Copyright

by

Jason Patrick Kurtz

2007

The Dissertation Committee for Jason Patrick Kurtz
certifies that this is the approved version of the following dissertation:

Fully Automatic *hp*-Adaptivity for Acoustic and Electromagnetic Scattering in Three Dimensions

Committee:

Leszek Demkowicz, Supervisor

Ivo M. Babuška

Luis A. Caffarelli

Hao Ling

J. Tinsley Oden

Robert A. van de Geijn

Fully Automatic *hp*-Adaptivity for Acoustic and Electromagnetic Scattering in Three Dimensions

by

Jason Patrick Kurtz, M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2007

To my parents.

Without your love, support and encouragement,
none of this would have been possible.

Acknowledgments

None of this would have been possible without the foundation and guidance of my advisor and friend, Professor Leszek Demkowicz. I could not hope for a better role model, both in academics and in life. I am also very grateful to my committee members, and other professors at ICES and The University of Texas at large, for the many classes and seminars I had the pleasure to attend over the past few years.

Financial support for this work was provided by the Air Force Office of Scientific Research, for which I am particularly grateful to Dr. Arje Nachman.

Finally, I thank the many professors and friends that contributed to my early education and growth at Clarkson University. I owe a special debt of gratitude to Professor Kevin Dempsey, who taught me the beauty of mathematics and convinced me to pursue a deeper study, and to Professor Christos Xenophontos, who introduced me to the finite element method. Most importantly, I never would have survived without the support and friendship of the triumvirate - you know who you are.

Jason Patrick Kurtz

The University of Texas at Austin
May 2007

Fully Automatic *hp*-Adaptivity for Acoustic and Electromagnetic Scattering in Three Dimensions

Publication No. ____________________

Jason Patrick Kurtz, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Leszek Demkowicz

We present an algorithm for fully automatic *hp*-adaptivity for finite element approximations of elliptic and Maxwell boundary value problems in three dimensions. The algorithm automatically generates a sequence of coarse grids, and a corresponding sequence of fine grids, such that the energy norm of the error decreases exponentially with respect to the number of degrees of freedom in either sequence. At each step, we employ a discrete optimization algorithm to determine the refinements for the current coarse grid such that the projection-based interpolation error for the current fine grid solution decreases with an optimal rate with respect to the number of degrees of freedom added by the refinement. The refinements are restricted only by the requirement that the resulting mesh is at most 1-irregular, but they may be anisotropic in both element size h and order of approximation p.

While we cannot prove that our method converges at all, we present numerical evidence of exponential convergence for a diverse suite of model problems from acoustic and electromagnetic scattering. In particular we show that our method is well suited to the automatic resolution of exterior problems truncated by the introduction of a perfectly matched layer. To enable and accelerate the solution of these problems on commodity hardware, we include a detailed account of three critical aspects of our implementation, namely an efficient implementation of sum factorization, several efficient interfaces to the direct multi-frontal solver MUMPS, and some fast direct solvers for the computation of a sequence of nested projections.

Contents

Acknowledgments **v**

Abstract **vi**

Chapter 1 Introduction **1**

1.1 Motivation 1

1.2 Review of *hp* and other high-order methods 2

1.3 Scope of this work 5

Chapter 2 Formulation of the Problems of Interest **7**

2.1 Elliptic PDE in a bounded domain 8

2.2 Acoustic scattering 9

2.2.1 Truncation by infinite elements 10

2.2.2 Truncation by a perfectly matched layer 14

2.3 Electromagnetic scattering 19

2.3.1 Truncation by an impedance boundary condition 20

2.3.2 Truncation by infinite elements 24

2.3.3 Truncation by a perfectly matched layer 28

Chapter 3 Compatible *hp*-FEM **32**

3.1 The exact sequence property and de Rham diagram 32

3.2 H^1-conforming *hp* elements 33
3.3 H(curl)-conforming *hp* elements 35

Chapter 4 Fully-Automatic *hp*-Adaptivity for Elliptic Equations 38
4.1 Projection-based interpolation in H^1 39
4.2 Algorithm for fully-automatic *hp*-adaptivity 43
  4.2.1 Overview 44
  4.2.2 The edge refinement algorithm 45
  4.2.3 The face refinement algorithm 50
  4.2.4 The brick refinement algorithm 54
  4.2.5 Mesh reconciliation 55
4.3 Abstract framework for computing projections 57

Chapter 5 Numerical Results 61
5.1 Fichera's corner 61
5.2 An exact solution with a "shock" 65
5.3 Acoustic scattering from a sphere 70
5.4 Acoustic scattering from a cone-sphere 80
5.5 Acoustic scattering from a thin square plate 87

Chapter 6 Fully-Automatic *hp*-Adaptivity for Maxwell's Equations 92
6.1 Projection-based Interpolation in $\boldsymbol{H}(curl)$ 92
6.2 Computational issues 95
6.3 Numerical Results II 97

Chapter 7 High Performance Computation 120
7.1 Sum factorization 121
7.2 Optimized interface to the MUltifrontal Massively Parallel Solver (MUMPS) 127

7.2.1 The symmetric positive-definite case 128

7.2.2 A domain decomposition solver for acoustics and electromagnetics 131

7.3 Fast solvers for projection-based interpolation 134

7.3.1 The symmetric positive-definite case 135

7.3.2 The symmetric indefinite case 138

Chapter 8 Concluding Remarks **143**

Bibliography **145**

Vita **152**

Chapter 1

Introduction

1.1 Motivation

Since the late 60's and early 70's, the development of low-observable aircraft has been a major focus for defense organizations the world over. The idea is to design aircraft which can evade detection by a variety of systems, including infrared, optical, acoustic and electromagnetic sensors. It is a common misconception that these efforts were successful and that the resulting aircraft are entirely invisible to radar and other methods of detection. In fact, radar technology has developed apace with stealth technology and there is a constant battle between attempting to detect, and attempting to evade detection.

If we focus in particular on low observability to radar then the goal is to design aircraft with small radar cross section (RCS). The RCS of a target is a range independent quantity that measures the ratio of the energy reflected from a target (toward a receiving radar platform) to the energy incident on the target (from a transmitting radar platform, modeled as a plane wave). If the transmitter and receiver are far apart, it is known as bi-static RCS; if they are close together or even one and the same, it is known as mono-static RCS. Apart from the obvious dependence on direction, it is well-known [35] that RCS depends on the frequency and polarization of the incident wave and on the geometry and composition of the target.

For example, it is well-known that a corner reflector (a re-entrant, 90 degree angle) produces a large, focused response in the direction of the transmitter, and that flat surfaces or edges produce focused responses in the direction of reflection governed by Snell's law. When such geometry cannot be avoided, say for aerodynamic reasons, a common practice is to use a thin coating of an absorbing material to mask the response.

To accelerate the design cycle for low-observable aircraft, the need for computational simulation of RCS is obvious. The ultimate goal is the simulation of scattering from an entire aircraft. However, much can be learned from the scattering characteristics of much simpler (and electrically smaller) objects. Moreover, a very high degree of accuracy is required from these simulations in an age where radar platforms are able to detect such insignificant scatterers as a swarm of insects [35].

1.2 Review of *hp* and other high-order methods

The finite element method can be seen most simply as a method for discretizing and obtaining an approximate solution to a partial differential equation on a bounded domain. The domain is subdivided into elements and the approximate solution defined as a piecewise polynomial with respect to the elements. As soon as one wants to have some quantitative measure of, and control over, the discretization error, refinement strategies enter the picture.

The most classical strategy is the h-version of the finite element method, where convergence is achieved by decreasing, either uniformly or in an adaptive way, the size h of the elements used in the discretization. The polynomial order of approximation p, is fixed and usually quite low. When the refinements are uniform, the h-version converges algebraically with a rate limited by the order of approximation *and* by the regularity of the solution. When the solution is singular, the refinements can be performed adaptively to restore the optimal convergence rate dictated by the order of approximation alone [6].

By the late 70's a new refinement strategy, dubbed the p-version, was under investigation ([8], and later [7]). Here, the finite element mesh is fixed and convergence is achieved by increasing p, either uniformly or in an adaptive way. It was demonstrated that in the presence of singularities, uniform p-enrichment achieved twice the (algebraic) rate of convergence for (uniform or quasi-uniform) h-refinement. Moreover, for a smooth solution, the p-version converges *exponentially.*

The hp-version of the finite element method combines these two refinement strategies. First proposed by Babuška and Dorr in 1981 [5], hp methods have been a hot topic for research ever since. The pioneering contributions of Gui and Babuška [32] and Guo and Babuška [33] set the stage. In particular it was shown that with a proper combination of local h and p refinement, exponential convergence could be achieved even in the presence of singularities. In the following years there was an explosion of theoretical developments related to hp-methods (see e.g. [54] and the references therein).

Apart from difficulties related to analysis, hp methods present many challenges in terms of their practical implementation. Devloo, Oden and Pattani [30] were the first to successfully implement a limited hp code in the context of 2D compressible flow. A pioneering effort by Demkowicz, Oden and Rachowicz [24, 40, 49] soon produced the first 3D hp code. This early code was limited to hexahedral elements and supported only *isotropic* hp-refinements. On the commercial side, this code provided the starting point for the general hp kernel PHLEX[1], which extended

---

[1] PHLEX is a trademark of Altair Engineering

the functionality to general unstructured meshes and *anisotropic hp*-refinements. On the academic side, this was the starting point for over 15 years of research in *hp*-methods and code development by Demkowicz and collaborators. In an iterative process the code was successively rewritten adding anisotropic *hp*-refinements [21] and a nodal data structure [26] making these refinements more manageable. Independent *hp* code development efforts include those of Schöberl (Netgen/NGSolve [53, 52]) and Frauenfelder (Concepts [31]).

While *hp*-methods were initially applied mainly to problems in mechanics, theoretical and practical advances have brought them to bear on electromagnetics. Demkowicz and Vardapetyan introduced a class of high-order $H(curl)$-conforming finite elements [29] that could be extended by infinite elements [25] to solve scattering problems. The first 3D implementation was completed by Rachowicz, Cecot and Demkowicz [47, 48, 15]. This code has been extended by Rachowicz and Zdunek to solve some non-trivial scattering problems with *hp*-adaptivity [51].

The availability of general (i.e. fully anisotropic) *hp*-codes has motivated a parallel line of research on fully-automatic *hp*-adaptivity. The goal here is the development of a problem-independent algorithm to determine (from a given initial grid) a sequence of optimal *hp*-grids such that the corresponding FE solutions actually exhibit the exponential convergence indicated by the theory. Such methods are classified as *energy driven* when the measure of convergence is the energy norm, and as *goal driven* when the error is measured only in some quantity of interest. Energy driven methods have been developed for both elliptic problems [27, 50] and Maxwell's equations [19, 51]. For the generalization of these algorithms to goal driven adaptivity, see e.g. [42, 51] and the references therein.

1.3 Scope of this work

In the sequel we will describe a general approach for fully-automatic (energy-driven) *hp*-adaptive finite element methods. The underlying principles are relevant for problems posed in one or two spatial dimensions, but our focus is to demonstrate their relevance for full three dimensional computations. In an abstract form, our algorithm can be applied to variational problems posed in any of the spaces H^1, $\boldsymbol{H}(\mathrm{curl})$, $\boldsymbol{H}(\mathrm{div})$ and L^2, related by the standard three dimensional exact sequence but our present discussion and implementation are limited to H^1 and $\boldsymbol{H}(\mathrm{curl})$ only. In this context, our focus is limited to methods that are H^1 or $\boldsymbol{H}(\mathrm{curl})$-conforming (technically, non-conformity is limited to approximate geometry or Dirichlet data) but our implementation is such that future application to non-conforming methods is feasible. Finally, for simplicity, our implementation is limited to hexahedral meshes only.

Within this limited context, our method is quite general. Most notably, our refinement strategy is in general anisotropic in both element size h and order of approximation p. That is, an element may be broken in any of seven ways, and the order of approximation may be set independently for each of the three coordinate directions, in each of the resulting son elements. This freedom enables the generation of optimal meshes for problems with diverse features such as corner and edge singularities and boundary or interior layers.

In Chapter 2 we derive standard variational formulations for the problems to be considered in the sequel, namely, elliptic boundary value problems, and the exterior problems associated with acoustic and electromagnetic scattering. In Chapter 3 we recall the main ideas behind the construction of finite dimensional subspaces of H^1 and $\boldsymbol{H}(\mathrm{curl})$ (and modified versions encountered in Chapter 2) that allow for local variation of the element size h and order of approximation p. In Chapter 4 we introduce our algorithm in the H^1 setting and discuss some of the critical features behind our implementation. Chapter 5 contains a thorough computational assessment of the algorithm for both regular and singular solutions to the Laplace equation, and for rigid acoustic scattering from both smooth and geometrically singular obstacles. In Chapter 6 we extend our algorithm to the $\boldsymbol{H}(\mathrm{curl})$ setting and present preliminary evidence of exponential convergence for electromagnetic scattering from a hexahedral obstacle imbedded in a waveguide. Chapter 7 discusses several technologies critical to our present implementation and certainly relevant for any high-order finite element method in three dimensions. Finally, we conclude in Chapter 8 with some final remarks and a look to the future.

Chapter 2

Formulation of the Problems of Interest

In this chapter we present variational formulations for the three main classes of problems targeted by this work. The problems represent an increasing level of difficulty for the development and application of automatic *hp*-adaptivity. The first class is a system of linear elliptic PDEs in a bounded domain, from which we will focus on the representative Laplace equation. The second is the scattering of time-harmonic acoustic waves from a bounded obstacle into an unbounded exterior region. Here, we will either model the exterior region with infinite elements (IE) or truncate it by introducing a perfectly matched layer (PML). Finally, we turn to the time-harmonic Maxwell equations and formulate the scattering problem either in an infinite rectangular waveguide truncated by an impedance boundary condition, or a general exterior domain truncated by infinite elements or PML.

2.1 Elliptic PDE in a bounded domain

Let $\Omega \subset I\!R^3$ be a bounded domain with Lipschitz boundary Γ. For simplicity we restrict our attention to Poisson's equation,

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega, && (2.1) \\ u &= g \text{ on } \Gamma_D, && (2.2) \\ \frac{\partial u}{\partial n} &= h \text{ on } \Gamma_N, && (2.3) \end{aligned}$$

where $\Gamma = \overline{\Gamma_D} \cup \overline{\Gamma_N}$, $\Gamma_D \cap \Gamma_N = \emptyset$. We assume $g \in H^{1/2}(\Gamma_D)$ so that it can be extended to $u_D \in H^1(\Omega)$. Then our test space is

$$V = \left\{ v \in H^1(\Omega) : \gamma v|_{\Gamma_D} = 0 \right\},$$

where $\gamma : H^1(\Omega) \to H^{1/2}(\Gamma)$ is the trace operator. The standard variational formulation reads:

$$\left\{ \begin{array}{l} \text{Find } u \in u_D + V \text{ such that} \\ b(u,v) \equiv \displaystyle\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \; dx = \int_\Omega f v \; dx + \int_{\Gamma_N} g v \; d\Gamma \equiv l(v) \text{ for all } v \in V. \end{array} \right. \tag{2.4}$$

For a given finite element space $V^{hp} \subset H^1(\Omega)$, we compute approximate Dirichlet data $u_D^{hp} \in V^{hp}$ and solve the discrete problem:

$$\left\{ \begin{array}{l} \text{Find } u^{hp} \in u_D^{hp} + V_0^{hp} \text{ such that} \\ b(u^{hp}, v) = l(v) \text{ for all } v \in V_0^{hp}, \end{array} \right.$$

where $V_0^{hp} = \left\{ v \in V^{hp} : v|_{\Gamma_D} = 0 \right\}$.

2.2 Acoustic scattering

For our second application we consider the problem of time-harmonic acoustic scattering from a bounded obstacle. Here, an obstacle is excited by an incident pressure wave and we seek the far-field pattern of the response. In principal, the incident wave could be generated by modeling an actual source, but under the assumption that the source is far from the obstacle, we approximate the incident wave by a plane wave,

$$p^{\mathrm{inc}}(\boldsymbol{x}) = p_0^{inc} e^{ik\widehat{\boldsymbol{e}}\cdot\boldsymbol{x}}.$$

Here, $k = \omega/c$ is the wave number, depending on the angular frequency ω and speed of sound c, and $\widehat{\boldsymbol{e}}$ is the direction of the source. The incident wave p^{inc} and the scattered wave p are complex-valued, with the associated time-dependent quantities given by,

$$P^{\mathrm{inc}}(\boldsymbol{x}, t) = \Re(p^{\mathrm{inc}}(\boldsymbol{x})e^{i\omega t}),\ P(\boldsymbol{x}, t) = \Re(p(\boldsymbol{x})e^{i\omega t}).$$

The obstacle is assumed to occupy a bounded region $\Omega_{\mathrm{int}} \subset I\!R^3$ with Lipschitz boundary Γ. The scattered pressure p satisfies the (homogeneous) Helmholtz equation outside the obstacle,

$$-\Delta p - k^2 p = 0 \text{ in } \Omega = I\!R^3 \setminus \Omega_{\mathrm{int}}, \tag{2.5}$$

along with the Neumann boundary condition (for the case of a rigid scatterer),

$$\frac{\partial p}{\partial n} = g = -\frac{\partial p^{\mathrm{inc}}}{\partial n} \text{ on } \Gamma, \tag{2.6}$$

and the Sommerfeld radiation condition,

$$\frac{\partial p}{\partial r} + ikp = w = o(r^{-1}) \text{ as } r \to \infty. \tag{2.7}$$

2.2.1 Truncation by infinite elements

Infinite elements, first introduced by P. Bettess [11], are a popular technique for truncating exterior problems like (2.5)–(2.7). A wide variety of formulations have since been developed that can be characterized as conjugated vs unconjugated, based on whether or not (2.5) is multiplied by the complex conjugate of a test function, and Bubnov–Galerkin vs Petrov–Galerkin, based on whether the same or different test and trial spaces are used. Here we recall the conjugated Bubnov–Galerkin formulation introduced in [28].

In the region exterior to a sphere large enough to enclose the obstacle, the solution to (2.5)–(2.7) can be represented by the Atkinson-Wilcox expansion,

$$p(\boldsymbol{r}) = \frac{e^{-ikr}}{r} \sum_{n=0}^{\infty} \frac{u_n(\psi, \theta)}{r^n} = \frac{e^{-ikr}}{r} P(\boldsymbol{r}),$$

where r, ψ, θ are the standard spherical coordinates (Figure 2.3). The main idea behind infinite elements is to remove the exponential phase factor e^{-ikr}/r and focus on the approximation of the remainder P.

We surround the obstacle with a truncating sphere S_a of radius a, and an additional sphere S_R of radius R, with the intent of allowing $R \to \infty$. In the end, we will discretize the problem with a finite element mesh in the bounded domain $\Omega^a = \{\boldsymbol{x} : |x| < a\} \setminus \overline{\Omega_{\text{int}}}$, coupled with a compatible infinite element mesh in the unbounded domain $\Omega_a^\infty = \{\boldsymbol{x} : |x| > a\}$. The geometry for finite R is illustrated in Figure 2.1.

We multiply equation (2.5) by the complex conjugate of a test function q, integrate over the truncated exterior domain $\Omega^a \cup \Omega_a^R$, and integrate by parts (making use of the boundary condition (2.6) and radiation condition (2.7)) to obtain,

$$\int_{\Omega^a \cup \Omega_a^R} \{\boldsymbol{\nabla} p \cdot \boldsymbol{\nabla} \overline{q} - k^2 p \overline{q}\}\, dx + ik \int_{S_R} p \overline{q}\, dS_R = \int_{\Gamma} g \overline{q}\, d\Gamma + \int_{S_R} w \overline{q}\, dS_R. \qquad (2.8)$$

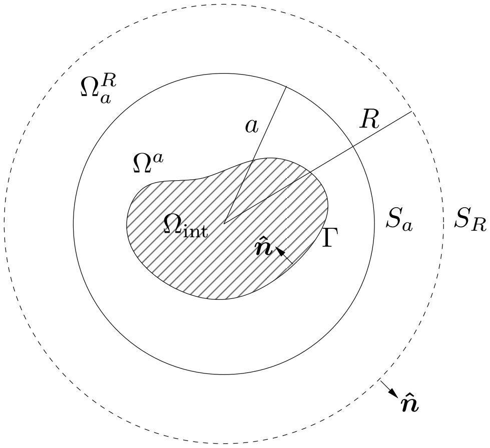


Figure 2.1: Geometry for rigid acoustic scattering

We assume $q = O(R^{-1})$ (the same rate as p) so that the right-most integral vanishes as $R \to \infty$. The limiting behavior of the remaining terms as $R \to \infty$ requires much more care. First, we split the sesquilinear form on the left hand side of (2.8) into the two contributions,

$$b(p, q) = b_a(p, q) + b_\infty(p, q), \tag{2.9}$$

where

$$b_a(p, q) = \int_{\Omega^a} \{\boldsymbol{\nabla} p \cdot \boldsymbol{\nabla} \overline{q} - k^2 p \overline{q}\} \; dx, \tag{2.10}$$

and

$$b_\infty(p, q) = \lim_{R \to \infty} \int_{\Omega_a^R} \{\boldsymbol{\nabla} p \cdot \boldsymbol{\nabla} \overline{q} - k^2 p \overline{q}\} \; dx + ik \int_{S_R} p \overline{q} \; dS_R. \tag{2.11}$$

To proceed with the interpretation of (2.11) we must first rewrite it in a coordinate system that illuminates the dependence on R. We begin with a parametrization of the truncating sphere S_a,

$$\boldsymbol{x} = \boldsymbol{x}_a(\xi_1, \xi_2), \;\; |x_a| = a$$

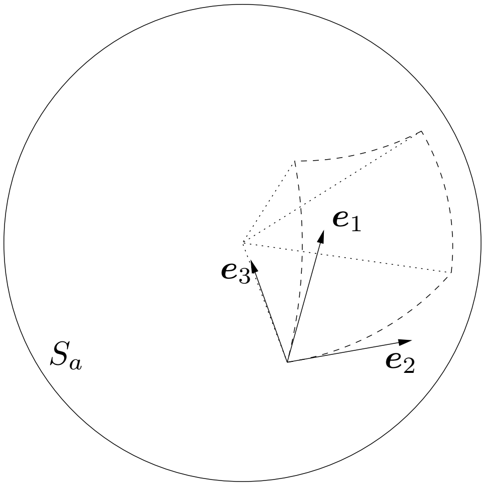


Figure 2.2: Infinite element coordinates

and extend it to define a parametrization of the exterior domain Ω_a^R,

$$\boldsymbol{x} = \xi_3^{-1}\boldsymbol{x}_a(\xi_1,\xi_2),\ a/R < \xi_3 < 1. \tag{2.12}$$

This parametrization, involving the scaled and inverted radial coordinate ξ_3, is admittedly non-intuitive, but we use it from the beginning in order to be consistent with the actual implementation, where $\boldsymbol{\xi}$ represents the reference coordinates for an infinite element. This parametrization introduces a (not necessarily orthogonal) system of curvilinear coordinates with basis $\{\boldsymbol{a}_i\}_{i=1}^3$ (and unit basis $\{\boldsymbol{e}_i\}_{i=1}^3$) given by,

$$\begin{aligned}
\boldsymbol{a}_1 = \frac{\partial \boldsymbol{x}}{\partial \xi_1} &= \frac{1}{\xi_3}\frac{\partial \boldsymbol{x}_a}{\partial \xi_1} &&= \frac{1}{\xi_3}\left|\frac{\partial \boldsymbol{x}_a}{\partial \xi_1}\right|\boldsymbol{e}_1 &&\Rightarrow\quad \boldsymbol{e}_1 = \left|\frac{\partial \boldsymbol{x}_a}{\partial \xi_1}\right|^{-1}\frac{\partial \boldsymbol{x}_a}{\partial \xi_1}\\
\boldsymbol{a}_2 = \frac{\partial \boldsymbol{x}}{\partial \xi_2} &= \frac{1}{\xi_3}\frac{\partial \boldsymbol{x}_a}{\partial \xi_2} &&= \frac{1}{\xi_3}\left|\frac{\partial \boldsymbol{x}_a}{\partial \xi_2}\right|\boldsymbol{e}_2 &&\Rightarrow\quad \boldsymbol{e}_2 = \left|\frac{\partial \boldsymbol{x}_a}{\partial \xi_2}\right|^{-1}\frac{\partial \boldsymbol{x}_a}{\partial \xi_2}\\
\boldsymbol{a}_3 = \frac{\partial \boldsymbol{x}}{\partial \xi_3} &= -\frac{1}{\xi_3^2}\boldsymbol{x}_a &&= \frac{a}{\xi_3^2}\boldsymbol{e}_3 &&\Rightarrow\quad \boldsymbol{e}_3 = -\frac{1}{a}\boldsymbol{x}_a
\end{aligned} \tag{2.13}$$

(possibly renumbered to form a right-handed triple as shown in Figure 2.2). The corresponding cobasis is then given by,

$$\boldsymbol{a}^1 = \xi_3 J_a^{-1} \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} \times \boldsymbol{e}_3, \ \boldsymbol{a}^2 = \xi_3 J_a^{-1} \boldsymbol{e}_3 \times \frac{\partial \boldsymbol{x}_a}{\partial \xi_1}, \ \boldsymbol{a}^3 = \frac{\xi_3^2}{a} \boldsymbol{e}_3,$$

where,

$$J_a = \boldsymbol{e}_3 \cdot \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \times \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} = \left| \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \times \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} \right|,$$

is the surface measure associated with S_a. In this system of coordinates, the gradient operator becomes,

$$\begin{aligned} \boldsymbol{\nabla} p = (\boldsymbol{\nabla} p \cdot \boldsymbol{a}_i) \boldsymbol{a}^i &= \frac{\xi_3^2}{a} \frac{\partial p}{\partial \xi_3} \boldsymbol{e}_3 + \xi_3 J_a^{-1} \left(\frac{\partial p}{\partial \xi_1} \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} \times \boldsymbol{e}_3 + \frac{\partial p}{\partial \xi_2} \boldsymbol{e}_3 \times \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right) \\ &= \frac{\xi_3^2}{a} \frac{\partial p}{\partial \xi_3} \boldsymbol{e}_3 + \xi_3 \boldsymbol{\nabla}_{S_a} p, \end{aligned} \tag{2.14}$$

and (2.11) can be rewritten as,

$$\begin{aligned} \lim_{R \to \infty} \int_{S_a} \Bigg\{ \int_{a/R}^1 \left(\frac{1}{a} \frac{\partial p}{\partial \xi_3} \frac{\partial \overline{q}}{\partial \xi_3} + \frac{a}{\xi_3^2} \boldsymbol{\nabla}_{S_a} p \cdot \boldsymbol{\nabla}_{S_a} \overline{q} - \frac{k^2 a}{\xi_3^4} p \overline{q} \right) d\xi_3 \\ + \frac{ik}{\xi_3^2} p \overline{q}|_{\xi_3 = a/R} \Bigg\} dS_a. \end{aligned} \tag{2.15}$$

We now remove the known asymptotic form for the solution (written in terms of the exterior coordinates (2.12) and scaled so that p and P are identical on the truncating sphere $\xi_3 = 1$),

$$p = \xi_3 e^{-ika(\xi_3^{-1} - 1)} P,$$

and to obtain a Bubnov-Galerkin formulation we make the same substitution for the test function,

$$\overline{q} = \xi_3 e^{+ika(\xi_3^{-1} - 1)} \overline{Q}.$$

With these substitutions (2.15) becomes

$$\begin{aligned}\lim_{R\to\infty}\int_{S_a}\Bigg\{\int_{a/R}^{1}\Bigg(\frac{1}{a}\left[(1+ika\xi_3^{-1})P+\xi_3\frac{\partial P}{\partial \xi_3}\right]\left[(1-ika\xi_3^{-1})\overline{Q}+\xi_3\frac{\partial \overline{Q}}{\partial \xi_3}\right]\\ +a\boldsymbol{\nabla}_{S_a}P\cdot\boldsymbol{\nabla}_{S_a}\overline{Q}-\frac{k^2a}{\xi_3^2}P\overline{Q}\Bigg)d\xi_3+ikP\overline{Q}|_{\xi_3=a/R}\Bigg\}dS_a.\end{aligned} \tag{2.16}$$

By interpreting the limit in the Cauchy principal value sense, we allow the cancellation of the non-integrable terms $\pm k^2aP\overline{Q}/\xi_3^2$, and after some additional cosmetic simplifications (2.16) takes the final form,

$$\begin{aligned} b_\infty(p,q) &= \frac{1}{a}\int_{S_a}\left\{\int_0^1\frac{\partial P}{\partial \xi_3}\left(\xi_3^2\frac{\partial \overline{Q}}{\partial \xi_3}-i2ka\overline{Q}\right)d\xi_3+(1+ika)P\overline{Q}\Big|_{\xi_3=1}\right\}dS_a\\ &+ a\int_{S_a}\int_0^1\nabla_{S_a}P\cdot\nabla_{S_a}\overline{Q}d\xi_3dS_a\end{aligned}$$

2.2.2 Truncation by a perfectly matched layer

An alternative method for truncating exterior problems, known as perfectly matched layers (PML), was first introduced by Berenger [9, 10] in the context of finite difference time domain (FDTD) computations for Maxwell equations. The main idea is to surround the obstacle with a layer of absorbing material such that an outgoing wave enters the layer without reflection and decays exponentially. Here, we simply recall the derivation in spherical coordinates from [39], based on the notion of complex coordinate stretching due to Chew and Weedon [16]. For more on the analysis of PML approximations please see [13, 12] and the references therein.

The derivation is motivated by the one dimensional problem,

$$\begin{aligned} -\frac{d^2p}{dx^2}-k^2p &= 0 \text{ for } x>0\\ p(0) &= p_0\\ \frac{dp}{dx}+ikp &= o(1/x) \text{ as } x\to\infty\end{aligned}$$

where we seek the solution p in the interior domain $(0, a)$. Here, we can observe that the exact solution $p = p_0 e^{-ikx}$ is analytic, and therefore its unique analytic extension $p(z) = p_0 e^{-ikz}$ satisfies the same equation,

$$-\frac{d^2 p}{dz^2} - k^2 p = 0$$

with x traded for the complex variable z. Now, if we trace p along a path $z(x) = x - i\beta(x)$ with $\beta \to \infty$ as $x \to \infty$ we observe the exponential decay $p(z(x)) = p_0 e^{-ikx} e^{-k\beta(x)}$. For example, given $b > a$ we can take

$$z(x) = x - i\beta(x),\ \beta(x) = \begin{cases} 0, & 0 < x < a \\ L\left(\frac{x-a}{b-a}\right)^n, & x > a \end{cases} \tag{2.17}$$

with L chosen so that $p(z(b))$ is as small as we wish, and n chosen so that the path $z \in C^{n-1}$ is as smooth as we wish. Along this path p satisfies the modified equation

$$-\frac{d}{dx}\left(\frac{1}{z'}\frac{dp}{dx}\right) - k^2 z' p = 0.$$

Multiplying by the complex conjugate of a test function q and integrating by parts we arrive at the variational problem: find $p \in \widetilde{p_0} + V_0$ such that

$$\int_0^\infty \left\{ \frac{1}{z'}\frac{dp}{dx}\frac{d\overline{q}}{dx} - k^2 z' p \overline{q} \right\} \, dx = 0$$

for all $q \in V_0$. Here, $\widetilde{p_0}$ is any finite energy extension of the Dirichlet data p_0 into the energy space

$$V = \left\{ q \ : \ |z'|^{-1/2}\frac{dq}{dx}, |z'|^{1/2} q \in L^2(0, \infty) \right\}$$

In the 3D setting the derivation is conceptually the same, but more alge-

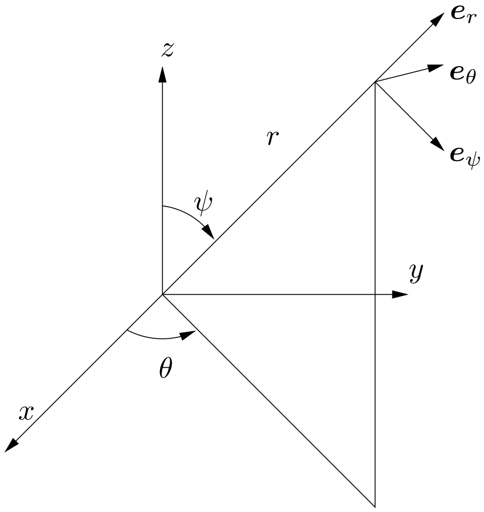


Figure 2.3: Spherical coordinates and unit basis vectors

braically complicated. We will work in spherical coordinates,

$$\boldsymbol{x} = \begin{pmatrix} r\sin(\psi)\cos(\theta) \\ r\sin(\psi)\sin(\theta) \\ r\cos(\psi) \end{pmatrix}.$$

(see Figure 2.3) Derivations in rectangular and cylindrical coordinates are similar. To facilitate the algebra we rewrite (2.5) as the first order system,

$$\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{u} + ikp &= 0, \\ \boldsymbol{\nabla}p + ik\boldsymbol{u} &= 0, \end{aligned} \tag{2.18}$$

or in spherical coordinates,

$$\begin{aligned} \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 u_r) + \frac{1}{r\sin\psi}\left(\frac{\partial}{\partial\psi}(\sin\psi u_\psi) + \frac{\partial u_\theta}{\partial\theta}\right) + ikp &= 0 \\ \frac{\partial p}{\partial r} + iku_r &= 0 \end{aligned}$$

$$
\begin{aligned}
\frac{1}{r}\frac{\partial p}{\partial \psi} + iku_\psi &= 0 \\
\frac{1}{r\sin\psi}\frac{\partial p}{\partial \theta} + iku_\theta &= 0
\end{aligned}
$$

Here, $\boldsymbol{u} = u_r\boldsymbol{e}_r + u_\psi\boldsymbol{e}_\psi + u_\theta\boldsymbol{e}_\theta$ is the velocity vector. We apply the complex coordinate stretching in the radial direction only ($r \leftarrow z = z(r)$) to obtain

$$
\begin{aligned}
\frac{1}{z'z^2}\frac{\partial}{\partial r}(z^2u_r) + \frac{1}{z\sin\psi}\left(\frac{\partial}{\partial\psi}(\sin\psi u_\psi) + \frac{\partial u_\theta}{\partial\theta}\right) + ikp &= 0 && (2.19)\\
\frac{1}{z'}\frac{\partial p}{\partial r} + iku_r &= 0 && (2.20)\\
\frac{1}{z}\frac{\partial p}{\partial\psi} + iku_\psi &= 0 && (2.21)\\
\frac{1}{z\sin\psi}\frac{\partial p}{\partial\theta} + iku_\theta &= 0 && (2.22)
\end{aligned}
$$

Then, multiplying (2.19) by $ikz'z^2/r^2$, and using (2.20)-(2.22) to eliminate the velocity vector, we obtain

$$
-\frac{1}{r^2}\frac{\partial}{\partial r}\left(\frac{z^2}{z'}\frac{\partial p}{\partial r}\right) - \frac{z'}{r^2\sin\psi}\frac{\partial}{\partial\psi}\left(\sin\psi\frac{\partial p}{\partial\psi}\right) - \frac{z'}{r^2\sin^2\psi}\frac{\partial^2 p}{\partial\theta^2} - k^2z'\frac{z^2}{r^2}p = 0 \quad (2.23)
$$

We observe that (2.23) reduces to (2.5) in the interior region where $z(r) = r$. To obtain the variational formulation, we multiply (2.23) by the complex conjugate of a test function q and integrate over the exterior domain $\Omega^a \cup \Omega_a^\infty$. For clarity, we separate the process into interior and exterior contributions. After integration by parts, the interior contribution is the expected,

$$
\int_{\Omega^a}\left\{\boldsymbol{\nabla}p\cdot\boldsymbol{\nabla}\overline{q} - k^2p\overline{q}\right\}dx - \int_{S_a}\frac{\partial u}{\partial r}\overline{q}\,dS_a = \int_\Gamma g\overline{q}\,d\Gamma,
$$

and under the assumption that p decays exponentially in r, the exterior contribution becomes,

$$\int_{\Omega_a^\infty} z' \left\{ \boldsymbol{\nabla} p \cdot \boldsymbol{\nabla} \overline{q} + \left(\left(\frac{z}{z' r} \right)^2 - 1 \right) \frac{\partial p}{\partial r} \frac{\partial \overline{q}}{\partial r} - k^2 \frac{z^2}{r^2} p \overline{q} \right\} dx + \frac{1}{z'(a^+)} \int_{S_a} \frac{\partial u}{\partial r} \overline{q} \; dS_a = 0$$

Here, $z'(a^+)$ refers to the limit of $z'(r)$ as $r \to a$ from above. For simplicity, we assume $z \in C^1$ so that $z'(a^+) = z'(a) = 1$ and the interface terms cancel out.

Then with

$$b_\infty(p, q) = \int_{\Omega_a^\infty} z' \left\{ \boldsymbol{\nabla} p \cdot \boldsymbol{\nabla} \overline{q} + \left(\left(\frac{z}{z' r} \right)^2 - 1 \right) \frac{\partial p}{\partial r} \frac{\partial \overline{q}}{\partial r} - k^2 \frac{z^2}{r^2} p \overline{q} \right\} dx \tag{2.24}$$

and b_a given once again by (2.10), the variational problem reads:

$$\left\{ \begin{array}{l} \text{Find } p \in V \text{ such that} \\ b(p, q) = b_a(p, q) + b_\infty(p, q) = l(q) \text{ for all } q \in V. \end{array} \right. \tag{2.25}$$

The energy space depends on the complex coordinate stretching z, as we must have,

$$V = \left\{ q : \; |z'|^{1/2} A(r) \boldsymbol{\nabla} q, \; |z'|^{1/2} \left| \frac{z}{r} \right| q \; \in \; L^2(\Omega) \right\}, \tag{2.26}$$

where

$$A(r)\boldsymbol{v} = \frac{z}{z' r} v_r \boldsymbol{e}_r + v_\psi \boldsymbol{e}_\psi + v_\theta \boldsymbol{e}_\theta .$$

The main advantage of the PML formulation is the simplicity of its discretization. We simply observe that the exponential decay of the solution in the exterior region Ω_a^∞ implies that after some sufficiently large radius b the remaining contribution to (2.24) is negligible. Thus our discrete trial (and test) space can be taken as a standard finite element space $V_0^{hp} \subset H^1(\Omega^b)$, where the subscript "0" indicates zero restriction to the outer sphere of radius b (implemented as a homogeneous Dirichlet boundary condition). Then clearly, $V_0^{hp} \subset V$, and our discrete problem reads:

$$\begin{cases} \text{Find } p^{hp} \in V_0^{hp} \text{ such that} \\ b(p^{hp}, q) = l(q) \text{ for all } q \in V_0^{hp}. \end{cases}$$

2.3 Electromagnetic scattering

For our final application we consider the time-harmonic Maxwell equations,

$$\begin{aligned} \boldsymbol{\nabla} \times \boldsymbol{E} &= -i\omega \boldsymbol{B} &(2.27)\\ \boldsymbol{\nabla} \times \boldsymbol{H} &= \boldsymbol{J}^{imp} + \sigma \boldsymbol{E} + i\omega \boldsymbol{D} &(2.28)\\ \boldsymbol{\nabla} \cdot \boldsymbol{D} &= \rho &(2.29)\\ \boldsymbol{\nabla} \cdot \boldsymbol{B} &= 0 &(2.30)\end{aligned}$$

Here, $\boldsymbol{E}$ and $\boldsymbol{H}$ are the electric and magnetic fields, $\boldsymbol{D}$ and $\boldsymbol{B}$ are the electric and magnetic flux, and $\boldsymbol{J}^{imp}$ and ρ are the impressed (electric) current and charge density, respectively, and ω is the angular frequency. In the present work, we will assume that the background medium is linear ($\boldsymbol{D} = \epsilon \boldsymbol{E}$ and $\boldsymbol{B} = \mu \boldsymbol{H}$, where ϵ, μ are the electric permittivity and magnetic permeability, respectively), isotropic ($\epsilon, \mu \in I\!R$) and non-conductive ($\sigma = 0$).

