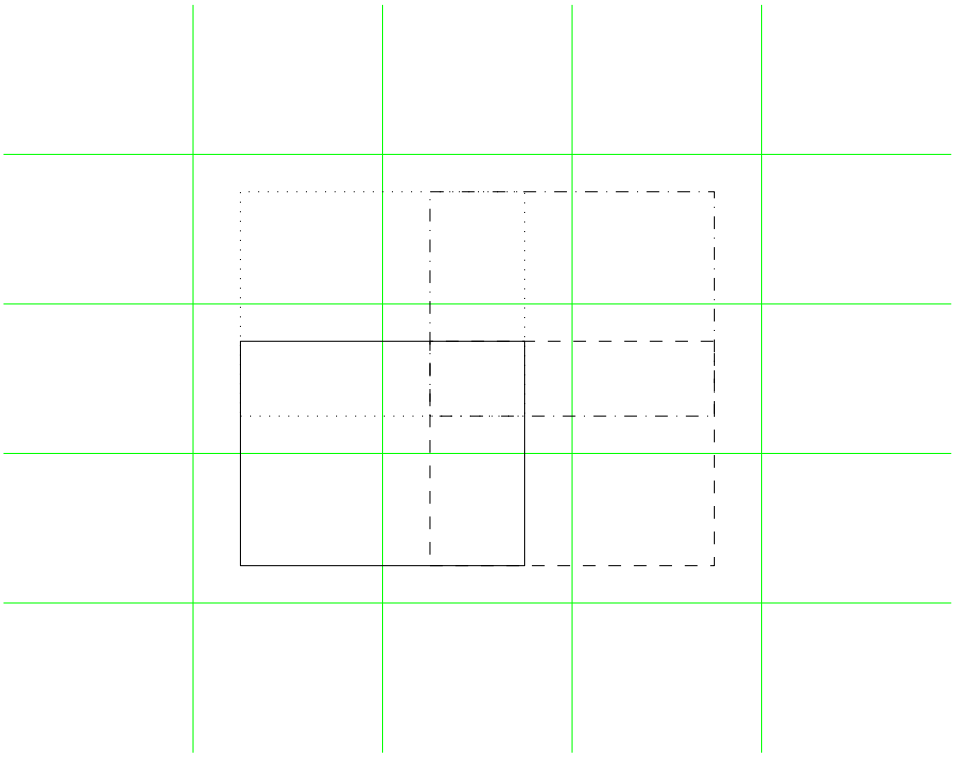


Figure 10.2: Four overlapping subregions in the vertex centered case: overlap of one half element.

multiplications and componentwise multiplication of matrices, and could be translated directly into BLAS level 3 calls in a C or FORTRAN implementation allowing the use of highly optimized numerical kernels. We also added an implementation of the O'Leary–Widlund conjugate gradient condition number estimator.

10.2 Numerical results: One level methods

Here we present two figures, 10.4 and 10.5. Both of them use T_{as1}. Figure 10.4 shows the dependence of the iteration count and the condition number on the number of spectral elements. We see that the iteration count seems to grow approximately linearly, and that the condition number grows superlinearly. In figure 10.5 we study the effect of increasing the degrees of the spectral elements while keeping their number fixed. Increasing the degree actually improves the condition number, which seems to converge to about 38.2 and the iteration count stays constant at 32.

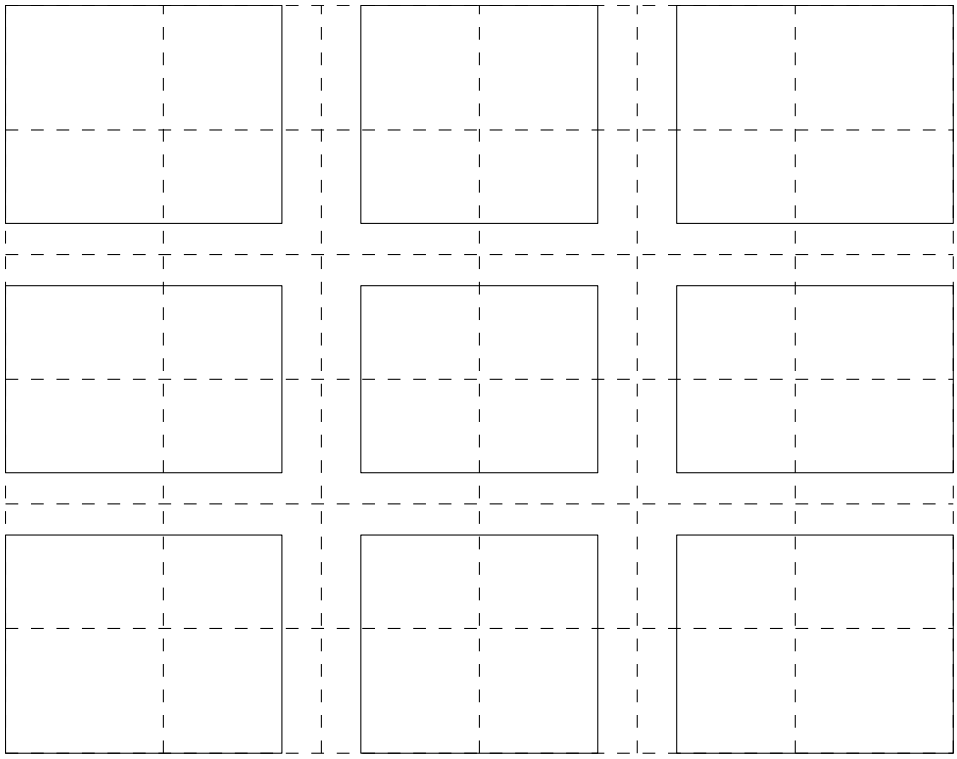


Figure 10.3: Interior and boundary subregions in the vertex centered case, overlap of one half element: the nine types of subregions, extended subregions on the boundary.

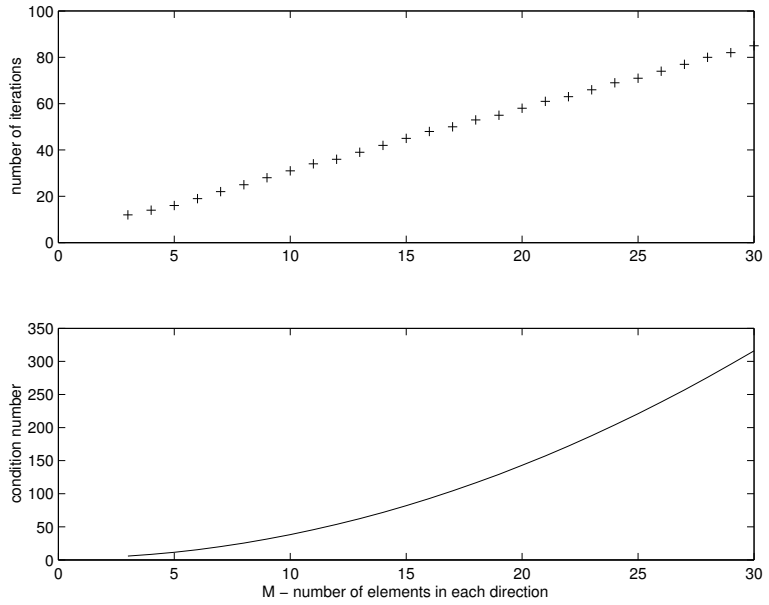


Figure 10.4: One-level method, varying number of spectral elements, degree 10×10

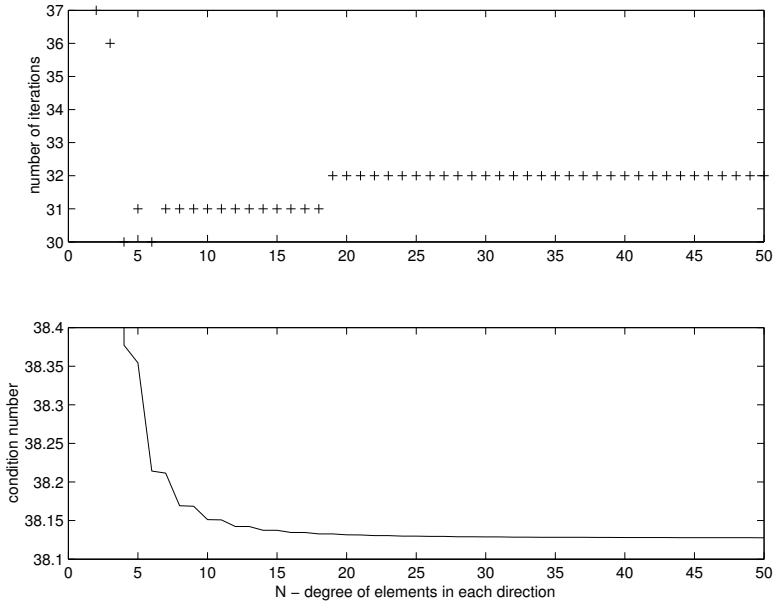


Figure 10.5: One-level method, varying degree, 10×10 spectral elements

10.3 Numerical results: Two level methods

We start with a table that summarizes some of the results shown in more detail later on, in table 10.4. In it we compare the different methods for three numbers of spectral elements, 20×20, 30×30, and 40×40. The degree of the spectral elements is always 10×10. We compare the one-level method of the previous section with two-level methods that differ in the degree of the coarse space. (We use the 2×2 vertex centered domain decomposition with element-wise overlap.)

Method	iter	$\kappa_{est}(K)$	$\|\|\mathbf{u}_{iter} - \mathbf{u}^*\|\|_\infty$	t_{CPU} in s
$M = 20$				
one-level	58	142.9	2.05e-06	74.4
two-level($N_0 = 2$)	15	4.84	1.46e-06	19.6
two-level($N_0 = 3$)	14	4.84	1.49e-06	18.9
two-level($N_0 = 4$)	15	4.85	5.56e-07	20.7
two-level($N_0 = 5$)	14	4.84	1.25e-06	20.7
$M = 30$				
one-level	85	316.0	1.59e-06	251
two-level($N_0 = 2$)	15	4.91	1.03e-06	47.2
two-level($N_0 = 3$)	15	4.93	3.74e-07	47.7
two-level($N_0 = 4$)	15	4.93	3.11e-07	49.7
two-level($N_0 = 5$)	15	4.93	2.83e-07	52.1
$M = 40$				
two-level($N_0 = 2$)	15	4.95	7.24e-07	98.3
two-level($N_0 = 3$)	15	4.96	2.66e-07	102.2
two-level($N_0 = 4$)	15	4.96	2.15e-07	106.0
two-level($N_0 = 5$)	15	4.96	1.83e-07	117.4

Table 10.4: Comparison of different methods for the 2×2 vertex centered domain decomposition for $\eta_1 = \eta_2 = 1$, $N = 10$, $M = 20, 30, 40$.

The performance of the one-level method detoriates with increasing number of spectral elements. The addition of a coarse space removes the dependence on the number of spectral elements. The choice of the degree N_0 of the coarse space does not seem to make much of a difference. The fastest method seems to be almost always the choice $N_0 = 2$ or $N_0 = 3$. Seeing that the exact form of the coarse space does not seem to matter would suggest testing coarse spaces of even lower dimension, maybe one or two well-chosen coarse basis functions per spectral element are already enough.

In figures 10.6 and 10.7 we show the case $n_0 = 2$, in 10.6 the dependence on the number of spectral elements, in 10.7 the dependence on the degree.