Assuming that no charge is being added to the system from an external source, we require that the existing charge is conserved so that

$$\boldsymbol{\nabla} \cdot \boldsymbol{J}^{imp} = -i\omega\rho. \tag{2.31}$$

Then the two Gauss laws (2.29) and (2.30) can be derived, respectively, by taking the divergence of Ampere's law (2.28) and Faraday's law (2.27), and the full Maxwell system (2.27)-(2.30) reduces to,

$$\boldsymbol{\nabla} \times \boldsymbol{E} = -i\omega\mu \boldsymbol{H} \tag{2.32}$$

$$\nabla \times \boldsymbol{H} \quad = \quad \boldsymbol{J}^{imp} + i\omega\epsilon\boldsymbol{E} \tag{2.33}$$

Finally, we can eliminate the magnetic field to obtain the reduced wave equation (for the electric field)

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \boldsymbol{E}\right) - \omega^2\epsilon\boldsymbol{E} = -i\omega\boldsymbol{J}^{imp} \tag{2.34}$$

For radiation problems, (2.34) is supplemented by appropriate boundary or radiation conditions and solved to determine the electric field $\boldsymbol{E}$ generated by a known current $\boldsymbol{J}^{imp}$. For scattering problems, we wish to determine the perturbation of a background electric field $\boldsymbol{E}^{inc}$ due to the presence of an obstacle.

2.3.1 Truncation by an impedance boundary condition

We consider the problem of scattering from a perfectly conducting obstacle placed inside an infinite rectangular waveguide. The motivation is that for a particular range of frequencies ω the waveguide geometry will only propagate a single pair (two pairs for a square waveguide) of transverse electric modes, moving in opposite directions. Any other mode introduced by a source within the waveguide decays exponentially and the infinite geometry can be truncated by a simple impedance boundary condition.

The geometry of the waveguide and a representative obstacle is shown in Figure 2.4. The interior of the waveguide is the unbounded domain $\Omega^w = (0, a) \times (0, b) \times (-\infty, \infty)$, and we will assume that $a \geq b$. We will denote the boundary of the waveguide by Γ^w. We will place a bounded, perfectly conducting obstacle Ω^{int} with Lipschitz boundary Γ, inside the waveguide, and denote by Ω the unbounded exterior domain $\Omega^w \setminus \overline{\Omega^{int}}$.

We first consider the waveguide without the obstacle. The walls of the waveguide Γ^w are assumed to be perfectly conducting so that the tangential component

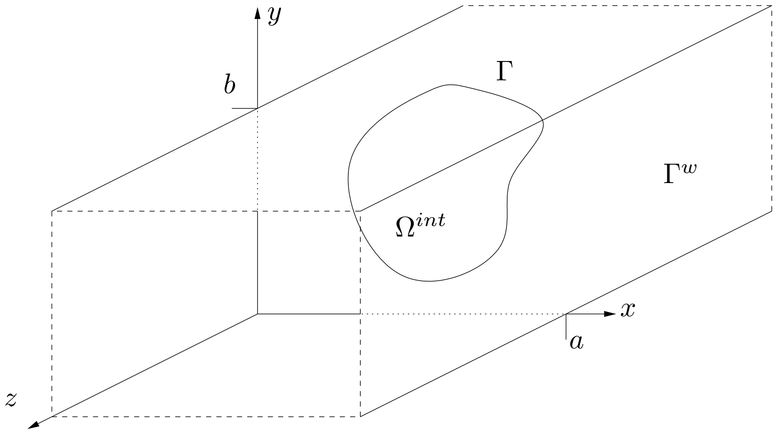


Figure 2.4: An obstacle Ω^{int} inside an infinite rectangular waveguide

of the electric field is zero, and we consider the eigenvalue problem,

$$\begin{aligned} \boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \boldsymbol{E} - \omega^2 \mu \epsilon \boldsymbol{E} &= 0 \text{ in } \Omega^w \qquad (2.35) \\ \widehat{\boldsymbol{n}} \times \boldsymbol{E} &= 0 \text{ on } \Gamma^w \qquad (2.36) \end{aligned}$$

For each pair of indices $m, n \geq 0$ (except $m = n = 0$) there exists a pair of transverse electric modes,

$$\boldsymbol{E}^{\pm}_{mn} = \begin{pmatrix} -\frac{n\pi}{b}\cos(m\pi x/a)\sin(n\pi y/b) \\ \frac{m\pi}{a}\sin(m\pi x/a)\cos(n\pi y/b) \\ 0 \end{pmatrix} e^{\mp i\beta_{mn} z}, \qquad (2.37)$$

where $\beta_{mn} = \sqrt{\omega^2\mu\epsilon - (m\pi/a)^2 - (n\pi/b)^2}$. When $\omega^2\mu\epsilon > (m\pi/a)^2 + (n\pi/b)^2$, β_{mn} is a positive real number, and $\boldsymbol{E}^{\pm}_{mn}$ is a pair of traveling waves with $\boldsymbol{E}^{+}_{mn}$ traveling in the $+z$ direction, and $\boldsymbol{E}^{-}_{mn}$ traveling in the $-z$ direction. When $\omega^2\mu\epsilon < (m\pi/a)^2 + (n\pi/b)^2$, the modes are evanescent. Hence we define the cutoff frequency,

$$\omega^c_{mn} = \frac{1}{\sqrt{\mu\epsilon}}\sqrt{(m\pi/a)^2 + (n\pi/b)^2}, \qquad (2.38)$$

below which a mode decays exponentially, and above which a mode travels without attenuation.

By observing that the cutoff frequency increases with both m and n we can select a range of operating frequencies ω such that the waveguide supports only the lowest order mode(s) and all higher order modes are attenuated. In case 1, $a > b$, we see that the smallest cutoff frequency is $\omega_{10}^c = \frac{\pi}{a\sqrt{\mu\epsilon}}$ and that the second smallest is $\omega_{01}^c = \frac{\pi}{b\sqrt{\mu\epsilon}}$. Hence, in the range $\omega_{10}^c < \omega < \omega_{01}^c$, the only traveling mode is

$$\boldsymbol{E}_{10}^{\pm} = \begin{pmatrix} 0 \\ \sin(\pi x/a) \\ 0 \end{pmatrix} e^{\mp i\beta_{10} z}, \tag{2.39}$$

(where we have dropped the constant π/a) and all others are attenuated. In case 2, $a = b$, the two lowest order modes, $1,0$ and $0,1$, have the same cutoff frequency, and the second smallest is $\omega_{11}^c = \frac{\pi}{a}\sqrt{\frac{2}{\mu\epsilon}}$. Hence, in the range $\omega_{10}^c = \omega_{01}^c < \omega < \omega_{11}^c$, there are only two pairs of traveling modes:

$$\boldsymbol{E}_{10}^{\pm} = \begin{pmatrix} 0 \\ \sin(\pi x/a) \\ 0 \end{pmatrix} e^{\mp i\beta_{10} z}, \quad \boldsymbol{E}_{01}^{\pm} = \begin{pmatrix} -\sin(\pi y/b) \\ 0 \\ 0 \end{pmatrix} e^{\mp i\beta_{01} z}. \tag{2.40}$$

The existence of such a range of operating frequencies allows us to derive a simple truncation of the waveguide for scattering computations. Consider case 1 and take the incident field $\boldsymbol{E}^{inc} = \boldsymbol{E}_{10}^{+}$, produced by a source far away in the negative z-direction. This field satisfies the reduced wave equation (2.35) in the entire waveguide Ω^w, and the PEC boundary condition (2.36) on the waveguide boundary Γ^w. In the presence of the obstacle Ω^{int}, the total electric field $\boldsymbol{E}^{tot}$ satisfies the same equation (2.35) but only in the exterior domain Ω, while inside the obstacle, $\boldsymbol{E}^{tot} = 0$. Moreover, $\boldsymbol{E}^{tot}$ satisfies the same PEC boundary condition

(2.36) but on both the waveguide boundary Γ^w and the boundary of the obstacle Γ.

Now, because the waveguide supports only modes $\boldsymbol{E}_{10}^{\pm}$, the total field has the following asymptotic form,

$$\boldsymbol{E}^{tot} = \boldsymbol{E}_{10}^{+} + R\boldsymbol{E}_{10}^{-} \text{ as } z \to -\infty, \quad \boldsymbol{E}^{tot} = T\boldsymbol{E}_{10}^{+} \text{ as } z \to \infty, \tag{2.41}$$

where the unknown constants R and T are reflection and transmission coefficients, respectively, for the scatterer. If we define the scattered field as $\boldsymbol{E} = \boldsymbol{E}^{tot} - \boldsymbol{E}^{inc}$, then the resulting asymptotic form for $\boldsymbol{E}$ is given by,

$$\boldsymbol{E} = R\boldsymbol{E}_{10}^{-} \text{ as } z \to -\infty, \quad \boldsymbol{E} = (T-1)\boldsymbol{E}_{10}^{+} \text{ as } z \to \infty. \tag{2.42}$$

Since the constants R and T are not known a-priori, these relations cannot be used directly to truncate the computational domain. However, we can eliminate R and T by observing that, as $z \to -\infty$,

$$\hat{\boldsymbol{n}} \times \boldsymbol{\nabla} \times \boldsymbol{E} = \hat{\boldsymbol{n}} \times \boldsymbol{\nabla} \times R\boldsymbol{E}_{10}^{-} = i\beta_{10} R\boldsymbol{E}_{10}^{-} = i\beta_{10}\boldsymbol{E}, \tag{2.43}$$

where $\hat{\boldsymbol{n}} = -\hat{\boldsymbol{e}}_z$, and likewise as $z \to \infty$,

$$\hat{\boldsymbol{n}} \times \boldsymbol{\nabla} \times \boldsymbol{E} = \hat{\boldsymbol{n}} \times \boldsymbol{\nabla} \times (T-1)\boldsymbol{E}_{10}^{+} = i\beta_{10}(T-1)\boldsymbol{E}_{10}^{+} = i\beta_{10}\boldsymbol{E}, \tag{2.44}$$

where $\hat{\boldsymbol{n}} = \hat{\boldsymbol{e}}_z$. For finite z relations (2.43) and (2.44) are off by a term that decays exponentially in z. Hence, we introduce a sufficiently large truncating distance l and impose (2.43) and (2.44) as an impedance boundary condition on the finite boundary $z = -l$ and $z = l$ (denoted by Γ^l). In case 2, the general incident field is a linear combination of the modes $\boldsymbol{E}_{10}^{+}$ and $\boldsymbol{E}_{01}^{+}$, and similar considerations yield the same impedance boundary condition.

In summary, after truncation the scattered field $\boldsymbol{E}$ satisfies the problem,

$$\begin{aligned}
\boldsymbol{\nabla} \times \left(\frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E} \right) - \omega^2 \epsilon \boldsymbol{E} &= 0 \text{ in } \Omega^l \\
\hat{\boldsymbol{n}} \times \boldsymbol{E} &= -\hat{\boldsymbol{n}} \times \boldsymbol{E}^{inc} \text{ on } \Gamma \\
\hat{\boldsymbol{n}} \times \boldsymbol{E} &= 0 \text{ on } \Gamma^w \\
\hat{\boldsymbol{n}} \times \boldsymbol{\nabla} \times \boldsymbol{E} &= i\beta_{10} \boldsymbol{E} \text{ on } \Gamma^l
\end{aligned}$$

So, let $\boldsymbol{E}_D \in \boldsymbol{H}(curl, \Omega^l)$ be such that $\hat{\boldsymbol{n}} \times \boldsymbol{E}_D = -\hat{\boldsymbol{n}} \times \boldsymbol{E}^{inc}$ on Γ and $\hat{\boldsymbol{n}} \times \boldsymbol{E}_D = 0$ on Γ^w and let $\boldsymbol{V} = \{\boldsymbol{F} \in \boldsymbol{H}(curl, \Omega^l) \ : \ \hat{\boldsymbol{n}} \times \boldsymbol{F} = 0 \text{ on } \Gamma \cup \Gamma^w\}$. Then the standard variational formulation reads:

$$\left\{ \begin{array}{l} \text{Find } \boldsymbol{E} \in \boldsymbol{E}_D + \boldsymbol{V} \text{ such that} \\ \displaystyle\int_\Omega \left\{ \frac{1}{\mu} \boldsymbol{\nabla} \times \boldsymbol{E} \cdot \boldsymbol{\nabla} \times \boldsymbol{F} - \omega^2 \epsilon \boldsymbol{E} \cdot \boldsymbol{F} \right\} \, dx + \frac{i\beta_0}{\mu} \int_{\Gamma^l} \boldsymbol{E} \cdot \boldsymbol{F} \; d\Gamma = 0 \text{ for all } \boldsymbol{F} \in \boldsymbol{V} \end{array} \right. \tag{2.45}$$

2.3.2 Truncation by infinite elements

In the previous section we considered the scattering from an obstacle in a waveguide because of the simple geometry and method of truncation. Presently we turn to the problem of scattering from a bounded obstacle into the entire exterior domain. In this case, the excitation comes from an incident electromagnetic wave generated by the transmitting antenna in a radar system. The transmitting antenna is assumed to be far away from the obstacle so that the asymptotically spherical wave produced can be approximated by an incident, plane, linearly polarized TEM wave. Thus, the incident wave has the form,

$$\boldsymbol{E}^{inc} = \boldsymbol{E}_0 e^{ik\hat{\boldsymbol{e}} \cdot \boldsymbol{r}}. \tag{2.46}$$

As before, the obstacle occupies a bounded domain Ω^{int} with Lipschitz boundary Γ. The scattered electric field $\boldsymbol{E}$ satisfies the reduced wave equation in the entire exterior domain,

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \boldsymbol{E} - k^2 \boldsymbol{E} = 0 \text{ in } \Omega = I\!R^3 \setminus \Omega^{int},$$

along with the Dirichlet boundary condition (for a perfect electric conductor),

$$\hat{\boldsymbol{n}} \times \boldsymbol{E} = -\hat{\boldsymbol{n}} \times \boldsymbol{E}^{inc} \text{ on } \Gamma,$$

and the Silver-Müller radiation condition,

$$\boldsymbol{e}_r \times \boldsymbol{\nabla} \times \boldsymbol{E} - ik\boldsymbol{E}_t = \boldsymbol{W} = o(1/r) \text{ as } r \to \infty,$$

which expresses the fact that the scattered electric field should move outward. As for the acoustic case we will again consider truncation by infinite elements and by the introduction of a PML.

We again split the exterior domain into a bounded part Ω^a between the scatterer and the sphere S_a of radius a, and the unbounded part Ω_a^∞. Multiplying by a test function $\boldsymbol{F}$ such that $\boldsymbol{n} \times \boldsymbol{F} = 0$ on Γ and integrating first over Ω^a we obtain, after integration by parts,

$$\begin{aligned} 0 &= \int_{\Omega^a} \{\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \boldsymbol{E} - k^2 \boldsymbol{E}\} \cdot \overline{\boldsymbol{F}}\, dx \\ &= \int_{\Omega^a} \{\boldsymbol{\nabla} \times \boldsymbol{E} \cdot \boldsymbol{\nabla} \times \overline{\boldsymbol{F}} - k^2 \boldsymbol{E} \cdot \overline{\boldsymbol{F}}\}\, dx + \int_{S_a} \boldsymbol{e}_r \times \boldsymbol{\nabla} \times \boldsymbol{E} \cdot \overline{\boldsymbol{F}}\, dS_a \end{aligned}$$

The integration over Ω_a^∞ will again be interpreted in the Cauchy principal value sense.

$$0 = \int_{\Omega_a^R} \{\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \boldsymbol{E} - k^2 \boldsymbol{E}\} \cdot \overline{\boldsymbol{F}}\, dx$$

$$
\begin{aligned}
= & \int_{\Omega_a^R} \{\boldsymbol{\nabla}\times\boldsymbol{E}\cdot\boldsymbol{\nabla}\times\overline{\boldsymbol{F}} - k^2\boldsymbol{E}\cdot\overline{\boldsymbol{F}}\}\,dx - \int_{S_a} \boldsymbol{e}_r\times\boldsymbol{\nabla}\times\boldsymbol{E}\cdot\overline{\boldsymbol{F}}\,dS_a \\
+ & \ ik\int_{S_R}\boldsymbol{E}_t\cdot\overline{\boldsymbol{F}}_t\,dS_R + \int_{S_R}\boldsymbol{W}\cdot\overline{\boldsymbol{F}}\,dS_R
\end{aligned}
$$

As before, the term involving the unknown field $\boldsymbol{W}$ vanishes in the limit, and we employ the interface condition

$$
\boldsymbol{e}_r\times[\boldsymbol{\nabla}\times\boldsymbol{E}] = i\omega[\mu\boldsymbol{H}]\times\boldsymbol{e}_r = 0,
$$

to cancel the integrals over S_a. Adding these contributions we obtain the standard variational formulation

$$
\left\{
\begin{array}{l}
\text{Find } \boldsymbol{E} \text{ such that } \boldsymbol{n}\times\boldsymbol{E} = -\boldsymbol{n}\times\boldsymbol{E}^{inc} \text{ on } \Gamma, \text{ and} \\
\displaystyle\int_{\Omega^a}\{\boldsymbol{\nabla}\times\boldsymbol{E}\cdot\boldsymbol{\nabla}\times\overline{\boldsymbol{F}} - k^2\boldsymbol{E}\cdot\overline{\boldsymbol{F}}\}\,dx \\
\displaystyle + \lim_{R\to\infty}\left(\int_{\Omega_a^R}\{\boldsymbol{\nabla}\times\boldsymbol{E}\cdot\boldsymbol{\nabla}\times\overline{\boldsymbol{F}} - k^2\boldsymbol{E}\cdot\overline{\boldsymbol{F}}\}\,dx + ik\int_{S_R}\boldsymbol{E}_t\cdot\overline{\boldsymbol{F}}_t\,dS_R\right) = 0 \\
\text{for all } \boldsymbol{F} \text{ such that } \boldsymbol{n}\times\boldsymbol{F} = 0 \text{ on } \Gamma
\end{array}
\right.
\tag{2.47}
$$

We have omitted the details about the energy space, apart from the obvious requirement that $\boldsymbol{E},\boldsymbol{F}\in\boldsymbol{H}_{loc}(curl,\Omega)$. The additional assumptions on $\boldsymbol{E},\boldsymbol{F}$ to follow, are required to guarantee the existence of the above limit.

We will work with the same system of coordinates as for the acoustic case, but the far-field ansatz is slightly different. Here, we will make the substitution ("overloading" the symbols $\boldsymbol{E}$ and $\boldsymbol{F}$),

$$
\boldsymbol{E} \leftarrow e^{-ika(\xi_3^{-1}-1)}\boldsymbol{E},\ \overline{\boldsymbol{F}} \leftarrow e^{+ika(\xi_3^{-1}-1)}\overline{\boldsymbol{F}}
$$

Then, using the elementary formula $\boldsymbol{\nabla}\times(\psi\boldsymbol{E}) = \boldsymbol{\nabla}\psi\times\boldsymbol{E} + \psi\boldsymbol{\nabla}\times\boldsymbol{E}$, and recalling the formula for the gradient of a scalar function (2.14), we get

$$\begin{aligned}\boldsymbol{\nabla} \times \boldsymbol{E} &\leftarrow e^{-ika(\xi_3^{-1}-1)}[ik\boldsymbol{e}_3 \times \boldsymbol{E} + \boldsymbol{\nabla} \times \boldsymbol{E}],\\ \boldsymbol{\nabla} \times \overline{\boldsymbol{F}} &\leftarrow e^{+ika(\xi_3^{-1}-1)}[-ik\boldsymbol{e}_3 \times \overline{\boldsymbol{F}} + \boldsymbol{\nabla} \times \overline{\boldsymbol{F}}]\end{aligned}$$

Making these substitutions, and utilizing the identity,

$$(\boldsymbol{e}_3 \times \boldsymbol{E}) \cdot (\boldsymbol{e}_3 \times \overline{\boldsymbol{F}}) = \boldsymbol{E} \cdot \overline{\boldsymbol{F}} - (\boldsymbol{e}_3 \cdot \boldsymbol{E})(\boldsymbol{e}_3 \cdot \overline{\boldsymbol{F}}),$$

the limit in (2.47) becomes,

$$\begin{aligned}\lim_{R\to\infty} \int_{S_a} \Bigg(\int_{a/R}^{1} \; & \Big\{ (\boldsymbol{\nabla} \times \boldsymbol{E}) \cdot (\boldsymbol{\nabla} \times \overline{\boldsymbol{F}}) \\ & -ik[(\boldsymbol{\nabla} \times \boldsymbol{E}) \cdot (\boldsymbol{e}_3 \times \overline{\boldsymbol{F}}) - (\boldsymbol{e}_3 \times \boldsymbol{E}) \cdot (\boldsymbol{\nabla} \times \overline{\boldsymbol{F}})] \\ & -k^2 (\boldsymbol{e}_3 \cdot \boldsymbol{E})(\boldsymbol{e}_3 \cdot \overline{\boldsymbol{F}}) \Big\} \frac{a}{\xi_3^4} d\xi_3 + ik \frac{R^2}{a^2} \boldsymbol{E}_t \cdot \boldsymbol{F}_t|_{\xi_3 = \frac{a}{R}} \Bigg) dS_a\end{aligned} \tag{2.48}$$

Finally, we assume that $\boldsymbol{E}$ (and $\boldsymbol{F}$) is obtained by a transformation consistent with the standard exact sequence property, namely that $\boldsymbol{E}$ transforms like the gradient of a scalar function (2.14),

$$\boldsymbol{E} = \xi_3 J_a^{-1} \left(E_1 \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} \times \boldsymbol{e}_3 + E_2 \boldsymbol{e}_3 \times \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right) + \frac{\xi_3^2}{a} E_3 \boldsymbol{e}_3,$$

and therefore the curl transforms in the standard way

$$\begin{aligned}\boldsymbol{\nabla} \times \boldsymbol{E} &= \left| \frac{d\boldsymbol{x}}{d\boldsymbol{\xi}} \right|^{-1} \frac{\partial \boldsymbol{x}}{\partial \xi_i} (\hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{E}})_i \\ &= \frac{\xi_3^3}{a J_a} \left\{ (\hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{E}})_1 \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} + (\hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{E}})_2 \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} + \frac{a}{\xi_3} (\hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{E}})_3 \boldsymbol{e}_3 \right\}\end{aligned}$$

$$
\begin{aligned}
= \quad \frac{\xi_3^3}{aJ_a} & \left\{ \left(\frac{\partial E_3}{\partial \xi_2} - \frac{\partial E_2}{\partial \xi_3} \right) \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right. \\
& + \left(\frac{\partial E_1}{\partial \xi_3} - \frac{\partial E_3}{\partial \xi_1} \right) \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} \\
& \left. + \left(\frac{\partial E_2}{\partial \xi_1} - \frac{\partial E_1}{\partial \xi_2} \right) \frac{a}{\xi_3} \boldsymbol{e}_3 \right\}
\end{aligned}
$$

The remaining terms in (2.48) take the following forms (paying particular attention to the dependence on ξ_3)

$$
\begin{aligned}
\boldsymbol{e}_3 \times \boldsymbol{E} &= \xi_3 J_a^{-1} \left(E_1 \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} - E_2 \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right) \\
\boldsymbol{e}_3 \cdot \boldsymbol{E} &= \frac{\xi_3^2}{a} E_3 \\
\boldsymbol{E}_t &= \xi_3 J_a^{-1} \left(E_1 \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} \times \boldsymbol{e}_3 + E_2 \boldsymbol{e}_3 \times \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right)
\end{aligned}
$$

We see that the limit converges as $R \to \infty$ provided that $\hat{\boldsymbol{E}}, \hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{E}}, \hat{\boldsymbol{F}}, \hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{F}} = O(1)$ as $\xi_3 \to 0$. The final form of (2.48) becomes

$$
\begin{aligned}
\int_{S_a} \Bigg(\int_0^1 \; & \Big\{ (\boldsymbol{\nabla} \times \boldsymbol{E}) \cdot (\boldsymbol{\nabla} \times \overline{\boldsymbol{F}}) \\
& -ik[(\boldsymbol{\nabla} \times \boldsymbol{E}) \cdot (\boldsymbol{e}_3 \times \overline{\boldsymbol{F}}) - (\boldsymbol{e}_3 \times \boldsymbol{E}) \cdot (\boldsymbol{\nabla} \times \overline{\boldsymbol{F}})] - \frac{k^2}{a^2} \xi_3^4 E_3 \overline{F}_3 \Big\} \frac{a}{\xi_3^4} d\xi_3 \\
& + \frac{ik}{J_a^2} \left(E_1 \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} - E_2 \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right) \cdot \left(F_1 \frac{\partial \boldsymbol{x}_a}{\partial \xi_2} - F_2 \frac{\partial \boldsymbol{x}_a}{\partial \xi_1} \right) \Bigg|_{\xi_3 = 0} \Bigg) dS_a
\end{aligned}
\tag{2.49}
$$

2.3.3 Truncation by a perfectly matched layer

Once again we subdivide the exterior domain $\Omega = I\!R^3 \setminus \overline{\Omega^{\text{int}}}$ into an interior region $\Omega^a = \{|x| < a\} \setminus \overline{\Omega^{\text{int}}}$, and an exterior region $\Omega_a^R = \{a < |x| < R\}$, with the intent to allow $R \to \infty$. For the interior region Ω^a we proceed as usual to multiply Ampere's law (2.33) by a test function F, integrate over Ω^a and integrate by parts to obtain,

$$
0 = \int_{\Omega^a} (-i\omega \boldsymbol{\nabla} \times \boldsymbol{H} - \omega^2 \epsilon \boldsymbol{E}) \cdot \boldsymbol{F} dx
$$

$$
\begin{aligned}
&= \int_{\Omega^a} (-i\omega \boldsymbol{H} \cdot \boldsymbol{\nabla} \times \boldsymbol{F} - \omega^2 \epsilon \boldsymbol{E} \cdot \boldsymbol{F}) dx - i\omega \int_{S_a} (\boldsymbol{e}_r \times \boldsymbol{H}) \cdot \boldsymbol{F} dS_a \\
&= \int_{\Omega^a} \left(\mu^{-1} \boldsymbol{\nabla} \times \boldsymbol{E} \cdot \boldsymbol{\nabla} \times \boldsymbol{F} - \omega^2 \epsilon \boldsymbol{E} \cdot \boldsymbol{F} \right) dx + \int_{S_a} \boldsymbol{e}_r \times (\mu^{-1} \boldsymbol{\nabla} \times \boldsymbol{E}) \cdot \boldsymbol{F} dS_a
\end{aligned}
$$

Here, we have assumed that the scatterer is a perfect conductor ($\hat{\boldsymbol{n}} \times \boldsymbol{E}$ is prescribed so we choose a test function $\boldsymbol{F}$ such that $\hat{\boldsymbol{n}} \times \boldsymbol{F} = 0$ on Γ), and used Faraday's law (2.32) to eliminate the magnetic field.

We follow the same steps to introduce a spherical PML in Ω_a^R. First we write Ampere's law (2.33) in spherical coordinates,

$$
\begin{aligned}
\frac{1}{r \sin\psi} \left[\frac{\partial}{\partial \psi} (\sin\psi H_\theta) - \frac{\partial H_\psi}{\partial \theta} \right] &= i\omega\epsilon E_r \\
\frac{1}{r} \left[\frac{1}{\sin\psi} \frac{\partial H_r}{\partial \theta} - \frac{\partial}{\partial r}(r H_\theta) \right] &= i\omega\epsilon E_\psi \\
\frac{1}{r} \left[\frac{\partial}{\partial r}(r H_\psi) - \frac{\partial H_r}{\partial \psi} \right] &= i\omega\epsilon E_\theta
\end{aligned}
$$

and then apply the complex coordinate stretching $r \leftarrow z = z(r)$ to obtain,

$$
\begin{aligned}
\frac{1}{z \sin\psi} \left[\frac{\partial}{\partial \psi} (\sin\psi H_\theta) - \frac{\partial H_\psi}{\partial \theta} \right] &= i\omega\epsilon E_r && (2.50) \\
\frac{1}{z} \left[\frac{1}{\sin\psi} \frac{\partial H_r}{\partial \theta} - \frac{1}{z'} \frac{\partial}{\partial r}(z H_\theta) \right] &= i\omega\epsilon E_\psi && (2.51) \\
\frac{1}{z} \left[\frac{1}{z'} \frac{\partial}{\partial r}(z H_\psi) - \frac{\partial H_r}{\partial \psi} \right] &= i\omega\epsilon E_\theta && (2.52)
\end{aligned}
$$

Here, we have used the physical components,

$$
\boldsymbol{E} = E_r \boldsymbol{e}_r + E_\psi \boldsymbol{e}_\psi + E_\theta \boldsymbol{e}_\theta, \ \boldsymbol{H} = H_r \boldsymbol{e}_r + H_\psi \boldsymbol{e}_\psi + H_\theta \boldsymbol{e}_\theta.
$$

Multiplying (2.50)-(2.52) by $i\omega z' \frac{z^2}{r^2} \boldsymbol{F}$, integrating over Ω_a^R and integrating by parts we obtain,

$$
-\omega^2 \int_{\Omega_a^R} \epsilon z' \frac{z^2}{r^2} \boldsymbol{E} \cdot \boldsymbol{F} dx \quad = \quad i\omega \int_{\Omega_a^R} \frac{z}{r^2} \left[\frac{\partial}{\partial r}(z H_\psi) F_\theta - \frac{\partial}{\partial r}(z H_\theta) F_\psi \right] dx
$$

$$\begin{aligned}
&+ \quad i\omega \int_{\Omega_a^R} \frac{zz'}{r^2 \sin\psi} \left[\frac{\partial}{\partial\psi}(\sin\psi H_\theta) F_r - \frac{\partial H_r}{\partial\psi} \sin\psi F_\theta \right] dx \\
&+ \quad i\omega \int_{\Omega_a^R} \frac{zz'}{r^2 \sin\psi} \left[\frac{\partial H_r}{\partial\theta} F_\psi - \frac{\partial H_\psi}{\partial\theta} F_r \right] dx \\
&= \quad i\omega \int_{\Omega_a^R} \frac{z'z}{r^2 \sin\psi} H_r \left[\frac{\partial}{\partial\psi}(\sin\psi F_\theta) - \frac{\partial F_\psi}{\partial\theta} \right] dx \\
&+ \quad i\omega \int_{\Omega_a^R} \frac{z'z}{r^2} H_\psi \left[\frac{\partial}{\partial\psi}(\sin\psi H_\theta) F_r - \frac{\partial H_r}{\partial\psi} \sin\psi F_\theta \right] dx \\
&+ \quad i\omega \int_{\Omega_a^R} \frac{z'z}{r^2} H_\theta \left[\frac{1}{z'} \frac{\partial}{\partial r}(zF_\psi) - \frac{\partial F_r}{\partial\psi} \right] dx \\
&+ \quad i\omega \int_{S_a} (H_\theta F_\psi - H_\psi F_\theta) dS_a
\end{aligned}$$

Assuming the complex coordinate stretching does indeed cause an exponential decay in $\boldsymbol{H}$ we have omitted the contribution from the outer boundary S_R. We will subsequently omit the contribution from S_a by observing that it cancels with the contribution from the interior domain. Finally using Faraday's law (modified with the same complex coordinate stretching),

$$\begin{aligned}
i\omega z H_r &= -\frac{1}{\mu \sin\psi} \left[\frac{\partial}{\partial\psi}(\sin\psi E_\theta) - \frac{\partial E_\psi}{\partial\theta} \right] \\
i\omega z H_\psi &= -\frac{1}{\mu} \left[\frac{1}{\sin\psi} \frac{\partial E_r}{\partial\theta} - \frac{1}{z'} \frac{\partial}{\partial r}(zE_\theta) \right] \\
i\omega z H_\theta &= -\frac{1}{\mu} \left[\frac{1}{z'} \frac{\partial}{\partial r}(zE_\psi) - \frac{\partial E_r}{\partial\psi} \right]
\end{aligned}$$

to eliminate the magnetic field we obtain,

$$\begin{aligned}
\int_{\Omega_a^R} \Bigg\{ \frac{z'}{\mu} \Bigg[&\frac{1}{r^2 \sin^2\psi} \left(\frac{\partial}{\partial\psi}(\sin\psi E_\theta) - \frac{\partial E_\psi}{\partial\theta} \right) \left(\frac{\partial}{\partial\psi}(\sin\psi F_\theta) - \frac{\partial F_\psi}{\partial\theta} \right) \\
&+ \frac{1}{r^2} \left(\frac{1}{\sin\psi} \frac{\partial E_r}{\partial\theta} - \frac{1}{z'} \frac{\partial}{\partial r}(zE_\theta) \right) \left(\frac{1}{\sin\psi} \frac{\partial F_r}{\partial\theta} - \frac{1}{z'} \frac{\partial}{\partial r}(zF_\theta) \right) \\
&+ \frac{1}{r^2} \left(\frac{1}{z'} \frac{\partial}{\partial r}(zE_\psi) - \frac{\partial E_r}{\partial\psi} \right) \left(\frac{1}{z'} \frac{\partial}{\partial r}(zF_\psi) - \frac{\partial F_r}{\partial\psi} \right) \Bigg] \\
&- \omega^2 \epsilon z' \frac{z^2}{r^2} \boldsymbol{E} \cdot \boldsymbol{F} \Bigg\} dx = 0
\end{aligned} \tag{2.53}$$

We see that the first three terms form a modified ("stretched") version of the usual $\boldsymbol{\nabla} \times \boldsymbol{E} \cdot \boldsymbol{\nabla} \times \boldsymbol{F}$ term. To recover a more familiar form we make the substitution

$$E_r = \frac{1}{z'}\tilde{E}_r,\ E_\psi = \frac{r}{z}\tilde{E}_\psi,\ E_\theta = \frac{r}{z}\tilde{E}_\theta. \tag{2.54}$$

With this ansatz (2.53) becomes

$$\begin{aligned}\int_\Omega \Big\{ &\frac{1}{z'\mu}\Big[\boldsymbol{\nabla} \times \tilde{\boldsymbol{E}} \cdot \boldsymbol{\nabla} \times \tilde{\boldsymbol{F}} + ((rz'/z)^2 - 1)(e_r \cdot \boldsymbol{\nabla} \times \tilde{\boldsymbol{E}})(e_r \cdot \boldsymbol{\nabla} \times \tilde{\boldsymbol{F}})\Big] \\ &-\omega^2 \epsilon z' \Big[\tilde{\boldsymbol{E}} \cdot \tilde{\boldsymbol{F}} + ((rz'/z)^{-2} - 1)(e_r \cdot \tilde{\boldsymbol{E}})(e_r \cdot \tilde{\boldsymbol{F}})\Big]\Big\}\, dx\end{aligned} \tag{2.55}$$

Chapter 3

Compatible *hp*-FEM

The preceding chapter contains a variety of variational problems set in the Sobolev spaces H^1 and $\boldsymbol{H}(\mathrm{curl})$, possibly modified by certain weights for problems posed in exterior domains. We presently take up a discussion of the construction of conforming, finite dimensional subspaces suitable for the discretization of these problems by *hp*-FEM.

3.1 The exact sequence property and de Rham diagram

It is well known that on a simply connected, Lipschitz domain Ω the following function spaces and differential operators form an exact sequence,

$$I\!R \xrightarrow{id} H^1(\Omega) \xrightarrow{\boldsymbol{\nabla}} \boldsymbol{H}(\mathrm{curl},\Omega) \xrightarrow{\boldsymbol{\nabla}\times} \boldsymbol{H}(\mathrm{div},\Omega) \xrightarrow{\boldsymbol{\nabla}\cdot} L^2(\Omega) \longrightarrow \{0\}, \tag{3.1}$$

that is, the range of each operator corresponds to the null space of the operator on the right. Moreover, it is possible to reproduce this sequence on a discrete level, that is by defining finite dimensional subspaces $W_{hp}(\Omega) \subset H^1(\Omega)$, $\boldsymbol{Q}_{hp}(\Omega) \subset \boldsymbol{H}(\mathrm{curl},\Omega)$,

$\boldsymbol{V}_{hp}(\Omega) \subset \boldsymbol{H}(\mathrm{div}, \Omega)$ and $Y_{hp}(\Omega) \subset L^2(\Omega)$ such that the sequence,

$$\mathbb{R} \xrightarrow{id} W_{hp}(\Omega) \xrightarrow{\boldsymbol{\nabla}} \boldsymbol{Q}_{hp}(\Omega) \xrightarrow{\boldsymbol{\nabla}\times} \boldsymbol{V}_{hp}(\Omega) \xrightarrow{\boldsymbol{\nabla}\cdot} Y_{hp}(\Omega) \longrightarrow \{0\}, \tag{3.2}$$

is also exact. By assuming some additional regularity, it is possible to relate these two sequences by defining projection-based interpolation operators Π^{grad}, Π^{curl}, Π^{div} and P such that the resulting de Rham diagram,

$$\begin{tikzcd}
H^r(\Omega) \arrow[r, "\boldsymbol{\nabla}"] \arrow[d, "\Pi^{grad}"] & \boldsymbol{H}^{r-1}(\mathrm{curl}, \Omega) \arrow[r, "\boldsymbol{\nabla}\times"] \arrow[d, "\Pi^{curl}"] & \boldsymbol{H}^{r-1}(\mathrm{div}, \Omega) \arrow[r, "\boldsymbol{\nabla}\cdot"] \arrow[d, "\Pi^{div}"] & H^{r-1}(\Omega) \arrow[d, "P"] \\
W_{hp}(\Omega) \arrow[r, "\boldsymbol{\nabla}"] & \boldsymbol{Q}_{hp}(\Omega) \arrow[r, "\boldsymbol{\nabla}\times"] & \boldsymbol{V}_{hp}(\Omega) \arrow[r, "\boldsymbol{\nabla}\cdot"] & Y_{hp}(\Omega)
\end{tikzcd} \tag{3.3}$$

(with $r > 3/2$) commutes. That is,

$$\begin{aligned}
\boldsymbol{\nabla}(\Pi^{grad}u) &= \Pi^{curl}(\boldsymbol{\nabla}u) && \forall u \in H^r(\Omega), \\
\boldsymbol{\nabla} \times (\Pi^{curl}\boldsymbol{E}) &= \Pi^{div}(\boldsymbol{\nabla} \times \boldsymbol{E}) && \forall \boldsymbol{E} \in \boldsymbol{H}^{r-1}(\mathrm{curl}, \Omega), \\
\boldsymbol{\nabla} \cdot (\Pi^{div}\boldsymbol{v}) &= P(\boldsymbol{\nabla} \cdot \boldsymbol{v}) && \forall \boldsymbol{v} \in \boldsymbol{H}^{r-1}(\mathrm{div}, \Omega).
\end{aligned}$$

This commutativity is critical for establishing a discrete compactness result which in turn is used to prove convergence results for the Maxwell eigenvalue problem and the mixed formulation for Maxwell equations. Here we wish only to motivate the construction of the spaces $W_{hp}(\Omega)$ and $\boldsymbol{Q}_{hp}(\Omega)$ that form the basis of our present work.

3.2 H^1-conforming *hp* elements

Let $\hat{K} = (0,1)^3$ be the master hexahedron. The master element of uniform (but possibly anisotropic) order $\boldsymbol{p} = (p,q,r)$ is defined as the tensor product space,

$$\mathcal{Q}^{(p,q,r)}(\hat{K}) = \mathcal{P}^p \otimes \mathcal{P}^q \otimes \mathcal{P}^r, \tag{3.4}$$

where $\mathcal{P}^p$ denotes 1D polynomials of order p. The master element of variable order is a subspace of $\mathcal{Q}^{(p,q,r)}(\hat{K})$ where restrictions to faces and edges are of possibly lower order. We will use the (admittedly incomplete) notation,

$$W_p(\hat{K}) = \mathcal{Q}^{(p,q,r)}_{(p_f,q_f),(p_f,r_f),(q_f,r_f),p_e,q_e,r_e}(\hat{K}). \tag{3.5}$$

Here, the letters p, q and r denote orders of approximation corresponding to the master element coordinate directions $1, 2$ and 3, respectively. Then the first subscript (p_f, q_f) indicates the possibility of a lower order restriction to either of the two faces orthogonal to the third axis. These orders may be different for the two such faces, but they have been combined into a single symbol to arrive at a more convenient notation. Likewise the fourth subscript p_e indicates the possibility of a lower order restriction to any of the four edges parallel to the first axis.

The physical element K is then defined in terms of a smooth invertible parametrization $\boldsymbol{x}_K : \hat{K} \to K$, with the corresponding space (of possibly non-polynomials),

$$W_p(K) = \{v = \hat{v} \circ \boldsymbol{x}_K^{-1} : \ \hat{v} \in W_p(\hat{K})\}. \tag{3.6}$$

The physical domain Ω is then subdivided into a regular mesh $\mathcal{M}$ consisting of a finite number of elements K,

$$\overline{\Omega} = \bigcup_{K \in \mathcal{M}} \overline{K},$$

and we define the global space,

$$W_p(\Omega) = \{v \in H^1(\Omega) \ : \ v|_K \in W_p(K) \ \forall K \in \mathcal{M}\}.$$

The requirement that $v \in H^1(\Omega)$ reduces to the requirement that v is continuous at element interfaces (since clearly $v \in H^1(K)$). To see this, consider the generic interface Γ between two subdomains Ω_1 and Ω_2 as shown in Figure 3.1. Let $v|_{\Omega_i} = v_i$,

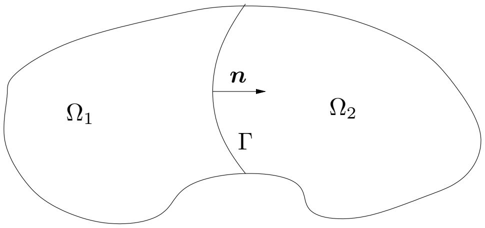


Figure 3.1: An interface Γ between subdomains

where $v_i \in H^1(\Omega_i)$, and let $\boldsymbol{\phi}$ be an arbitrary test function (C^∞ with compact support) in $\Omega = \operatorname{int}(\overline{\Omega_1} \cup \overline{\Omega_2})$. Then,

$$\begin{aligned}
\langle \boldsymbol{\nabla} v, \boldsymbol{\phi} \rangle_\Omega &\equiv -\langle v, \boldsymbol{\nabla} \cdot \boldsymbol{\phi} \rangle_\Omega \\
&= -\int_{\Omega_1} v_1 \boldsymbol{\nabla} \cdot \boldsymbol{\phi} \; dx - \int_{\Omega_2} v_2 \boldsymbol{\nabla} \cdot \boldsymbol{\phi} \; dx \\
&= \int_{\Omega_1} \boldsymbol{\nabla} v_1 \cdot \boldsymbol{\phi} \; dx + \int_{\Omega_2} \boldsymbol{\nabla} v_2 \cdot \boldsymbol{\phi} \; dx + \int_\Gamma (v_2 - v_1)(\boldsymbol{\phi} \cdot \boldsymbol{n}) \; dS,
\end{aligned}$$

so that $\boldsymbol{\nabla} v \in L^2(\Omega)$ if and only if $v_1 = v_2$ on Γ.

Though the restriction of orders for faces and edges is in principle arbitrary, our implementation assumes the convention, known as the minimum rule, that orders for faces and edges are set to the minimum of the parallel orders for adjacent elements. Finally, observe that the definition of the global space $W_p(\Omega)$ is also suitable for the case of an irregular mesh $\mathcal{M}$. In this case, the corresponding global space will be denoted by $W_{hp}(\Omega)$.

3.3 $H(\mathrm{curl})$-conforming hp elements

Motivated by the desired exact sequence property, the master element of uniform order is obtained as the image of the space (3.4) under the gradient, i.e.

$$\boldsymbol{Q}_p(\hat{K}) = \mathcal{Q}^{(p-1,q,r)}(\hat{K}) \times \mathcal{Q}^{(p,q-1,r)}(\hat{K}) \times \mathcal{Q}^{(p,q,r-1)}(\hat{K}). \tag{3.7}$$

Once again, the master element of variable order is introduced to enable a non-uniform order of approximation.

$$\begin{aligned}\boldsymbol{Q}_p(\hat{K}) &= \mathcal{Q}^{(p-1,q,r)}_{(p_f-1,q_f),(p_f-1,r_f),(q,r),p_e-1,q_f,r_f}(\hat{K}) \\ &\times \mathcal{Q}^{(p,q-1,r)}_{(p_f,q_f-1),(p,r),(q_f-1,r_f),p_f,q_e-1,r_f}(\hat{K}) \\ &\times \mathcal{Q}^{(p,q,r-1)}_{(p,q),(p_f,r_f-1),(q_f,r_f-1),p_f,q_f,r_e-1}(\hat{K}) \qquad (3.8)\end{aligned}$$

The choice of orders for restrictions to faces and edges is quite complex and is motivated by the final goal of constructing a discrete space of vector-valued functions such that only the tangential component is continuous between elements. Hence, considering the first component of (3.8), there is no restriction on the order of the faces orthogonal to direction 1 since the normal component of the function on these faces is identified with the *interior* of the element rather than the face. Likewise, the order for edges parallel to direction 2 corresponds to the order for the adjacent face in the 1-2 plane.

The definition of the transformation to the physical element K, such that the exact sequence property is preserved, is motivated by the action of the gradient on $W_p(K)$. If $v = \hat{v} \circ \boldsymbol{x}_K^{-1}$ then

$$\boldsymbol{\nabla} v = \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right)^T \hat{\boldsymbol{\nabla}} \hat{v},$$

where $\boldsymbol{\xi}$ denotes the master element coordinates and $\hat{\boldsymbol{\nabla}}$ is the gradient with respect to $\boldsymbol{\xi}$. This leads to the definition,

$$\boldsymbol{Q}_p(K) = \left\{ \boldsymbol{F} = \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right)^T (\hat{\boldsymbol{F}} \circ \boldsymbol{x}_K^{-1}) \ : \ \hat{\boldsymbol{F}} \in \boldsymbol{Q}_p(\hat{K}) \right\}. \qquad (3.9)$$

Then the curl satisfies the transformation rule [17],

$$\boldsymbol{\nabla} \times \boldsymbol{F} = \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right|^{-1} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} \hat{\boldsymbol{\nabla}} \times \hat{\boldsymbol{F}}.$$

Once again, the computational domain Ω is partitioned into a regular mesh $\mathcal{M}$ of curvilinear hexahedra and the global space is defined by,

$$\boldsymbol{Q}_p(\Omega) = \{\boldsymbol{F} \in \boldsymbol{H}(\text{curl}, \Omega) \ : \ \boldsymbol{F}|_K \in \boldsymbol{Q}_p(K) \ \forall K \in \mathcal{M}\}.$$

To obtain the continuity requirement for $\boldsymbol{F} \in \boldsymbol{H}(\text{curl}, \Omega)$, we reconsider the situation depicted in Figure 3.1. Now $\boldsymbol{F}_i \in \boldsymbol{H}(\text{curl}, \Omega_i)$, and

$$\begin{aligned}
\langle \boldsymbol{\nabla} \times \boldsymbol{F}, \boldsymbol{\phi} \rangle_\Omega &\equiv \langle \boldsymbol{F}, \boldsymbol{\nabla} \times \boldsymbol{\phi} \rangle_\Omega \\
&= \int_{\Omega_1} \boldsymbol{F}_1 \cdot \boldsymbol{\nabla} \times \boldsymbol{\phi} \, dx + \int_{\Omega_2} \boldsymbol{F}_2 \cdot \boldsymbol{\nabla} \times \boldsymbol{\phi} \, dx \\
&= \int_{\Omega_1} \boldsymbol{\nabla} \times \boldsymbol{F}_1 \cdot \boldsymbol{\phi} \, dx + \int_{\Omega_2} \boldsymbol{\nabla} \times \boldsymbol{F}_2 \cdot \boldsymbol{\phi} \, dx \\
&\quad + \int_\Gamma (\boldsymbol{n} \times \boldsymbol{F}_2 - \boldsymbol{n} \times \boldsymbol{F}_1) \cdot \boldsymbol{\phi} \, dS,
\end{aligned}$$

so that $\boldsymbol{\nabla} \times \boldsymbol{F} \in L^2(\Omega)$ if and only if $\boldsymbol{n} \times \boldsymbol{F}_1 = \boldsymbol{n} \times \boldsymbol{F}_2$ on Γ. This continuity requirement sheds some additional light on the practical importance of the exact sequence property and the transformation in (3.9). Indeed, if $\hat{f}$ is a face of the master element with image f then the transformation in (3.9) maps the component of $\hat{\boldsymbol{F}}$ tangential to $\hat{f}$ onto the component of $\boldsymbol{F}$ tangential to f. Hence, the continuity of the tangential component can be enforced directly in terms of the master element with no reference to the parametrization $\boldsymbol{x}_K$.

Chapter 4

Fully-Automatic *hp*-Adaptivity for Elliptic Equations

In this chapter we describe the mathematics behind, and implementation of, an algorithm for fully-automatic *hp*-adaptivity for elliptic problems in three dimensions. In the presented version, the adaptivity is driven by the minimization of the error in the H^1 norm. However, with relatively minor modifications [42], the algorithm extends to a more general goal-oriented adaptivity, driven by the minimization of an upper bound for the error in a quantity of interest.

The algorithm is quite general in that the search for optimal refinements allows for full anisotropy both in the element size h and the order of approximation p. It is also problem independent in the sense that it requires no explicit information about material data or the geometry of the computational domain. Rather, the algorithm automatically detects singularities (and smooth regions) in the solution and devises an appropriate refinement strategy to resolve them.

The central mathematical tool behind our automatic *hp* algorithm is the projection-based interpolation operator. We begin this section by recalling its definition and fundamental properties. We then turn to an overview of the algorithm for *hp*-adaptivity. Finally we discuss some of the details and computational issues behind its implementation.

4.1 Projection-based interpolation in H^1

The projection-based interpolation operator has been the subject of much research in recent years (see e.g. [23], [18], [22], [14], [20] and the references therein). Here, we only wish to recall its definition and basic properties. We first need to introduce a few underlying polynomial spaces.

For each hexahedral element K we denote by $\mathcal{Q}^p_{p_f,p_e}(K)$ the usual element space of shape functions. In this notation, $p = (p_1, p_2, p_3)$ is the possibly anisotropic order of approximation for the middle node, and p_f, p_e represent the possibly lower orders of approximation for restrictions to element faces and edges, respectively. For each of the six faces f of K we denote by $\mathcal{Q}^{p_f}_{-1}(f)$ the space of polynomials of possibly anisotropic order $p_f = (p_{f,1}, p_{f,2})$ whose restrictions to the boundary ∂f are zero. Finally, for each of the twelve edges e of K we denote by $\mathcal{P}^{p_e}_{-1}(e)$ the space of polynomials of order p_e whose values at the endpoints of e are zero.

Now, for a given function $u \in H^{3/2+\epsilon}(K)$, where $\epsilon > 0$ is arbitrary, the projection-based interpolant $u^p = \Pi_K u \in \mathcal{Q}^p_{p_f,p_e}(K)$ is defined by a four stage process:

- First we define u_0 to be the trilinear vertex interpolant of u, i.e. $u_0 \in \mathcal{Q}^1(K)$ such that $u_0(v) = u(v)$ for all eight vertices v of K. We note that by the Sobolev imbedding theorem, u is ϵ-Hölder continuous, and pointwise evaluation of u is a well-defined operation.

- Next for each edge e of K we solve the minimization problem:

$$\begin{cases} \text{Find } u_{1,e} \in \mathcal{P}^{p_e}_{-1}(e) : \\ \|(u - u_0) - u_{1,e}\|_{0,e} \to \min. \end{cases} \tag{4.1}$$

These twelve single-edge projections are then combined by extending each one with a bilinear blending factor in the plane orthogonal to the edge and adding them to obtain u_1.

- Next for each face f of K we solve the minimization problem:

$$\begin{cases} \text{Find } u_{2,f} \in \mathcal{Q}^{p_f}_{-1}(f) : \\ |(u - u_0 - u_1) - u_{2,f}|_{1/2,f} \to \min. \end{cases} \tag{4.2}$$

  These six single-face projections are then combined by extending each one with a linear blending factor in the direction orthogonal to the face and adding them to obtain u_2.

- Finally we solve the interior minimization problem:

$$\begin{cases} \text{Find } u_3 \in \mathcal{Q}^{p}_{-1}(K) : \\ |(u - u_0 - u_1 - u_2) - u_3|_{1,K} \to \min. \end{cases} \tag{4.3}$$

The minimization problems (4.1), (4.2) and (4.3) can also be expressed in the equivalent variational forms,

$$\begin{cases} \text{Find } u_{1,e} \in \mathcal{P}^{p_e}_{-1}(e) : \\ b_e(u_{1,e}, v) = b_e(u - u_0, v) \ \forall v \in \mathcal{P}^{p_e}_{-1}(e) \end{cases} \tag{4.4}$$

$$\begin{cases} \text{Find } u_{2,f} \in \mathcal{Q}^{p_f}_{-1}(f) : \\ b_f(u_{2,f}, v) = b_f(u - u_0 - u_1, v) \ \forall v \in \mathcal{Q}^{p_f}_{-1}(f) \end{cases} \tag{4.5}$$

$$\begin{cases} \text{Find } u_3 \in \mathcal{Q}^{p}_{-1}(K) : \\ b_K(u_3, v) = b_K(u - u_0 - u_1 - u_2, v) \ \forall v \in \mathcal{Q}^{p}_{-1}(K) \end{cases} \tag{4.6}$$

where

$$\begin{aligned} b_e(u,v) &= \int_e u(x)v(x)\ dx \\ b_f(u,v) &= \int_f\int_f \frac{(u(x)-u(y))(v(x)-v(y))}{|x-y|^3}\ dxdy \\ b_K(u,v) &= \int_K \boldsymbol{\nabla} u(x)\cdot\boldsymbol{\nabla} v(x)\ dx \end{aligned}$$

Then the projection-based interpolant u^p of u is given by the sum of these four contributions,

$$u^p = u_0 + u_1 + u_2 + u_3.$$

The use of bilinear extensions for edges, and linear extensions for faces, in the above algorithm is arbitrary since the final u^p is independent of the chosen extension. For a function u defined on the entire computational domain, discretized by a *regular* FE grid, its projection-based interpolant is the union of the element interpolants defined above.

The definition of the projection-based interpolant, and its application in automatic *hp*-adaptivity, is motivated by four main properties:

- It is optimal in the sense that it delivers the same convergence rate in both element size h and order of approximation p as the best approximation error.
- The definition is local in the sense that it only requires information about the function u restricted to a single element.
- The global projection is H^1 conforming.
- In conjunction with the $H(curl)$-conforming projection-based interpolation operator, the de Rham diagram [23] commutes.

For general *hp*-grids the last three properties are in conflict with each other and require further comment.

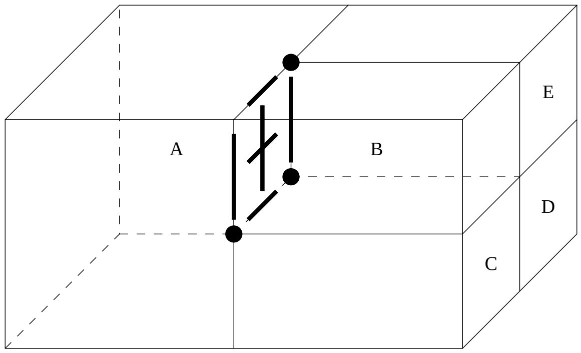


Figure 4.1: A typical situation involving constrained nodes in a 1-irregular mesh

For a general *hp*-grid we need to modify the definition of the projection-based interpolant to account for the presence of constrained (or hanging) nodes in the mesh in order to maintain H^1-conformity. Consider the situation depicted in Figure 4.1. Here, the highlighted vertex, edge and face nodes for element B are actually constrained by edge and face nodes for element A. If we apply the local definition of projection-based interpolation given above, separately on elements A and B, then we cannot guarantee that their union is in H^1.

To guarantee that the interpolant is in H^1 we have to slightly weaken our notion of locality. For the constrained nodes in element B, we can replace the local procedure by the following:

- Compute the usual projection for the constraining node from element A.
- Restrict the result to the constrained node for element B.
- Extend the result by the usual procedure into element B.

With this modification the interpolant thus obtained is globally in H^1, but its restriction to element B requires information about u, not only on element B, but also on the constraining face and edges of element A. Moreover, this definition violates the commutativity of the de Rham diagram. (An alternative definition recovers the commutativity of the de Rham diagram by further weakening locality. The projection can be defined over "partially refined" element patches. In Figure 4.1, the patch would include elements B, C, D and E. However, the resulting projection is impractical in the context of driving *hp*-refinements, since we wish to test the effects of further refinements to the elements)

The consequences of this slight non-locality, in terms of the practical implementation of projection-based interpolation (i.e. for automatic *hp*-adaptivity), are quite far-reaching. First, it is clear that the implementation will have to make frequent and explicit reference to the underlying data structures supporting the *hp*-mesh (node connectivities, constraints, etc.). Thus the code implementing automatic *hp*-adaptivity will be strongly tied to the underlying finite element code on which it is built. Secondly, the complexity of the code alone rules out the possibility for its implementation in a parallel, distributed memory, environment.

These main factors (code simplicity, portability, and potential for parallelization) have motivated a slightly different approach in the present work. The alternative is to trade H^1-conformity for true locality. The ramifications for each of these three issues will be made clear in the sequel.

4.2 Algorithm for fully-automatic *hp*-adaptivity

The main idea is that we search for optimal h and p refinements for a given coarse grid ignoring the fact that it may contain hanging or constrained nodes. This freedom allows all of the mesh optimization code to be essentially separated from the code used to maintain the mesh data structure. Once an optimal refinement strategy has been determined for every element, a mesh reconciliation algorithm is applied to ensure that the refinements can be performed while maintaining the 1-irregularity of the mesh. In this section we describe some of the details of the algorithm.

4.2.1 Overview

A single step of *hp*-adaptivity takes the following form.

1. Solve the problem on the current coarse grid and dump the coarse grid to disk.
2. Perform a global *hp*-refinement, breaking each element isotropically into eight sons and enriching the polynomial order of approximation by one.
3. Solve the problem on the resulting fine grid.
4. Compute the norm of the difference between the coarse and fine grid solutions as a global error estimate. Stop if the error is small enough.
5. Determine an optimal refinement strategy for edges.
6. Using edge refinements as the starting point, determine an optimal refinement strategy for faces.
7. Using face refinements as the starting point, determine an optimal refinement strategy for element interiors.
8. Possibly enrich the optimal refinements in order to preserve the 1-irregularity of the resulting mesh.
9. Load the coarse grid from disk and perform first h and then p-refinements to produce the new coarse grid.

(The stopping criterion, *item 4*, based on the coarse grid error estimate, will in the future be replaced by a-posteriori error estimation for the fine-grid solution.)

To achieve the aforementioned separation of the mesh optimization code, the fine grid solution is stored in an interface data structure in the refined-element fashion. For each coarse grid element, the solution is stored for all eight of its fine grid sons. This is the central data structure for mesh optimization. During mesh

Figure 4.2: An unrefined edge of order p, and an h-refined edge with order p_l on the left and order p_r on the right

optimization (steps 5-7), the fine grid solution is projected onto the coarse grid and onto a sequence of intermediate grids to investigate the relative benefits of h and p refinement. To compute the projection-based interpolant of the fine grid solution u, the first step (not listed above) is the removal of the coarse grid vertex interpolant u_0. With that completed, we are ready to proceed with edges.

4.2.2 The edge refinement algorithm

Each coarse grid edge is visited by looping through coarse grid elements, and then looping through the twelve edges of the element. The restriction of the modified fine grid solution $u - u_0$ to the coarse grid edge is retrieved from the interface data structure (observe that it is zero at the endpoints of the edge).

Our goal is to determine whether h or p-refinement, or no refinement, is appropriate for each edge. We accomplish this by projecting the modified fine grid solution onto a p-refined edge, and onto a sequence of h-refined edges (to be defined more precisely below). A schematic of the potential types of edges is shown in Figure 4.2. In order to discuss the h-refined edge in Figure 4.2, we denote the corresponding piecewise polynomial space by $\mathcal{P}_{-1}^{p_l,p_r}(e)$, with the subscript -1 indicating as usual zero values at the endpoints.

We begin by investigating the effect of p-enrichment. If the coarse grid edge e has order p, we project the fine grid solution onto the coarse grid space $P_{-1}^{p}(e)$ *and* onto the p-refined space $P_{-1}^{p+1}(e)$. For each projection we record the associated projection error, building a database for the edge.

To investigate the effect of h-refinement, we begin by projecting onto the piecewise linear space $P^{1,1}_{-1}(e)$. The projection error is computed separately for each edge son. If the error in the first son is largest, then we virtually increase the order of approximation for only the first son, and the next step is to project onto the space $P^{2,1}_{-1}(e)$. Otherwise, the second son's order is increased and we project onto $P^{1,2}_{-1}(e)$. Proceeding in the same way, a nested sequence of potential edge refinements is generated, from piecewise linears all the way up to the fine grid space $P^{p+1,p+1}_{-1}(e)$, and the associated projection errors are recorded (of course, the last one is always zero). We refer to this sequence as the *largest son error refinement path* since at each step the order is increased in the son with largest error.

An example maximum son error refinement path for a coarse edge of order five is represented schematically by the black dots in Figure 4.3, where the horizontal axis records the order for the left son, and the vertical axis is the order for the right son. For clarity, the solid diagonal line spans all potential h-refinements that are competitive (in terms of the number of local degrees of freedom, i.e. four) with the coarse edge, the dashed line spans those that are competitive with the p-refined edge (five degrees of freedom), and the dotted bounding box encloses all potential h-refinements imbedded in the fine grid. In principle, we could simply compute projection errors for all combinations with $1 \leq p_l, p_r \leq p+1$, however the present approach is designed to discover the optimal combinations of (p_l, p_r) while computing as few projections as possible (a strategy that becomes critical in higher dimensions).

It is useful to visualize the database thus collected. In Figure 4.4 we plot the edge projection errors with respect to the number of local degrees of freedom for the same example edge. The vertical solid and dashed lines indicate the number of degrees of freedom in the coarse, and in the p-refined edge, respectively. The projection errors for the coarse and p-refined edge are marked with squares, and projection errors along the maximum son error refinement path (from Figure 4.3) are marked with circles.

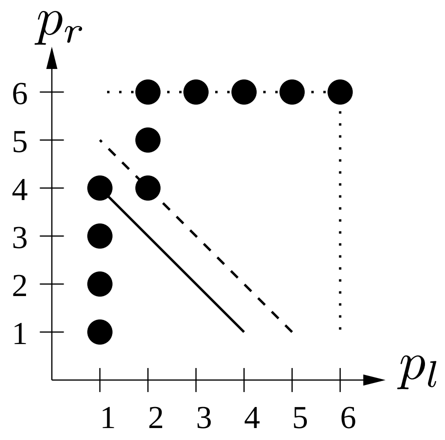


Figure 4.3: An example maximum son error refinement path for a coarse edge of order 5.

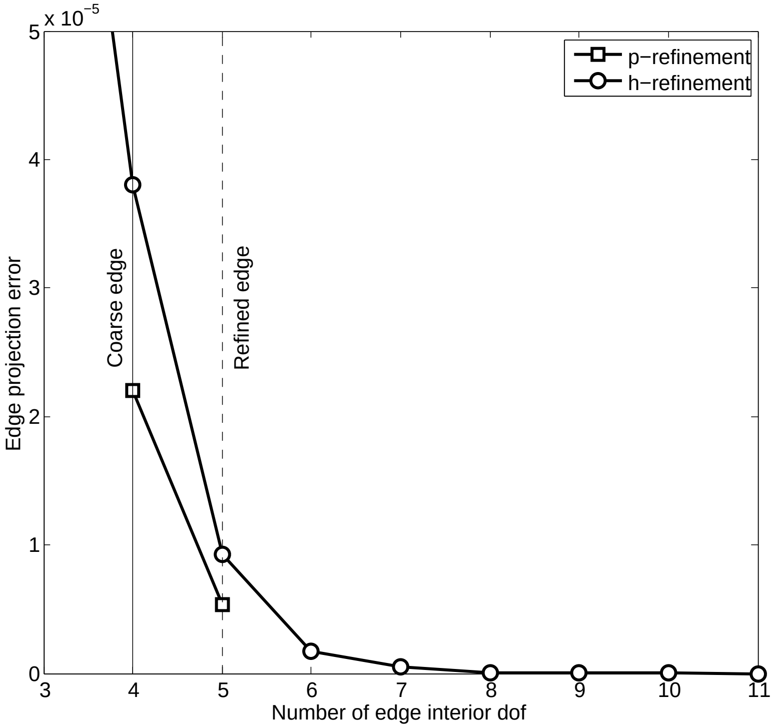


Figure 4.4: An example database of projection errors for a coarse edge of order 5.

We have to be ready to answer three questions:

1. Is this edge worthy of investment?
2. If not, should it remain unchanged?
3. If so, which of the h or p-refinements is best?

To answer question 2 we search the database to find the mesh with the fewest degrees of freedom among all meshes with projection error less than or equal to the coarse edge projection error. This is identified as the local *reference* mesh. Defined in this way, the reference mesh may simply be the coarse mesh, but includes the possibility of finding an h-refinement that achieves a lower error level than the coarse mesh with fewer degrees of freedom. For the example in Figure 4.4, the reference mesh is the coarse mesh, since h-refinement requires one additional dof to reproduce the coarse mesh error level.

To answer question 3 we identify the so-called best competitive refinement. For every mesh with more degrees of freedom than the reference mesh ($\mathrm{nrdof}_{\mathrm{ref}}$) but not more than the p-refinement (p) we compute the associated projection error decrease rate with respect to the number of local degrees of freedom added,

$$\mathrm{rate} = \frac{\mathrm{error}_{\mathrm{ref}} - \mathrm{error}}{\mathrm{nrdof} - \mathrm{nrdof}_{\mathrm{ref}}}.$$

The one delivering the largest error decrease rate is identified as the best *competitive* refinement. This is where (and how) the decision between h and p refinement is made. Clearly, in Figure 4.4, the best competitive refinement is p-refinement.

Finally we compute the maximum of this rate over all meshes in the database (even those with more degrees of freedom than the p-refinement). This rate represents a lower bound on the benefit of investing multiple degrees of freedom into this edge, and is therefore called the *guaranteed rate* for the edge. It quite often coincides with the best competitive refinement (clearly the case in Figure 4.4).

With this data collected for each edge, we are ready to answer question 1. The edges are entered into a global competition for investment. We compute the global maximum of the guaranteed-rate observed for each edge, and those that deliver rates above 70% of the max rate are deemed worthy of investment. In this case we will add degrees of freedom in the competitive *direction* (h or p) until the associated rate falls below 70% of the global max rate (if p-refinement wins the local competition only one degree of freedom can be added). All edges that lose in the global competition select their local reference mesh to ensure that the coarse grid error level is not exceeded.

There are three main features to note. First, we decide which direction (h or p) is best by comparing only refinements which add 1 degree of freedom (i.e. h-refinements adding more than one degree of freedom are not considered). This is because we are limited (for reasons to be discussed in § 4.3) to computing projections onto edges which are locally imbedded in the fine grid. To make this decision on a competitive basis h-refinements adding more than one degree of freedom would have to be compared with an unrefined edge of order $p + 2$ (or more), which is not imbedded in the fine grid. Second, the decision to invest in an edge is made based on the guaranteed rate for the edge, and not on the rate associated with the competitive refinement. This strategy addresses the so-called "case of a missing scale" [19]: It may happen that adding only one dof fails to resolve a dominant scale of the fine grid solution. Then the competitive refinement will deliver only a small error decrease rate. However, the error is still large, and convergence of the algorithm will "stall" through several steps until the global maximum rate is brought down to this level. By comparing guaranteed rates we can detect this situation and force the addition of one dof in the hope that a future iteration will then perform a more appropriate refinement. Finally, while we do not globally control the number of degrees of freedom that can be added in a single step, we can increase the cutoff

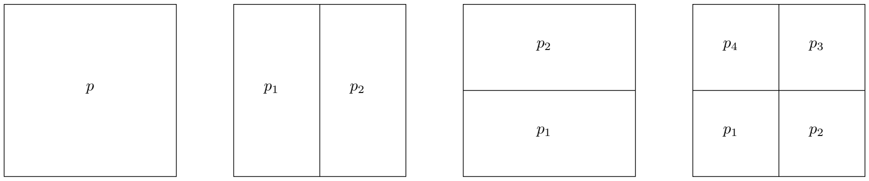


Figure 4.5: An unrefined face of (possibly anisotropic) order p, and the possible h-refinements.

(70%) to make more careful investments at the expense of requiring more iterations.

In a final step, the chosen edge projections are lifted into the element by a bilinear blending function in the plane orthogonal to the edge and subtracted from the fine grid solution in the interface data structure.

4.2.3 The face refinement algorithm

In a previous implementation of automatic hp-adaptivity [50], the determination of h-refinements was left entirely to edges and the so-called isotropy flags. At this point, the algorithm interfaces back to the mesh data structure and performs h-refinements, breaking elements according to whether or not their edges requested h-refinement. In the present work, we continue, independent of the mesh data structure, and stage a similar competition for faces.

The starting point for the face refinement algorithm is the ending point for the edge refinement algorithm. If none of the four edges of a given face have selected h-refinement, then we will presently consider all four possibilities shown in Figure 4.5. If however, some edges have selected h-refinement, then we will restrict our search to include only face refinements that also break the corresponding edges. For example, in Figure 4.6, the bottom edge selected h-refinement, so we restrict our search to the two h-refinement types shown on the right. Moreover, the optimal orders chosen for edges determine minimal orders for faces. We simply apply the familiar minimum rule in reverse (the maximum rule). In Figure 4.6, the face on the

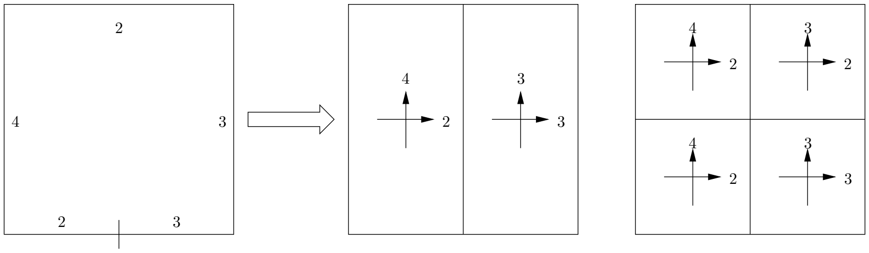


Figure 4.6: Using edges to restrict the search for optimal face refinements.

left indicates optimal orders of approximation chosen in the edge refinement step. The two faces on the right show the corresponding minimal orders of approximation for the face interiors, dictated by the edge refinements.

There are two main reasons for this coupling of edges to faces. First, since face projections are more expensive to compute than edge projections we want to use all of the information available to restrict the search as we go. Second, as will be seen below, the ultimate end of the algorithm is an optimal refinement strategy for *element interiors*, with the final edge and face orders determined by the minimum rule. Using the above strategy, we can guarantee that refinements chosen by edges alone are only potentially enriched by the algorithm using the additional information gained from faces and element interiors.

In our actual computations we replace the $H^{1/2}$-seminorm from §4.1 (dictated by the trace theorem) by a weighted H^1-seminorm. This choice is motivated mainly by the locality of the H^1-seminorm, i.e. it can be restricted to a single sub-element independent of others. Moreover, the H^1-seminorm splits nicely into separate contributions that can be examined to detect anisotropy in the error. Finally, we are not completely discarding the convergence theory since, in the presence of additional regularity, the H^1-projection still yields an optimal convergence rate (with respect to p) in the $H^{1/2}$-seminorm [22].

Optimal convergence with respect to h is recovered by selecting an appropriate weight function to interpolate between the scaling of the L^2 norm and H^1 seminorm. Suppose the (possibly curved) face f is parametrized by the map $\boldsymbol{x} : \hat{f} \ni \boldsymbol{\xi} \to \boldsymbol{x}(\boldsymbol{\xi}) \in f$, where $\hat{f} = (0,1)^2$ is the reference square. Then introducing tangent vectors $\boldsymbol{a}_i$ (and unit tangent $\boldsymbol{e}_i$),

$$\boldsymbol{a}_i = \frac{\partial \boldsymbol{x}}{\partial \xi_i} = h_i \boldsymbol{e}_i \ (i = 1, 2)$$

(no summation) we obtain the cobasis vectors,

$$\boldsymbol{a}^1 = h_1^{-1} \frac{\boldsymbol{e}_1 - (e_1 \cdot e_2)\boldsymbol{e}_2}{1 - (e_1 \cdot e_2)^2}, \ \boldsymbol{a}^2 = h_2^{-1} \frac{\boldsymbol{e}_2 - (e_1 \cdot e_2)\boldsymbol{e}_1}{1 - (e_1 \cdot e_2)^2},$$

and the face gradient,

$$\boldsymbol{\nabla}_f u = (\boldsymbol{\nabla} u \cdot a_1) a^1 + (\boldsymbol{\nabla} u \cdot a_2) a^2 = \frac{\partial u}{\partial \xi_1} \boldsymbol{a}^1 + \frac{\partial u}{\partial \xi_2} \boldsymbol{a}^2,$$

where $h_i = |\partial x / \partial \xi_i|$. Then we obtain the scaling of the L^2 norm and H^1 seminorm as

$$\begin{aligned} \|u\|_{0,f}^2 &= \int_f |u|^2 dx \\ &= \int_{\hat{f}} |u|^2 h_1 h_2 |e_1 \times e_2| \ d\xi \\ |u|_{1,f}^2 &= \int_f |\nabla_f u|^2 dx \\ &= \int_{\hat{f}} \left\{ \frac{h_2}{h_1} \left(\frac{\partial u}{\partial \xi_1} \right)^2 - 2(e_1 \cdot e_2) \frac{\partial u}{\partial \xi_1} \frac{\partial u}{\partial \xi_2} + \frac{h_1}{h_2} \left(\frac{\partial u}{\partial \xi_2} \right)^2 \right\} \frac{d\xi}{|e_1 \times e_2|}. \end{aligned}$$

To obtain a scaling like the $H^{1/2}$ seminorm, we interpolate half-way between the two (for each term) and define the face seminorm,

$$|u|_f^2 = \int_{\hat{f}} \left\{ h_2 \left(\frac{\partial \hat{u}}{\partial \xi_1} \right)^2 - 2(e_1 \cdot e_2) \sqrt{h_1 h_2} \frac{\partial u}{\partial \xi_1} \frac{\partial u}{\partial \xi_2} + h_1 \left(\frac{\partial \hat{u}}{\partial \xi_2} \right)^2 \right\} \frac{d\xi}{|e_1 \times e_2|}, \quad (4.7)$$

which reduces for a rectangular face to,

$$|u|_f^2 = \int_{\hat{f}} \left\{ h_2 \left(\frac{\partial \hat{u}}{\partial \xi_1} \right)^2 + h_1 \left(\frac{\partial \hat{u}}{\partial \xi_2} \right)^2 \right\} \, d\xi. \tag{4.8}$$

In fact, for a curvilinear face f, we compute average values for the anisotropic element size as,

$$h_1 = \int_{\hat{f}} \left| \frac{\partial x}{\partial \xi_1} \right| d\xi, \;\; h_2 = \int_{\hat{f}} \left| \frac{\partial x}{\partial \xi_2} \right| d\xi,$$

and use (4.8) to define the bilinear form for the projection,

$$b(u,v) = \int_{\hat{f}} \left\{ h_2 \frac{\partial \hat{u}}{\partial \xi_1} \frac{\partial \hat{v}}{\partial \xi_1} + h_1 \frac{\partial \hat{u}}{\partial \xi_2} \frac{\partial \hat{v}}{\partial \xi_2} \right\} \, d\xi. \tag{4.9}$$

The "correct" seminorm (4.7) for the curved face is only used to compute the associated projection error, but not to define the projection operator. This is because the simplified form (4.9) can be assembled from pre-computed stiffness matrices.

For faces, we cannot illustrate the algorithm graphically (as in Figure 4.3 for edges) because the maximum son error refinement path now involves up to eight different orders of approximation (anisotropic orders for up to four sons). However, we proceed in an analogous fashion to generate a local database of potential refinements and corresponding projection errors. For each admissible h-refinement type the fine grid solution (now with the coarse grid vertex interpolant and optimal edge projections removed) is projected onto a nested sequence of face interiors. The first projection corresponds to the minimal orders of approximation dictated by the edges. The next grid in the sequence is determined by carefully examining the projection error.

Consider first the case of a coarse face with order (p_1, p_2). We can observe that the norm (4.7) has three contributions: the first measures variation in ξ_1, the second is isotropic, and the third measures variation in ξ_2. If the first contribution to the error is large relative to the others, then we postulate that the best way to decrease the error is by increasing the order of approximation in the first direction (i.e. to $(p_1 + 1, p_2)$). If the three terms are relatively balanced we will increase the order isotropically (i.e. directly to the fine grid order $(p_1 + 1, p_2 + 1)$). For potential h-refinements, the local sequence is generated by enriching the son with the largest contribution to the error, using a similar strategy to detect anisotropy. By this process, we generate a sequence of potential refinements spanning the gap from the minimal order (dictated by edges) to the fine grid order for each admissible h-refinement type, recording the resulting projection errors in a local database.

This database is processed, much like it was for edges, and we identify a *reference mesh*, the best *competitive refinement* and the *guaranteed rate* for each face. The *guaranteed rates* are entered into a global competition, and those faces that deliver rates within 70% of the global max are selected for investment. The rest select the local reference mesh.

In a final step, the chosen projections are lifted into the element by a linear blending function in the direction normal to the face and subtracted from the fine grid solution in the interface data structure.