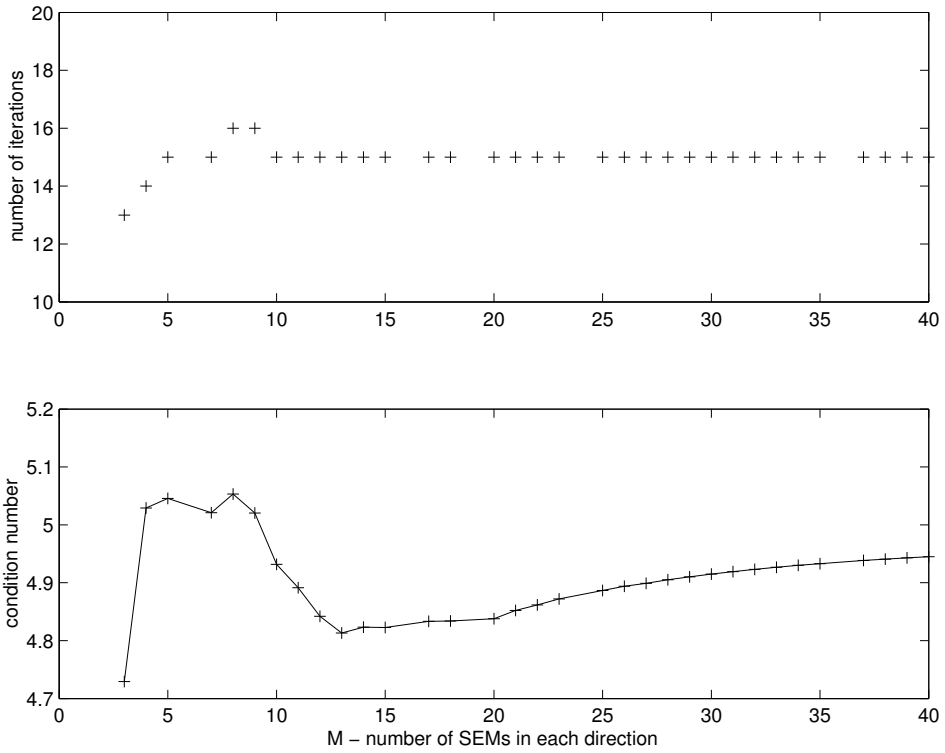


Figure 10.6: Two-level method, varying number of spectral elements, degree 10×10, $n_0 = 2$

In the computations for figure 10.6 we encountered several badly conditioned eigensystems. We mark the data points for the well-conditioned eigensystems with a $+$ and do not report the results for the near-singular cases. It seems that the coarse solver with $n_0 = 2$ is more prone to such problems, we did not observe badly conditioned eigensystems in any other case in our tests. The iteration count stays constant at 15 after $M = 10$, the condition number approaches 4.95. Increasing the degree in 10.7 results in increasing the iteration count to 16 at $N = 12$, but there seems to be no further increase, and the condition number goes to 4.95 after some initial oscillations.

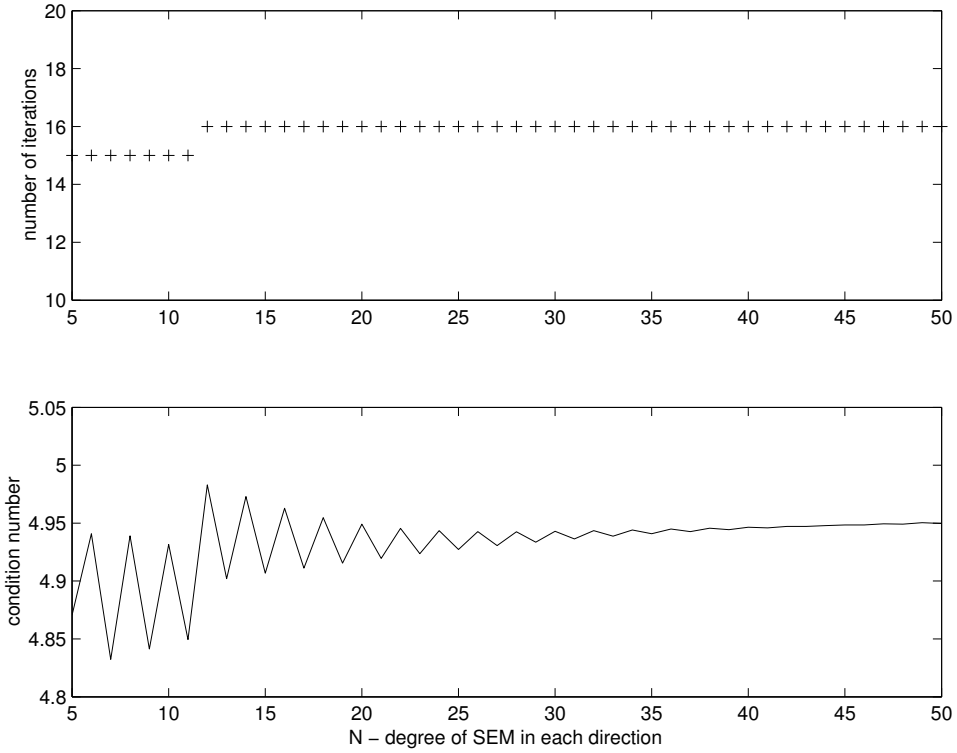


Figure 10.7: Two-level method, varying degree, 10×10 spectral elements, $n_0 = 2$

In figures 10.8, 10.9, and 10.10, we show the dependence on the number of spectral elements for $n_0 = 3$, $n_0 = 4$, and $n_0 = 5$, respectively. Increasing the degree of the coarse space seems to improve the results for small numbers of spectral elements, but it does not seem to change the bound for large M for the iteration count nor the condition number.

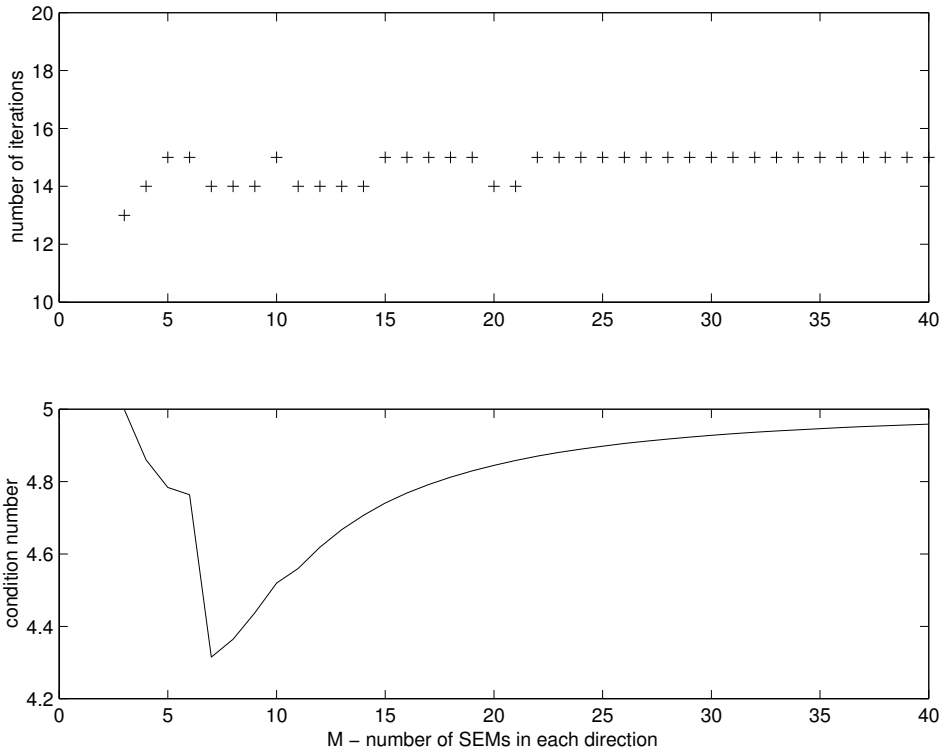


Figure 10.8: Two-level method, varying number of spectral elements, degree 10×10, $n_0 = 3$

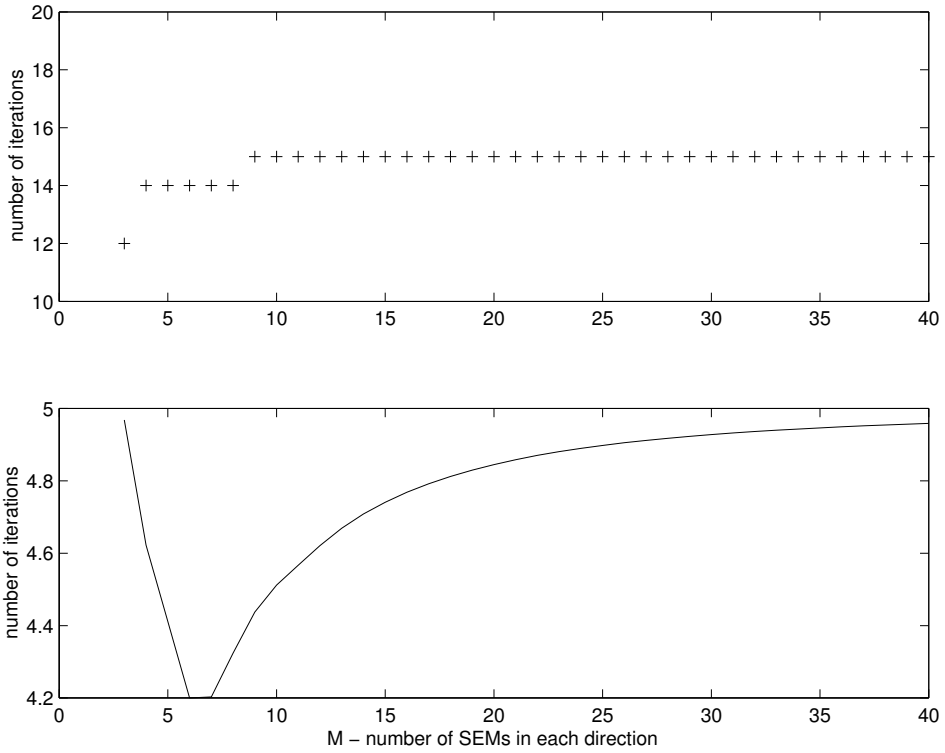


Figure 10.9: Two-level method, varying number of spectral elements, degree 10×10, $n_0 = 4$

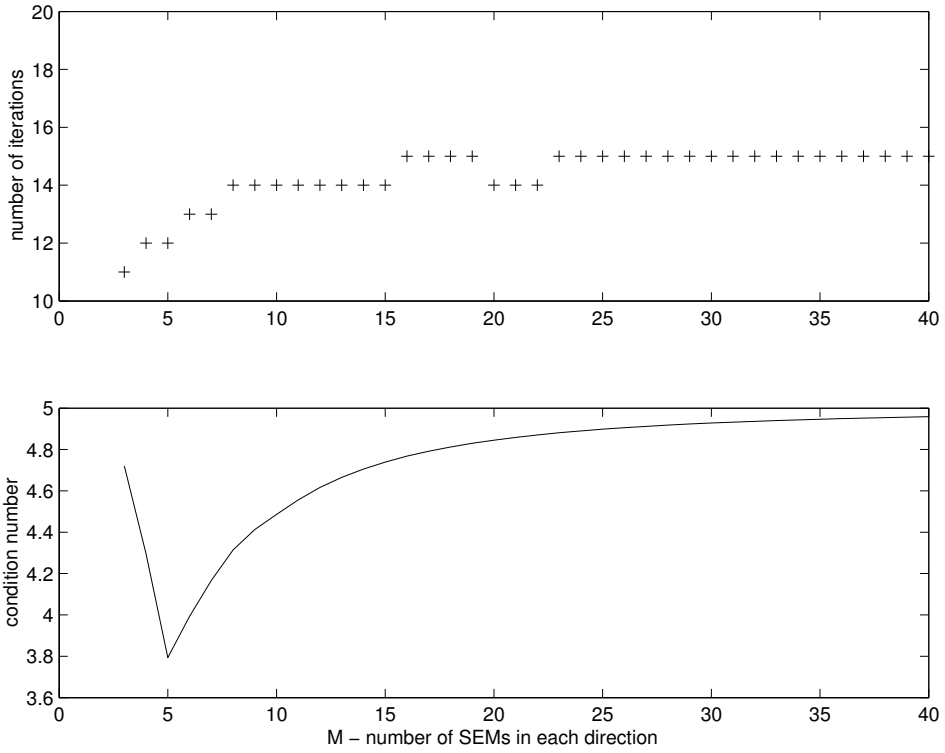


Figure 10.10: Two-level method, varying number of spectral elements, degree 10×10, $n_0 = 5$

In figures 10.11, 10.12, and 10.13, we show the dependence on the degree of spectral elements for $n_0 = 3$, $n_0 = 4$, and $n_0 = 5$, respectively. Increasing the degree of the coarse space does not seem to improve the iteration count, but it improves the initial condition number, and, to a smaller extent, the condition number at $N = 50$.