4.2.4 The brick refinement algorithm

With the above details on choosing face refinements, the algorithm for choosing refinements for element interiors is clear. However, the implementation is significantly more complex. We must be prepared to compute projections corresponding to up to eight different h-refinement types, as shown in Figure 4.7. To keep the discussion as simple as possible, we describe our approach in an abstract setting in § 4.3, which encompasses the computation of projections for edges, faces and element interiors.

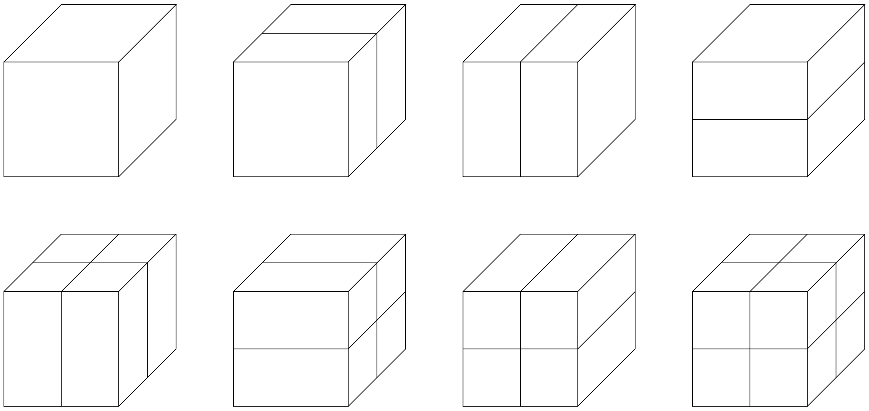


Figure 4.7: All potential h-refinement types for an element interior.

4.2.5 Mesh reconciliation

The output of the above algorithm is an h-refinement flag for each coarse grid element, and new orders of approximation for the corresponding element sons. The final step is to implement these refinements by modifying the underlying mesh data structure. In our case, the underlying data structure is managed by the code `3Dhp` [26]. We must first perform the requested h-refinements, followed by p-enrichment.

Before performing the h-refinements, we must ensure that they can, in fact, be performed while maintaining the 1-irregularity of the mesh. To accomplish this, it may be necessary to upgrade some of the requested h-refinements by adding so-called "unwanted refinements". Our approach is to first perform the refinements "virtually" by setting refinement flags in the data structure. These flags are then iteratively upgraded (if necessary) by repeated calls to a new subroutine: `prerefine`. This subroutine implements the logic of the old refinement routine, `refine` (see [26] for a complete discussion), but operates only with the flags and performs no actual refinements.

In addition to the previous logic from `refine`, `prerefine` eliminates inconsistent refinements for equal-sized neighbors. An example is shown in Figure 4.8.

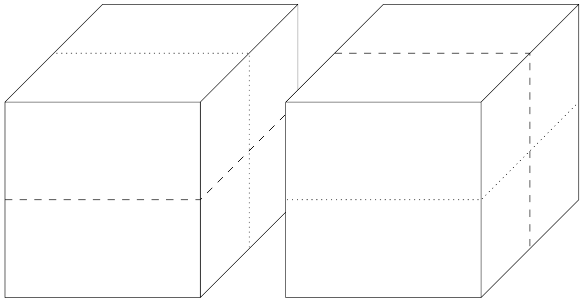


Figure 4.8: Inconsistent refinements for equal-sized neighbors (dashed lines), and how to upgrade them (dotted lines)

Here, the requested h-refinements (dashed lines) for two neighboring elements are inconsistent at the common face. The inconsistency is eliminated by upgrading both refinements (dotted lines). The procedure is repeated until no flags are modified by a given iteration. The process is guaranteed to terminate (in the worst case, the final refinements will be uniform and isotropic).

Now, the elements can be broken using the upgraded h-refinement flags, and we are ready to perform p-refinements. For the case of an element whose h-refinement flag was upgraded by the mesh reconciliation step, we have to determine new orders of approximation for the element sons. This is done by examining the database of projection errors for the element. First we identify a refinement that duplicates the *reference* error level for that element, and then see if there is a *competitive* refinement with the same h-refinement type. If not, the reference orders are used: if so, the competitive orders are used. Once orders have been chosen for all element middle nodes, orders are set for mid-face and mid-edge nodes according to the minimum rule.

The mesh reconciliation step is essentially the only portion of the code that must be treated differently in a distributed memory, parallel implementation (see [46, 45] for details of the parallel implementation). In the present algorithm, the search for optimal refinements can be carried out almost completely in parallel. The only need for communication between subdomains (prior to mesh reconciliation) is in the computation of global max error decrease rates.

4.3 Abstract framework for computing projections

The central computational problem behind our algorithm for fully-automatic *hp*-adaptivity is that of computing a sequence of projections. We need to compute projections onto edges, faces and element interiors, for a variety of admissible h-refinement types, and for orders of approximation varying between given minimal and maximal orders. For example, in $3D$, we must be prepared to compute projections onto element interiors with up to eight different h-refinement types (one for no h-refinement, three for $h2$-refinement, three for $h4$-refinement and one for isotropic $h8$-refinement).

We pose the problem in terms of four different piecewise polynomial spaces, each one containing the next, but with imbeddings of different type. We denote the spaces as $U \subset V \subset W \subset X$. The space X is the fine grid piecewise polynomial space restricted to a single coarse grid edge, face or element. We represent elements of X by their expansion in terms of a basis of hierarchical, piecewise polynomial shape functions $\{x_i\}_{i=1}^{N_X}$. We also require that this basis can be separated as usual into so-called interior or bubble modes, which are zero on the boundary of the coarse grid element, and exterior or boundary modes, which are nonzero on the boundary.

The subspace $W \subset X$ is spanned by only the interior degrees of freedom from X. That is $W = \text{span}\{w_i\}_{i=1}^{N_W}$, and there is an extraction vector (injection), $\text{extw} : \{1, \ldots, N_W\} \rightarrow \{1, \ldots, N_X\}$, such that $w_i = x_{\text{extw}(i)}$. The new symbol w_i for shape functions is introduced only to indicate this renumbering.

The subspace $V \subset W$ is introduced to compute projections onto elements of different h-refinement types. For the simplest example, consider the imbedding of a

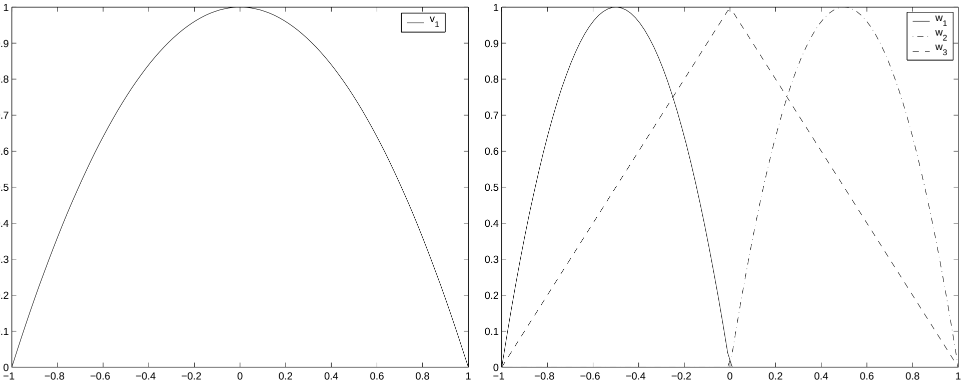


Figure 4.9: On the left, the single shape function spanning V; on the right, the three shape functions spanning W.

second order edge interior V into its h-refined counterpart W. The basis functions for V and W are shown in Figure 4.9. We can represent the imbedding $V \subset W$ by introducing an extension matrix E with one column for each shape function for $V = \text{span}\{v_i\}_{i=1}^{N_V}$, listing the coefficients for its expansion in terms of shape functions for W. In this example we have only a single column and the corresponding expansion,

$$v_1(x) = \frac{1}{4} w_1(x) + \frac{1}{4} w_2(x) + w_3(x).$$

In general we have an expansion of the form,

$$v_i = \sum_{j=1}^{N_W} E_{j,i} w_j, \ i = 1, \ldots, N_V.$$

Finally, the subspace $U \subset V$ is introduced to compute projections onto elements having a lower order of approximation. Consequently the imbedding $\text{span}\{u_i\}_{i=1}^{N_U} = U \subset V$ is represented by an additional extraction vector $\text{extu} : \{1, \ldots, N_U\} \to \{1, \ldots, N_V\}$, such that, $u_i = v_{\text{extu}(i)}$.

We are now ready to express the problem of interest: projecting an element of X onto U. Let $x \in X$ be represented by the expansion $x(\xi) = \sum_{j=1}^{N_X} x^j x_j(\xi)$, and its projection onto U by the expansion $u(\xi) = \sum_{j=1}^{N_U} w^j u_j(\xi)$. If we take $b : X \times X \to R$ to be the appropriate bilinear form for edge, face or element interior projection, then we can state the problem as: find coefficients $\{u^j\}_{j=1}^{N_U}$ such that,

$$\sum_{j=1}^{N_U} B^U_{i,j} u^j = \sum_{j=1}^{N_U} b(u_j, u_i) u^j = \sum_{j=1}^{N_X} b(x_j, u_i) x^j = l^U_i,$$

for each $i = 1, \ldots, N_U$.

Using the extraction vector extu, the stiffness matrix B^U and load vector l^U can be extracted from the corresponding stiffness matrix B^V and load vector l^V for computing a projection onto V, i.e.

$$B^U_{i,j} = b(u_j, u_i) = b(v_{\mathrm{extu}(j)}, v_{\mathrm{extu}(i)}) = B^V_{\mathrm{extu}(i),\mathrm{extu}(j)},$$

and

$$l^U_i = \sum_{j=1}^{N_X} b(x_j, u_i) x^j = \sum_{j=1}^{N_X} b(x_j, v_{\mathrm{extu}(i)}) x^j = l^V_{\mathrm{extu}(i)}.$$

Now, using the extension matrix E, B^V and l^V can be computed from B^W and l^W, the stiffness matrix and load vector for projection onto W, i.e.

$$\begin{aligned}
B^V_{i,j} = b(v_j, v_i) &= b\left(\sum_{l=1}^{N_W} E_{l,j} w_l, \sum_{k=1}^{N_W} E_{k,i} w_k\right) \\
&= \sum_{l,k=1}^{N_W} E_{k,i} b(w_l, w_k) E_{l,j} \\
&= \sum_{l,k=1}^{N_W} E_{k,i} B^W_{k,l} E_{l,j}, \\
l^V_i = \sum_{j=1}^{N_X} b(x_j, v_i) x^j &= \sum_{j=1}^{N_X} b\left(x_j, \sum_{k=1}^{N_W} E_{k,i} w_k\right) x^j \\
&= \sum_{k=1}^{N_W} E_{k,i} \sum_{j=1}^{N_X} b(x_j, w_k) x^j
\end{aligned}$$

$$= \sum_{k=1}^{N_W} E_{k,i} l_k^W .$$

In short, $B^V = E^T B^W E$ and $l^V = E^T l^W$.

Finally, the stiffness matrix B^W and load vector l^W are assembled using precomputed stiffness matrices for a single element (face or edge) and the fine grid solution $x \in X$.

One drawback to this approach is the need for extension matrices. In 3D, it would seem that extension matrices must be precomputed for imbedding seven types of unrefined or partially-refined elements into the isotropically refined element with a maximal order of approximation. The relevant extension matrix E above, can then be extracted on the fly (even this is only possible with a hierarchical basis). However, for a maximal order of 9, the resulting extension matrices (in double precision) for element interiors require over 400MB of memory! To overcome this significant obstacle, the seven maximal extension matrices were replaced by seven subroutines (each with two integer arguments that represent array indices) that use the tensor product structure of 3D shape functions and the extension matrix for an edge (which requires less than 1KB of memory) to compute entries for the 3D extension matrices *without explicitly storing them.*

Chapter 5

Numerical Results

This chapter presents numerical evidence of the exponential convergence of the *hp*-algorithm for elliptic problems in three dimensions. We begin with the classical model problems for the Laplace equation, namely the Fichera corner domain, where singularities are induced by a re-entrant corner and edges, and the so-called shock problem where the manufactured exact solution has an interior layer with arbitrarily steep gradients. The algorithm is then applied to the exterior acoustic scattering problem for a sphere, a cone-sphere and a thin rectangular plate. To demonstrate the practicality of the implementation, all computations were performed on a Compaq Presario laptop with 2GB RAM.

5.1 Fichera's corner

Our first model problem is Laplace's equation ((2.1) with $f = 0$). The domain $\Omega = (-1,1)^3 \backslash [0,1)^3$, known as Fichera's corner (see Figure 5.1), has three re-entrant edges and one re-entrant corner. As in [50] we use homogeneous Dirichlet data, and the problem is driven by an inhomogeneous Neumann boundary condition. The Neumann data is derived by superposing three singular solutions for the analogous

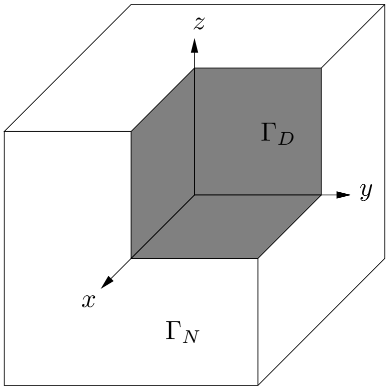


Figure 5.1: Fichera's corner domain with Dirichlet (shaded) and Neumann portions of the boundary

problem for the L-shaped domain in 2D. This choice of Neumann data is to focus our study on the edge and vertex singularities due to the re-entrant corner, and not on additional singularities due to incompatible Dirichlet and Neumann data at the interface between Γ_D and Γ_N. The initial mesh consists of seven hexahedral elements of second order.

While the exact solution for this problem is unknown, an "overkill" solution u can be used to approximate the energy norm of the error for a given *hp*-grid. In fact, if the *hp*-grid is imbedded in the "overkill"-grid, then the energy norm of the difference is given by,

$$\|u - u^{hp}\|_E^2 = b(u - u^{hp}, u - u^{hp}) = \|u\|_E^2 - \|u^{hp}\|_E^2.$$

By storing the energy norm for each coarse and fine grid obtained by successive *hp*-adaptive iterations, we can use the energy of the final fine grid to plot the error with respect to the number of degrees of freedom. Figure 5.2, shows the estimated percent relative error (in a logarithmic scale) with respect to the number of degrees of freedom (in the algebraic scale $N^{1/5}$), for both the *hp* coarse and fine grids. For

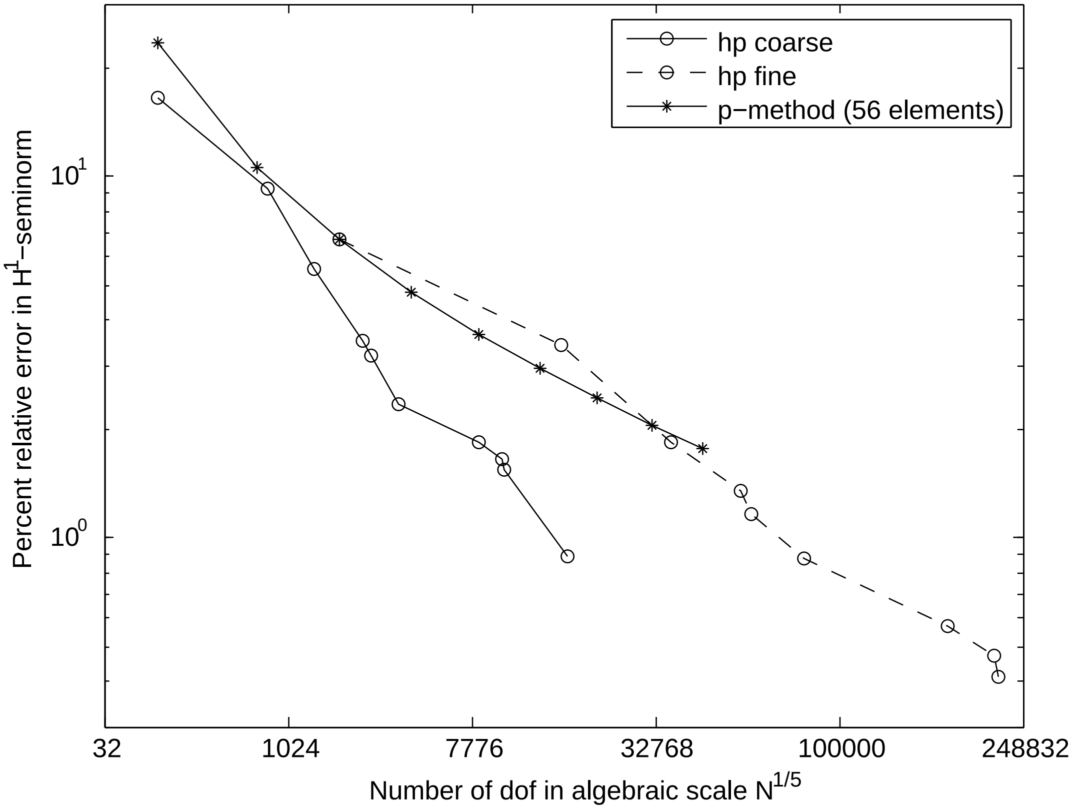


Figure 5.2: Estimated convergence curves for the *hp* coarse and fine grids, and the *p* method with 56 elements, for the Fichera problem

reference, we have included the error for the p-method on a uniform grid of 56 elements. The nearly linear plots indicate exponential convergence of the form,

$$\|u - u^{hp}\|_E \sim \alpha e^{-\beta N^{1/5}}.$$

The convergence of the p-method is clearly algebraic, and in fact we observe that,

$$\|u - u^{p}\|_E \approx 24p^{-19/16}$$

The final hp coarse grid plotted in Figure 5.2 has an estimated error of 0.9% with 17K degrees of freedom (which requires at least $p = 15$ or around 200K degrees of freedom for the p-method). The final hp fine grid has an estimated error of 0.5% with 220K degrees of freedom (which requires at least $p = 26$ or around 1 million degrees of freedom for the p-method). The "overkill" solution used to estimate the energy of the exact solution had over 325K degrees of freedom.

The final coarse grid solution is shown in Figure 5.3, and the final coarse grid with colors indicating the order of approximation is shown in Figure 5.4. The color scheme in Figure 5.4 (and all meshes to follow) requires further comment. Because the order of approximation is in general anisotropic, a single color cannot be used to represent the order for any given face. Moreover, because of the minimum rule, the orders of approximation for edges adjacent to a given face may be lower than the order for the face. Hence, the color scheme in general subdivides each face into 6 regions, shown in Figure 5.5. In this case the face has horizontal order $p_f^h = 2$ and vertical order $p_f^v = 3$, while the order for horizontal edges has been restricted to $p_e^1 = p_e^3 = 1$ and for vertical edges to $p_e^2 = p_e^4 = 2$.

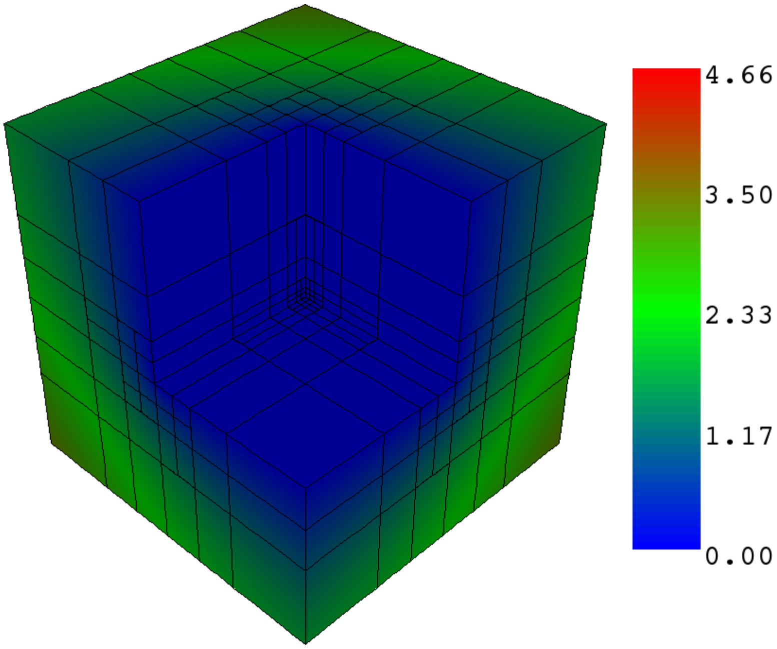
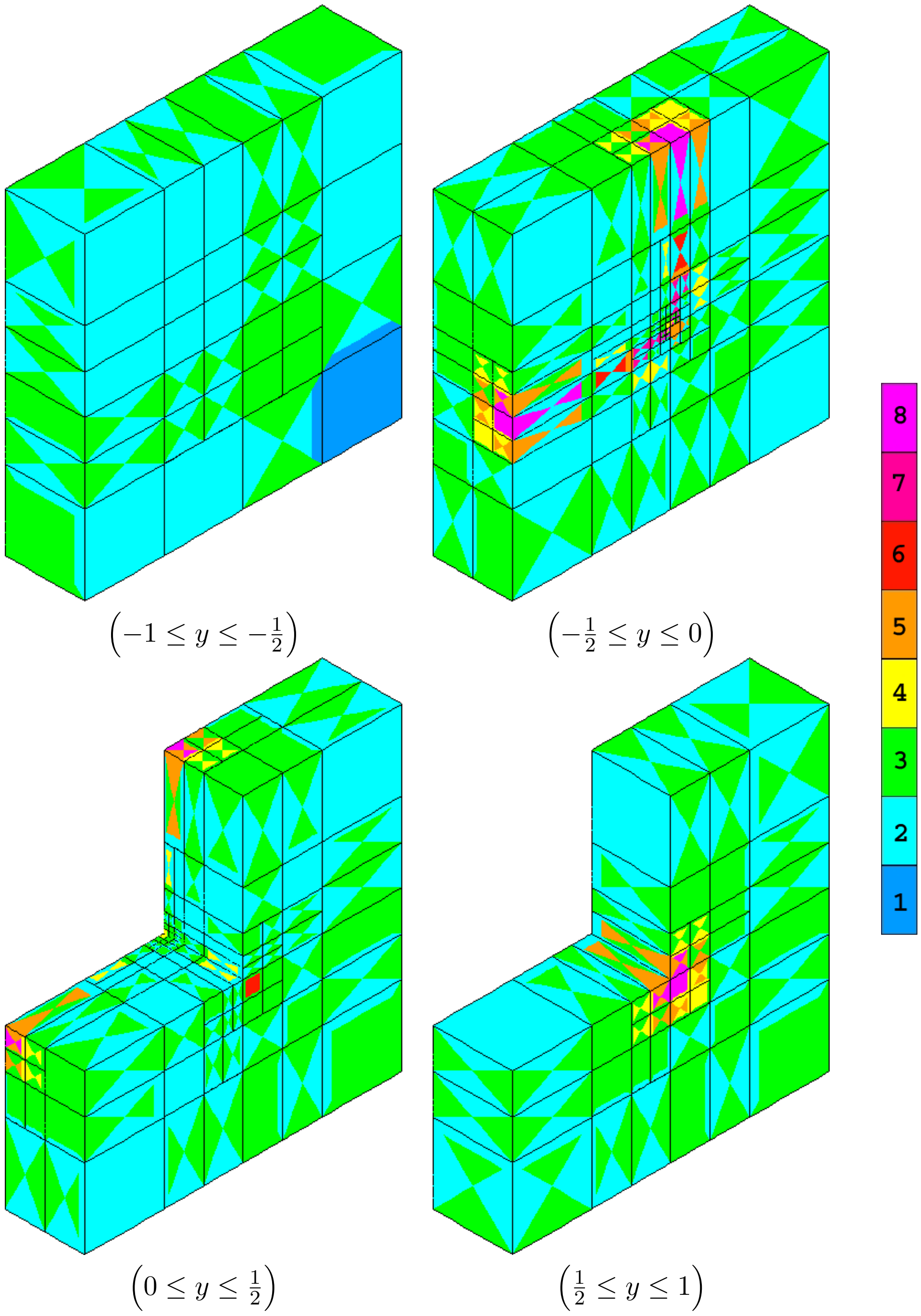


Figure 5.3: Solution to the Fichera problem

5.2 An exact solution with a "shock"

The second model problem is Poisson's equation in the unit cube $(0,1)^3$ with mixed Dirichlet and Neumann boundary conditions (equations (2.1)-(2.3)). The data f, g, h are manufactured from an exact solution chosen to have an arbitrarily sharp internal layer on the surface of a sphere. Figure 5.6 displays the generating function $f(x) = \tan^{-1}(\alpha x)$ with the parameter $\alpha = 20, 40$ and 60. Clearly, increasing α leads to steeper gradients in a more focused region. The corresponding 3D exact solution is $u(r) = \tan^{-1}(\alpha(r - r_0))$, where $r = |\boldsymbol{x} - (1/4, 1/4, 1/4)|$ is a recentered radial coordinate, and $r_0 = \sqrt{3}$ is the radius of the spherical layer. To drive the problem with this exact solution we take $f = -\Delta u$, $g = u$ and $h = \frac{\partial u}{\partial n}$. The Dirichlet data g is prescribed on the planes $x = 0$, $y = 0$ and $z = 0$, while the Neumann data h is prescribed on the remainder of the boundary. Because the data f and h in (2.4) are



$\left(-1 \leq y \leq -\frac{1}{2}\right)$

$\left(-\frac{1}{2} \leq y \leq 0\right)$

$\left(0 \leq y \leq \frac{1}{2}\right)$

$\left(\frac{1}{2} \leq y \leq 1\right)$

Figure 5.4: The final coarse grid for the Fichera problem divided into four slabs along the y-axis in order to expose the refinements in the interior of the domain.

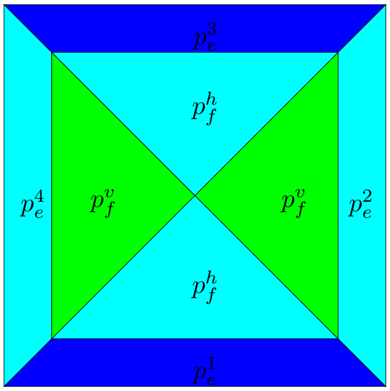


Figure 5.5: Color scheme for displaying the (in general anisotropic) orders of approximation for a single quadrilateral face. In this case the face has horizontal order $p_f^h = 2$ and vertical order $p_f^v = 3$, while the order for horizontal edges has been restricted to $p_e^1 = p_e^3 = 1$ and for vertical edges to $p_e^2 = p_e^4 = 2$

nearly singular, the corresponding volume and surface contributions to the element load vector are computed using adaptive quadrature schemes.

In Figures 5.7–5.9 we plot the exact percent relative error in the H^1 seminorm (in a logarithmic scale) with respect to the number of degrees of freedom (in the algebraic scale $N^{1/3}$) for both the sequence of coarse and fine grids and for increasing parameter values $\alpha = 20, 40$ and 60. In each case, the initial coarse grid is just a single trilinear element. The scale $N^{1/3}$ is used since the exact solution is analytic (though it may appear to be singular in the pre-asymptotic regime). For reference, the figures include the convergence of the p-method ($p = 1, \ldots, 9$) on a uniform grid of 64 elements. Indeed, we do observe an exponential convergence for the p-method of the form,

$$\text{error} = C 10^{-\gamma N^{1/3}},$$

with roughly,

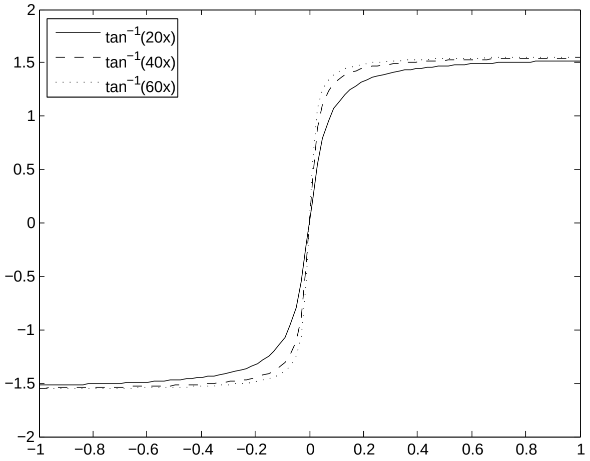


Figure 5.6: A plot of the 1D generating function $\tan^{-1}(\alpha x)$

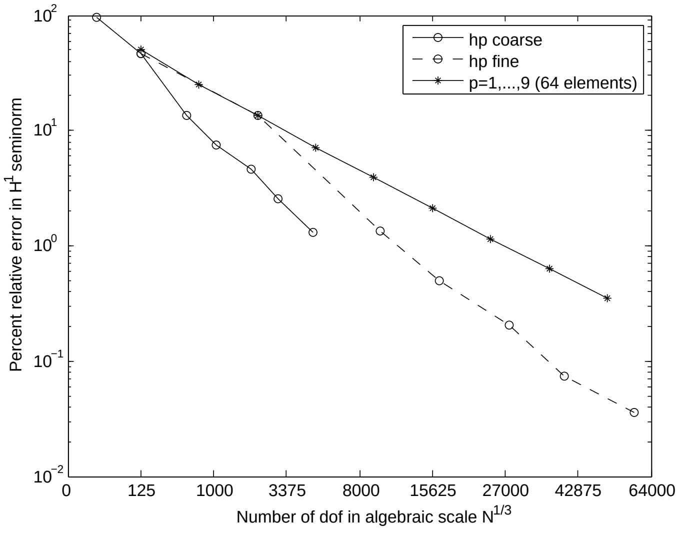


Figure 5.7: Convergence history for the hp coarse and fine grids compared with the p-method on a uniform grid of 64 elements, for the "shock" problem ($\alpha = 20$)

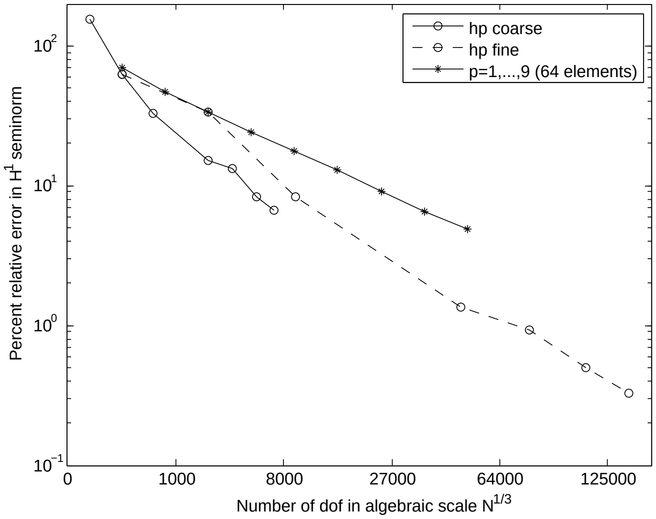


Figure 5.8: Convergence history for the hp coarse and fine grids compared with the p-method on a uniform grid of 64 elements, for the "shock" problem ($\alpha = 40$)

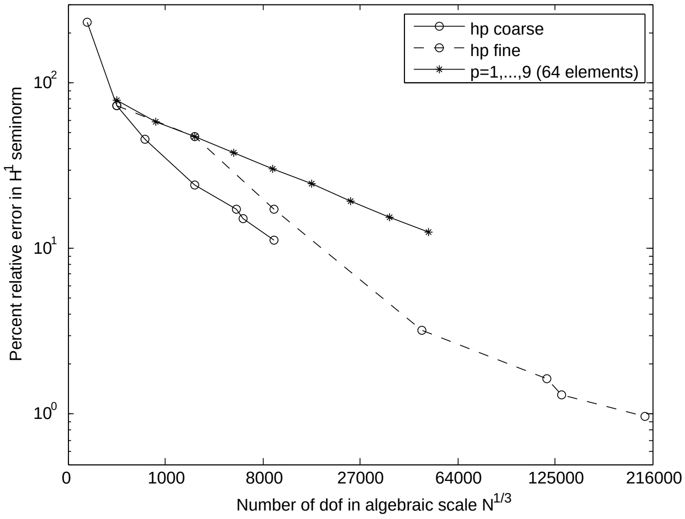


Figure 5.9: Convergence history for the hp coarse and fine grids compared with the p-method on a uniform grid of 64 elements, for the "shock" problem ($\alpha = 60$)

α	Fine grid % error	Fine grid nrdof	p-method nrdof
20	0.0355	58435	140000
40	0.3265	140469	370000
60	0.9787	208029	520000

Figure 5.10: Summary of final fine grid results for the “shock” problem and projected problem size to match the same error level with the p-method

α	C	γ
20	100	$\frac{1}{15}$
40	80	$\frac{1}{30}$
60	100	$\frac{1}{40}$

The *hp* coarse and fine grids appear to asymptotically approach the same rate, but they do much better in the pre-asymptotic regime. Figure 5.10 contains a summary of the results for the final *hp* fine grids, and projects the number of degrees of freedom required by the p-method to achieve the same error level.

Figure 5.11 compares the exact solution along with the final *hp* coarse grid for $\alpha = 20, 40$ and 60. For the smoothest case $\alpha = 20$, the algorithm selects only two levels of adaptive h-refinement followed by only adaptive p-enrichment. As α is increased, the algorithm selects h-refinements more readily in order to resolve and isolate the layer.

5.3 Acoustic scattering from a sphere

We consider the problem of rigid acoustic scattering of an incident plane wave by a sphere of radius a centered at the origin. For simplicity, the incident wave is assumed to travel toward the origin from the positive z direction. With the assumption of

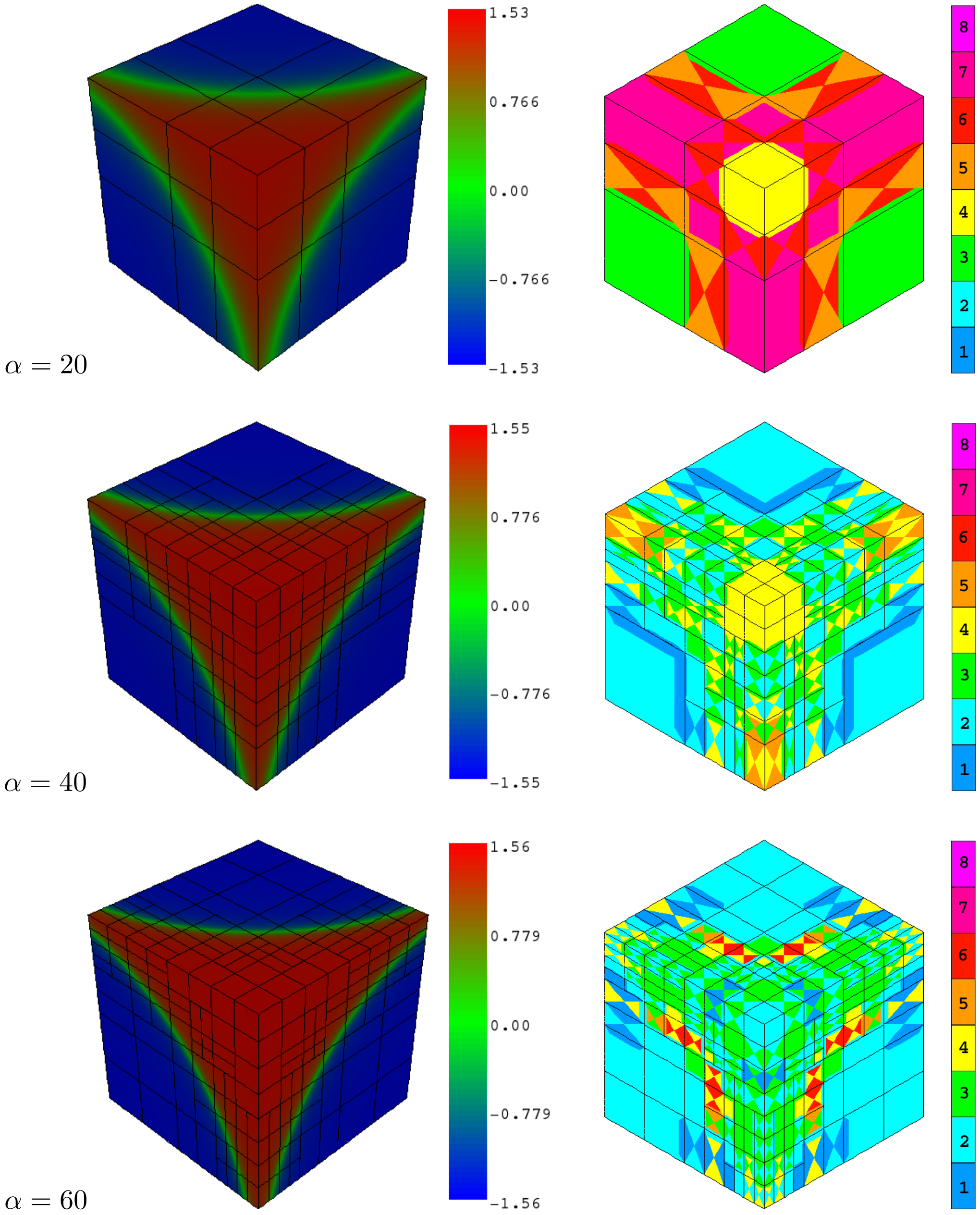


Figure 5.11: Exact solution (left) and sixth coarse grid (right) for the "shock" problem with parameter $\alpha = 20, 40$ and 60

an $e^{+i\omega t}$ time-dependence, the incident wave is given by

$$p^{inc} = p_0^{inc} e^{ikz} = p_0^{inc} e^{ikr\cos\theta} \tag{5.1}$$

where $k = \omega/c$ is the wave number, c is the speed of sound in the surrounding homogeneous medium and ω the angular frequency. The scattered pressure p satisfies (2.5)–(2.7).

Using separation of variables one finds that the scattered pressure has the general form (see, e.g. [34])

$$p = \sum_{n=0}^{\infty} \sum_{m=0}^{n} h_n^{(2)}(kr) P_n^m(\cos(\theta)) \left\{ a_{nm}\cos(m\phi) + b_{nm}\sin(m\phi) \right\},$$

where $h_n^{(2)}$ is the (n-th) spherical Hankel function of the second kind, and P_n^m is the Legendre function of the first kind of degree n and order m (see [1]). Because of the symmetry of (5.1) about the z-axis, this expansion simplifies to

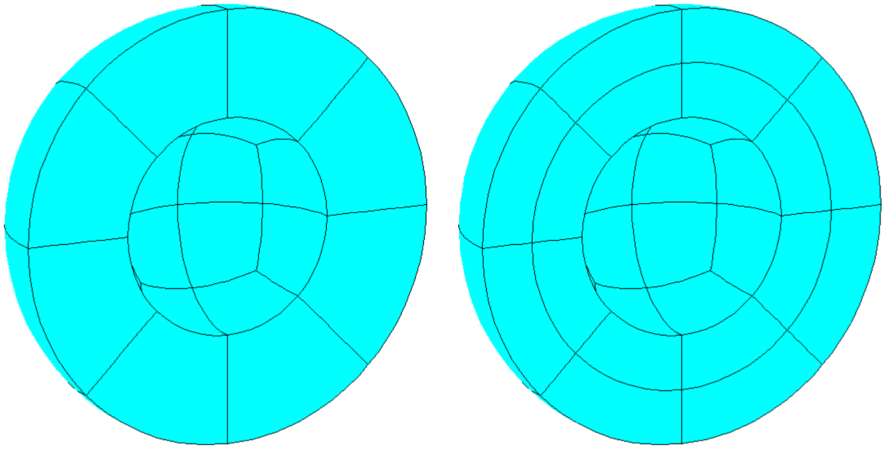
$$p = \sum_{n=0}^{\infty} c_n h_n^{(2)}(kr) P_n(\cos\theta), \tag{5.2}$$

where P_n is the Legendre polynomial of degree n. The coefficients c_n are determined by the Neumann boundary condition (2.6). Here we make use of the identity [1, 10.1.47],

$$e^{ikr\cos\theta} = \sum_{n=0}^{\infty} (2n+1) i^n j_n(kr) P_n(\cos\theta),$$

(j_n being the n-th spherical Bessel function of the first kind) to obtain,

$$c_n = -u_0^{\text{inc}} (2n+1) i^n \frac{j_n'(ka)}{h_n^{(2)\prime}(ka)}. \tag{5.3}$$



IE mesh A IE mesh B

Figure 5.12: Cutaway view of two initial meshes for infinite element discretization (only finite elements are shown)

For an incident wave *from* an arbitrary direction $\hat{e}$,

$$p^{inc} = p_0^{inc} e^{ik\hat{e}\cdot x}$$

we simply evaluate (5.2) at $r = |x|$ and $\cos(\theta) = \hat{e} \cdot x/|x|$. Of course, only the truncated sum,

$$p \approx \sum_{n=0}^{N} c_n h_n^{(2)}(kr) P_n(\cos\theta), \tag{5.4}$$

is evaluated, and we use the usual guideline [34] $N \approx 2k$. In the following computational results, we will consider a scatterer of radius 1 wavelength.

Since the exact solution is analytic in the exterior domain we will begin with a comparison of the infinite element and PML truncations. In both cases, we surround the scatterer with a layer of finite elements of thickness λ (i.e. extending from $r = \lambda$ to $r = 2\lambda$), and initial order $p = 2$. This mesh is then either extended by infinite elements (see Figure 5.12(a)) or by another layer of finite elements of thickness λ (i.e. extending from $r = 2\lambda$ to $r = 3\lambda$) where we employ the complex

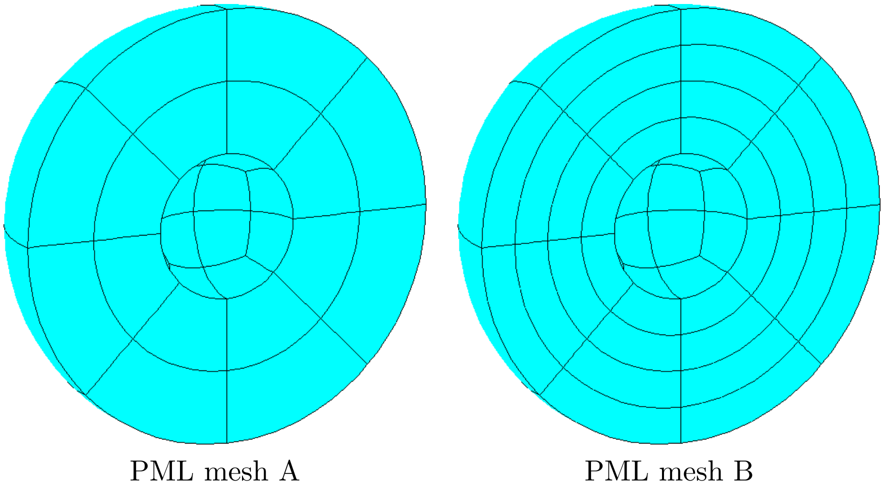


Figure 5.13: Cutaway view of two initial meshes for PML discretization (the PML is the outer layer of elements in mesh (A) and outer two layers in mesh (B))

coordinate stretching (2.17) and truncate with a homogeneous Dirichlet boundary condition (see Figure 5.13(a)). In Figure 5.14 we plot the percent relative error in H^1-seminorm evaluated over the interior domain ($\lambda < r < 2\lambda$) vs the total number of degrees of freedom. With the error in a logarithmic scale, and the number of degrees of freedom in the algebraic scale $N^{1/3}$, we observe exponential convergence for the infinite element method with respect to p, as expected.

For the PML truncation, we plot results for different profiles $n = 2, \ldots, 8$ with higher n indicating a smoother transition into the PML (compare (2.17)). We see that the results are not as clean as they are for infinite elements. First, the error within the interior region is unstable with respect to p, and may actually increase. Second, none of the tested profiles emerges as a clear winner over the others. The difficulty stems from the fact that the complex coordinate stretching introduces (by design) a layer within the PML where the solution switches from oscillation to exponential decay. These tests indicate that the p-method alone inadequately resolves this layer (at least pre-asymptotically), and the solution is adversely effected in the interior region as well.

In an attempt to capture this interior layer, we begin with a single radial h-refinement to obtain the initial meshes shown in Figure 5.12(b) and Figure 5.13(b), and again apply the p-method. The improved convergence curves are shown in Figure 5.15. We can observe that the radial h-refinement successfully captures the interior layer, and that the convergence generally improves for smoother profiles. Figure 5.16 shows the real and imaginary parts of the solution for the infinite element truncation with $p = 9$, and we can observe that the scattered pressure waves pass cleanly through the truncating sphere. Figure 5.17 shows the real and imaginary parts of the solution for the PML truncation with profile $n = 7$ and order $p = 9$, and we can observe that the scattered pressure waves cleanly decay into the absorbing layer.

The preceding observations motivate the use of our fully-automatic hp algorithm for the resolution of irregularities introduced by PML truncation. Our algorithm should automatically detect the need for radial h-refinements within the PML, and select an optimal distribution of anisotropic order p. We have applied the algorithm using the initial coarse grid from Figure 5.13(a) (where the uniform p-method has significant difficulty) and the first four PML profiles $n = 2, \ldots, 5$. The results are shown in Figures 5.18–5.21. In each case we plot the exact error for both sequences of hp-coarse and fine grids. For comparison, we have also included the convergence of the p-method for infinite elements on mesh (a) and for PML on both meshes (a) and (b).

First, the hp-adaptivity successfully delivers exponential convergence for the coarse grid, and in fact delivers results comparable to the infinite elements despite the irregularity introduced by the PML. Moreover, the exponential convergence is achieved, not only asymptotically but throughout the range of N. This is actually somewhat surprising because the refinements in the coarse grid can only be as good

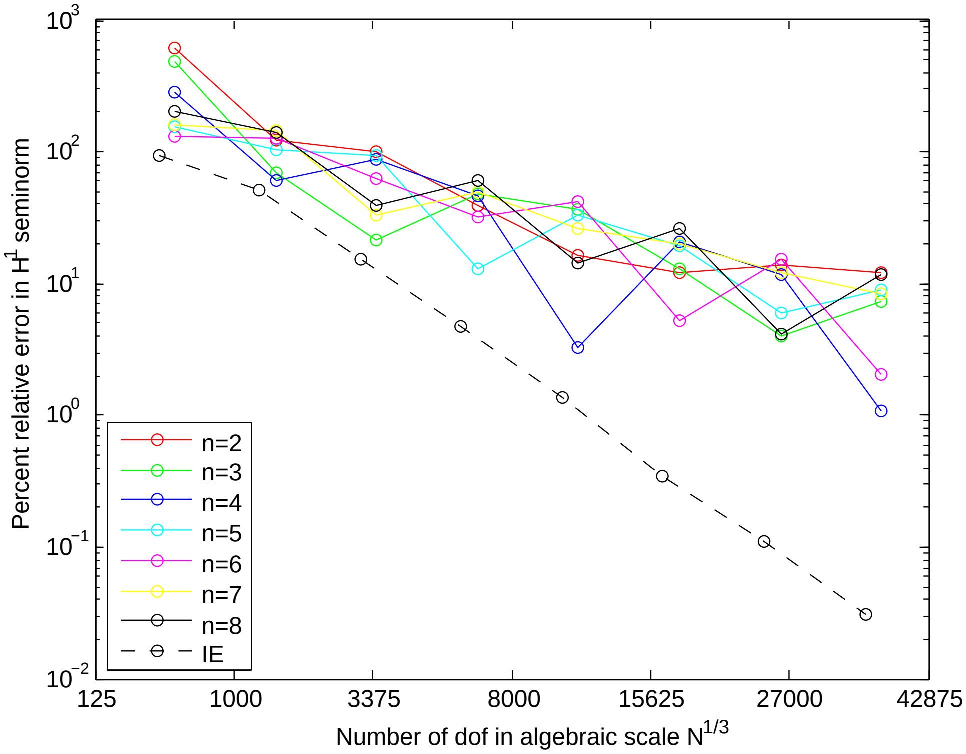


Figure 5.14: Convergence of percent relative error evaluated over the interior region for mesh (a)

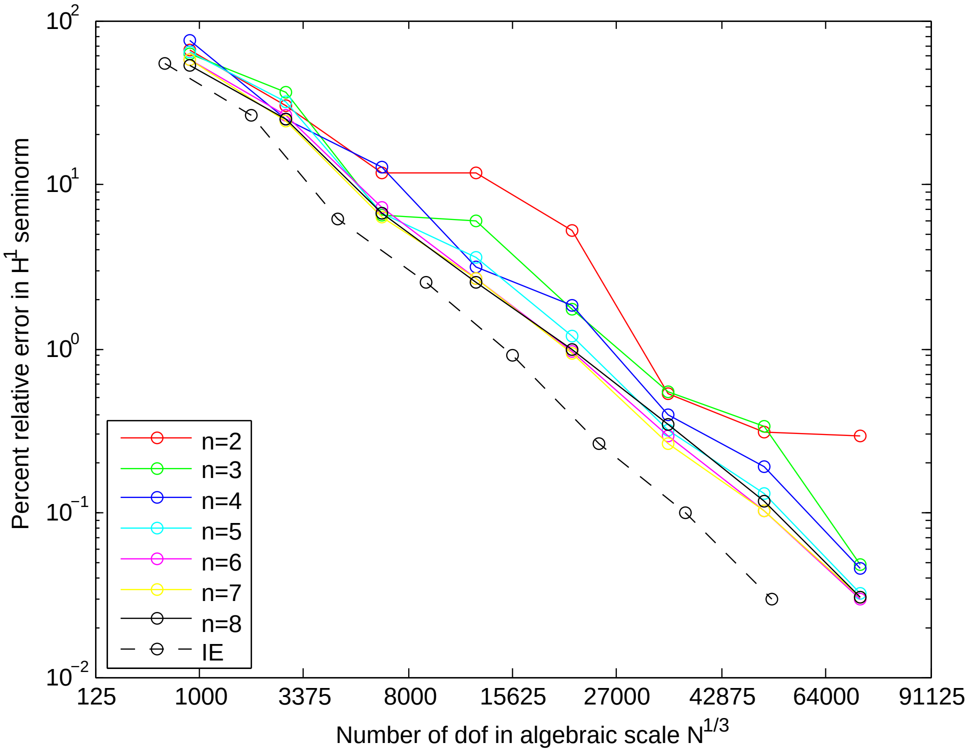


Figure 5.15: Convergence of percent relative error evaluated over the interior region for mesh (b)

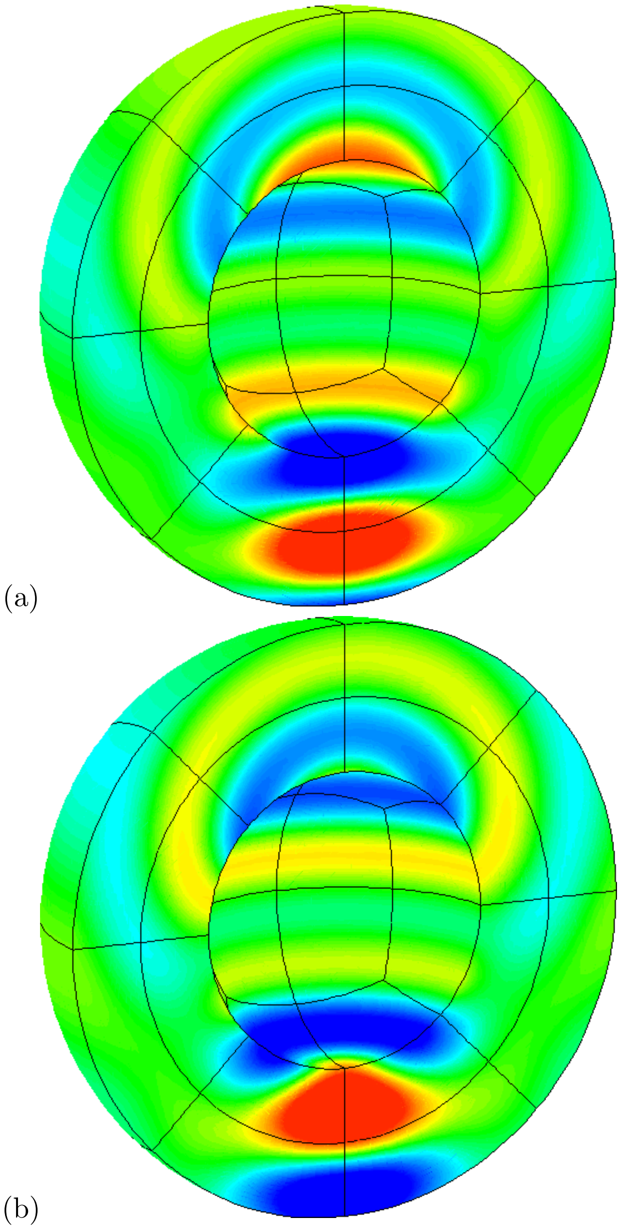


Figure 5.16: Real (a) and imaginary (b) parts of infinite element solution on mesh (b) with $p = 9$, scaled to range $[-1, 1]$

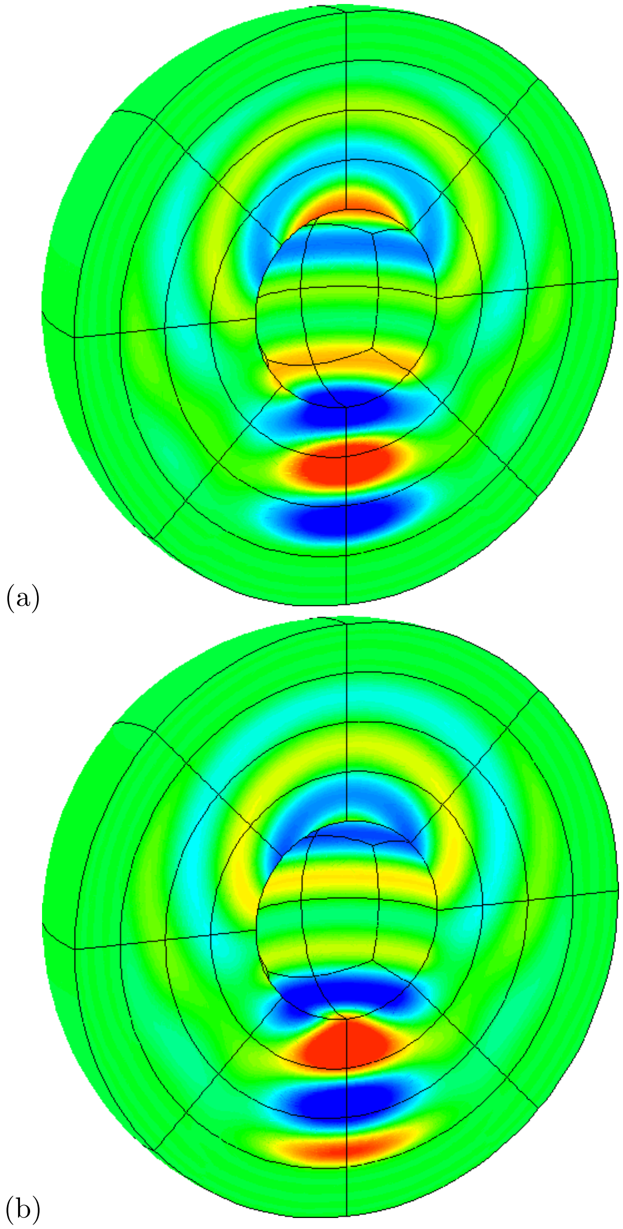


Figure 5.17: Real (a) and imaginary (b) parts of PML solution on mesh (b) with $p = 9$, scaled to range $[-1, 1]$

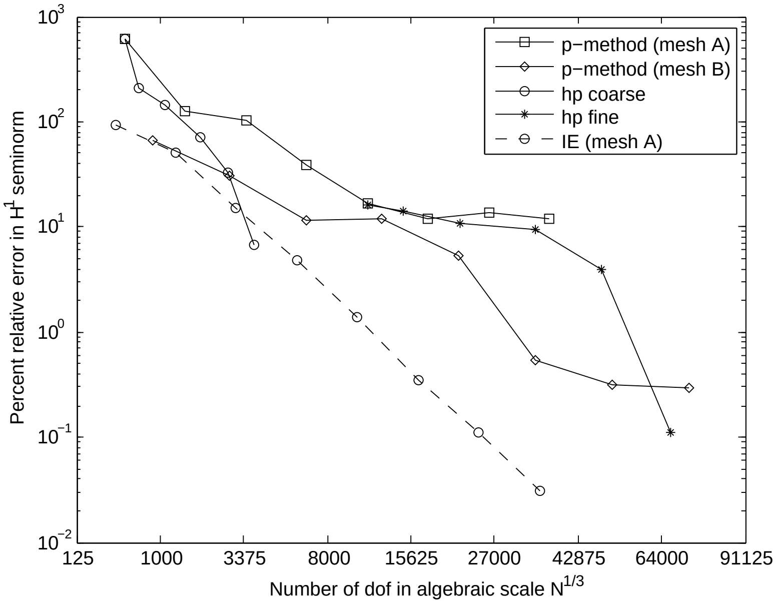


Figure 5.18: Convergence for *hp* coarse and fine grids with profile $n = 2$

as the fine grid solution used to generate them, but here the initial fine grid has as much as 30% error! Second, we generally observe exponential convergence for the fine grid as well, though the line is shifted to the right and it may “stall” in the pre-asymptotic range (as in Figure 5.18). This is not too surprising since isotropic h-refinement is far from optimal for this problem.

5.4 Acoustic scattering from a cone-sphere

We now consider acoustic scattering from the so-called cone-sphere obstacle. The geometry of the cone-sphere (shown in Figure 5.22) is parametrized by the interior angle α of the cone, and the radius c of the sphere, and fixed by requiring that the cone and sphere meet tangentially. The origin of the system of coordinates is then placed mid-way between the vertex of the cone and the south pole of the sphere. The obstacle is then surrounded with a PML of inner radius a and outer radius $b > a$.

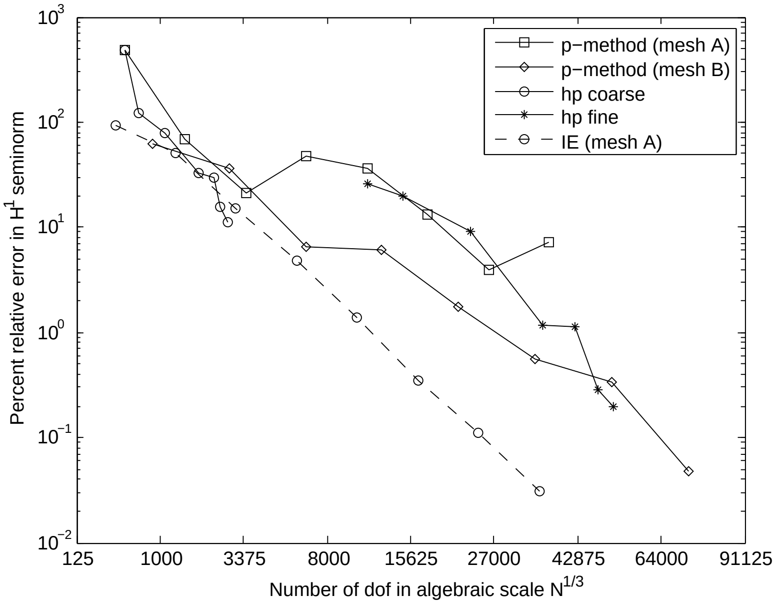


Figure 5.19: Convergence for *hp* coarse and fine grids with profile $n = 3$

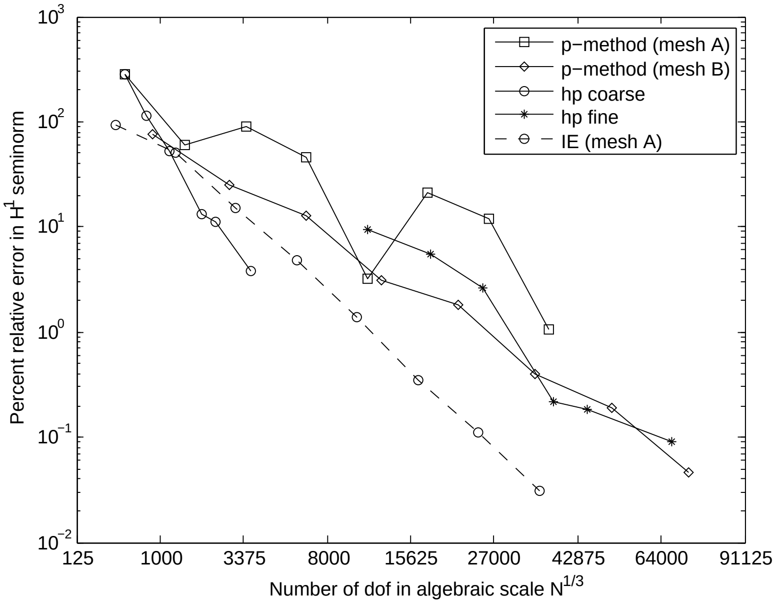


Figure 5.20: Convergence for *hp* coarse and fine grids with profile $n = 4$

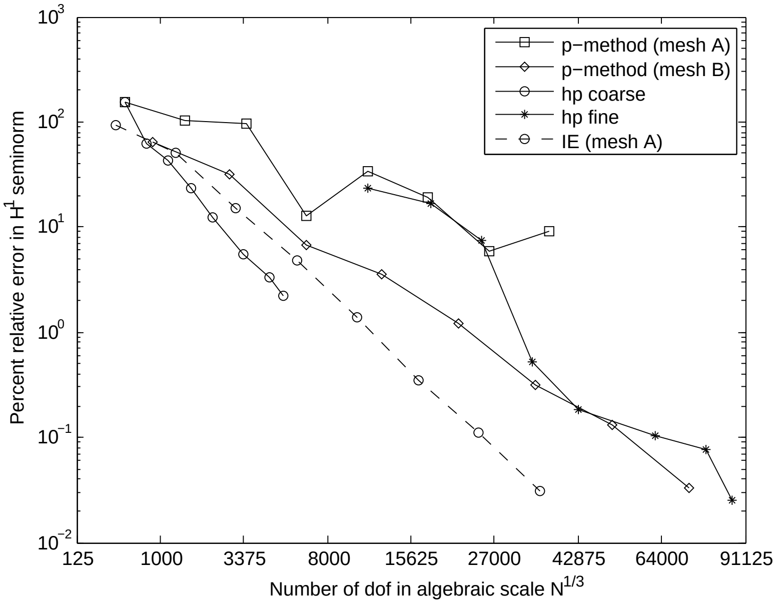


Figure 5.21: Convergence for *hp* coarse and fine grids with profile $n = 5$

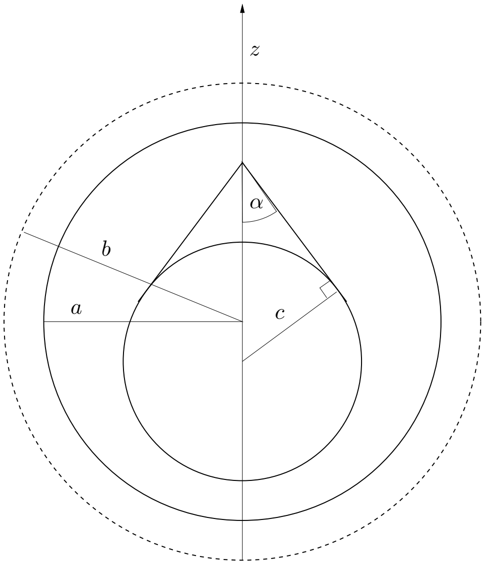


Figure 5.22: Geometry for acoustic scattering from a cone-sphere

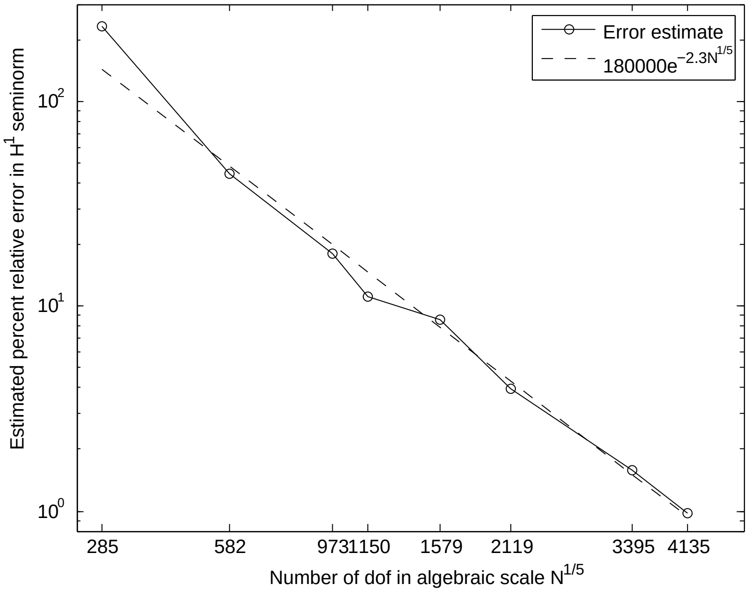


Figure 5.23: Estimated percent relative error for the sequence of *hp* coarse grids for the cone-sphere with incident wave from above

Presently we consider this problem with parameters $\alpha = \pi/4$, $c = \lambda/2$, $a = 3\lambda/2$ and $b = 2\lambda$, where $\lambda = 2\pi/k$ is the wavelength. The PML profile is determined by the complex coordinate stretching (2.17) with $n = 3$.

In Figure 5.23 we plot error estimates for the first eight coarse grids (shown in Figures 5.24 and 5.25) when the incident wave hits the cone from above. The error estimates are obtained as the H^1-seminorm of the difference between the coarse and corresponding fine grid solutions, evaluated over only the interior region $r < a$ (excluding the obstacle). The reported number of dof includes those in the PML region. The observed convergence is clearly exponential and a reference line is included to indicate the observed rate (obtained as a least-squares best fit of the final four points). Contour plots of the real and imaginary parts of the final fine-grid solution are shown in Figure 5.26, with the color scale applied to the range $[-1, 1]$.

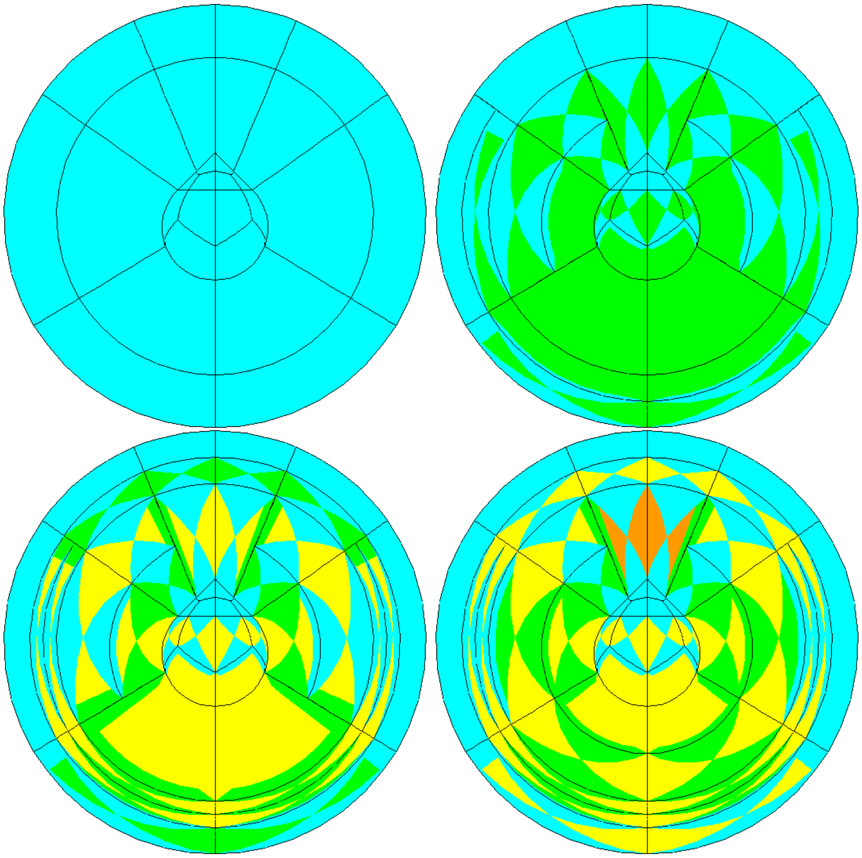


Figure 5.24: First four *hp* coarse grids for the cone-sphere with incident wave from above

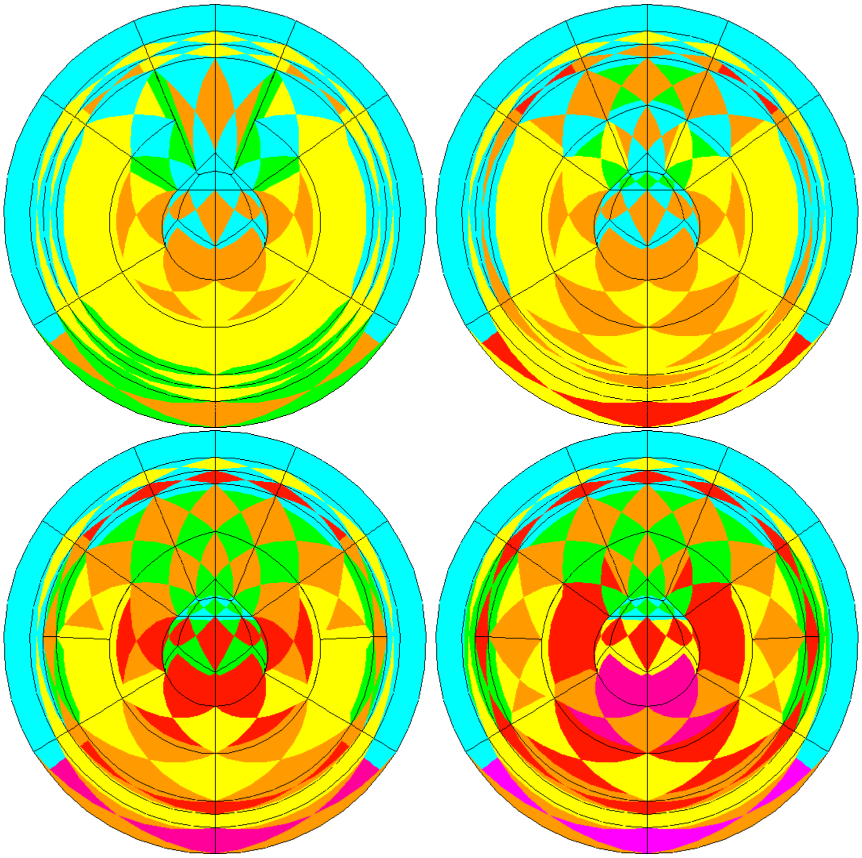


Figure 5.25: Fifth through eighth *hp* coarse grids for the cone-sphere with incident wave from above

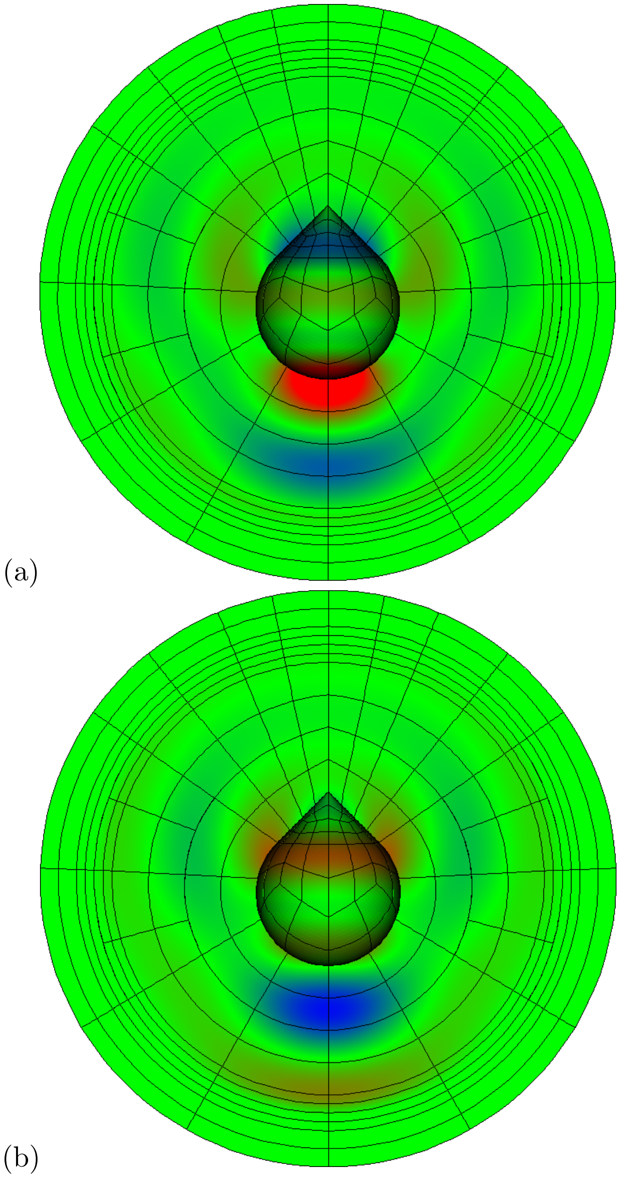


Figure 5.26: Real (a) and imaginary (b) parts of solution on the final fine grid scaled to the range $[-1, 1]$

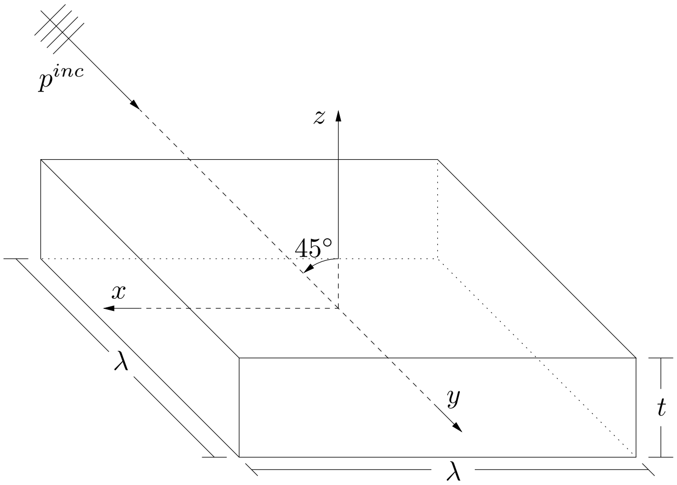


Figure 5.27: Geometry for acoustic scattering from a thin square plate

5.5 Acoustic scattering from a thin square plate

Finally we consider the rigid acoustic scattering from a hexagonal obstacle with dimensions $\lambda \times \lambda \times t$, as shown in Figure 5.27. For the present example, the thickness of the obstacle is set to $t = \lambda/5$, and we surround the obstacle with a spherical PML with inner radius $a = 3\lambda/2$ and outer radius $b = 2\lambda$. For the PML, we used the complex coordinate stretching (2.17), this time with $n = 6$.

Figure 5.28 shows the estimated percent relative error for the sequence of *hp* coarse grids evaluated over the interior region $r < a$ (excluding the obstacle). The reported number of degrees on freedom however, includes those used in the PML. It is clear that the algorithm delivers exponential convergence only asymptotically for this problem. Singularities at all twelve edges and eight vertices of the obstacle, combined with exponential decay in the PML, make this problem particularly difficult. With this in mind, the results are quite encouraging since the error decays with a *faster* rate in the pre-asymptotic range. The lower, asymptotic rate of convergence is also shown in Figure 5.28 for comparison. The first ten coarse grids are

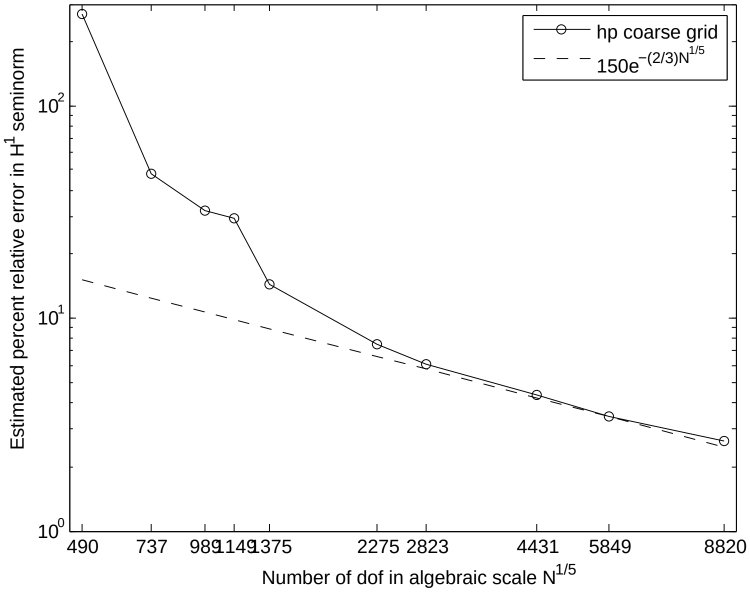


Figure 5.28: Estimated percent relative error for the sequence of *hp* coarse grids for the thin box

shown in Figures 5.29 and 5.30, and the real and imaginary parts of the solution in the final fine grid are shown in Figure 5.31.