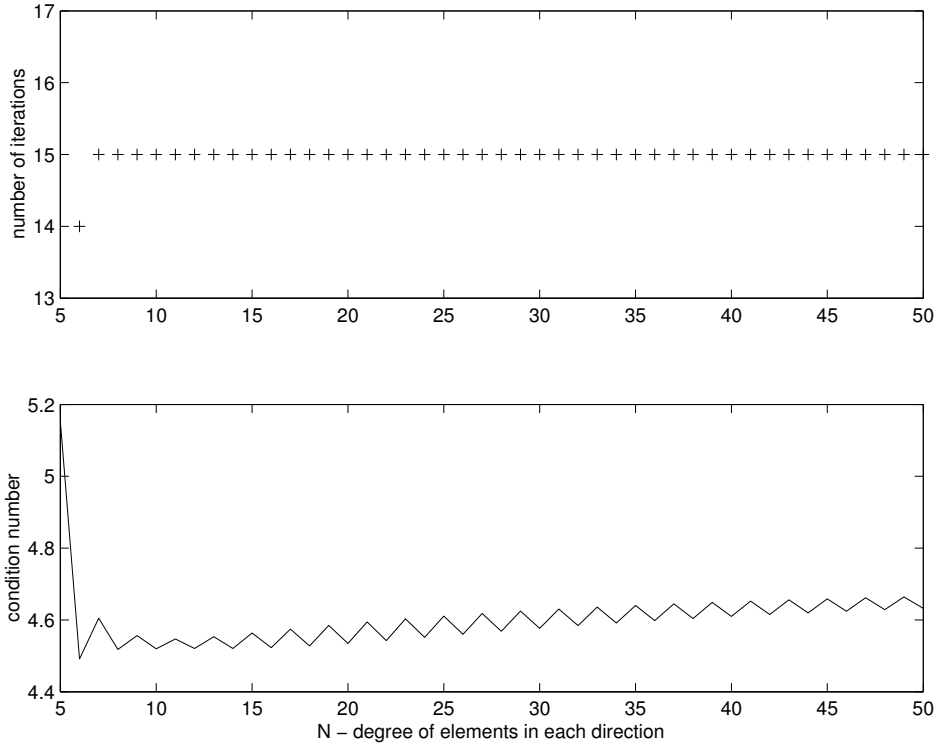


Figure 10.11: Two-level method, varying degree, 10×10 spectral elements, $n_0 = 3$

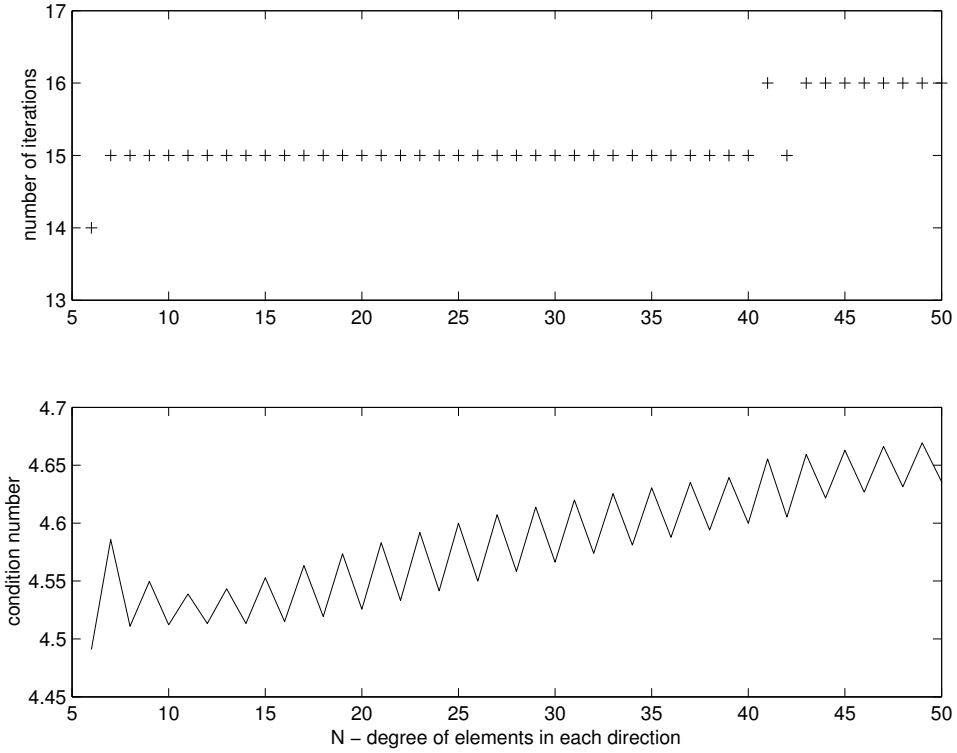


Figure 10.12: Two-level method, varying degree, 10×10 spectral elements, $n_0 = 4$

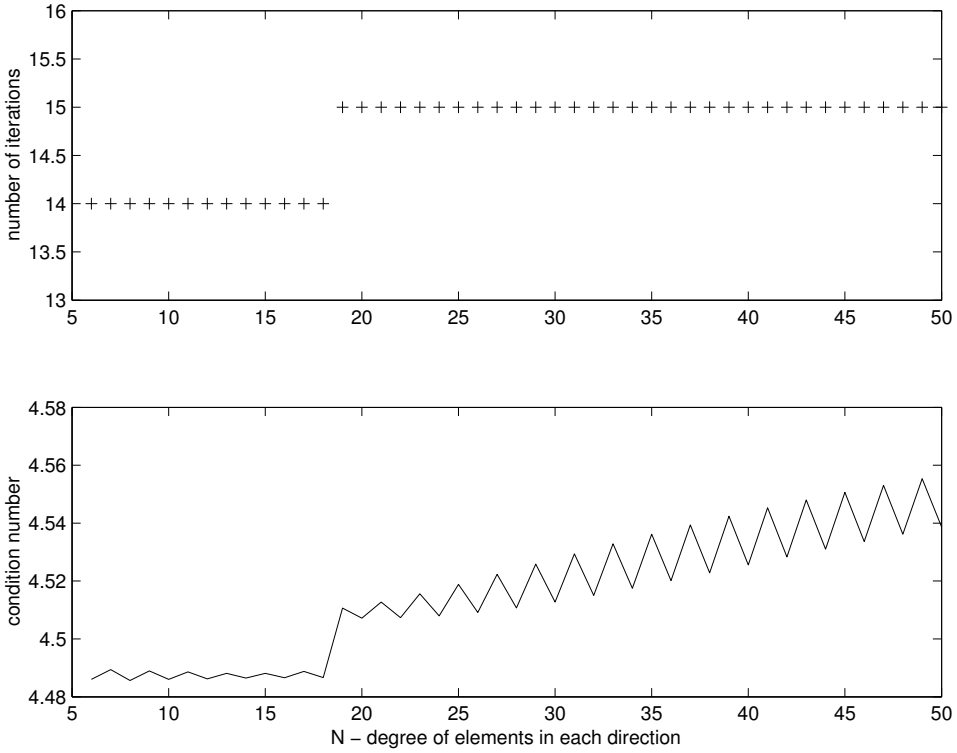


Figure 10.13: Two-level method, varying degree, 10×10 spectral elements, $n_0 = 5$

Lastly, we show two examples with overlap smaller than one element in figure 10.14 and 10.15. In figure 10.14 we show the case of spectral elements of local degree 10×10, and in figure 10.15 the case of local degree 20×20; both on 10×10 spectral elements. We give results for four different δ in both figures. Decreasing overlap yields an increase in condition number and iteration count. Increased degree seems to result in larger iteration counts and larger condition numbers. It is not possible to guess from the figures what the "empirical" c and d in the condition number estimate $N^c(1+\frac{H}{\delta})^d$ should be. More tests are needed, and we will present further analysis and numerical evidence in future work.

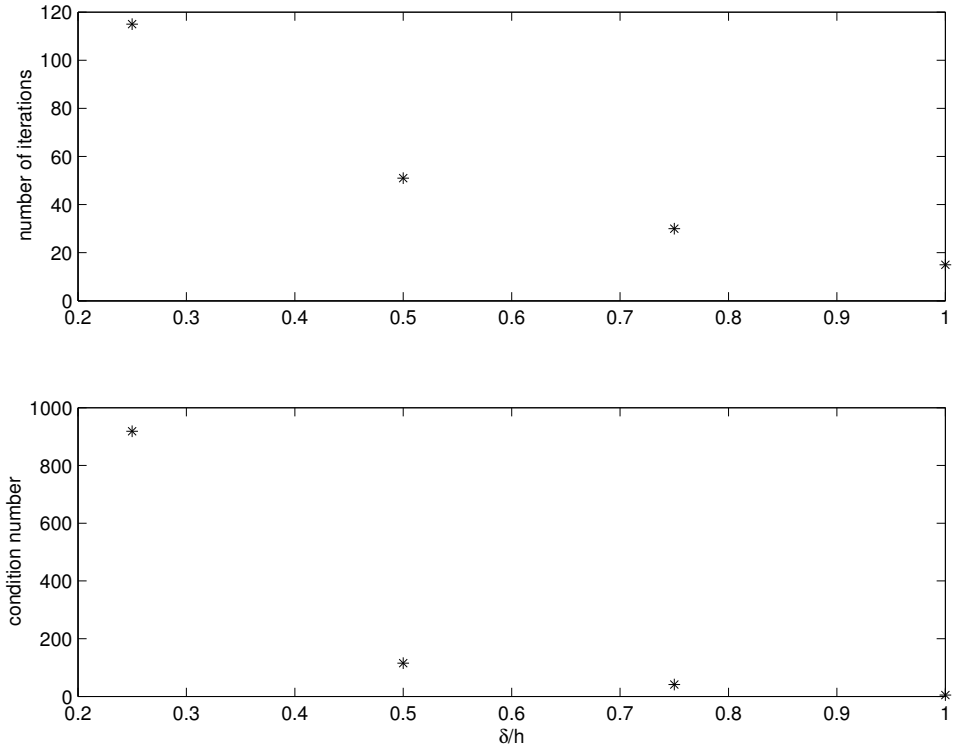


Figure 10.14: Two-level method, 10×10 spectral elements of degree 10×10, $n_0 = 2$, varying overlap

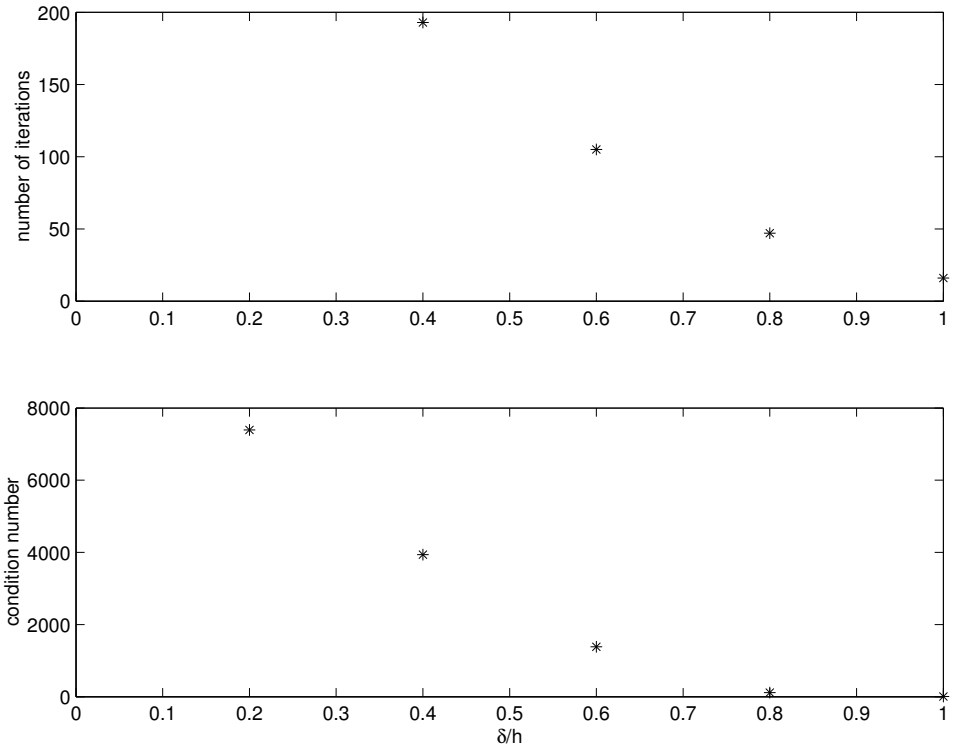


Figure 10.15: Two-level method, 10×10 spectral elements of degree 20×20, $n_0 = 2$, varying overlap

Chapter 11

Overlapping Schwarz methods: Theory

In this chapter we will prove a bound on the condition number of the two-level overlapping Schwarz method in the two-dimensional and three-dimensional case, following the general outline of the proof from Toselli [98] with some changes and extensions necessitated by our use of spectral elements and by our desire to obtain bounds that are explicit in their dependence on N.