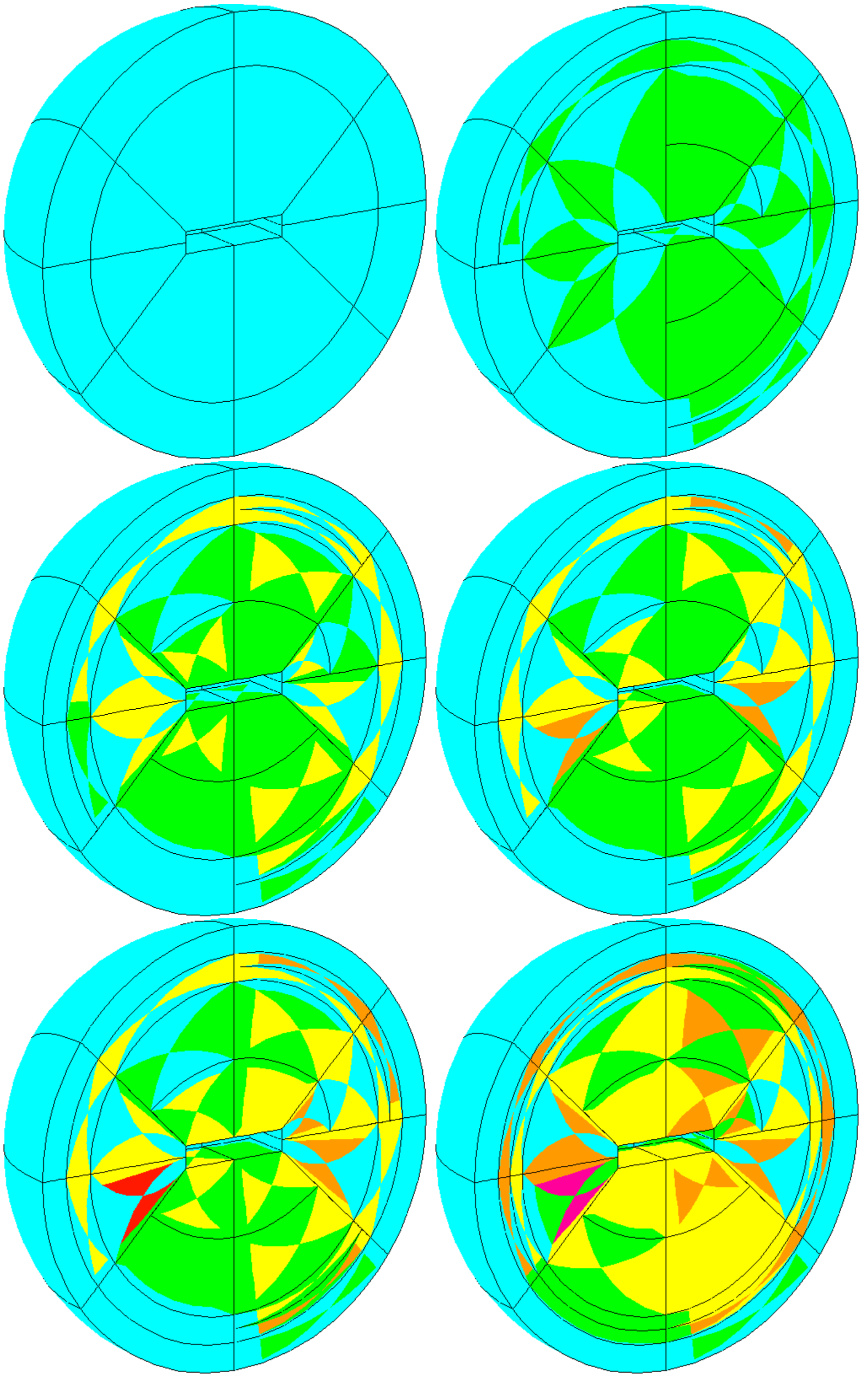


Figure 5.29: First six *hp* coarse grids for the thin box with incident wave from 45°

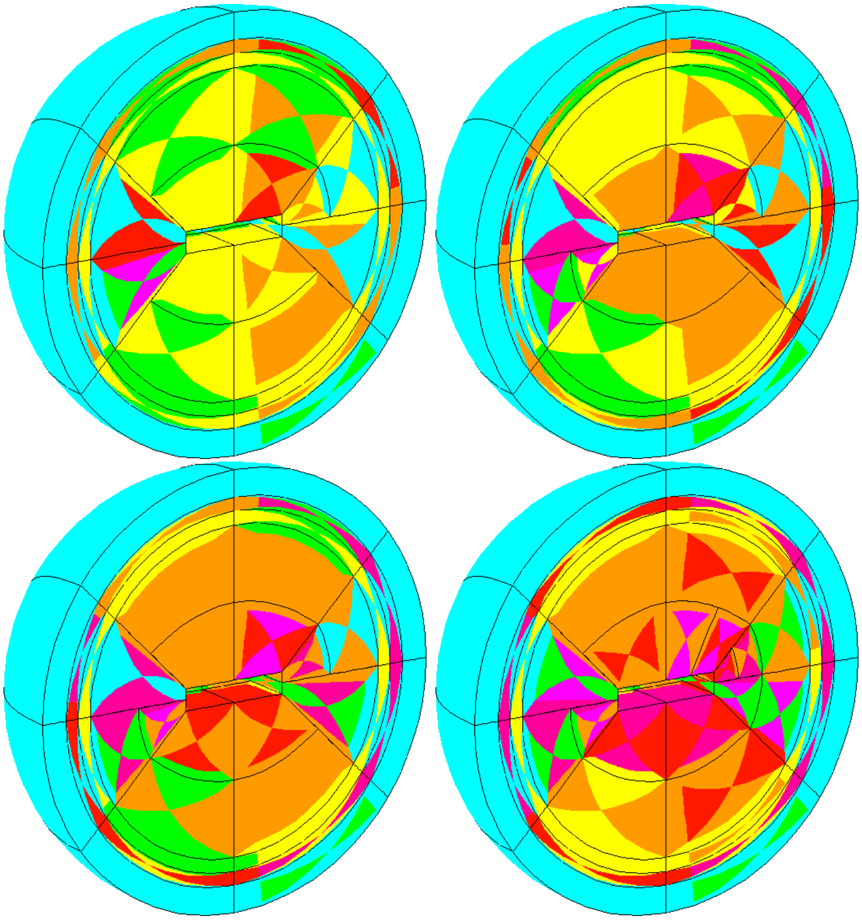


Figure 5.30: Seventh through tenth *hp* coarse grids for the thin box with incident wave from 45°

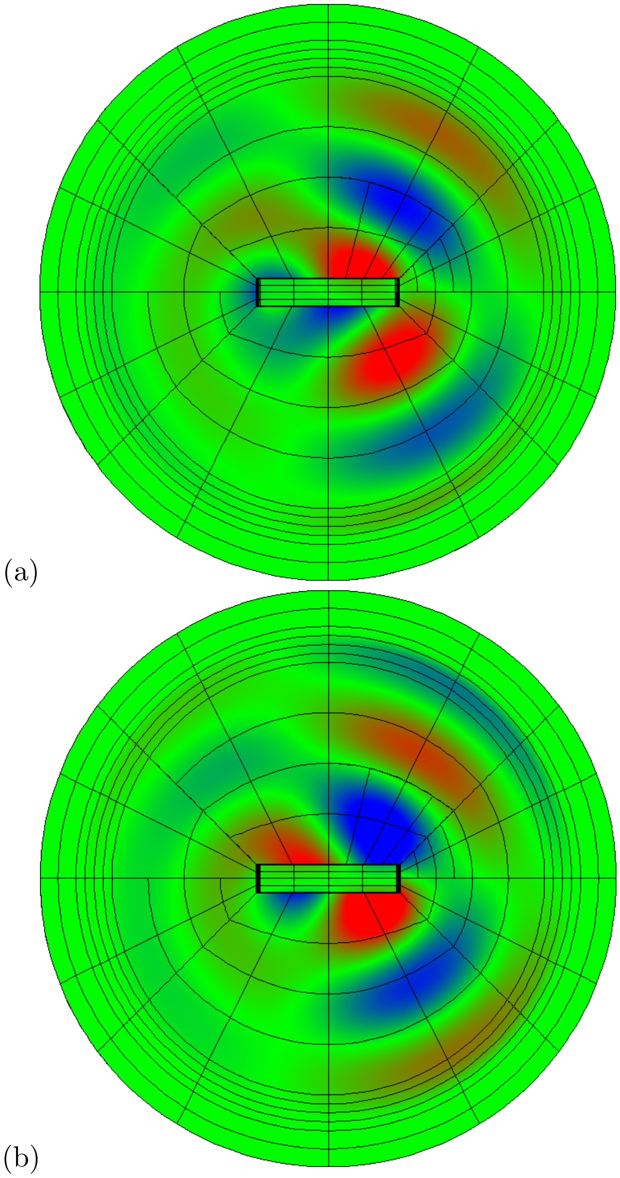


Figure 5.31: Real (a) and imaginary (b) parts of solution on the final fine grid scaled to the range $[-1, 1]$

Chapter 6

Fully-Automatic *hp*-Adaptivity for Maxwell's Equations

This chapter summarizes the key concepts behind the extension of the *hp* algorithm and its implementation to Maxwell's equations. Superficially the only difference is that the H^1 version of the projection-based interpolation operator is replaced with the slightly more complicated $\boldsymbol{H}(curl)$ version. However, there are several fundamental differences that require additional comment. We will begin by recalling the definition of the $\boldsymbol{H}(curl)$ version of the projection-based interpolation operator. Once again, the definition will be modified for computations.

6.1 Projection-based Interpolation in $\boldsymbol{H}(curl)$

We begin by recalling the definition of Nédélec's spaces of vector-valued polynomials for a quadrilateral face f and a hexahedral element K. In the absence of any restrictions on the order for faces and edges, the Nédélecspaces are just the gradient of the corresponding space of scalar-valued polynomials, i.e.

$$\begin{aligned}
\boldsymbol{Q}^p(f) &= \boldsymbol{\nabla} Q^{p_1,p_2}(f) = Q^{p_1-1,p_2}(f) \times Q^{p_1,p_2-1}(f) \\
\boldsymbol{Q}^p_{p_e}(f) &= \{\boldsymbol{F} \in \boldsymbol{Q}^p(f) : F_t \in \mathcal{P}^{p_e-1}(e)\ \forall \text{ edges } e\} \\
\boldsymbol{Q}^p(K) &= \boldsymbol{\nabla} Q^{p_1,p_2,p_3}(K) = Q^{p_1-1,p_2,p_3}(K) \times Q^{p_1,p_2-1,p_3}(K) \times Q^{p_1,p_2,p_3-1}(K) \\
\boldsymbol{Q}^p_{p_f,p_e}(K) &= \{\boldsymbol{F} \in \boldsymbol{Q}^p(K) : \boldsymbol{F}_t \in \boldsymbol{Q}^{p_f}(f)\ \forall \text{ faces } f, F_t \in \mathcal{P}^{p_e-1}(e)\ \forall \text{ edges } e\}
\end{aligned}$$

Given a function $\boldsymbol{E} \in \boldsymbol{H}^{\epsilon}(curl, K) \cap \boldsymbol{H}^{1/2+\epsilon}(K)$ the projection-based interpolant $\boldsymbol{E}^p = \Pi^{\text{curl}}\boldsymbol{E} \in \boldsymbol{Q}^p_{p_f,p_e}(K)$ is defined in four stages.

- The lowest order Whitney interpolant $\boldsymbol{E}_0 \in \boldsymbol{Q}^{1,1,1}(K)$ is defined to match the average value of the tangential component of $\boldsymbol{E}$ over each edge, i.e.

  $$\int_e \boldsymbol{E}_{0,t}\ ds = \int_e \boldsymbol{E}_t\ ds\ \forall \text{ edges } e.$$

  Observe that if $E_t = du/ds$ for some function u, then $E_{0,t}$ is the derivative of the linear vertex interpolant of u, i.e. the de Rham diagram commutes.

- The edge contributions $\boldsymbol{E}_{1,e} \in \mathcal{P}^{p_e-1}(e)$ are defined by observing that the remainder $\boldsymbol{E}_t - \boldsymbol{E}_{0,t} \in \boldsymbol{H}^{-1/2+\epsilon}(e)$, and there exists a unique potential $\psi \in H_0^{1/2+\epsilon}(e)$ such that

  $$\frac{d\psi}{ds} = \boldsymbol{E}_t - \boldsymbol{E}_{0,t}.$$

  We project this potential onto the edge interior space,

  $$\begin{cases} \psi^{p_e} \in \mathcal{P}^{p_e}_{-1}(e), \\ \|\psi - \psi^{p_e}\|_{0,e} \to \min \end{cases}$$

  and differentiate to get the individual edge contribution,

  $$E_{1,e} = \frac{d\psi^{p_e}}{ds}.$$

The edge contributions are extended into the element with a blending factor that is bilinear in the plane orthogonal to the edge, and summed to define the edge projection $\boldsymbol{E}_1$.

- The face contributions $\boldsymbol{E}_{2,f}$ are defined by solving the constrained minimization problem,

$$\begin{cases} \boldsymbol{E}_{2,f} \in \boldsymbol{Q}^{p_f}_{-1}(f), \\ \|\mathrm{curl}_f((\boldsymbol{E} - \boldsymbol{E}_0 - \boldsymbol{E}_1)_t - \boldsymbol{E}_{2,f})\|_{-1/2,f} \to \min, \\ ((\boldsymbol{E} - \boldsymbol{E}_0 - \boldsymbol{E}_1)_t - \boldsymbol{E}_{2,f}, \boldsymbol{\nabla}_f \phi)_{-1/2,f} = 0 \quad \forall \phi \in Q^{p_f}_{-1}(f) \end{cases}$$

  The face contributions are extended into the element with a blending factor that is linear in the direction orthogonal to the face, and summed to define the face projection $\boldsymbol{E}_2$.

- Finally, the interior contribution is defined by solving the constrained minimization problem,

$$\begin{cases} \boldsymbol{E}_3 \in \boldsymbol{Q}^{p}_{-1,-1}(K), \\ \|\boldsymbol{\nabla} \times ((\boldsymbol{E} - \boldsymbol{E}_0 - \boldsymbol{E}_1 - \boldsymbol{E}_2) - \boldsymbol{E}_3)\|_{0,K} \to \min, \\ ((\boldsymbol{E} - \boldsymbol{E}_0 - \boldsymbol{E}_1 - \boldsymbol{E}_2) - \boldsymbol{E}_3, \boldsymbol{\nabla} \phi)_{0,K} = 0 \quad \forall \phi \in Q^{p}_{-1,-1}(K) \end{cases}$$

The definition of the face contribution is equivalent to the mixed variational formulation,

$$\begin{cases} \boldsymbol{E}_{2,f} \in \boldsymbol{Q}^{p}_{-1}(f), p \in Q^{p}_{-1}(f), \\ a_f(\boldsymbol{E}_{2,f}, \boldsymbol{F}) + c_f(p, \boldsymbol{F}) & = a_f((\boldsymbol{E} - \boldsymbol{E}_0 - \boldsymbol{E}_1)_t, \boldsymbol{F}) \quad \forall \boldsymbol{F} \in \boldsymbol{Q}^{p}_{-1}(f) \\ c_f(q, \boldsymbol{E}_{2,f}) & = c_f(q, (\boldsymbol{E} - \boldsymbol{E}_0 - \boldsymbol{E}_1)_t) \quad \forall q \in Q^{p}_{-1}(f) \end{cases} \tag{6.1}$$

where

$$\begin{aligned} a_f(\boldsymbol{E},\boldsymbol{F}) &= (\mathrm{curl}_f \boldsymbol{E}, \mathrm{curl}_f \boldsymbol{F})_{-1/2,f} \\ c_f(p,\boldsymbol{F}) &= (\boldsymbol{\nabla}_f p, \boldsymbol{F})_{-1/2,f} \end{aligned}$$

Likewise, the interior contribution satisfies the mixed problem,

$$\begin{cases} \boldsymbol{E}_3 \in \boldsymbol{Q}^p_{-1,-1}(K), p \in Q^p_{-1,-1}(K), & & \\ a_K(\boldsymbol{E}_3,\boldsymbol{F}) + c_K(p,\boldsymbol{F}) & = a_K(\boldsymbol{E}-\boldsymbol{E}_0-\boldsymbol{E}_1-\boldsymbol{E}_2,\boldsymbol{F}) & \forall \boldsymbol{F} \in \boldsymbol{Q}^p_{-1,-1}(K) \\ c_K(q,\boldsymbol{E}_3) & = c_K(q,\boldsymbol{E}-\boldsymbol{E}_0-\boldsymbol{E}_1-\boldsymbol{E}_2) & \forall q \in Q^p_{-1,-1}(K) \end{cases} \tag{6.2}$$

where

$$\begin{aligned} a_K(\boldsymbol{E},\boldsymbol{F}) &= (\boldsymbol{\nabla}\times\boldsymbol{E}, \boldsymbol{\nabla}\times\boldsymbol{F})_{0,K} \\ c_K(p,\boldsymbol{F}) &= (\boldsymbol{\nabla} p, \boldsymbol{F})_{0,K} \end{aligned}$$

6.2 Computational issues

For computations, these projection operators are once again modified by replacing curvilinear geometry with rectilinear geometry and replacing the fractional seminorm for faces by a stronger seminorm modified by the appropriate weight. The lowest order interpolant is treated as a pre-processing step and subtracted from the fine grid solution in the interface data structure.

At least for edges, the code developed for computing H^1 projections (which reduces to L^2 for edges) can be directly re-used for the $\boldsymbol{H}(\mathrm{curl})$ case. This is achieved by working directly with the potential ψ. Let E be the tangential component of the fine grid solution (with the lowest order interpolant removed) restricted to a coarse grid edge e of order p. Then E is a discontinuous, piecewise polynomial of order p on each son, with zero average over the whole edge, and ψ is a continuous, piecewise polynomial of order $p+1$ on each son, with zero values at the endpoints. Hence, E can be represented in terms of Legendre polynomials,

$$E = \begin{cases} E^1 = \sum_{i=1}^{p+1} E_i^1 P_{i-1} & \text{on } e_1 \\ E^2 = \sum_{i=1}^{p+1} E_i^2 P_{i-1} & \text{on } e_2 \end{cases}$$

where $E_1^1 = -E_1^2$ since the average value of E is zero. Likewise, ψ can be represented in terms of integrated Legendre polynomials,

$$\psi = \begin{cases} \psi^1 = \sum_{i=1}^{p+2} \psi_i^1 \phi_i & \text{on } e_1 \\ \psi^2 = \sum_{i=1}^{p+2} \psi_i^2 \phi_i & \text{on } e_2 \end{cases}$$

where $\psi_1^1 = \psi_2^2 = 0$ since ψ vanishes at the endpoints. Then since $\phi_1' = -P_0$ and $\phi_i' = P_{i-2}$ for $i \geq 2$, the condition $\psi' = E$ yields the coefficients of ψ as,

$$\begin{aligned} \psi_1^1 = 0, \quad \psi_2^1 = E_1^1, \quad \psi_i^1 = E_{i-1}^1 \ (i \geq 3), \\ \psi_1^2 = -E_1^2, \quad \psi_2^2 = 0, \quad \psi_i^2 = E_{i-1}^2 \ (i \geq 3). \end{aligned} \tag{6.3}$$

This representation for ψ can be fed directly to the edge optimization code for the H^1 case. Once the optimal L^2 projection ψ^p is returned, the relations (6.3) are used once again to get the coefficients for E^p. The bilinear extension of each edge contribution is then subtracted from the fine grid solution in the interface data structure.

Unfortunately, there are no similar tricks for face and element interiors. The details will be omitted in favor of some general observations. First, the same logical structure described in Section 4.3 for the H^1 case, extends to the $\boldsymbol{H}(\text{curl})$ case as well. That is, the stiffness matrix and load vector for problems (6.1) and (6.2), with the trial and test spaces of piecewise polynomials, are constructed by an assembly procedure, followed by the application of extension matrices. We first assemble the system corresponding to the projection of the fine grid solution onto the fine-grid space, but with zero tangential component on the boundary of the *coarse* grid face or element. Once again, we replace the curvilinear geometry with a rectilinear geometry using the average values of h_1, h_2 (for faces, and h_3 for elements), so that the stiffness matrix and right hand side can be assembled from precomputed matrices. As before, systems for other h-refinement types are generated using extension operators, and systems of lower order are generated by extraction (based on the use of a hierarchical basis).

6.3 Numerical Results II

The implementation of the $\boldsymbol{H}(\text{curl})$ version of projection-based interpolation is the most recent contribution to this work, and to date it has only been applied to one model electromagnetic scattering problem. Here we present results for the waveguide problem introduced in section §2.3.1. This problem was selected because of the simplicity of the geometry and method of truncation, and the very nontrivial nature of the solution. We have selected parameters $a = b = l = 1$ so that the truncated waveguide occupies the region $[0,1] \times [0,1] \times [-1,1]$ and has a square cross-section. For this geometry, the smallest cutoff frequency is the pair $\omega^c_{10} = \omega^c_{01} = \pi$ and the second smallest is $\omega^c_{11} = \sqrt{2}\pi$. We have selected the midpoint $\omega = (1 + \sqrt{2})\pi/2$ so that both modes in (2.40) are propagated and all higher modes decay exponentially. The incident wave E^{inc} is set to E^+_{10}, which travels in the $+z$ direction and has nonzero y component. The obstacle is a square plate in the center of the waveguide, $[1/3, 2/3] \times [1/3, 2/3] \times [-t/2, t/2]$, and we will consider two values for the thickness, $t = 1/3$ and $1/10$. We expect better accuracy for the thicker obstacle since in the limit $t \to 0$ we arrive at the severe case of diffraction from a screen. Still for finite thickness we expect strong singularities at each of the 12 edges and 8 vertices of the obstacle.

In Figures 6.1 and 6.2 we plot the estimated convergence curves for the

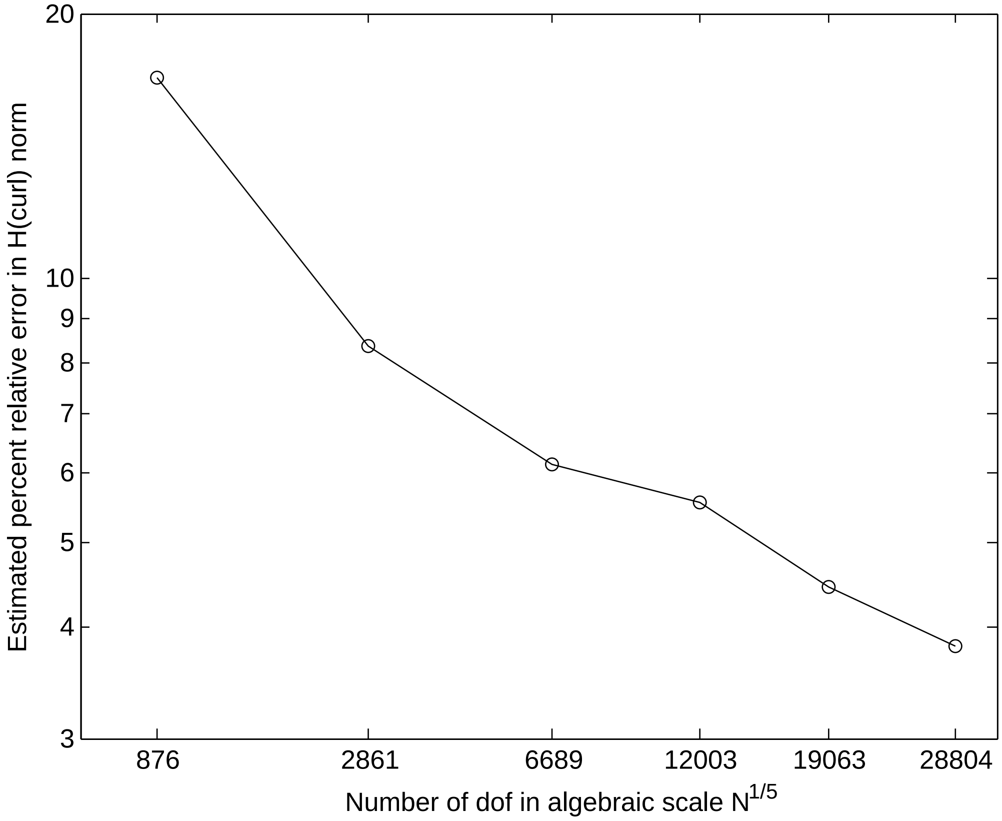


Figure 6.1: Estimated convergence of the *hp* coarse grid for the waveguide problem with thickness $t = 1/3$

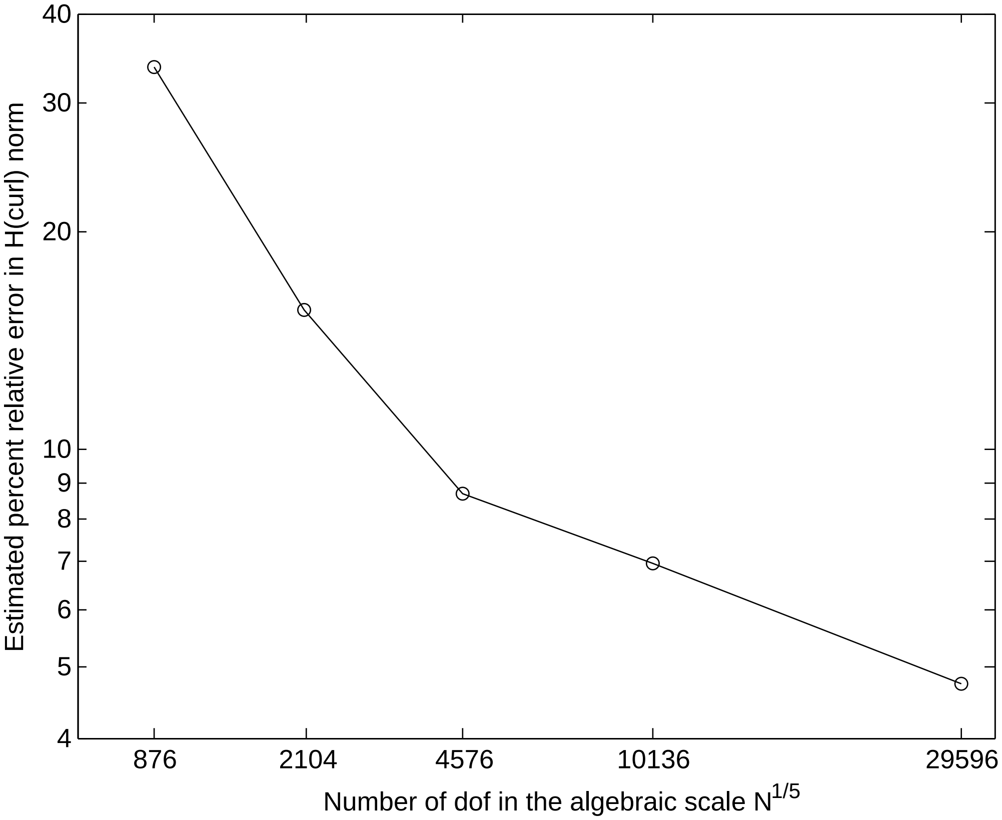


Figure 6.2: Estimated convergence of the *hp* coarse grid for the waveguide problem with thickness $t = 1/10$

hp coarse grid when the obstacle has thickness $t = 1/3$ and $1/10$, respectively. Indeed, the final coarse grids are roughly the same size and the one for the thick obstacle delivers roughly 1% lower error than for the thin obstacle. As was the case for acoustic scattering from a square plate, the exponential convergence is only observed asymptotically. In the first steps, the algorithm is essentially *h*-adaptive, as it attempts to resolve the singularities near the obstacle. We are once again encouraged by the fact that the pre-asymptotic convergence is faster than the rate observed asymptotically.

If we exclude the initial mesh for the thick obstacle, and the first two for the thin obstacle, then we do observe a roughly linear trend to Figures 6.1 and 6.2, and we offer this as preliminary evidence of exponential convergence. Unfortunately, we are presently unable to push our algorithm further as the final fine grid for either problem has stretched the limits of our direct solver and computer resources (memory in particular). The problems were solved on a 64 bit workstation with 16 GB physical memory, and Figure 6.3 shows the memory and time used for the sequence of fine grids. For the final fine grids we have

t	Nrdof	Memory	Time
1/3	685214	23.3 GB	74 min
1/10	653960	17.6 GB	43 min

Investigating this problem further will require either the development of a parallel direct solver ([45], extended for electromagnetic problems) or the development of a suitable iterative solver ([41] and [44], extended for 3D problems).

Based on the available data, we estimate that the coarse grid solution converges with the approximate rate,

$$E_c \approx 30e^{-0.265N^{1/5}},$$

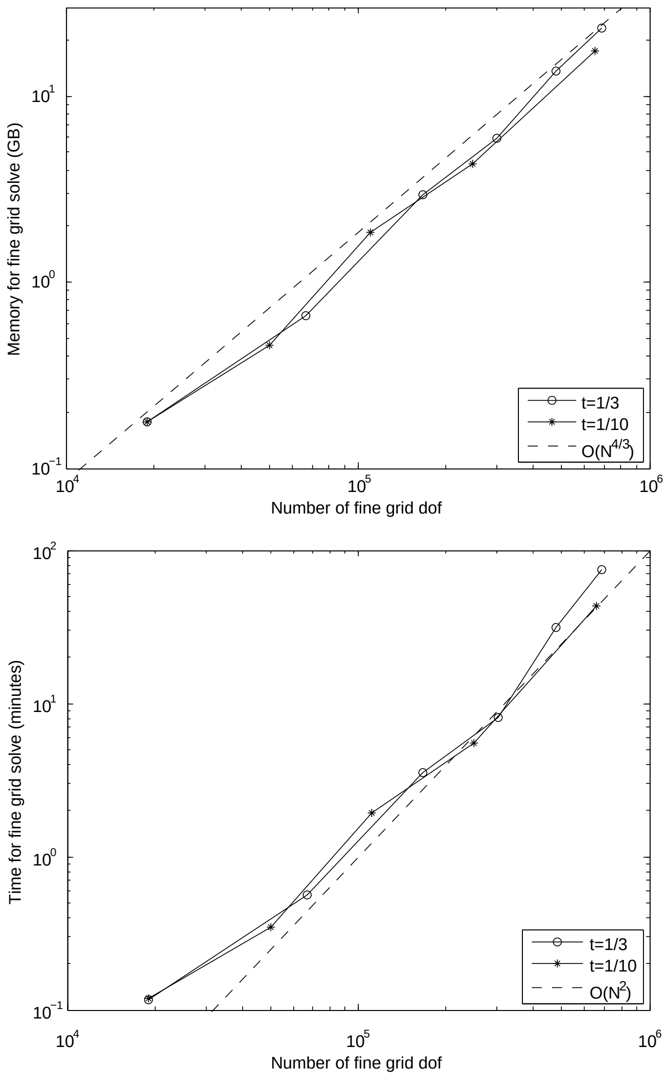


Figure 6.3: Memory and time used by the direct solver for the sequence of fine grids for the waveguide problem.

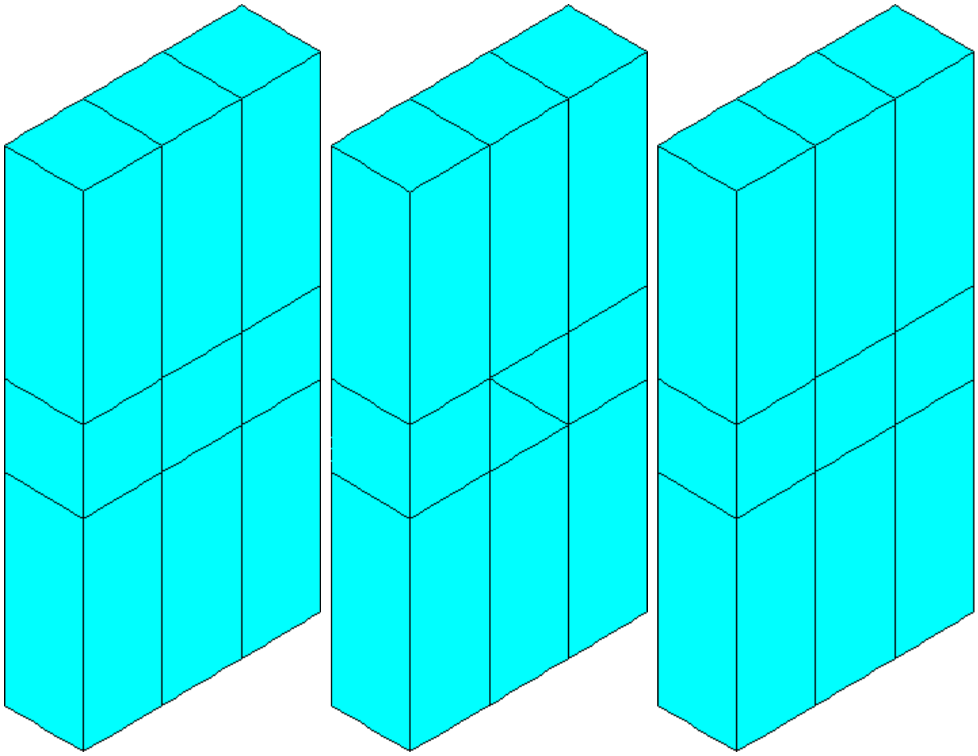
for $t = 1/3$, and

$$E_c \approx 33e^{-0.248N^{1/5}},$$

for $t = 1/10$. Then assuming that the fine grid converges with the same *rate*, and that the error in the second-to-last fine grid is comparable to the last coarse grid (in fact it should be *much* lower) we can extrapolate the conservative upper-bound of 2.5% for the error in the final fine grids.

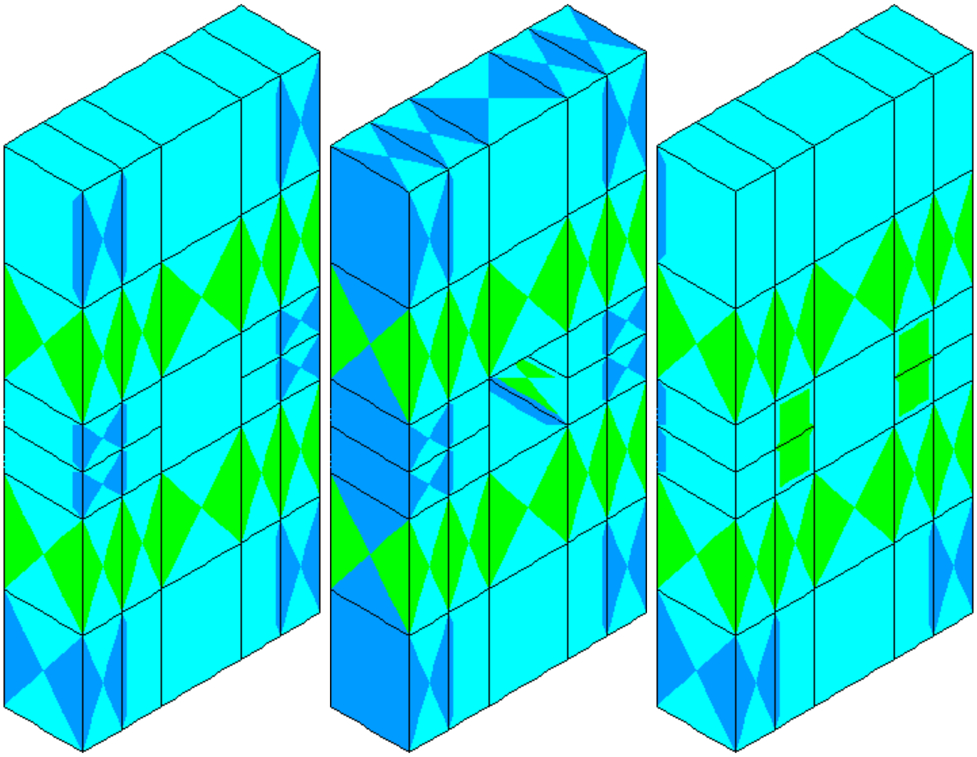
Figures 6.4–6.9 display the sequence of hp coarse grids for $t = 1/3$ and Figures 6.10–6.14 display the same for $t = 1/10$. We have divided each mesh into three slabs along the y-axis to expose the refinements in the interior of the waveguide and particularly near the obstacle. We observe the expected trend of $h2$-refinements toward the faces of the obstacle, $h4$-refinements toward the edges, and isotropic $h8$-refinements toward the corners. Anisotropic orders of approximation are generally selected with higher orders along the length of the waveguide.

Figures 6.15–6.17 display the real and imaginary parts of each component of the final fine grid solution for $t = 1/3$, and Figures 6.18–6.20 display the same for $t = 1/10$. Though the maximum amplitude is larger, the colors only vary in the range -1 to 1 in order to see the field more clearly away from the obstacle. The singularities in the field clearly mark the edges of the obstacle. The x and z components of $\boldsymbol{E}$ decay away from the obstacle, and only the y component is propagated as a wave.



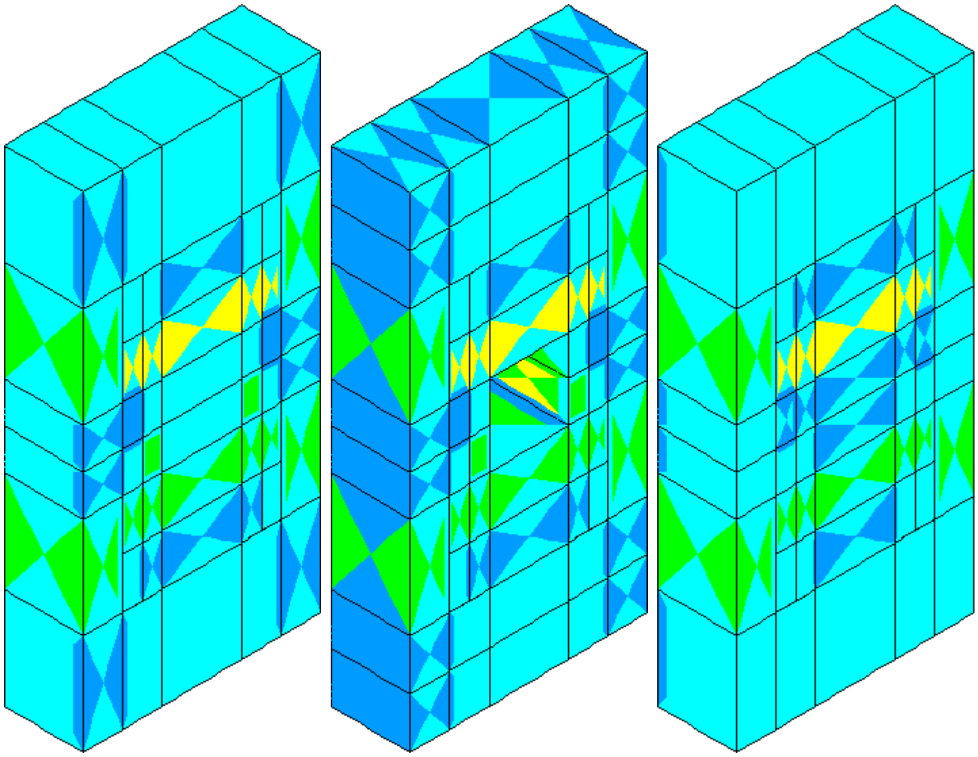
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.4: Slabs along the y-axis of the initial coarse grid for the waveguide problem ($t = 1/3$) having 876 dof, 16.9% error (the corresponding fine grid has 19044 dof).



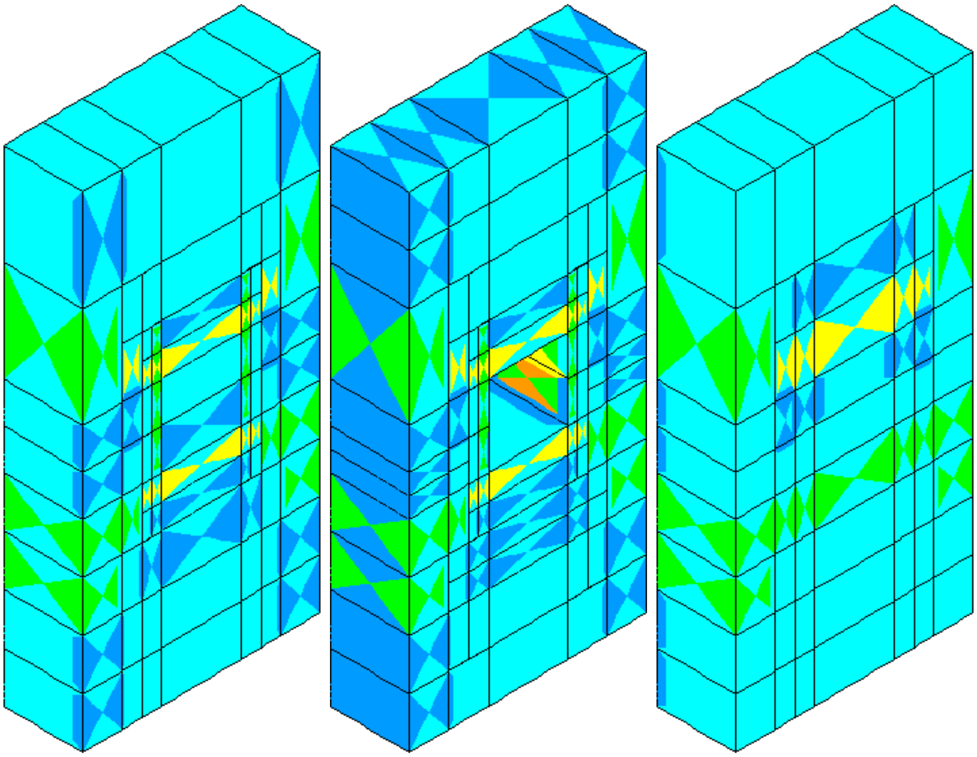
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.5: Slabs along the y-axis of the first hp coarse grid for the waveguide problem ($t = 1/3$) having 2861 dof, 8.4% error (the corresponding fine grid has 66738 dof).