In the first subsection, we state the problem, the domain decomposition, and the overlapping additive Schwarz method for which we will prove the condition number estimate. Our main result is given in terms of three estimates that we present and discuss in the second section. In the third section, we introduce some operators, prove a lemma and give a result that we need in the proof of the main result. The last section presents the estimate and its proof. The estimate is then explicated for two choices of overlap and possible improvements are noted.

In the following, recall that $||\cdot||_r$ is the H^r-norm, $|\cdot|_r$ is the H^r-seminorm, $||\cdot||_{s,p}$ is the $W^{s,p}$-norm and, in particular, $||\cdot||_{0,p}$ is the L^p-norm.

11.1 Variational problem and overlapping method

We solve the model problem in the constant coefficient case on a bounded and convex polyhedron Ω of diameter $H_\Omega = O(1)$, i.e., the variational problem is for some computational subspace $V = V_N(\Omega)$ of $H_0(\mathbf{curl}, \Omega)$

$$?\mathbf{u} \in V : \forall \mathbf{v} \in V : a(\mathbf{u}, \mathbf{v}) := \eta_1(\mathbf{u}, \mathbf{v})_0 + \eta_2(\mathbf{curl}\,\mathbf{u}, \mathbf{curl}\,\mathbf{v})_0 = (\mathbf{f}, \mathbf{u})_0$$

The domain is covered by a shape-regular and quasi-uniform mesh T_H of quadrilateral elements of size H. Those elements are further subdivided into spectral elements of size h

and degree N, constituting a shape-regular fine mesh T_h. There are several ways to obtain the overlapping subregions. In one of them, the subdomains Ω_i, $i = 1, \cdots, J$, correspond to the elements of the coarse mesh, and they are extended by some distance δ_i to yield the overlapping regions Ω_i'.

See figure 11.1 for an example. We show one of the Ω_i', with some surrounding elements from T_h.

A second way is to combine elements (and parts of elements) in such a way that the constructed subregion still has a diameter of $O(H)$ and overlaps other subregions with a geometric overlap δ. We chose such a setting for the implementation in chapter 10, and refer to the explanation and figures there.

The overlap parameter δ is the minimal distance between $\partial\Omega_i'$ and Ω_i, and therefore equal to $\min_i \delta_i$. In the element-wise overlap case it will be a multiple of h.

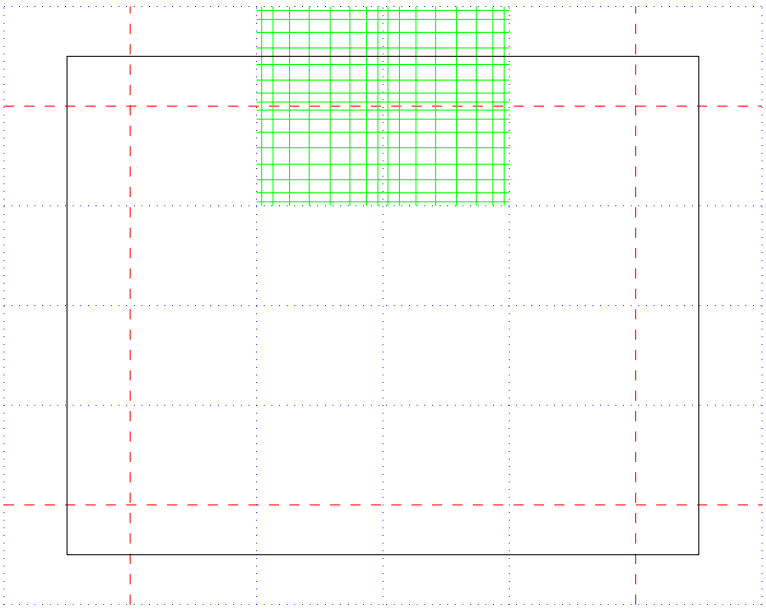


Figure 11.1: An overlapping subregion for the domain decomposition method. $h = H/4$, $\delta = h/2$, $N = 10$. Broken lines: subdomain mesh T_H. Dotted lines: element mesh T_h. Solid enclosure: $\partial\Omega_i'$. We also show the GLL mesh associated to a degree 10 spectral element in four of the elements of size h.

The global computational subspace $V = V_N(\Omega)$, in which the variational problem is discretized, is chosen as $V = \mathbb{ND}_N^I(\Omega, T_h)$.

In the element-wise overlap case, the local spaces V_i are the subspaces of functions in V that have support in Ω_i'. For general (smaller) overlap, V_i is the subspace of functions in V

spanned by the spectral element basis functions in $\mathbb{ND}_N^I(\Omega, T_h)$ that are associated to GLL points that are inside Ω_i'. (Or equivalently, the subspace of functions in V with spectral element degrees of freedom outside of Ω_i' set to zero.) In the element-wise overlap case, the support of functions in V_i is Ω_i'; for general overlap, the support is Ω_i^*, equal to the union of all the elements in T_h that intersect Ω_i'.

The coarse space V_0 is chosen as $\mathbb{ND}_{N_0}^I(\Omega, T_H) \subset \mathbb{ND}_N^I(\Omega, T_h)$. (Different choices for the fine-to-coarse mapping in the algorithms of chapter 11 correspond either to a different space V_0, or to a different system, i.e., different $a_0(\cdot,\cdot)$, posed on V_0.) For any fixed N_0 we have the estimates in lemma 11.2. For the sake of simplicity, we will not try to explicate the dependence of the condition number of the operator on N_0.

The global space V admits a non-unique decomposition $V = \sum_{i=0}^J V_i$.

We will use exact solvers in the subspaces, i.e., the bilinear form for all problems will be $a(\cdot,\cdot)$. The proof could be extended to inexact solvers with standard arguments. (See, e.g., Smith, Bjørstad, and Gropp [91].)

We introduce the local projections $T_i : V \to V_i, u \mapsto T_i u$ defined by

$$?T_i u : \forall \mathbf{v} \in V_i : a(T_i \mathbf{u}, \mathbf{v}) = a(\mathbf{u}, \mathbf{v})$$

Using these projections, many domain decomposition methods can be defined (see chapter 5 and, for instance, [91, pages 149–153]).

We define two operators, an additive one-level operator given by

$$T_{as1} = \sum_{i=1}^{J} T_i$$

and an additive two-level operator

$$T_{as2} = T_0 + \sum_{i=1}^{J} T_i = T_0 + T_{as1}$$

See chapter 5 for further explanations on the implementation of such methods, and the previous chapter for an implementation of these methods in the two-dimensional case.

We will prove a condition number estimate for T_{as2}. With similar techniques and the same kind of estimates, results for T_{as1}, multiplicative and hybrid methods could be proven, see, e.g., Smith, Bjørstad, and Gropp [91, pages 155–158].

To use the standard coloring arguments (see, e.g., [91, bottom of page 165 and proof of theorem 1 on page 167], and also chapter 5), we need an assumption about the covering of Ω by the overlapping regions Ω_i':

Coloring assumption: *The overlapping regions $\{\Omega_i'\}$ can be colored using N_c colors, in such a way that regions with the same color do not intersect.*

11.2 Required estimates

Our final estimate depends on the following three estimates.

Estimate 1 (Interpolation property for divergence-free vector fields) *There is a constant C independent of N, h and $\mathbf{u}$, and a function $f_1(N)$ such that for $\mathbf{u} \in H_0^{\perp}(\mathbf{curl})$ with $\mathbf{curl}\,\mathbf{u} \in \mathbb{W}_N(\Omega, T_h)$ there is a bound*

$$||(I - \mathbf{\Pi}_N^{ND,I})\mathbf{w}||_0 \leq Ch f_1(N)||\,\mathbf{curl}\,\mathbf{w}||_0 \tag{11.1}$$

Estimate 2 (L_2-stability of the local splitting) *Let χ_i be the interpolated partition of unity used to define the local splitting. Then, there exist a constant C independent of N, h and $\mathbf{u}$, and a function $f_2(N)$ independent of h and $\mathbf{u}$ such that for all $\mathbf{u} \in \mathbb{ND}_N^I$*

$$||\mathbf{\Pi}_N^{ND,I}(\chi_i\mathbf{u})||_0 \leq C f_2(N)||\chi_i\mathbf{u}||_0 \tag{11.2}$$

Estimate 3 ($\mathbf{curl}$-stability of the local splitting) *Let χ_i be the interpolated partition of unity used to define the local splitting. Then, there exist a constant C independent of N, h and $\mathbf{u}$, and a function $f_3(N)$ independent of h and $\mathbf{u}$ such that for all $\mathbf{u} \in \mathbb{ND}_N^I$*

$$||\,\mathbf{curl}\left(\mathbf{\Pi}_N^{ND,I}(\chi_i\mathbf{u})\right)||_0 \leq C f_3(N)||\,\mathbf{curl}(\chi_i\mathbf{u})||_0 \tag{11.3}$$

We proved the interpolation property in lemma 7.17 in section 7.7. There we showed that (11.1) holds with $f_1(N) = 1 + C(\epsilon)N^{-1+\epsilon}$ where $C(\epsilon)$ is related to the regularity of a certain $\mathbf{curl}$ potential problem. In the proof of that result we also indicated that an improved or optimal interpolation estimate for the Nédélec interpolation operator (which is not yet proven in the three-dimensional case) would imply $f_1(N) = C(\epsilon)N^{-1+g(\epsilon)}$.

The properties of the local splitting (11.2) and (11.3) are usually proven in a way that makes as little use of the special form of χ_i as possible, and are based on estimates for $\mathbf{\Pi}_N^{ND,I}$ on certain polynomial spaces of higher degree in which $\chi_i\mathbf{u}$ lies. If that space is denoted $\mathbb{V}_{N+}$, then it would be enough to prove

$$\forall v_{N+} \in \mathbb{V}_{N+}: \qquad ||\mathbf{\Pi}_N^{ND,I}v_{N+}||_0 \leq C f_2(N)||v_{N+}||_0$$

$$\forall v_{N+} \in \mathbb{V}_{N+}: \quad ||\,\mathbf{curl}\left(\mathbf{\Pi}_N^{ND,I}v_{N+}\right)||_0 \leq C f_3(N)||\,\mathbf{curl}\,v_{N+}||_0$$

The second estimate is reduced to an estimate of a different interpolation operator on a different space using the commuting diagram property: Let $\mathbb{C}_{N+}$ be a space containing $\mathbf{curl}\,\mathbb{V}_{N+}$. Set $\mathbb{T}_N = \mathbb{W}_N(\Omega, T_h)$ for the two-dimensional case, and $\mathbb{T}_N = \mathbb{RT}_N(\Omega, T_h)$ for the threedimensional case and let $\mathbf{\Pi}_N^T$ be the commuting interpolant in $\mathbb{T}_N$. Then the

commuting diagram property (lemma 7.1 in section 7.3) implies that $\mathbf{curl}\left(\mathbf{\Pi}_N^{ND,I} v_{N+}\right) = \mathbf{\Pi}_N^T(\mathbf{curl}\, v_{N+})$. Therefore proving

$$\forall v_{N+} \in \mathbb{V}_{N+}: \qquad ||\mathbf{\Pi}_N^{ND,I} v_{N+}||_0 \leq C f_2(N)||v_{N+}||_0$$

$$\forall w_{N+} \in \mathbb{C}_{N+}: \qquad ||\mathbf{\Pi}_N^{T} w_{N+}||_0 \leq C f_3(N)||w_{N+}||_0$$

implies the estimates (11.2) and (11.3).

There are several candidates for the interpolated partition of unity χ_i. One possible choice for the χ_i is as a polynomial inside each small element K_h in T_h. In the case of element-wise overlap we can just choose a linear function inside each element, and $\chi_i \mathbf{u}$ would be in $\mathbb{V}_{N+} = \mathbb{ND}_{N+1}^I$, and $\mathbb{C}_{N+} = \mathbb{T}_{N+1}$.

For this case the numerical results from section 7.6 show that (11.2) and (11.3) are satisfied with $f_1(N) = f_2(N) = 1$ and a small constant C.

For overlaps δ smaller than h, we need to construct an interpolated partition of unity χ_i with

$$||\chi_i||_{0,\infty} < C \qquad\qquad ||\,\mathbf{grad}\,\chi_i||_{0,\infty} < \frac{C}{\delta} \tag{11.4}$$

We also need to assume that the sets $\Omega_i^* := \text{supp}\chi_i$ can be colored by N_c colors so that Ω_i^* with the same color do not intersect. Even though that in general in theory this coloring assumption is stronger than the one with Ω_i' given above, in almost all cases in practice the number of colors needed will stay the same or will increase very slightly.