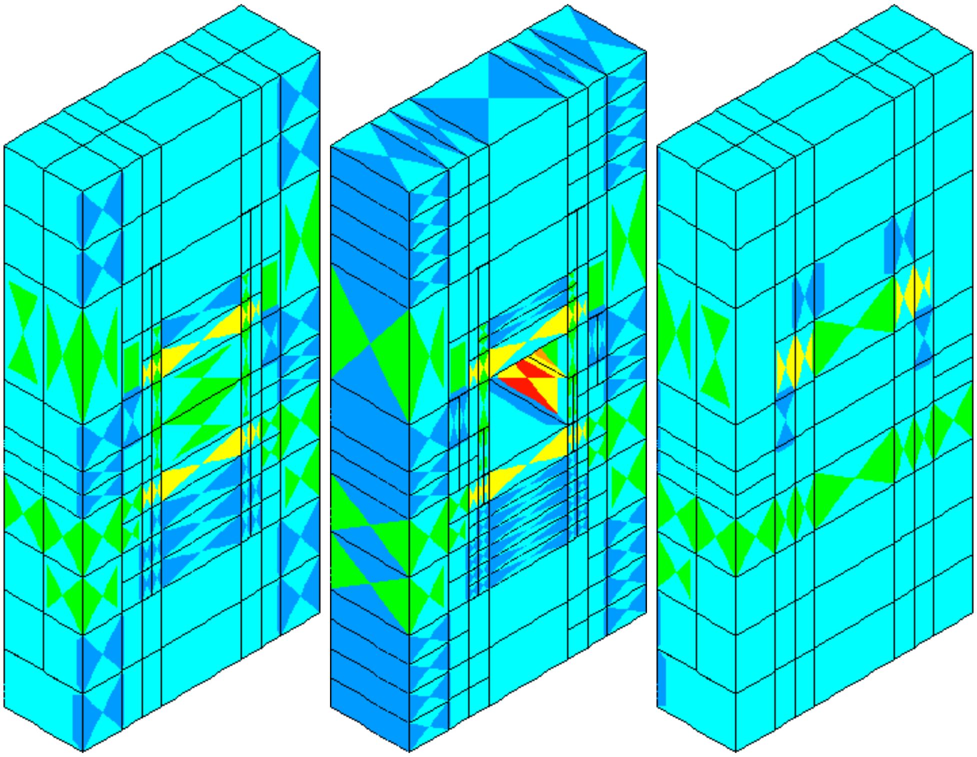
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.6: Slabs along the y-axis of the second hp coarse grid for the waveguide problem ($t = 1/3$) having 6689 dof, 6.1% error (the corresponding fine grid has 166650 dof).



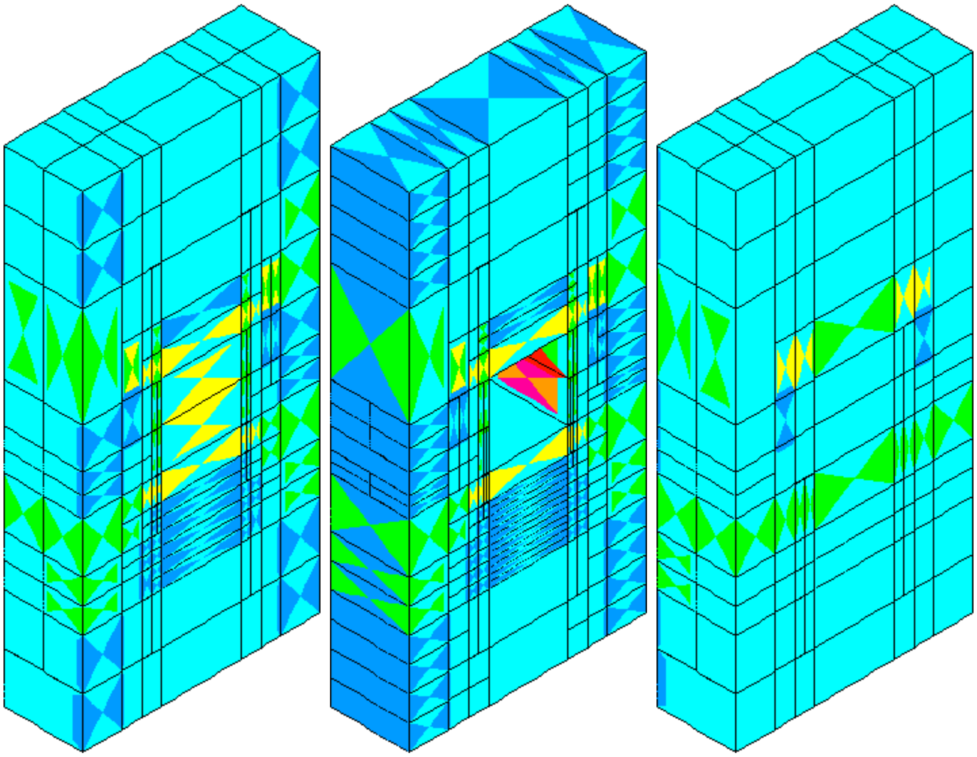
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.7: Slabs along the y-axis of the third hp coarse grid for the waveguide problem ($t = 1/3$) having 12003 dof, 5.6% error (the corresponding fine grid has 300932 dof).



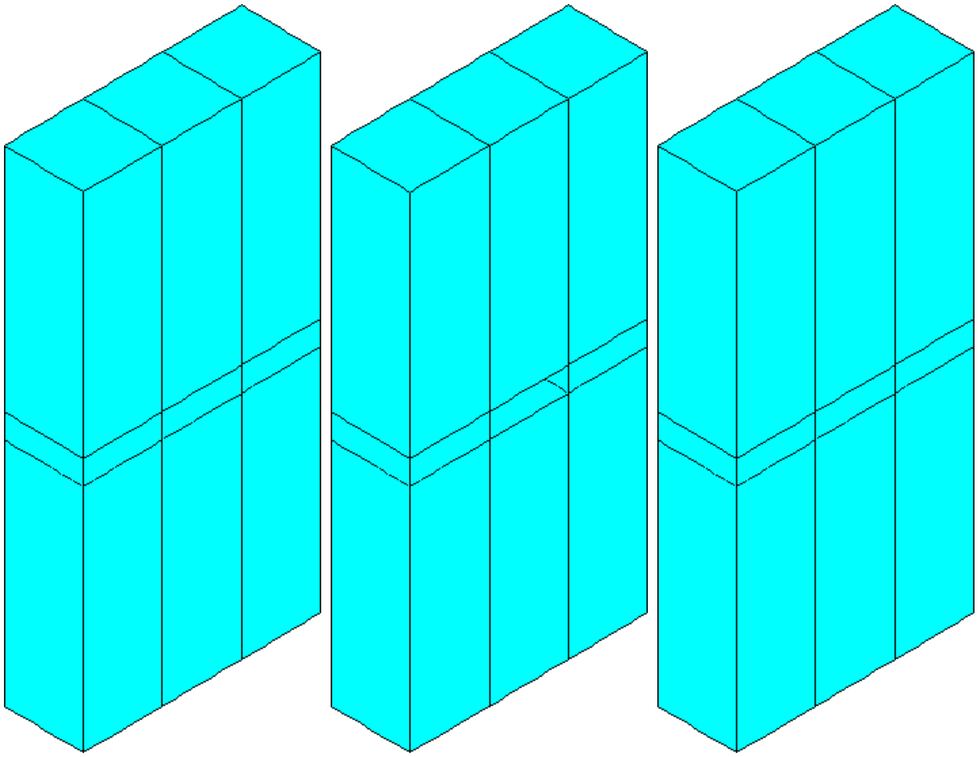
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.8: Slabs along the y-axis of the fourth hp coarse grid for the waveguide problem ($t = 1/3$) having 19063 dof, 4.5% error (the corresponding fine grid has 476822 dof).



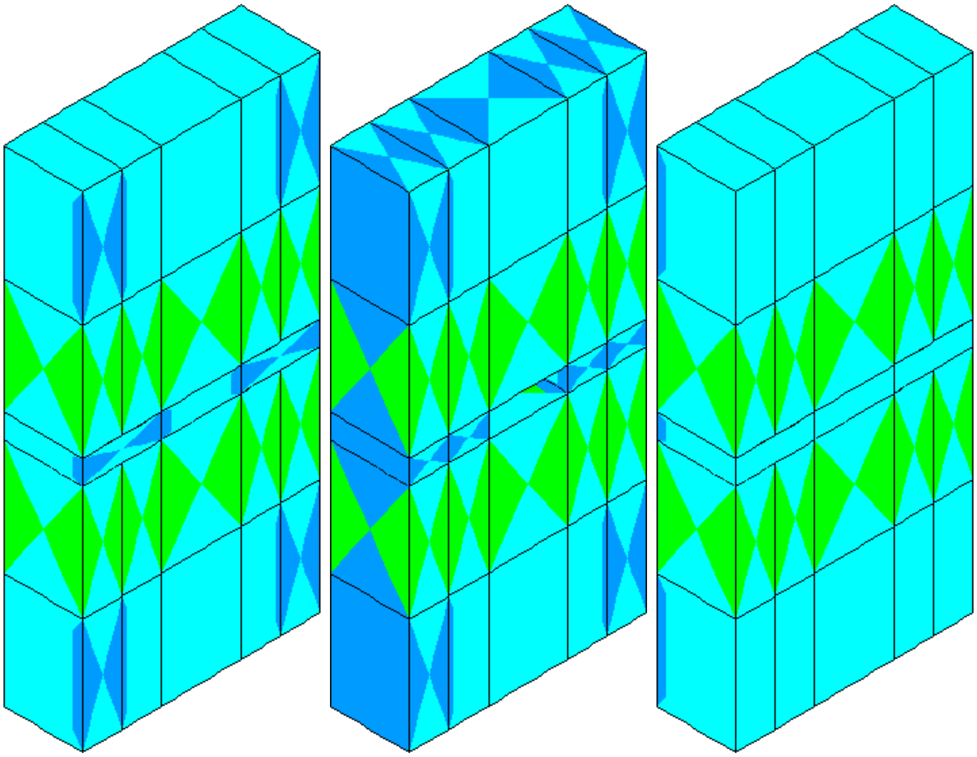
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.9: Slabs along the y-axis of the fifth hp coarse grid for the waveguide problem ($t = 1/3$) having 28804 dof, 3.8% error (the corresponding fine grid has 685214 dof).



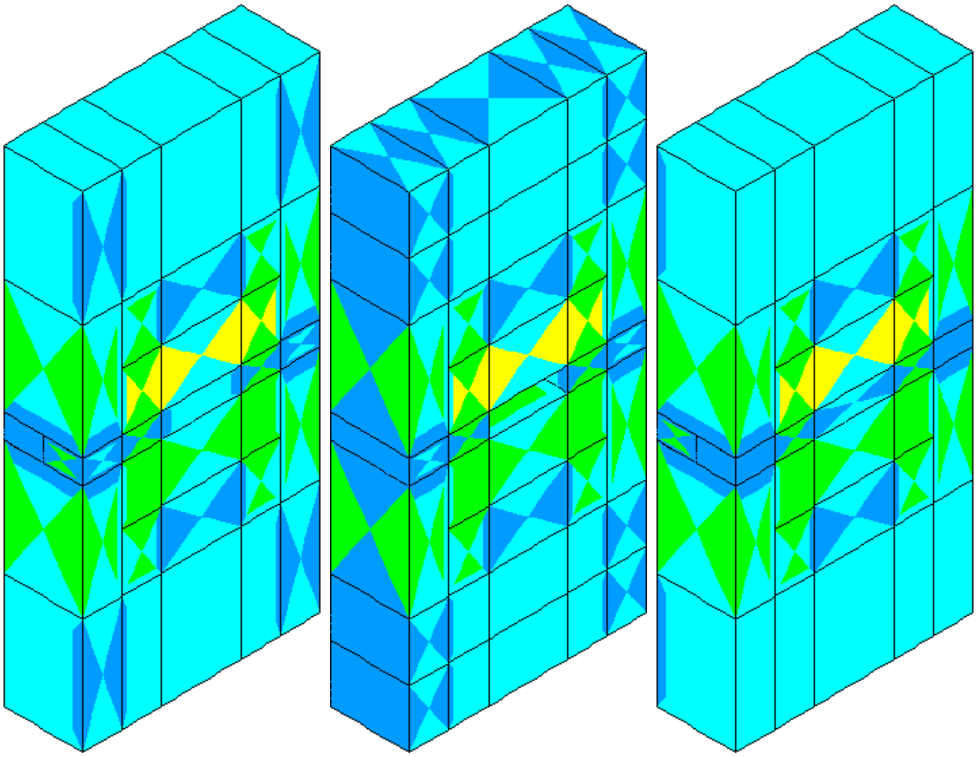
$0 < y < 1/3$ $\quad$ $1/3 < y < 2/3$ $\quad$ $2/3 < y < 1$

Figure 6.10: Slabs along the y-axis of the initial coarse grid for the waveguide problem ($t = 1/10$) having 876 dof, 33.7% error (the corresponding fine grid has 19044 dof).



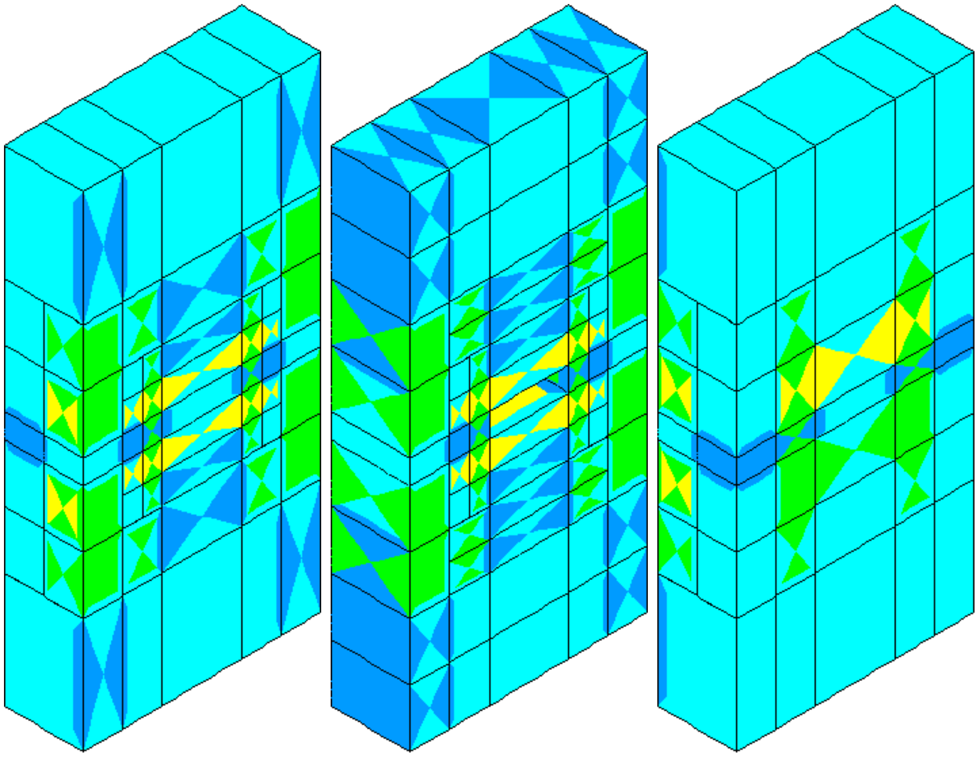
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.11: Slabs along the y-axis of the first hp coarse grid for the waveguide problem ($t = 1/10$) having 2104 dof, 15.6% error (the corresponding fine grid has 49948 dof).



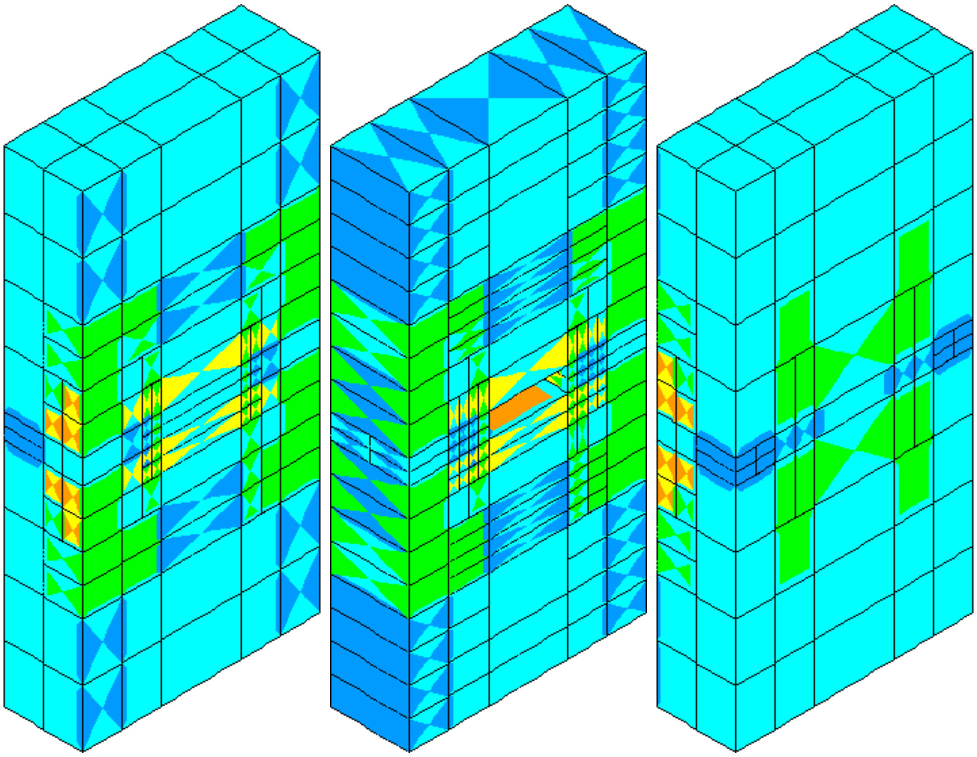
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.12: Slabs along the y-axis of the second *hp* coarse grid for the waveguide problem ($t = 1/10$) having 4576 dof, 8.7% error (the corresponding fine grid has 110800 dof).



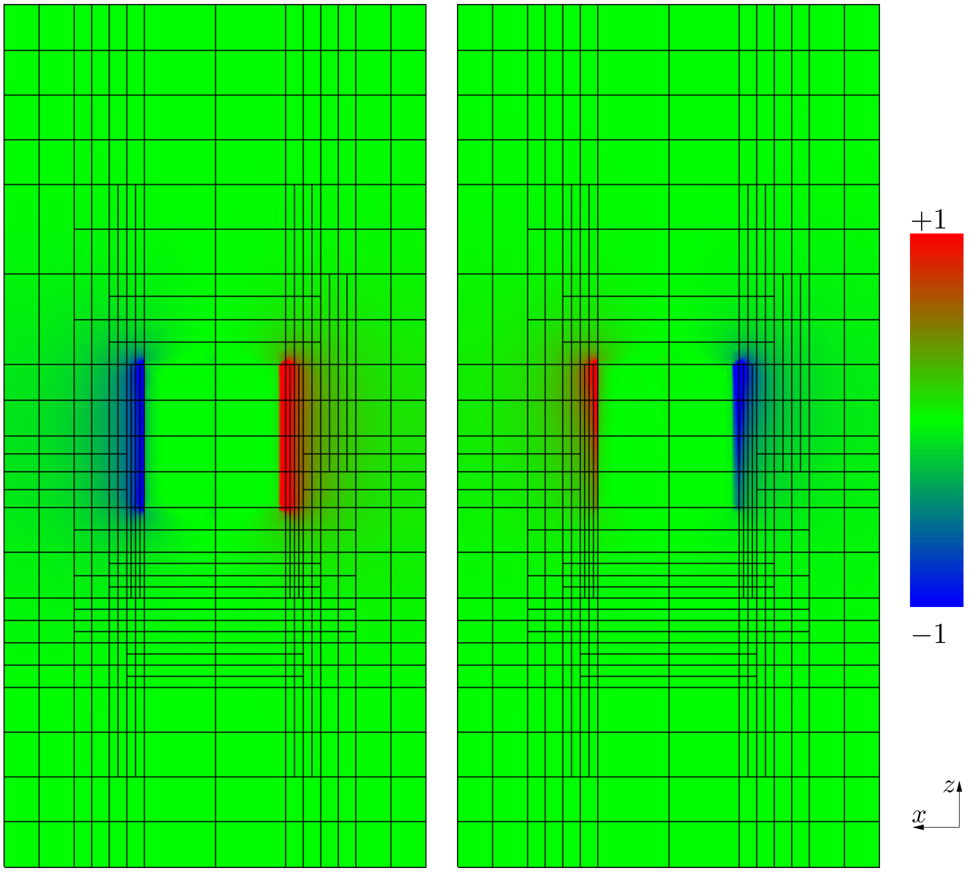
$0 < y < 1/3$ $1/3 < y < 2/3$ $2/3 < y < 1$

Figure 6.13: Slabs along the y-axis of the third hp coarse grid for the waveguide problem ($t = 1/10$) having 10136 dof, 6.9% error (the corresponding fine grid has 248960 dof).



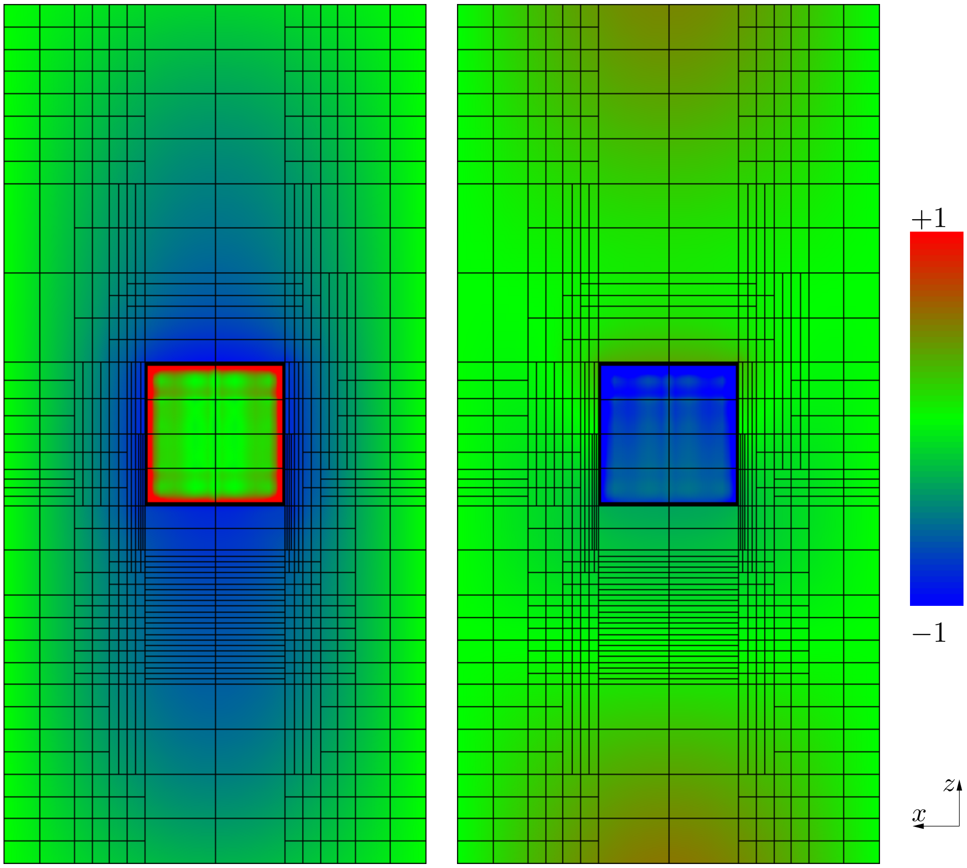
$0 < y < 1/3$ $\quad$ $1/3 < y < 2/3$ $\quad$ $2/3 < y < 1$

Figure 6.14: Slabs along the y-axis of the fourth hp coarse grid for the waveguide problem ($t = 1/10$) having 29596 dof, 4.7% error (the corresponding fine grid has 653960 dof).



$\Re(E_x)$ $\Im(E_x)$

Figure 6.15: Real and imaginary parts of the x-component of the scattered electric field for $t = 1/3$ in the plane $y = 1/3$ ($\|E_x\|_\infty \approx 7.8$).



$\Re(E_y)$ $\Im(E_y)$

Figure 6.16: Real and imaginary parts of the y-component of the scattered electric field for $t = 1/3$ in the plane $y = 1/2$ ($\|E_y\|_\infty \approx 6.3$).

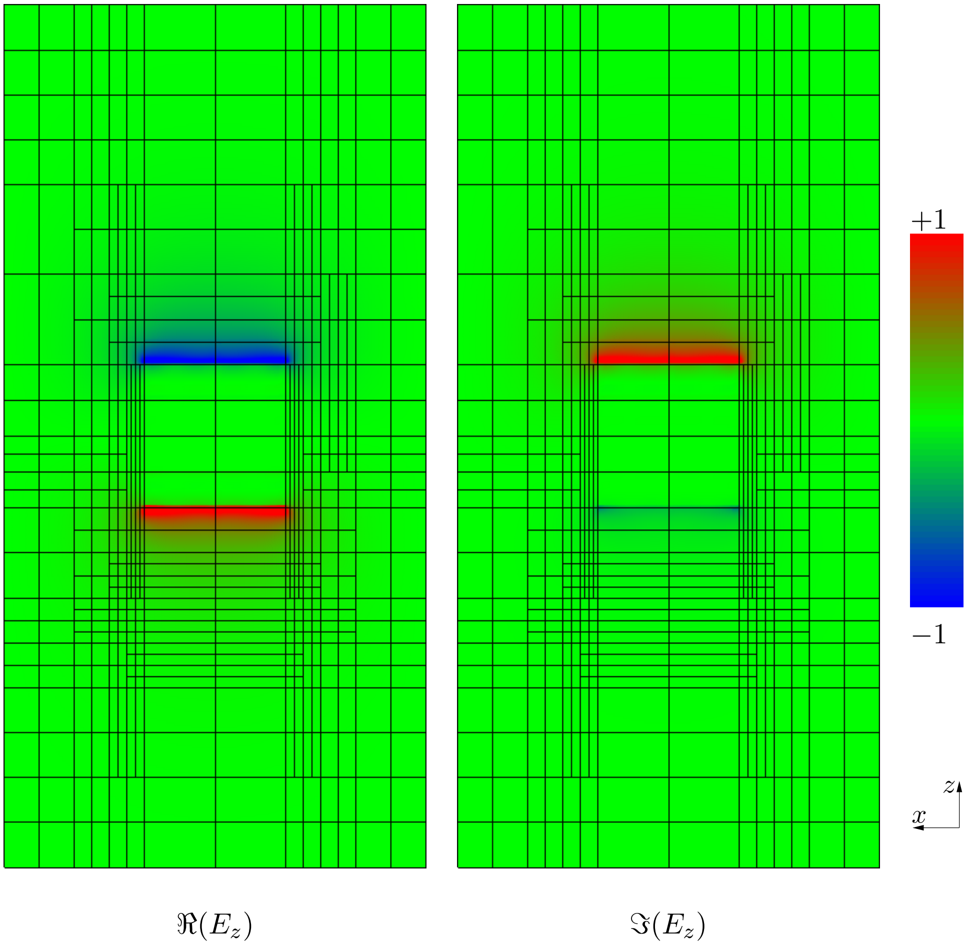


Figure 6.17: Real and imaginary parts of the z-component of the scattered electric field for $t = 1/3$ in the plane $y = 1/3$ ($\|E_z\|_\infty \approx 6.8$).

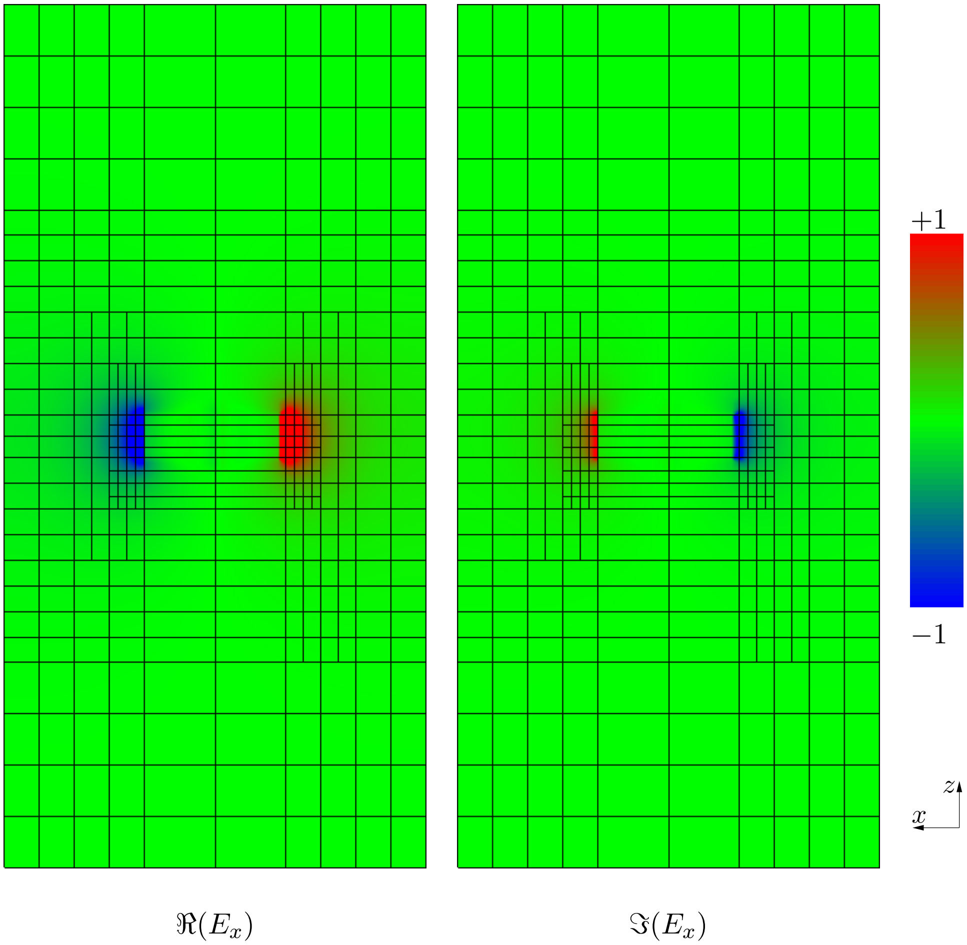
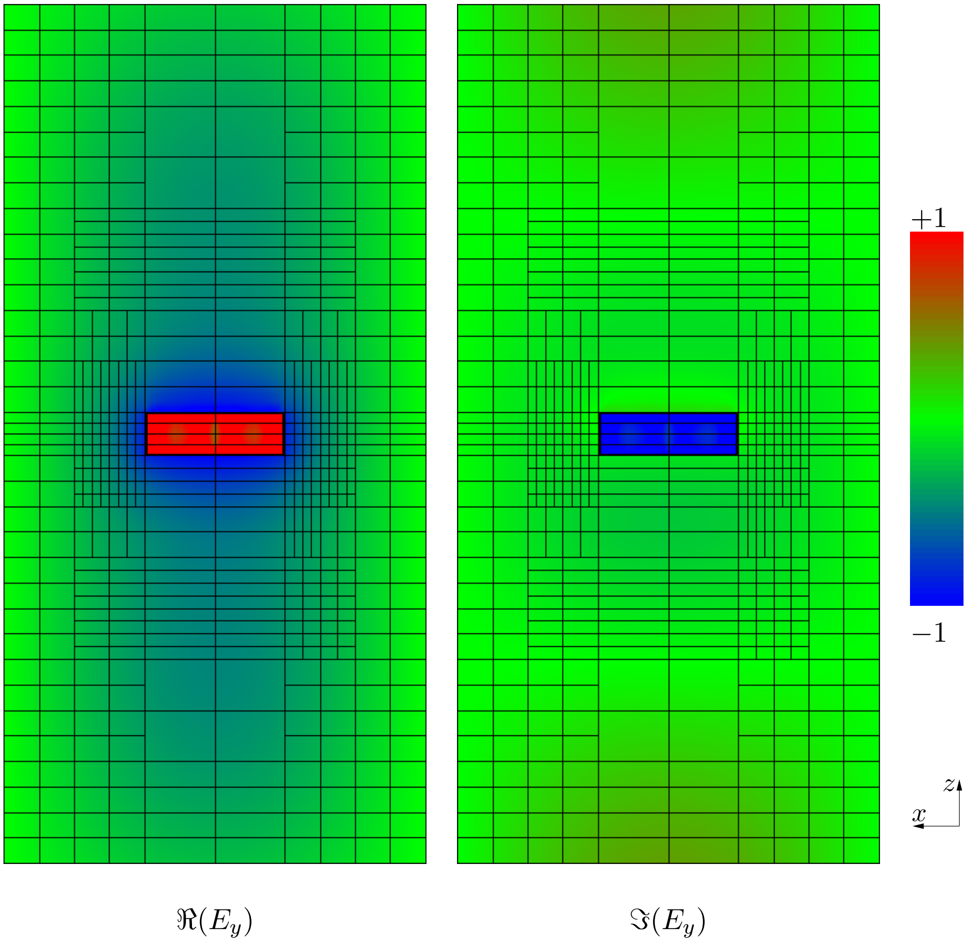


Figure 6.18: Real and imaginary parts of the x-component of the scattered electric field for $t = 1/10$ in the plane $y = 1/3$ ($\|E_x\|_\infty \approx 10.3$).



$\Re(E_y)$ $\Im(E_y)$

Figure 6.19: Real and imaginary parts of the y-component of the scattered electric field for $t = 1/10$ in the plane $y = 1/2$ ($\|E_y\|_\infty \approx 9.2$).

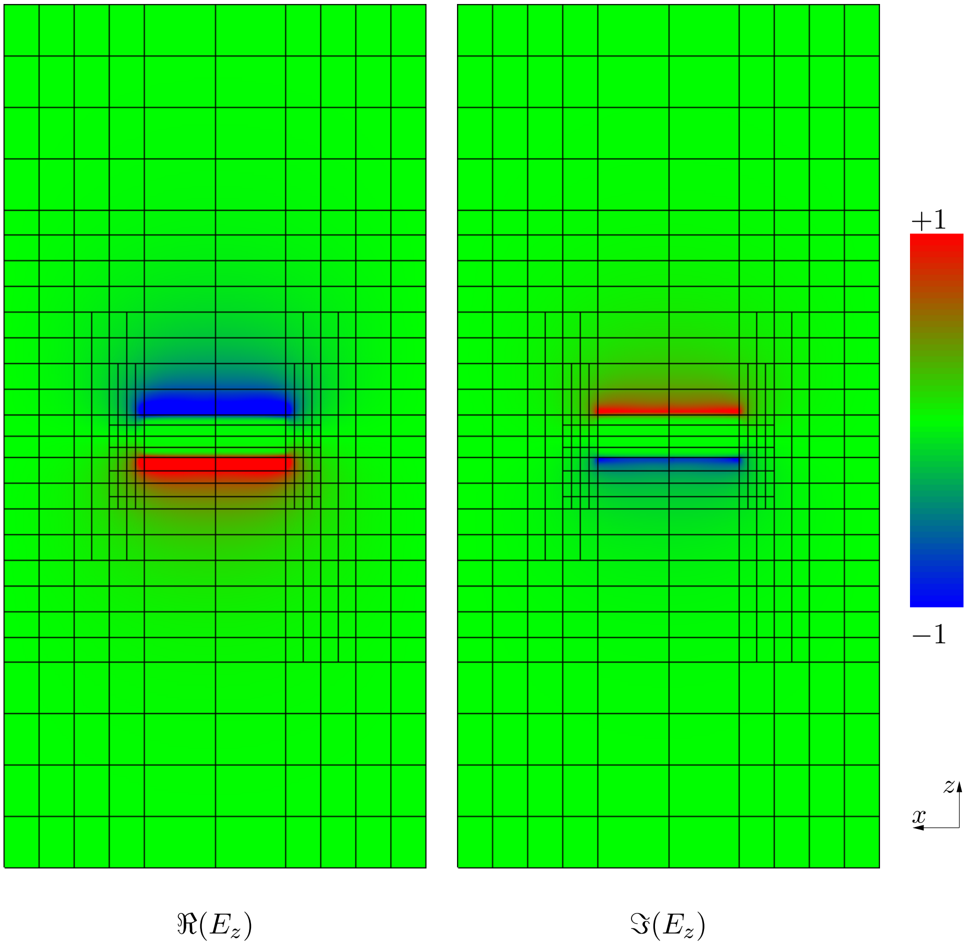


Figure 6.20: Real and imaginary parts of the z-component of the scattered electric field for $t = 1/10$ in the plane $y = 1/3$ ($\|E_z\|_\infty \approx 9.5$).

Chapter 7

High Performance Computation

Probably the most challenging aspect of *hp*-adaptive finite element computations is controlling the complexity of the implementation. For practical applications however, the efficiency of the implementation is a very close second. In this chapter we identify three areas where a straight-forward implementation proves inadequate and must be replaced by a slightly more complex, but dramatically more efficient alternative. The first area is the computation of element stiffness matrices, where we present an implementation of the classical sum-factorization algorithm that achieves an optimal speedup through the efficient use of fast cache memory. In the second section we discuss our interface with the direct multifrontal solver MUMPS and several tricks we have employed to accelerate the factorization of the global stiffness matrix. Finally we conclude with a discussion of fast algorithms for the fundamental operation behind our *hp*-refinement strategy, namely the problem of projecting a given function onto a nested sequence of (piecewise) polynomial spaces.