The standard choice for the small overlap case in the finite element context is the piece-wise linear interpolated partition of unity χ_i^{PL}. For rectangles, χ_i^{PL} can be constructed as a tensor product of one-dimensional function like in figure 11.2, which obviously meet the requirements (and the Ω_i^* are identical to the Ω_i'). For spectral elements, χ_i^{PL} is not a polynomial inside the element for $\delta < h$. To be able to study the properties of $\chi_i \mathbf{u}$ inside a framework of polynomial spaces, we have two choices: either we work with a polynomial interpolation of the piecewise linear χ_i^{PL} such as the Gauss-Lobatto-Legendre interpolant χ_i^M associated to some degree M and the corresponding Gauss-Lobatto-Legendre mesh GLL_M inside the element T_h – then $\chi_i \mathbf{u}$ will be in a standard Nédélec spectral element space $\mathbb{V}_{N+} = \mathbb{ND}_{N+M}^I$ (and $\mathbb{C}_{N+} = \mathbb{T}_{N+M}^I$); or we work with the original piecewise linear χ_i^{PL} and chose piecewise polynomial spaces $\mathbb{V}_{N+}$ and $\mathbb{C}_{N+}$.

We will first discuss the case of χ_i^M: to convince ourselves that the Gauss-Lobatto-Legendre interpolated partition of unity has the requisite properties, we performed some numerical experiments shown in the figures 11.3, 11.4, 11.5, and 11.6. In figures 11.3, 11.4, and 11.5 we show the ratio of the maximal gradient and maximal value[1] of (the one-dimensional

[1] We evaluated the gradient and the value of χ_i^M on an uniform grid of 2000 grid points inside the element. The subdomain part of χ_i^M is $\chi_i^M|_{\Omega_i}$, the border part is $\chi_i^M|_{\Omega-\Omega_i}$.

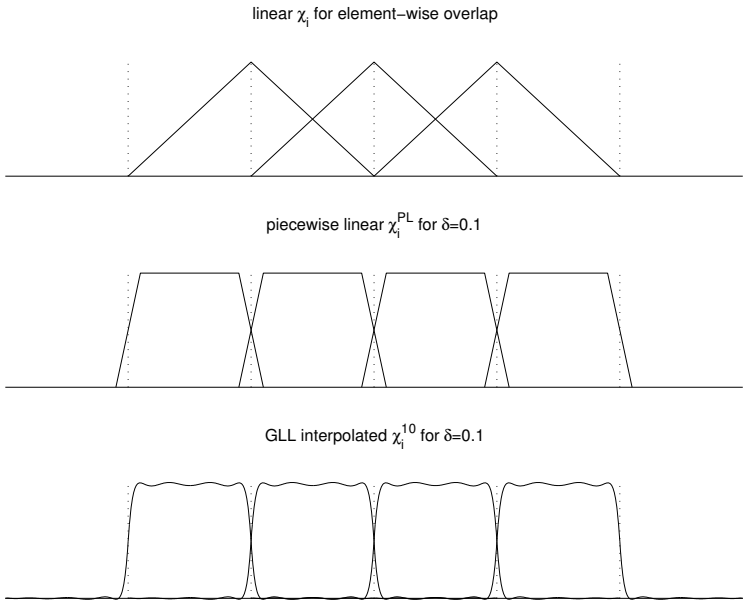


Figure 11.2: One-dimensional partitions of unity: Upper panel: linear χ_i for the case of element-wise overlap. Middle panel: piecewise linear χ_i^{PL} for $\delta = 0.1$. Bottom panel: GLL interpolated χ_i^{10} for $\delta = 0.1$.

version of) χ_i^M over χ_i^{PL} for the choices $\delta = 0.5$, $\delta = 0.1$ and $\delta = 0.01$. Since the Gauss-Lobatto-Legendre mesh on each element has a spacing of $O(h/N^2)$ close to the boundary, the smallest possible δ is also of that size. To test if the properties of χ_i^M detoriate for the smallest possible δ, we test overlaps of a small number of Gauss-Lobatto-Legendre cells in figure 11.6. In all the tested cases, the Gauss-Lobatto-Legendre interpolated χ_i^M satisfies (11.4) with bounds worse by a factor of at most 2 when compared to the piecewise linear χ_i^{PL}.

To use χ_i^M in the proof, we need L^2-bounds for the interpolation operators on the spaces $\mathbb{V}_{N+} = \mathbb{ND}^I_{N+M}$ and $\mathbb{C}_{N+} = \mathbb{T}^I_{N+M}$. We refer to section 7.6 where we computed such bounds numerically, especially to Observation 7.1 on page 94: for M a constant, $f_2(N) = f_3(N) = 1$. For $M = N$ or $M = cN$, we obtained $f_2(N) = \sqrt{N}$ and $f_3(N) = \sqrt{N}$.

That translates into conditions on the overlap. If we have the case of fixed overlap, i.e., $\frac{\delta}{h} > C_{ov}$, then we can find a fixed $M \sim \sqrt{\frac{h}{\delta}} \sim \sqrt{C_{ov}}$ so that (11.2) and (11.3) are satisfied with $f_2(N) = f_3(N) = 1$.

For minimal overlap, i.e., $\delta \sim \frac{h}{N^2}$ we need $M \sim N$, and therefore $f_2(N) = f_3(N) = \sqrt{N}$.

The second choice — using χ_i^{PL} as interpolated partition of unity and piecewise polynomial

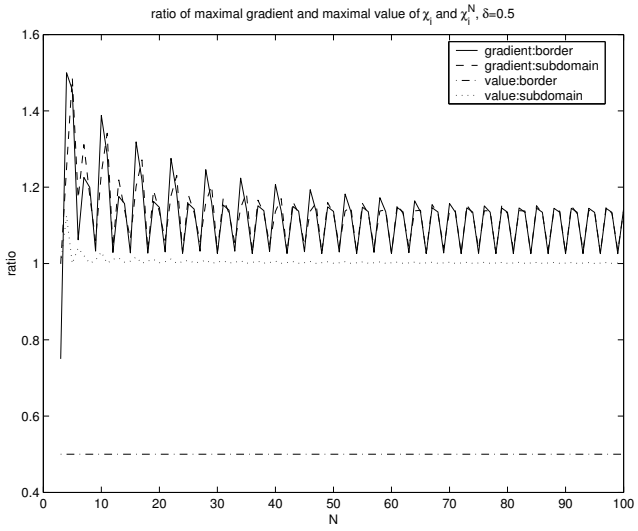


Figure 11.3: Comparing Gauss-Lobatto-Legendre and piecewise linear interpolated partitions of unity, $\delta = 0.5$.

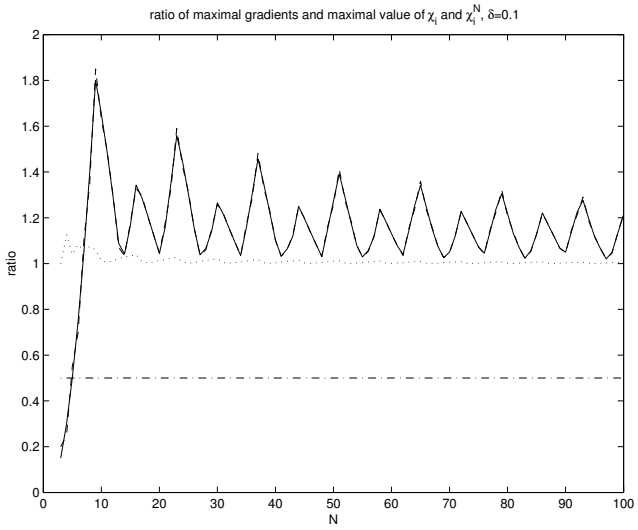


Figure 11.4: Comparing Gauss-Lobatto-Legendre and piecewise linear interpolated partitions of unity, $\delta = 0.1$.

spaces — holds some promise for better estimates.

We will explain the main idea in the following: Since the interpolation is defined element-

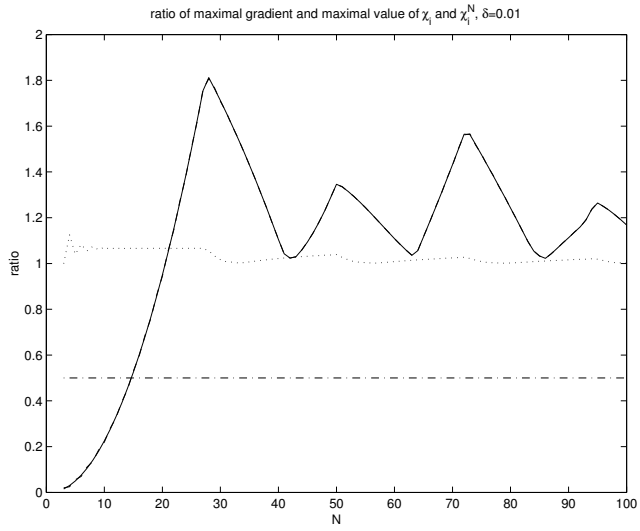


Figure 11.5: Comparing Gauss-Lobatto-Legendre and piecewise linear interpolated partitions of unity, $\delta = 0.01$.

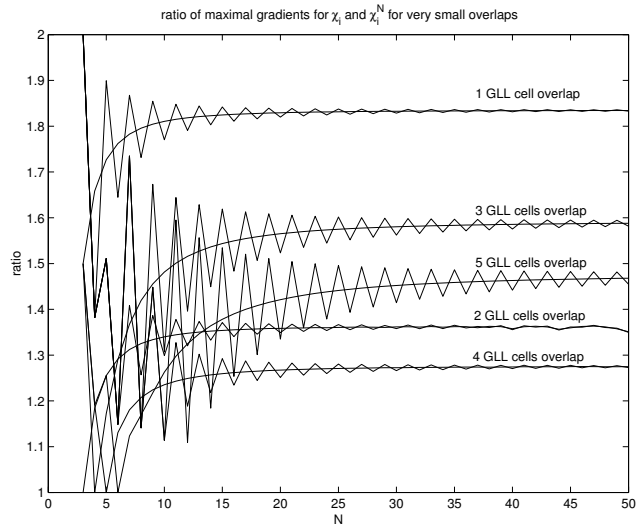


Figure 11.6: Comparing Gauss-Lobatto-Legendre and piecewise linear interpolated partitions of unity, minimal overlap on GLL grid.

wise and global bounds will easily follow from local bounds, we can restrict ourselves to

the case of the reference element $[-1,1]^d$ with $d = 2, 3$. Now define the piecewise space

$$\mathbb{Q}_{N,\delta}([-1,1]) = \mathbb{Q}_{N+1}([-1,-1+\delta]) \cap \mathbb{Q}_N([-1+\delta,1-\delta]) \cap \mathbb{Q}_{N+1}([1-\delta,1])$$

and also its tensorized versions $\mathbb{Q}_{m,n,\delta}$ and $\mathbb{Q}_{l,m,n,\delta}$ for the two-dimensional and three-dimensional case. From these spaces we build the piecewise analogue of ND^I_{N+1}, and T_{N+1}, and call them $ND^I_{N,\delta}$ and $T_{N,\delta}$. We can retrace the derivations in section 7.6, and we will obtain very similar expressions for the Nédélec type interpolants from the piecewise space to the standard space ND^I_N and T_N. (The main difference being different interpolation matrices and mass matrices, the former corresponding to the piecewise Gauss-Lobatto-Legendre interpolation on $\mathbb{Q}_{N,\delta}([-1,1])$ and the latter corresponding to the subassembled mass matrix on $\mathbb{Q}_{N,\delta}([-1,1])$.) We could use then the same numerical and analytical approaches as in section 7.6 to study the local splitting. Unfortunately we lack both time and space to follow this idea in the context of this thesis, but we will treat it in future work.

11.3 Technical tools

To introduce and analyze a stable projection into the coarse space, we need several operators. One of them is the orthogonal projection into the weakly divergence-free space

$$\Theta : H_0(\mathbf{curl}) \rightarrow H_0^{\perp}(\mathbf{curl})$$

defined by

$$\Theta\mathbf{u} := \mathbf{u} - \mathbf{grad}\, q$$

where $q \in H^1_0(\Omega)$ is the unique solution of

$$?q : \forall p \in H^1_0(\Omega) : (\mathbf{grad}\, q, \mathbf{grad}\, p) = (\mathbf{u}, \mathbf{grad}\, p)$$

It follows easily that Θ leaves the $\mathbf{curl}$ of its argument unchanged, and is also an orthogonal projection in $(L^2(\Omega))^3$.

We use Θ now to define the finite dimensional subspace

$$V^{\perp} = \Theta(\mathbb{ND}^{I,+}_N(\Omega, T_h)) \subset H_0^{\perp}(\mathbf{curl}).$$

Even though $V^{\perp}$ is not a spectral element space, the $\mathbf{curl}$ of functions in $V^{\perp}$ will be a piecewise polynomial vector field. We recall that we showed in chapter 7 that the Nédélec interpolant has better bounds on such functions.

Next, we define a projection P_N onto $V^{\perp}$:

$$\begin{aligned} P_N : H_0(\mathbf{curl}) &\rightarrow V^{\perp} \\ ?P_N\mathbf{u} \in V^{\perp} : \forall \mathbf{v} \in V^{\perp} &: (\mathbf{curl}(P_N\mathbf{u} - \mathbf{u}), \mathbf{curl}\, \mathbf{v}) = 0 \end{aligned}$$

Since $||\,\mathbf{curl}\cdot||_0$ is an equivalent norm to $||\cdot||_0$ on $H_0(\mathbf{curl})$, P_N is well-defined.

Remark On $\mathbb{ND}_N^{I,+}(\Omega, T_h)$, $P_N\mathbf{u}$ coincides with $\Theta\mathbf{u}$, and it is clear that

$$\mathbf{curl}\, P_N\mathbf{u} = \mathbf{curl}\,\Theta\mathbf{u} = \mathbf{curl}\,\mathbf{u}$$

Next, we prove an error estimate for P_N using the interpolation properties of the Nédélec interpolant. This lemma corresponds to [98, Lemma 3.3]. We follow the idea of the proof from that paper, but instead of invoking [98, Lemma 3.1], we use lemma 7.17 in section 7.7.

Lemma 11.1 *Let Ω be convex. Then, the operator P_N satisfies the following error estimate for all $\mathbf{u} \in \mathbb{ND}_N^{I,+}(\Omega, T_h)$ with C independent of h, N and $\mathbf{u}$:*

$$||\mathbf{u} - P_N\mathbf{u}||_0 \leq Chf_1(N)||\,\mathbf{curl}\,\mathbf{u}||_0 \tag{11.5}$$

Proof: Let $\mathbf{u} \in \mathbb{ND}_N^{I,+}(\Omega, T_h)$. Thanks to the remark after the definition of $P_N\mathbf{u}$, $\mathbf{curl}(\mathbf{u} - P_N\mathbf{u}) = 0$, and therefore

$$\mathbf{u} - P_N\mathbf{u} = \mathbf{grad}\, q$$

with some $q \in H_0^1(\Omega)$. Now the appropriate version of the commuting diagram property guarantees that

$$\mathbf{u} - \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u} = \mathbf{\Pi}_N^{ND,I}(\mathbf{u} - P_N\mathbf{u}) = \mathbf{grad}\, q_N \tag{11.6}$$

with some $q_N \in \mathbb{W}_N(\Omega)$. (See section 7.3 on the commuting diagram property.)

We rewrite

$$||\mathbf{u} - P_N\mathbf{u}||_0^2 = (\mathbf{u} - P_N\mathbf{u}, \mathbf{u} - \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u} + \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u} - P_N\mathbf{u})$$

and use that $\mathbf{u}$ and $P_N\mathbf{u}$, by (11.6), are orthogonal to $\mathbf{u} - \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u}$:

$$\begin{aligned} ||\mathbf{u} - P_N\mathbf{u}||_0^2 &= (\mathbf{u} - P_N\mathbf{u}, \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u} - P_N\mathbf{u}) \\ &\leq ||\mathbf{u} - P_N\mathbf{u}||_0 ||P_N\mathbf{u} - \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u}||_0 \end{aligned}$$

to obtain

$$||\mathbf{u} - P_N\mathbf{u}||_0 \leq ||P_N\mathbf{u} - \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u}||_0$$

We use lemma 7.17 and the remark to estimate

$$||P_N\mathbf{u} - \mathbf{\Pi}_N^{ND,I} P_N\mathbf{u}||_0 \leq Chf_1(N)||\,\mathbf{curl}\, P_N\mathbf{u}||_0 = Chf_1(N)||\,\mathbf{curl}\, P_N\mathbf{u}||_0$$

and thus obtain the estimate in the lemma. ∎

We also need the L^2-projection

$$Q_0 : (L^2(\Omega))^3 \to V_0$$

onto the coarse space V_0. We require some estimates for Q_0, which can be proven exactly as in Toselli [98]:

Lemma 11.2 *Let T_H be shape-regular and quasi-uniform. Then, the following estimates hold with constants independent of $\mathbf{u}$ and H:*

$$\forall \mathbf{u} \in (H^1(\Omega))^3 : \qquad ||\,\mathbf{curl}\, Q_0 \mathbf{u}||_0 \leq C|\mathbf{u}|_1 \tag{11.7}$$

$$\forall \mathbf{u} \in (H^1(\Omega))^3 : \qquad ||\mathbf{u} - Q_0\mathbf{u}||_0 \leq CH|\mathbf{u}|_1 \tag{11.8}$$

11.4 Condition number bound

We will use the abstract Schwarz theory; see section 5.2 for a short introduction and the theorems that we will use here, and Smith, Bjørstad, and Gropp [91, chapter 5] for an introduction in textbook form, discussing the h-versions of standard algorithms.

We will give an upper bound for the inverse of the smallest eigenvalue by the standard decomposition argument, and C_0^{-2} will then be a lower bound for the smallest eigenvalue. As discussed in the first section, we assume a coloring with N_c colors for the overlapping regions Ω_i' (respective Ω_i^*). Using a standard argument (see, e.g., Smith, Bjørstad, and Gropp [91, proof of theorem 1 on page 167]), this implies an upper bound for the eigenvalues of T_{as1} of N_c and of T_{as2} of $N_c + 1$. Therefore, the bound C_0^2 proven in the next theorem will imply a bound of $(N_c + 1)C_0^2$ for the condition number of T_{as2}.

First we will prove the theorem using the general forms of the estimates (11.1), (11.2) and (11.3). Afterwards we will discuss the estimate for specific cases, and give shorter forms.

Theorem 11.3 (Lower bound) *For every $\mathbf{u} \in V$ there is a splitting $\mathbf{u} = \sum \mathbf{u}_i$ with $\sum a(\mathbf{u}_i, \mathbf{u}_i) \leq C_0^2 a(\mathbf{u}, \mathbf{u})$ with a C_0^2 of the form*

$$C \max \left\{ N_c \left(1 + \frac{H}{\delta}\right), \frac{\max(\eta_1, \eta_2)}{\min(\eta_1, \eta_2)} \max \left(1 + N_c f_2^2(N), 1 + N_c f_3^2(N) \left(1 + \left(\frac{H + h f_1(N)}{\delta} \right)^2 \right) \right) \right\}$$

Proof: First, we use the discrete Helmholtz decomposition (see section 7.3 in chapter 7) in $\mathbb{ND}_N^I$ to split $\mathbf{u}$ into a sum $\mathbf{grad}\, q + \mathbf{w}$, where $q \in \mathbb{S}_N$ and $\mathbf{w} \in \mathbb{ND}_N^{I,+}$. The two parts are orthogonal in $H(\mathbf{curl})$ and also with respect to the bilinear form $a(\cdot, \cdot)$, so that we can decompose and estimate them separately. For gradients, the second term in $a(\cdot, \cdot)$ vanishes, and $a(\mathbf{grad}\, q, \mathbf{grad}\, q) = (\mathbf{grad}\, q, \mathbf{grad}\, q)_0$ is the bilinear form for the Laplace operator in q. Therefore, we can use the domain decomposition theory for scalar elliptic operators and results for the spectral element case for the Laplace equation. Casarin [25, Theorem 3.5.2] proves a bound on the condition number of the additive two-level overlapping Schwarz

preconditioner that coresponds to our preconditioner on the $\mathbf{grad}\, q$ part. His result implies that there is a decomposition $\sum_i q_i$ of q such that

$$\begin{aligned}\sum_i a(\mathbf{grad}\, q_i, \mathbf{grad}\, q_i) &= \eta_1 \sum_i |q_i|_1^2 \leq C N_C \left(1 + \frac{H}{\delta}\right) \eta_1 |q|_1^2 \\ &= C N_C \left(1 + \frac{H}{\delta}\right) a(\mathbf{grad}\, q, \mathbf{grad}\, q) \end{aligned} \tag{11.9}$$

Remark: *Casarin uses the spectral equivalence of a finite element preconditioner on the Gauss-Lobatto-Legendre mesh to the spectral element preconditioner for the spectral elements associated to that mesh, which he proves in his thesis. We do not know of any direct proof by exhibiting a splitting and verifying the assumptions in section 5.2. If one follows the standard proof for generous overlap, one obtains* $C N_C \left(1 + (\frac{H}{\delta})\right)^2$ *instead of* $C N_C \left(1 + \frac{H}{\delta}\right)$*. It would be interesting to see if a direct proof could be constructed extending the small-overlap theory for the* h*-version by Dryja and Widlund [44].*

Now, we will decompose $\mathbf{w}$. First we note that for any decomposition $\mathbf{w} = \sum_i \mathbf{w}_i$ we have

$$\begin{aligned}\sum_i a(\mathbf{w}_i, \mathbf{w}_i) &\leq \max(\eta_1, \eta_2) \sum_i (\mathbf{w}_i, \mathbf{w}_i)_{\mathbf{curl}} = \\ &\leq \max(\eta_1, \eta_2) \sum_i (||\mathbf{w}_i||^2 + ||\,\mathbf{curl}\, \mathbf{w}_i||^2)\end{aligned}$$

and that

$$(||\mathbf{w}||_0^2 + ||\,\mathbf{curl}\, \mathbf{w}||_0^2) \leq \frac{1}{\min(\eta_1, \eta_2)} a(\mathbf{w}, \mathbf{w})$$

Therefore, if we can decompose $\mathbf{w} = \sum_i \mathbf{w}_i$ so that

$$\sum_i (||\mathbf{w}_i||_0^2 + ||\,\mathbf{curl}\, \mathbf{w}_i||_0^2) \leq C_n (||\mathbf{w}||_0^2 + ||\,\mathbf{curl}\, \mathbf{w}||_0^2)$$

the same decomposition will satisfy

$$\sum_i a(\mathbf{w}_i, \mathbf{w}_i) \leq \frac{\max(\eta_1, \eta_2)}{\min(\eta_1, \eta_2)} C_n a(\mathbf{w}, \mathbf{w}) \tag{11.10}$$

In the following, we will introduce a decomposition $\mathbf{w} = \sum_i \mathbf{w}_i$ for which we can estimate C_n.

We will start with the coarse space. To define w_0, we first use the projection P_N into the semicontinuous divergence-free space $V^\perp$, followed by the L^2-projection Q_0 into the coarse space $\mathbb{ND}_{N_0}^I$:

$$\mathbf{w} = \mathbf{w}_0 + \mathbf{v} \qquad \mathbf{w}_0 := Q_0(P_N \mathbf{w}) \in V_0 \qquad \mathbf{v} = (I - Q_0 P_N)\mathbf{w}$$

We will decompose the remainder $\mathbf{v}$ by multiplying it with the partition of unity χ_i and using the Nédélec interpolant to project it back into the local space:

$$\mathbf{w}_i := \mathbf{\Pi}_N^{ND,I}(\chi_i \mathbf{v}) \in V_i$$

First, we estimate $||\,\mathbf{curl}\,\mathbf{w}_0||_0^2$:

$$\begin{aligned} ||\,\mathbf{curl}\,\mathbf{w}_0||_0^2 &= ||\,\mathbf{curl}\,Q_0(P_N\mathbf{w})||_0^2 \leq C|P_N\mathbf{w}|_1^2 \\ &\leq C||\,\mathbf{curl}\,P_N\mathbf{w}||_0^2 \leq C||\,\mathbf{curl}\,\mathbf{w}||_0^2 \end{aligned} \tag{11.11}$$

The first inequality uses property (11.7) from lemma 11.2 on the L^2-projection, the second follows from the imbedding of $H_T(\Omega)$ in $H^1(\Omega)$ (see section 2.7), and the last one by noticing that P_N leaves the $\mathbf{curl}$ of its argument unchanged.

Then, we bound $||\,\mathbf{curl}\,\mathbf{w}_i||_0^2$:

$$\begin{aligned} ||\,\mathbf{curl}\,\mathbf{w}_i||_0^2 &= ||\,\mathbf{curl}\,\mathbf{\Pi}_N^{ND,I}(\chi_i\mathbf{v})||_0^2 \\ &\leq Cf_3^2(N)||\,\mathbf{curl}(\chi_i\mathbf{v})||_0^2 \\ &\leq Cf_3^2(N)\left(||\,\mathbf{grad}\,\chi_i \times \mathbf{v} + \chi_i\,\mathbf{curl}\,\mathbf{v}||_0^2\right) \\ &\leq Cf_3^2(N)\left(||\,\mathbf{grad}\,\chi_i||_{0,\infty,\Omega}^2||\mathbf{v}||_{0,\Omega_i'}^2 + ||\chi_i||_{0,\infty,\Omega_i'}^2||\,\mathbf{curl}\,\mathbf{v}||_{0,\Omega_i'}^2\right) \\ &\leq Cf_3^2(N)\left(\delta^{-2}||\mathbf{v}||_0^2 + ||\,\mathbf{curl}\,\mathbf{v}||_0^2\right) \end{aligned} \tag{11.12}$$

We realize that we have to bound $||\mathbf{v}||_0^2$ and $||\,\mathbf{curl}\,\mathbf{v}||_0^2$ to finish this estimate.

To bound the L^2-norm of $\mathbf{v} = \mathbf{w} - Q_0P_N\mathbf{w}$ we write $\mathbf{v} = \mathbf{w} - P_N\mathbf{w} + P_N\mathbf{w} - Q_0P_N\mathbf{w}$ and use the triangle inequality to obtain

$$||\mathbf{v}||_0^2 \leq ||\mathbf{w} - P_N\mathbf{w}||_0^2 + ||P_N\mathbf{w} - Q_0P_N\mathbf{w}||_0^2$$

We can estimate the first term by lemma 11.1 from the last section, and the second term by property (11.8) from lemma 11.2 on the L^2-projection and the arguments in (11.11):

$$\begin{aligned} ||\mathbf{v}||_0^2 &\leq Ch^2f_1^2(N)||\,\mathbf{curl}\,\mathbf{w}||_0^2 + CH^2||\,\mathbf{curl}\,\mathbf{w}||_0^2 \\ &\leq C(H + hf_1(N))^2||\,\mathbf{curl}\,\mathbf{w}||_0^2 \end{aligned}$$

To estimate $||\,\mathbf{curl}\,\mathbf{v}||_0^2$, we rewrite $\mathbf{curl}\,\mathbf{v} = \mathbf{curl}(\mathbf{w} - \mathbf{w}_0) = \mathbf{curl}\,\mathbf{w} - \mathbf{curl}\,\mathbf{w}_0$, use (11.11) and the triangle inequality to obtain

$$||\,\mathbf{curl}\,\mathbf{v}||_0^2 \leq C||\,\mathbf{curl}\,\mathbf{w}||_0^2$$

Substituting these bounds into (11.12) yields

$$|| \mathbf{curl}\, \mathbf{w}_i||_0^2 \leq C f_3^2(N) \left(1 + \left(\frac{H + h f_1(N)}{\delta}\right)^2\right) || \mathbf{curl}\, \mathbf{w}||_0^2 \tag{11.13}$$

The bound on $||\mathbf{w}_0||_0^2$ follows from the definitions of Q_0 and θ and the remark after their definition:

$$||\mathbf{w}_0||_0^2 = ||Q_0 P_N \mathbf{w}||_0^2 \leq ||P_N \mathbf{w}||_0^2 = ||\Theta \mathbf{w}||_0^2 \leq ||\mathbf{w}||_0^2 \tag{11.14}$$

Finally, using $\mathbf{w}_i = \mathbf{\Pi}_N^{ND,I}(\chi_i(\mathbf{w} - \mathbf{w}_0))$, (11.2), the triangle inequality, and (11.14), we obtain

$$\begin{aligned} ||\mathbf{w}_i||_0^2 &= ||\mathbf{\Pi}_N^{ND,I}(\chi_i(\mathbf{w} - \mathbf{w}_0))||_0^2 \\ &\leq C f_2^2(N) ||\mathbf{w} - \mathbf{w}_0||_0^2 \\ &\leq 2C f_2^2(N) ||\mathbf{w}||_0^2 \end{aligned} \tag{11.15}$$

Adding up (11.11), (11.13) and using the coloring assumption shows

$$\sum_{i=0}^{J} || \mathbf{curl}\, \mathbf{w}_i||_0^2 \leq C \left(1 + N_C f_3^2(N) \left(1 + \left(\frac{H + h f_1(N)}{\delta}\right)^2\right)\right) || \mathbf{curl}\, \mathbf{w}_0||_0^2 \tag{11.16}$$

Similarly, adding (11.14) and (11.15) shows

$$\sum_{i=0}^{J} ||\mathbf{w}_i||_0^2 \leq C(1 + N_C f_2^2(N)) ||\mathbf{w}||_0^2 \tag{11.17}$$

Combining the last two inequalities, we obtain an upper bound for C_n:

$$C_n \leq C \max\left(1 + N_C f_2^2(N), \left(1 + N_C f_3^2(N) \left(1 + \left(\frac{H + h f_1(N)}{\delta}\right)^2\right)\right)\right) \tag{11.18}$$

We derive a bound on the $\mathbf{w}$ part of the decomposition using (11.10). Finally, we combine this bound with the bound from the decomposition of $\mathbf{grad}\, q$ in (11.9) to obtain the bound given in the theorem. ∎

Domain decomposition methods for spectral elements are often used with the spectral elements constituting the subdomains, and therefore $H = h$.

In all cases, $f_1(N) = 1 + C(\epsilon)N^{-1+\epsilon}$, which allows an upper bound $f_1(N) = 1 + C(\epsilon)$. If a better bound on $f_1(N)$ would be proven as mentioned in the proof of 7.17, $f_1(N)$ would go to zero with increasing N. In any case,

$$\left(1 + \left(\frac{H + hf_1(N)}{\delta}\right)^2\right) \leq C\left(1 + \left(\frac{H}{\delta}\right)^2\right).$$

For both the element-wise overlap case and the fixed overlap case $f_2(N) = f_3(N) = 1$, and therefore we obtain after some easy computations:

Corollary 11.4 (Fixed and element-wise overlap) *In the case of element-wise or fixed overlap, the condition number of T_{as2} is bounded by*

$$\kappa(T_{as2}) \leq C(N_c + 1)\frac{\max(\eta_1, \eta_2)}{\min(\eta_1, \eta_2)}\left(1 + N_c\left(1 + \left(\frac{H}{\delta}\right)^2\right)\right)$$

This result corresponds to the result in Toselli [98] and differs only in that it is explicit in N_c (and N).

For the minimal overlap case we obtain the most probably not optimal

Corollary 11.5 (Minimal overlap) *For general δ, an upper bound of the condition number of T_{as2} is given by*

$$\kappa(T_{as2}) \leq C(N_c + 1)N\frac{\max(\eta_1, \eta_2)}{\min(\eta_1, \eta_2)}\left(1 + N_c\left(1 + \left(\frac{H}{\delta}\right)^2\right)\right)$$

For overlaps corresponding to a minimal overlap in the uniform finite element case, i.e. $\delta \sim \frac{1}{N}$, we obtain with numerically estimated (for the case $M = \sqrt{N}$ in section 7.6, following from numerical results not given there) $f_2(N) = f_3(N) = o(N^{0.2})$ a power $N^{0.4}$ instead of N. Any improvement in the bounds of $f_2(N)$ and $f_3(N)$ in the minimal overlap case – possibly using the piecewise spaces $ND^I_{N,\delta}$ and $T_{N,\delta}$ in the interpolation estimates as indicated in section 12.2 – directly results in a improved power of N^c, or even $N^0 = 1$.

It has been noted in Toselli [96, section 3.6] that the estimates for the minimal eigenvalue obtained from a small number of tests with different H/δ allowed the conjecture that also for the h-version for the model problem, the power of $\left(\frac{H}{\delta}\right)$ could be reduced. A numerical test of this conjecture for both the h- and the N-version is possible within our implementation, we intend to perform such tests in future work. The initial tests that we performed for the N-version were inconclusive.

We refer to chapter 10 for some numerical results. There we give numbers of iterations and condition numbers for several settings that explore the condition number estimate in different regimes for N, $H = O(h)$, N_0, and δ.

Bibliography

[1] R. A. Adams. *Sobolev spaces*. Academic Press, New York-London, 1975. Pure and Applied Mathematics, Vol. 65.

[2] M. Ainsworth and J. Coyle. Hierarchic hp-edge element families for Maxwell's equations on hybrid quadrilateral/triangular meshes. *Comp. Meth. Appl. Mech. Eng.*, to appear.

[3] A. Alonso and A. Valli. Some remarks on the characterization of the space of tangential traces of $H(\mathrm{rot};\omega)$ and the construction of an extension operator. *Manuscripta Math.*, 89(2):159–178, 1996.

[4] A. Alonso and A. Valli. An optimal domain decomposition preconditioner for low-frequency time-harmonic Maxwell equations. *Math. Comp.*, 68(226):607–631, 1999.

[5] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault. Vector potentials in three-dimensional non-smooth domains. *Math. Methods Appl. Sci.*, 21(9):823–864, 1998.

[6] D. N. Arnold, R. S. Falk, and R. Winther. Preconditioning in $H(\mathrm{div})$ and applications. *Math. Comp.*, 66(219):957–984, 1997.

[7] D. N. Arnold, R. S. Falk, and R. Winther. Multigrid preconditioning in $H(\mathrm{div})$ on non-convex polygons. *Comput. Appl. Math.*, 17(3):303–315, 1998.

[8] D. N. Arnold, R. S. Falk, and R. Winther. Multigrid in $H(\mathrm{div})$ and $H(\mathrm{curl})$. *Numer. Math.*, 85(2):197–217, 2000.

[9] I. Babuška and B. Q. Guo. Approximation properties of the h-p version of the finite element method. *Comput. Methods Appl. Mech. Engrg.*, 133(3-4):319–346, 1996.

[10] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith. PETSc home page. `http://www.mcs.anl.gov/petsc`, 2001.

[11] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.0, Argonne National Laboratory, 2001.

[12] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994. Available online under http://www.netlib.org/linalg/html_templates/Templates.html.

[13] R. Bartels and G. W. Stewart. Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*, 15(9):820–826, 1972.

[14] R. Beck, R. Hiptmair, R. H. W. Hoppe, and B. Wohlmuth. Residual based a posteriori error estimators for eddy current computation. *M2AN Math. Model. Numer. Anal.*, 34(1):159–182, 2000.

[15] F. Ben Belgacem and C. Bernardi. Spectral element discretization of the Maxwell equations. *Math. Comp.*, 68(228):1497–1520, 1999.

[16] F. Ben Belgacem, A. Buffa, and Y. Maday. The mortar method for the Maxwell's equations in 3D. *C. R. Acad. Sci. Paris Sér. I Math.*, 329(10):903–908, 1999.

[17] C. Bernardi and Y. Maday. Spectral methods. In *Handbook of numerical analysis, Vol. V*, pages 209–485. North-Holland, Amsterdam, 1997.

[18] C. Bernardi, Y. Maday, and A. T. Patera. Domain decomposition by the mortar element method. In *Asymptotic and numerical methods for partial differential equations with critical parameters (Beaune, 1992)*, pages 269–286. Kluwer Acad. Publ., Dordrecht, 1993.

[19] C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991)*, pages 13–51. Longman Sci. Tech., Harlow, 1994.

[20] F. A. Bornemann. An adaptive multilevel approach to parabolic equations I. general theory and 1d-implementation. *IMPACT Comput. Sci. Engrg.*, 2:279–317, 1990.

[21] J. H. Bramble. *Multigrid methods*. Longman Scientific & Technical, Harlow, 1993.

[22] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*. Springer-Verlag, New York, 1994.

[23] P. E. Buis and W. R. Dyksen. Efficient vector and parallel manipulation of tensor products. *ACM Trans. Math. Software*, 22(1):18–23, 1996.

[24] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods in fluid dynamics*. Springer-Verlag, New York, 1988.

[25] M. A. Casarin Jr. *Schwarz Preconditioners for Spectral and Mortar Finite Element Methods with Applications to Incompressible Fluids*. PhD thesis, Courant Institute, New York University, February 1996, also Technical Report, Nr. 717, Department of Computer Science, Courant Institute.

[26] P. Ciarlet, Jr. and J. Zou. Fully discrete finite element approaches for time-dependent Maxwell's equations. *Numer. Math.*, 82(2):193–219, 1999.

[27] P. G. Ciarlet. Basic error estimates for elliptic problems. In *Handbook of numerical analysis, Vol. II*, pages 17–351. North-Holland, Amsterdam, 1991.

[28] P. Clément. Approximation by finite element functions using local regularization. *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér.*, 9(R-2):77–84, 1975.

[29] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3):251–280, 1990.

[30] R. Courant and D. Hilbert. *Methoden der Mathematischen Physik. Vols. I, II*. Interscience Publishers, Inc., N.Y., 1943.

[31] M. Dauge. *Elliptic boundary value problems on corner domains*. Springer-Verlag, Berlin, 1988. Smoothness and asymptotics of solutions.

[32] R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 2*. Springer-Verlag, Berlin, 1988. Functional and variational methods, With the collaboration of Michel Artola, Marc Authier, Philippe Bénilan, Michel Cessenat, Jean Michel Combes, Hélène Lanchon, Bertrand Mercier, Claude Wild and Claude Zuily, Translated from the French by Ian N. Sneddon.

[33] R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 1*. Springer-Verlag, Berlin, 1990. Physical origins and classical methods, With the collaboration of Philippe Bénilan, Michel Cessenat, André Gervat, Alain Kavenoky and Hélène Lanchon, Translated from the French by Ian N. Sneddon, With a preface by Jean Teillac.

[34] R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 3*. Springer-Verlag, Berlin, 1990. Spectral theory and applications, With the collaboration of Michel Artola and Michel Cessenat, Translated from the French by John C. Amson.

[35] R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 4*. Springer-Verlag, Berlin, 1990. Integral equations and numerical methods, With the collaboration of Michel Artola, Philippe Bénilan, Michel Bernadou, Michel Cessenat, Jean-Claude Nédélec, Jacques Planchard and Bruno Scheurer, Translated from the French by John C. Amson.

[36] R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 5*. Springer-Verlag, Berlin, 1992. Evolution problems. I, With the collaboration of Michel Artola, Michel Cessenat and Hélène Lanchon, Translated from the French by Alan Craig.

[37] R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 6*. Springer-Verlag, Berlin, 1993. Evolution problems. II, With the collaboration of Claude Bardos, Michel Cessenat, Alain Kavenoky, Patrick Lascaux, Bertrand Mercier, Olivier Pironneau, Bruno Scheurer and Rémi Sentis, Translated from the French by Alan Craig.

[38] P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Academic Press Inc., Orlando, FL, second edition, 1984.

[39] C. de Boor. Corrigenda: "Efficient computer manipulation of tensor products" (ACM Trans. Math. Software **5** (1979), no. 2, 173–182). *ACM Trans. Math. Software*, 5(4):525, 1979.

[40] C. de Boor. Efficient computer manipulation of tensor products. *ACM Trans. Math. Software*, 5(2):173–182, 1979.

[41] L. Demkowicz. Edge finite elements of variable order for Maxwell's equations - a discussion. Technical Report TICAM Report 00-32, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 2000.

[42] L. Demkowicz and I. Babuška. Optimal p interpolation error estimates for edge finite elements of variable order in 2D. Technical Report TICAM Report 01-11, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 2001.

[43] M. Dryja, B. F. Smith, and O. B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.*, 31(6):1662–1694, 1994.

[44] M. Dryja and O. B. Widlund. Domain decomposition algorithms with small overlap. *SIAM J. Sci. Comput.*, 15(3):604–620, 1994. Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992).

[45] X. Feng. Absorbing boundary conditions for electromagnetic wave propagation. *Math. Comp.*, 68(225):145–168, 1999.

[46] P. R. Garabedian. *Partial differential equations*. John Wiley & Sons Inc., New York, 1964.

[47] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler. Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Trans. Math. Software*, 18(2):223–231, 1992.

[48] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations*. Springer-Verlag, Berlin, 1986. Theory and algorithms.

[49] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[50] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1997.

[51] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numerische Mathematik*, 70:163–180, 1995.

[52] P. Grisvard. *Elliptic problems in nonsmooth domains*. Pitman, Boston, MA, 1985.

[53] W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssyteme*. Teubner, 1993.

[54] W. Hackbusch. *Iterative solution of large sparse systems of equations*. Springer-Verlag, New York, 1994.

[55] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I.* Springer-Verlag, Berlin, second edition, 1993. Nonstiff problems.

[56] B. Hientzsch. *Theoretische und numerische Untersuchungen zur Burgersgleichung (German) [Theoretical and numerical investigations on the Burgers equation]*. GMD-Studien [GMD Studies], 269. GMD—Forschungszentrum Informationstechnik GmbH, St. Augustin, 1995. With a foreword by Ulrich Trottenberg.

[57] R. Hiptmair. *Multilevel preconditioning for mixed problems in three dimensions*. PhD thesis, Mathematisches Institut, Universität Augsburg, Germany, March 1996.

[58] R. Hiptmair. Multigrid method for $\mathbf{H}(\mathrm{div})$ in three dimensions. *Electron. Trans. Numer. Anal.*, 6(Dec.):133–152 (electronic), 1997. Special issue on multilevel methods (Copper Mountain, CO, 1997).

[59] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM J. Numer. Anal.*, 36(1):204–225 (electronic), 1999.

[60] R. Hiptmair and A. Toselli. Overlapping and multilevel Schwarz methods for vector valued elliptic problems in three dimensions. In P. E. Bjørstad and M. Luskin, editors, *Parallel Solution of Partial Differential Equations*, pages 181–208, New York, 2000. Springer Verlag.

[61] X. Huang and V. Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complexity*, 14(2):257–299, 1998.

[62] B. Kågström and P. Poromaa. LAPACK-style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs. *ACM Trans. Math. Software*, 22(1):78–103, 1996.

[63] P. A. Knight. Fast rectangular matrix multiplication and QR decomposition. *Linear Algebra Appl.*, 221:69–81, 1995.

[64] J. Laderman, V. Pan, and X. H. Sha. On practical algorithms for accelerated matrix multiplication. *Linear Algebra Appl.*, 162/164:557–588, 1992. Directions in matrix theory (Auburn, AL, 1990).

[65] P. D. Lax. *Linear algebra*. John Wiley & Sons Inc., New York, 1997. A Wiley-Interscience Publication.

[66] Q. Lin and N. Yan. Global superconvergence for Maxwell's equations. *Math. Comp.*, 69(229):159–176, 2000.

[67] R. E. Lynch, J. R. Rice, and D. H. Thomas. Direct solution of partial difference equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.

[68] Y. Maday and E. M. Rønquist. Optimal error analysis of spectral methods with emphasis of non-constant coefficients and deformed geometries. Technical report, Université Pierre et Marie Curie, Centre National de la Recherche Scientifique, January 1990. Publications du Laboratoire D'Analyse Numérique R 89030.

[69] J. Mandel and G. S. Lett. Domain decomposition preconditioning for the p-version finite element version in three dimensions. *Int. J. Numer. Meth. Eng.*, 29:1095–1108, 1991.

[70] P. Monk. An analysis of Nédélec's method for the spatial discretization of Maxwell's equations. *J. Comput. Appl. Math.*, 47(1):101–121, 1993.

[71] P. Monk. On the p- and hp-extension of Nédélec's curl-conforming elements. *J. Comput. Appl. Math.*, 53(1):117–137, 1994.

[72] P. Monk. A posteriori error indicators for Maxwell's equations. *J. Comput. Appl. Math.*, 100(2):173–190, 1998.

[73] J. Nečas. *Les méthodes directes en théorie des équations elliptiques*. Masson et Cie, Éditeurs, Paris, 1967.

[74] J.-C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 35(3):315–341, 1980.

[75] J.-C. Nédélec. A new family of mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 50(1):57–81, 1986.

[76] S. Nicaise. Edge elements on anisotropic meshes and approximation of the Maxwell equations. *SIAM Journal on Numerical Analysis*, 39(3):784–816, 2001.

[77] D. P. O'Leary and O. B. Widlund. Capacitance matrix methods for the Helmholtz equation on general three dimensional regions. *Math. Comp.*, 33:849–879, 1979.

[78] P. Oswald. *Multilevel finite element approximation*. B. G. Teubner, Stuttgart, 1994. Theory and applications.

[79] V. Pan. How can we speed up matrix multiplication? *SIAM Rev.*, 26(3):393–415, 1984.

[80] V. Pan. *How to multiply matrices faster*. Springer-Verlag, Berlin, 1984.

[81] V. Pan. Complexity of computations with matrices and polynomials. *SIAM Rev.*, 34(2):225–262, 1992.

[82] L. F. Pavarino. *Domain Decomposition Algorithms for the p-version Finite Element Method for Elliptic Problems*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, September 1992. also Technical Report, Nr. 616, Department of Computer Science, Courant Institute.

[83] V. Pereyra and G. Scherer. Efficient computer manipulation of tensor products with applications to multidimensional approximation. *Math. Comput.*, 27:595–605, 1973.

[84] J. S. Przemieniecki. *Theory of matrix structural analysis*. Dover Publications Inc., New York, 1985. Reprint of the 1968 original.

[85] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer-Verlag, Berlin, 1994.

[86] Y. Saad. *Iterative methods for sparse linear systems*. PWS, 1996. A corrected second edition (2000) is available online under http://www-users.cs.umn.edu/~saad/books.html.

[87] Y. Saad and M. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7(3):856–869, 1986.

[88] M. Schemann and F. A. Bornemann. An adaptive Rothe method for the wave equation. *Computing and Visualization in Science*, 1(3):137–144, 1998.

[89] L. R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Math. Comp.*, 54(190):483–493, 1990.

[90] R. E. Showalter. *Hilbert space methods for partial differential equations*. Electronic Monographs in Differential Equations, San Marcos, TX, 1994. Electronic reprint of the 1977 original.

[91] B. F. Smith, P. E. Bjørstad, and W. D. Gropp. *Domain decomposition*. Cambridge University Press, Cambridge, 1996. Parallel multilevel methods for elliptic partial differential equations.

[92] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.

[93] M. Suri. On the stability and convergence of higher-order mixed finite element methods for second-order elliptic problems. *Math. Comp.*, 54(189):1–19, 1990.

[94] G. Szegő. *Orthogonal polynomials*. American Mathematical Society, Providence, R.I., fourth edition, 1975. Colloquium Publications, Vol. XXIII.

[95] A. F. Timan. *Theory of approximation of functions of a real variable*. A Pergamon Press Book. The Macmillan Co., New York, 1963.

[96] A. Toselli. *Domain Decomposition Methods for Vector Field Problems*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, May 1999. also Technical Report, Nr. 785, Department of Computer Science, Courant Institute.

[97] A. Toselli. Neumann-Neumann methods for vector field problems. *Electron. Trans. Numer. Anal.*, 11:1–24 (electronic), 2000.

[98] A. Toselli. Overlapping Schwarz methods for Maxwell's equations in three dimensions. *Numer. Math.*, 86(4):733–752, 2000.

[99] A. Toselli and A. Klawonn. A FETI domain decomposition method for Maxwell's equations with discontinuous coefficients in two dimensions. Technical Report 788, Courant Institute, New York University, September 1999. Appeared in: *SIAM J. Numer. Anal.* 39(3):932-956, 2001.

[100] A. Toselli and F. Rapetti. A FETI preconditioner for two dimensional edge element approximations of Maxwell's equations on non-matching grids. Technical Report 797, Courant Institute, New York University, January 2000. Appeared as: F. Rapetti, A. Toselli, *SIAM J. Sci. Comp.* 23(1):92-108, 2001.

[101] A. Toselli, O. B. Widlund, and B. I. Wohlmuth. An iterative substructuring method for Maxwell's equations in two dimensions. *Math. Comp.*, 70(235):935–949, 2001.

[102] O. B. Widlund. Domain decomposition methods for elliptic partial differential equations. In *Error control and adaptivity in scientific computing (Antalya, 1998)*, pages 325–354. Kluwer Acad. Publ., Dordrecht, 1999.

[103] B. I. Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM J. Numer. Anal.*, 38(3):989–1012 (electronic), 2000.

[104] B. I. Wohlmuth, A. Toselli, and O. B. Widlund. An iterative substructuring method for Raviart-Thomas vector fields in three dimensions. *SIAM J. Numer. Anal.*, 37(5):1657–1676 (electronic), 2000.

[105] H.-M. Yin. On Maxwell's equations in an electromagnetic field with the temperature effect. *SIAM J. Math. Anal.*, 29(3):637–651 (electronic), 1998.

[106] T. Zang and D. B. Haidvogel. The accurate solution of Poisson's equation by expansion in Chebyshev polynomials. *J. Comput. Phys.*, 30(2):167–180, 1979.