7.1 Sum factorization

For the Helmholtz equation (or Laplace's equation if $k = 0$), the finite element stiffness matrix has entries of the form,

$$B_{JI} = \int_K \{[\boldsymbol{\nabla}_x \phi_I(\boldsymbol{x})]^T \boldsymbol{\nabla}_x \phi_J(\boldsymbol{x}) - k^2 \phi_I(\boldsymbol{x})\phi_J(\boldsymbol{x})\} dx.$$

We make the typical change of variables,

$$B_{JI} = \int_{\widehat{K}} \left\{ [\boldsymbol{\nabla}_\xi \phi_I(\boldsymbol{\xi})]^T D \boldsymbol{\nabla}_\xi \phi_J(\boldsymbol{\xi}) - k^2 \phi_I(\boldsymbol{\xi})\phi_J(\boldsymbol{\xi}) \left| \frac{dx}{d\xi} \right| \right\} d\xi, \tag{7.1}$$

where D is the symmetric matrix,

$$D = \left| \frac{dx}{d\xi} \right| \frac{d\xi}{dx} \frac{d\xi}{dx}^T.$$

If the element has polynomial order of approximation $\boldsymbol{p} = [p_1, p_2, p_3]$ then a Gauss-Legendre quadrature rule of order $\boldsymbol{p} + 1 = [p_1 + 1, p_2 + 1, p_3 + 1]$ is used to guarantee exact integration in the case of an affine element. Then

$$B_{JI} = \sum_{l_1=1}^{p_1+1} \sum_{l_2=1}^{p_2+1} \sum_{l_3=1}^{p_3+1} \left\{ [\boldsymbol{\nabla}_\xi \phi_I(\boldsymbol{\xi}^{(l)})]^T D \boldsymbol{\nabla}_\xi \phi_J(\boldsymbol{\xi}^{(l)}) - k^2 \phi_I(\boldsymbol{\xi}^{(l)})\phi_J(\boldsymbol{\xi}^{(l)}) \left| \frac{dx}{d\xi} \right| \right\} w^{(l)} \tag{7.2}$$

where $\boldsymbol{\xi}^{(l)} = [\xi_1^{(l_1)}, \xi_2^{(l_2)}, \xi_3^{(l_3)}]$ and $w^{(l)} = w_1^{(l_1)} w_2^{(l_2)} w_3^{(l_3)}$ are the 1D Gauss-Legendre points and weights, respectively. A straight-forward algorithm for computing the entries in (7.2) is shown in Figure 7.1. Clearly the runtime complexity of this algorithm is $O(p^9)$.

Sum factorization is a well-known algorithm (e.g. see [37] and [38]) for accelerating the integration of element stiffness matrices. Here, we take a different approach from [38] by computing an auxiliary stiffness matrix for a promoted element, i.e. the element obtained by possibly increasing the orders for all edge and

```
initialize $B = 0$
for $l_3 = 1, \ldots, p_3 + 1$
for $l_2 = 1, \ldots, p_2 + 1$
for $l_1 = 1, \ldots, p_1 + 1$
    Evaluate shape functions $\phi$ and gradients $\boldsymbol{\nabla}_\xi \phi$ at point $\boldsymbol{\xi}^{(l)}$
    Evaluate Jacobian $|dx/d\xi|$ and metric $D$
    for $J, I = 1, \ldots, (p_1 + 1)(p_2 + 1)(p_3 + 1)$
        $B_{JI} = B_{JI} + \left\{ [\boldsymbol{\nabla}_\xi \phi_I]^T D \boldsymbol{\nabla}_\xi \phi_J - k^2 \phi_I \phi_J \left| \frac{dx}{d\xi} \right| \right\} w^{(l)}$
```

Figure 7.1: Classical algorithm for element stiffness matrix computation

face nodes to the parallel order for the element interior. As a final step, the relevant entries are then extracted to the actual stiffness matrix. This approach leads to a relatively clean implementation, free of logical gates, which attains the theoretical speedup.

The basic assumption is that $3D$ shape functions are obtained as tensor products of $1D$ shape functions, i.e.

$$\begin{aligned} \phi_I(\xi_1, \xi_2, \xi_3) &= \chi_{i_1}(\xi_1) \chi_{i_2}(\xi_2) \chi_{i_3}(\xi_3), \\ \phi_J(\xi_1, \xi_2, \xi_3) &= \chi_{j_1}(\xi_1) \chi_{j_2}(\xi_2) \chi_{j_3}(\xi_3). \end{aligned}$$

Now since the matrix D is symmetric, we enumerate its entries as,

$$D = \begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{pmatrix} = \begin{pmatrix} D_1 & D_2 & D_3 \\ D_2 & D_4 & D_5 \\ D_3 & D_5 & D_6 \end{pmatrix},$$

and introduce the symbol,

$$E = k^2 \left| \frac{dx}{d\xi} \right|.$$

Then the integrand in (7.1) has the expansion,

$$\begin{aligned}&\frac{\partial\phi_I}{\partial\xi_1}\left(\frac{\partial\phi_J}{\partial\xi_1}D_1+\frac{\partial\phi_J}{\partial\xi_2}D_2+\frac{\partial\phi_J}{\partial\xi_3}D_3\right)\\ &+\quad\frac{\partial\phi_I}{\partial\xi_2}\left(\frac{\partial\phi_J}{\partial\xi_1}D_2+\frac{\partial\phi_J}{\partial\xi_2}D_4+\frac{\partial\phi_J}{\partial\xi_3}D_5\right)\\ &+\quad\frac{\partial\phi_I}{\partial\xi_3}\left(\frac{\partial\phi_J}{\partial\xi_1}D_3+\frac{\partial\phi_J}{\partial\xi_2}D_5+\frac{\partial\phi_J}{\partial\xi_3}D_6\right)-\phi_I\phi_J E,\end{aligned}$$

which can be written in terms of the $1D$ shape functions as,

$$\begin{aligned}&\chi'_{i_1}\chi'_{j_1}\chi_{i_2}\chi_{j_2}\chi_{i_3}\chi_{j_3}D_1+\chi'_{i_1}\chi_{j_1}\chi_{i_2}\chi'_{j_2}\chi_{i_3}\chi_{j_3}D_2+\chi'_{i_1}\chi_{j_1}\chi_{i_2}\chi_{j_2}\chi_{i_3}\chi'_{j_3}D_3\\ +&\quad\chi_{i_1}\chi'_{j_1}\chi'_{i_2}\chi_{j_2}\chi_{i_3}\chi_{j_3}D_2+\chi_{i_1}\chi_{j_1}\chi'_{i_2}\chi'_{j_2}\chi_{i_3}\chi_{j_3}D_4+\chi_{i_1}\chi_{j_1}\chi'_{i_2}\chi_{j_2}\chi_{i_3}\chi'_{j_3}D_5\\ +&\quad\chi_{i_1}\chi'_{j_1}\chi_{i_2}\chi_{j_2}\chi'_{i_3}\chi_{j_3}D_3+\chi_{i_1}\chi_{j_1}\chi_{i_2}\chi'_{j_2}\chi'_{i_3}\chi_{j_3}D_5+\chi_{i_1}\chi_{j_1}\chi_{i_2}\chi_{j_2}\chi'_{i_3}\chi'_{j_3}D_6\\ -&\quad\chi_{i_1}\chi_{j_1}\chi_{i_2}\chi_{j_2}\chi_{i_3}\chi_{j_3}E),\end{aligned}$$

where $'$ indicates differentiation with respect to function argument. Finally we collect like terms in ξ_1 to get

$$\begin{aligned}\chi'_{i_1}\chi'_{j_1}\ \chi_{i_2}\chi_{j_2}\ \chi_{i_3}\chi_{j_3}D_1 &\quad +\\ \chi'_{i_1}\chi_{j_1}(\chi_{i_2}\chi'_{j_2}\ \chi_{i_3}\chi_{j_3}D_2 &\quad +\\ \chi_{i_2}\chi_{j_2}\ \chi_{i_3}\chi'_{j_3}D_3) &\quad +\\ \chi_{i_1}\chi'_{j_1}(\chi'_{i_2}\chi_{j_2}\ \chi_{i_3}\chi_{j_3}D_2 &\quad +\\ \chi_{i_2}\chi_{j_2}\ \chi'_{i_3}\chi_{j_3}D_3) &\quad +\\ \chi_{i_1}\chi_{j_1}(\chi'_{i_2}\chi'_{j_2}\ \chi_{i_3}\chi_{j_3}D_4 &\quad +\\ \chi'_{i_2}\chi_{j_2}\ \chi_{i_3}\chi'_{j_3}D_5 &\quad +\\ \chi_{i_2}\chi'_{j_2}\ \chi'_{i_3}\chi_{j_3}D_5 &\quad +\\ \chi_{i_2}\chi_{j_2}(\chi'_{i_3}\chi'_{j_3}D_6 &\quad -\quad \chi_{i_3}\chi_{j_3}E)).\end{aligned}$$

Now we are ready to express (7.1) as an iterated integral. The entries B_{JI} will be extracted from the auxiliary stiffness matrix $B^{\text{aux}} = B^{\text{aux}}(j_1, j_2, j_3, i_1, i_2, i_3)$, where B^{aux} is computed as follows,

$$\begin{aligned} B^{\text{aux}}(j_1, j_2, j_3, i_1, i_2, i_3) \;=\; & \int_0^1 \{\chi'_{i_1}\chi'_{j_1} B_1^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) + \\ & \chi'_{i_1}\chi_{j_1} B_2^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) + \\ & \chi_{i_1}\chi'_{j_1} B_3^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) + \\ & \chi_{i_1}\chi_{j_1} B_4^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1)\} d\xi_1 \end{aligned}$$

The new auxiliary matrix B^{aux1} has the form,

$$\begin{aligned} B_1^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi_{i_2}\chi_{j_2} B_1^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1)\} d\xi_2, \\ B_2^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi_{i_2}\chi'_{j_2} B_2^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) + \\ & \chi_{i_2}\chi_{j_2} B_3^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1)\} d\xi_2, \\ B_3^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi'_{i_2}\chi_{j_2} B_2^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) + \\ & \chi_{i_2}\chi_{j_2} B_4^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1)\} d\xi_2, \\ B_4^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi'_{i_2}\chi'_{j_2} B_5^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) + \\ & \chi'_{i_2}\chi_{j_2} B_6^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) + \\ & \chi_{i_2}\chi'_{j_2} B_7^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) + \\ & \chi_{i_2}\chi_{j_2} B_8^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1)\} d\xi_2, \end{aligned}$$

where,

$$\begin{aligned} B_1^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi_{i_3}\chi_{j_3} D_1(\xi_1, \xi_2, \xi_3)\} d\xi_3, \\ B_2^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi_{i_3}\chi_{j_3} D_2(\xi_1, \xi_2, \xi_3)\} d\xi_3, \\ B_3^{\text{aux2}}(; \xi_2, j_3, i_3, \xi_1) \;=\; & \int_0^1 \{\chi_{i_3}\chi'_{j_3} D_3(\xi_1, \xi_2, \xi_3)\} d\xi_3, \end{aligned}$$

$$\begin{aligned}
B_4^{\text{aux2}}(;\xi_2, j_3, i_3, \xi_1) &= \int_0^1 \{\chi'_{i_3}\chi_{j_3}D_3(\xi_1,\xi_2,\xi_3)\}d\xi_3,\\
B_5^{\text{aux2}}(;\xi_2, j_3, i_3, \xi_1) &= \int_0^1 \{\chi_{i_3}\chi_{j_3}D_4(\xi_1,\xi_2,\xi_3)\}d\xi_3,\\
B_6^{\text{aux2}}(;\xi_2, j_3, i_3, \xi_1) &= \int_0^1 \{\chi_{i_3}\chi'_{j_3}D_5(\xi_1,\xi_2,\xi_3)\}d\xi_3,\\
B_7^{\text{aux2}}(;\xi_2, j_3, i_3, \xi_1) &= \int_0^1 \{\chi'_{i_3}\chi_{j_3}D_5(\xi_1,\xi_2,\xi_3)\}d\xi_3,\\
B_8^{\text{aux2}}(;\xi_2, j_3, i_3, \xi_1) &= \int_0^1 \{\chi'_{i_3}\chi'_{j_3}D_6(\xi_1,\xi_2,\xi_3) - \chi_{i_3}\chi_{j_3}E(\xi_1,\xi_2,\xi_3)\}d\xi_3.
\end{aligned}$$

The integrals above are approximated using Gauss-Legendre quadrature of order $p+1$ (exact for affine elements), and the notation using ξ_j as an index is replaced by the index of a quadrature point in the ξ_j-direction (the significance of the ";" will be explained below). Clearly, this formulation can be evaluated in $O(p^7) = O(p^{2d+1})$-time, using $O(p^6) + O(p^5) + O(p^4)$ auxiliary storage. The auxiliary storage, though essentially of $O(p^6)$ for the storage of B^{aux}, can be reduced (without any extra computation) by an appropriate choice of order for the necessary nested loops. We implement the following loops,

```
for each $\xi_1, i_3, j_3$
    set $B^{\text{aux1}} = 0$
    for each $\xi_2$
        set $B^{\text{aux2}} = 0$
        for each $\xi_3$
            accumulate for $B^{\text{aux2}} = B^{\text{aux2}}(;\xi_2, j_3, i_3, \xi_1)$
        for each $i_2, j_2$
            accumulate for $B^{\text{aux1}} = B^{\text{aux1}}(j_2, i_2; j_3, i_3, \xi_1)$
    for each $i_2, j_2$
        for each $i_1, j_1$
            accumulate for $B^{\text{aux}} = B^{\text{aux}}(j_1, j_2, j_3, i_1, i_2, i_3)$
```

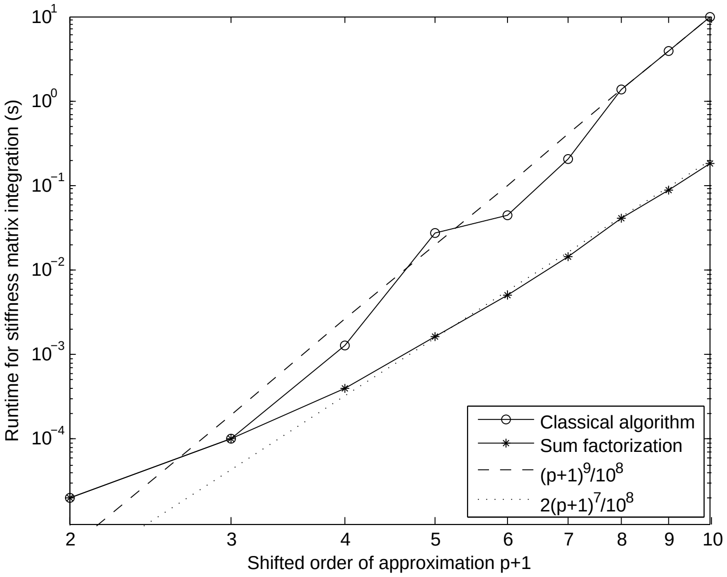


Figure 7.2: Runtime for element stiffness matrix integration

Now we can observe that all of the indices appearing to right of a ";" do not require explicit storage because of the order in which the arrays appear within the loops. This loop structure allows us to reduce the auxiliary storage to $O(p^6)+O(p^2)+O(1)$.

This algorithm is implemented in the subroutine `elem_sumfact`, with much attention given to the elimination of unnecessary computations. The runtimes for the classical algorithm and for sum factorization are shown in Figure 7.2. For comparison, curves are also plotted for $O(p^9)$ and $O(p^7)$. While there are some outliers, the expected trends are clear. For the maximum order of approximation $p = 9$, the runtime is reduced from 9.9 seconds to 0.18 seconds! This improvement is dramatic enough that stiffness matrix integration becomes negligible in comparison to the time for the direct solution of the global system of equations.

In [38], an alternative implementation is presented, based explicitly on the decomposition of element shape functions into vertex, edge, face and interior degrees of freedom. The stiffness matrix is accordingly decomposed into blocks with sum factorization applied to each block using lower order quadrature where possible (e.g. vertex-vertex interaction can be calculated with 2 Gauss points in each direction, rather than $p+1$). The implementation is more logically complex than that reported above (though the algorithmic complexity is still $O(p^7)$), and results in a speedup ratio of less than 5 for $p = 9$. Even the accelerated spectral-Galerkin algorithm (based on a non-hierarchical Lagrange basis, and resulting in an algorithmic complexity of $O(p^5)$) achieves a speedup ratio of less than 12 (for $p = 9$).

Apart from the `elem_sumfact` routine, sum factorization has been applied to other parts of the code. Fast routines were developed for adaptive integration of the element load vector including the volume contribution associated with an inhomogeneous right hand side and the boundary integral associated with a Neumann boundary condition.

7.2 Optimized interface to the MUltifrontal Massively Parallel Solver (MUMPS)

The limiting factor behind any finite element code is the solution of the global system of equations. In our algorithm, the coarse grid is negligible and only the performance of a given solver for the fine grid is relevant. For an overview of iterative and multigrid techniques adapted to *hp* methods we refer to [41], [43] and [44]. Here, we describe a variety of interfaces that we have developed for the direct solver MUMPS.

MUMPS is a general purpose solver for sparse linear systems. There are separate versions for single and double precision, and real or complex arithmetic, each of which can be compiled for serial execution on a single processor workstation, or for parallel execution on a distributed-memory parallel machine (using the message passing interface (MPI)). Each version implements the three factorizations, $A = LL^T$ for symmetric positive-definite problems, $A = LDL^T$ for symmetric indefinite problems, and $A = LU$ for unsymmetric problems. The matrix A can be specified in either a sparse assembled format, or as a collection of unassembled element matrices. Recently added features include support for multiple right-hand sides and the ability to compute only a partial factorization and the associated Schur complement, both of which are particularly useful for scattering problems. For more information on MUMPS please see [3], [2] and [4].

We will begin with a description of our interface for symmetric positive-definite problems, where we observe a significant gain from the explicit implementation of static condensation. We then present an interface that uses both the real and complex versions of MUMPS to build an efficient solver for scattering problems.

7.2.1 The symmetric positive-definite case

The interface with MUMPS is based mainly on specifying four arrays.

ELTPTR Specifies the locations in the ELTVAR array where data is stored for each element.

ELTVAR Stores the map from local element shape function numbers to corresponding global basis function numbers.

A_ELT Stores the unassembled element contributions to the global stiffness matrix.

RHS Stores the assembled right hand side and is overwritten with the solution.

The solution process is then separated into three stages.

analysis Automatically selects from up to six available algorithms (including the graph partitioner METIS) and determines an optimal or nearly optimal ordering for the assembly and factorization of the global stiffness matrix.

factorization Carries out the factorization determined in the analysis phase.

solve Performs back-substitution, overwriting the right hand side with the solution.

The efficiency of the solver hinges on both the optimality of the re-ordering determined in the analysis phase and the implementation of the factorization in terms of level III BLAS operations.

To test the optimality we have implemented a second interface that explicitly performs the static condensation of interior degrees of freedom. Let A and a be the stiffness matrix and right hand side for a given element, and let B and b be the assembled contributions from all other elements. Then if the interface degrees of freedom for the element are listed first, the global linear system has the form,

$$\begin{bmatrix} A_{11}+B_{11} & A_{12}^T & B_{31}^T \\ A_{21} & A_{22} & 0 \\ B_{31} & 0 & B_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_1+b_1 \\ a_2 \\ b_3 \end{bmatrix}. \tag{7.3}$$

Static condensation consists of eliminating the interior degrees of freedom by,

$$x_2 = A_{22}^{-1}(a_2 - A_{21}x_1), \tag{7.4}$$

so that the remaining global system becomes,

$$\begin{bmatrix} A_{11} - A_{21}^T A_{22}^{-1} A_{21} + B_{11} & B_{31}^T \\ B_{31} & B_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_1 - A_{21}^T A_{22}^{-1} a_2 + b_1 \\ b_3 \end{bmatrix}. \tag{7.5}$$

Since the modifications (compared with (7.3)) are local to each element, static condensation can be implemented in a pre-processing step, where we form the modified element stiffness matrix,

$$\tilde{A} = A_{11} - A_{21}^T A_{22}^{-1} A_{21},$$

and right hand side,

$$\tilde{a} = a_1 - A_{21}^T A_{22}^{-1} a_2,$$

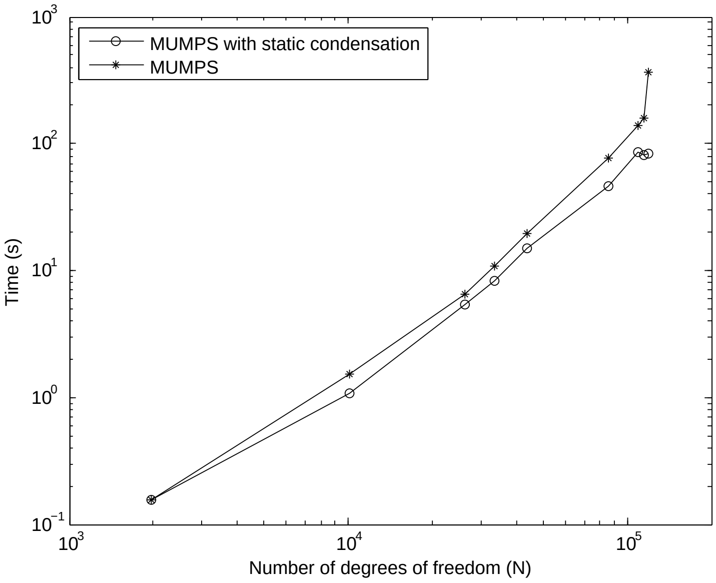


Figure 7.3: Impact of static condensation on total solve time for the sequence of fine grids for the Fichera problem

combined with a post-processing step where we solve for the middle node using (7.4). Some auxiliary memory is used to store,

$$\tilde{A}_{21} = A_{22}^{-1}A_{21},\ \tilde{a}_2 = A_{22}^{-1}a_2,$$

rather than recomputing them in the post-processing step.

In Figures 7.3 and 7.4 we present the total solve time and memory usage for the sequence of fine grids for the Fichera problem from section 5.1 using the two approaches described above (MUMPS alone and MUMPS with explicit static condensation). The reported memory for the case with static condensation includes both the internal memory used by MUMPS and the auxiliary memory used in the post-processing step (7.4). We observe that, for this example, static condensation delivers nearly 50% reduction in runtime and uses up to 30% less memory.

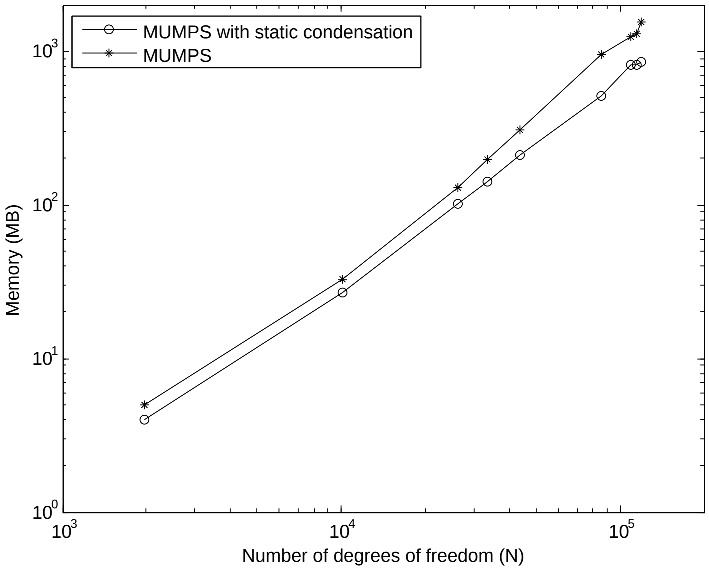


Figure 7.4: Impact of static condensation on total memory used for the sequence of fine grids for the Fichera problem

7.2.2 A domain decomposition solver for acoustics and electromagnetics

We observe that for each of the scattering problems (2.9), (2.25), (2.45), (2.47) and (2.55), the associated sesquilinear form can be split into an interior and an exterior contribution. In each case, the interior contribution is real-valued, symmetric and indefinite. For the infinite element methods, the exterior contribution is complex-valued and unsymmetric, while for PML, the exterior contribution is complex symmetric (not Hermitian) and indefinite. Our approach is to enumerate the global degrees of freedom such that interior dof are listed first, the interface dof are listed second and exterior dof are listed last (for the waveguide truncated with an impedance boundary condition (2.45) there are no exterior dof).

For the case of a PML truncation, the global linear system has the form,

$$\begin{bmatrix} A_{11} & A_{21}^T & 0 \\ A_{21} & A_{22}+Z_{22} & Z_{32}^T \\ 0 & Z_{32} & Z_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix}. \tag{7.6}$$

Here, the matrix A is real-valued, symmetric indefinite, and Z is complex-valued, symmetric indefinite. Then using the first row of (7.6) to eliminate x_1,

$$x_1 = A_{11}^{-1}(b_1 - A_{21}^T x_2), \tag{7.7}$$

the remaining system is,

$$\begin{bmatrix} A_{22} - A_{21}A_{11}^{-1}A_{21}^T + Z_{22} & Z_{32}^T \\ Z_{32} & Z_{33} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -A_{21}A_{11}^{-1}b_1 \\ 0 \end{bmatrix}. \tag{7.8}$$

Hence we can limit the complex arithmetic to the solution of the interface-exterior problem (7.8), and use the faster real arithmetic to form the (dense) Schur complement matrix $A_{22} - A_{21}A_{11}^{-1}A_{21}^T$, and solve for the interior degrees of freedom (7.7). Our algorithm reads as follows:

- Input the system,

  $$\begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \end{bmatrix},$$

  to the real symmetric indefinite solver (dmumps) listing x_2 as interface degrees of freedom, factor A_{11} and form the Schur complement matrix,

  $$\tilde{A}_{22} = A_{22} - A_{21}A_{11}^{-1}A_{21}^T$$

and modified right-hand side,

$$b_2 = -A_{11}^{-1} b_1.$$

- Compute the unassembled-matrix vector product $\tilde{b}_2 = A_{21} b_2$.
- Input the system (7.8) to the complex symmetric solver (zmumps), by adding the Schur complement $\tilde{A}_{22}$ as if it were a single (large) element stiffness matrix, and solve for the interface and exterior degrees of freedom x_2 and x_3.
- Modify the right hand side for the real solver by computing the unassembled-matrix vector product $\tilde{b}_1 = b_1 - A_{21}^T x_2$, and use the existing factorization of A_{11} to solve for x_1.

The algorithm for infinite elements, where the exterior matrix Z is unsymmetric, is very similar.

We remark that the original system (7.6) may either be the full system including middle nodes, or the reduced system resulting from static condensation of middle nodes. The latter case requires additional comment. Consider a single element K with the symmetric indefinite stiffness matrix,

$$A^K = \begin{bmatrix} A_{11}^K & A_{12}^K \\ A_{21}^K & A_{22}^K \end{bmatrix},$$

where $A_{12}^K = [A_{21}^K]^T$, and A_{22}^K is the middle node contribution. The natural choice is to form the modified stiffness matrix $\tilde{A}_{11}^K = A_{11}^K - A_{12}^K [A_{22}^K]^{-1} A_{21}^K$, by computing the symmetric factorization $A_{22}^K = LDL^T$. However, the LAPACK implementation `DSYTRF`, appears to suffer from poor cache performance, and is easily outperformed by the general LU decomposition with partial pivoting, $A_{22}^K = PLU$. That is, it is faster to treat the element as if it were unsymmetric, rather than trying to exploit symmetry, when eliminating middle nodes.

7.3 Fast solvers for projection-based interpolation

Computing the projection-based interpolant requires the solution of a linear system of equations. The matrix is real, symmetric and either positive definite (for all projections in H^1 and the edge contributions in $H(curl)$) or indefinite (for face and interior projections in $H(curl)$). The right-hand side (and hence the solution) may be real or complex-valued. In either case, our *hp* algorithm (applied to each edge, face and element interior, and for each admissible *h*-refinement type) requires the computation of not a single projection, but a dynamically-determined sequence of nested projections. The sequence is dynamically determined in the sense that we compute the projection for a given mesh and then examine the projection error to determine the next mesh. It is nested both because the meshes are nested and (since our shape functions are hierarchical) because the matrices are nested.

The straight-forward algorithm is to assemble the matrix A and right-hand side b for the maximal mesh V and then for each sub-mesh $V_k \subset V$:

- Build an extraction vector n^k such that degree of freedom i of mesh V_k corresponds to degree of freedom n^k_i of mesh V.
- Extract the matrix $A^k_{i,j} = A_{n^k_i, n^k_j}$ and right-hand side $b^k_i = b_{n^k_i}$ for projection onto V_k.
- Factor $A^k \rightarrow L_k L_k^T$ or $P_k L_k U_k$ and solve for $x^k = [A^k]^{-1} b^k$.

This is prohibitively expensive since there may be many terms in the sequence (10-20 is typical) and the problem size is as large as $(2p-1)^3 = 3375$ for H^1 and $(8p-1)(2p-1)^2 = 14175$ for $H(curl)$ projection onto an $h8$-refined element interior of uniform order $p = 8$.

We wish to accelerate this process by using the previously computed factorization and solution in the current step. The first modification is that rather than rebuilding the extraction vector at each step we will only append the new degrees of freedom to be added. In what follows, purely to simplify the notation, we will assume that the entire sequence is known apriori and that the maximal matrix and right-hand side have been reordered accordingly. That is, the linear system for mesh k $(1 <= k <= M)$ has the block structure,

$$\begin{bmatrix} A_{11} & \cdots & A_{1k} \\ \vdots & \ddots & \\ A_{k1} & & A_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}. \tag{7.9}$$

7.3.1 The symmetric positive-definite case

As the matrix is symmetric, we will only refer to the lower triangle of (7.9). In what follows, the function POTRF(B) returns the lower triangular matrix L such that $B = LL^T$. Then step k of our algorithm can be expressed as follows:

- Repartition:

$$A = \begin{bmatrix} B_{11} & & \\ B_{21} & B_{22} & \\ B_{31} & B_{32} & B_{33} \end{bmatrix}, \; b = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix},$$

  where,

$$B_{21} = [A_{k,1}, \cdots, A_{k,k-1}], \; B_{22} = A_{kk}, \; c_2 = b_k.$$

- Pre-condition: B_{11} is lower-triangular, and

$$B_{11}B_{11}^T = \begin{bmatrix} A_{11} & \cdots & A_{k-1,1}^T \\ \vdots & \ddots & \\ A_{k-1,1} & & A_{k-1,k-1} \end{bmatrix}, \text{ and } B_{11}B_{11}^T c_1 = \begin{bmatrix} b_1 \\ \vdots \\ b_{k-1} \end{bmatrix}.$$

- Perform in-place updates:

$$\begin{aligned}
\tilde{B}_{21} &= B_{21}B_{11}^{-T} \\
\tilde{B}_{22} &= \mathrm{POTRF}(B_{22} - \tilde{B}_{21}\tilde{B}_{21}^{T}) \\
\tilde{c}_2 &= \tilde{B}_{22}^{-T}\tilde{B}_{22}^{-1}(c_2 - \tilde{B}_{21}B_{11}^{T}c_1) \\
\tilde{c}_1 &= c_1 - B_{11}^{-T}\tilde{B}_{21}^{T}\tilde{c}_2
\end{aligned}$$

- Observe (post-condition):

$$\begin{aligned}
\begin{bmatrix} B_{11} & \\ \tilde{B}_{21} & \tilde{B}_{22} \end{bmatrix}
\begin{bmatrix} B_{11}^{T} & \tilde{B}_{21}^{T} \\ & \tilde{B}_{22}^{T} \end{bmatrix}
&= \begin{bmatrix} B_{11}B_{11}^{T} & B_{11}\tilde{B}_{21}^{T} \\ \tilde{B}_{21}B_{11}^{T} & \tilde{B}_{21}\tilde{B}_{21}^{T} + \tilde{B}_{22}\tilde{B}_{22}^{T} \end{bmatrix} \\
&= \left[\begin{array}{ccc|c}
A_{1,1} & \cdots & A_{k-1,1}^{T} & A_{k,1}^{T} \\
\vdots & \ddots & & \vdots \\
A_{k-1,1} & & A_{k-1,k-1} & A_{k,k-1}^{T} \\
\hline
A_{k,1} & \cdots & A_{k,k-1} & A_{k,k}
\end{array}\right]
\end{aligned}$$

$$\begin{aligned}
&\begin{bmatrix} B_{11} & \\ \tilde{B}_{21} & \tilde{B}_{22} \end{bmatrix}
\begin{bmatrix} B_{11}^{T} & \tilde{B}_{21}^{T} \\ & \tilde{B}_{22}^{T} \end{bmatrix}
\begin{bmatrix} \tilde{c}_1 \\ \tilde{c}_2 \end{bmatrix} \\
&= \begin{bmatrix} B_{11}B_{11}^{T}\tilde{c}_1 + B_{11}\tilde{B}_{21}^{T}\tilde{c}_2 \\ \tilde{B}_{21}B_{11}^{T}\tilde{c}_1 + (\tilde{B}_{21}\tilde{B}_{21}^{T} + \tilde{B}_{22}\tilde{B}_{22}^{T})\tilde{c}_2 \end{bmatrix}
= \left[\begin{array}{c} b_1 \\ \vdots \\ b_{k-1} \\ \hline b_k \end{array}\right]
\end{aligned}$$

Here, the pre-condition simply asserts that B_{11} is the Cholesky factorization and c_1 the solution associated with mesh $k-1$ (observe that at step $k = 1$, blocks B_{11} and c_1 are empty). The updates are performed in-place, i.e. the results overwrite

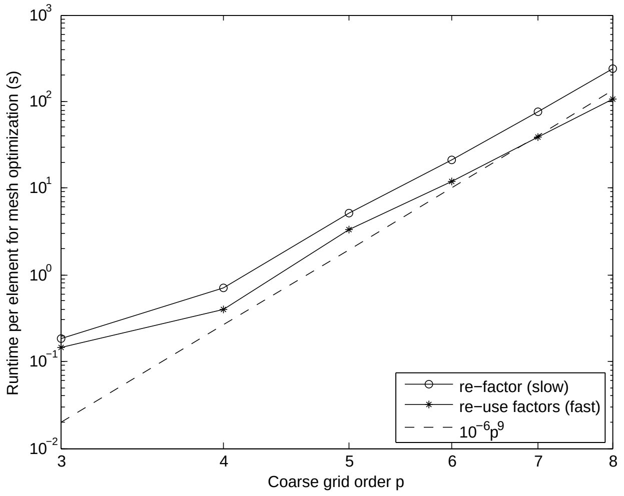


Figure 7.5: Runtime for mesh optimization as a function of coarse grid order p (symmetric positive definite case)

the corresponding locations in matrix A and right-hand side b. Finally, the post-condition (easily verified) asserts that A and b now satisfy the pre-condition for step $k+1$. Notice that the algorithm makes no reference to blocks B_{31}, B_{32}, B_{33} or c_3, and is therefore ideal for our application (since these blocks are not yet known at step k).

In Figure 7.5 we show the impact of this algorithm on the runtime of the whole mesh optimization process. The timing data was obtained by repeatedly running one step of *hp* refinement for the Fichera corner problem with an initial mesh of 7 elements of uniform order $p = 3, \ldots, 8$. We plot the runtime (per element) when two different solvers are used for projection-based interpolation: the slow one *re-factors* the entire matrix at each step, and the fast one *re-uses* the factors from the previous step. We can observe that these results are pre-asymptotic since both algorithms appear to scale faster than the expected rate of p^9. Still, the fast algorithm takes around half the time of the slow one, and by looking at the speedup ratio (Figure 7.6) we can observe that the pre-asymptotic *rate* is improved by around

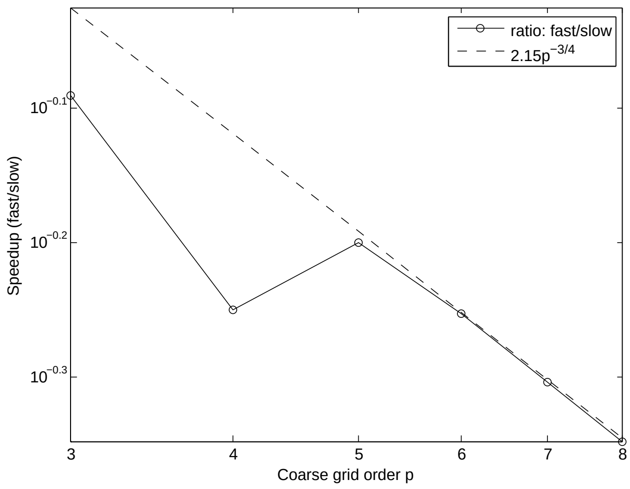


Figure 7.6: Speedup: ratio of fast runtime over slow runtime as a function of coarse grid order p (symmetric positive definite case)

3/4.

7.3.2 The symmetric indefinite case

Our approach is to disregard the symmetry of A and build our algorithm around the general LU-factorization with partial pivoting, $A = PLU$, where P is a permutation matrix, L is unit lower triangular and U is upper triangular. This is motivated by our experience that the symmetric factorization $A = LDL^T$ (where L is a permuted lower-triangular matrix and D is block 1-by-1 and 2-by-2 diagonal) suffers from poor utilization of cache memory. In what follows, we will use the notation $[\tilde{A}, P] = \text{GETRF}(A)$ to indicate that A is overwritten by is factors L and U, i.e. $\tilde{A} = [L\backslash U]$ and $PLU = A$. In order to build an algorithm in the spirit of the previous section, we will restrict the permutation matrix P to be block diagonal. Then step k of our algorithm can be expressed as follows:

- Repartition:

$$A = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}, \ b = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \ P = \begin{bmatrix} P_1 & & \\ & P_2 & \\ & & P_3 \end{bmatrix}$$

where,

$$B_{21} = [A_{k,1}, \cdots, A_{k,k-1}], \ B_{22} = A_{kk}, \ B_{12} = \begin{bmatrix} A_{1,k} \\ \vdots \\ A_{k-1,k} \end{bmatrix}, \ c_2 = b_k.$$

- Pre-condition: $B_{11} = [L_{11} \backslash U_{11}]$, where L_{11} is unit lower-triangular, U_{11} is upper-triangular, such that

$$L_{11}U_{11} = P_1^T \begin{bmatrix} A_{11} & \cdots & A_{1,k-1} \\ \vdots & \ddots & \\ A_{k-1,1} & & A_{k-1,k-1} \end{bmatrix}, \text{ and } L_{11}U_{11}c_1 = P_1^T \begin{bmatrix} b_1 \\ \vdots \\ b_{k-1} \end{bmatrix}.$$

Block P_2 is uninitialized.

- Perform in-place updates:

$$\begin{aligned} \tilde{B}_{12} &= L_{11}^{-1} P_1^T B_{12} \\ \left[\tilde{B}_{22}, P_2\right] &= \text{GETRF}(B_{22} - B_{21} U_{11}^{-1} \tilde{B}_{12}) \\ \tilde{B}_{21} &= P_2^T B_{21} U_{11}^{-1} \\ \tilde{c}_2 &= U_{22}^{-1} L_{22}^{-1} (P_2^T c_2 - \tilde{B}_{21} U_{11} c_1) \\ \tilde{c}_1 &= c_1 - U_{11}^{-1} \tilde{B}_{12} \tilde{c}_2 \end{aligned}$$

- Observe (post-condition):

$$\begin{bmatrix} B_{11} & \tilde{B}_{12} \\ \tilde{B}_{21} & \tilde{B}_{22} \end{bmatrix} = \begin{bmatrix} L_{11} \backslash U_{11} & \tilde{B}_{12} \\ \tilde{B}_{21} & L_{22} \backslash U_{22} \end{bmatrix}$$

$$\begin{aligned} \begin{bmatrix} L_{11} & \\ \tilde{B}_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & \tilde{B}_{12} \\ & U_{22} \end{bmatrix} &= \begin{bmatrix} L_{11} U_{11} & P_1^T B_{12} \\ P_2^T B_{21} & P_2^T B_{22} \end{bmatrix} \\ &= \begin{bmatrix} P_1^T & \\ & P_2^T \end{bmatrix} \left[\begin{array}{ccc|c} A_{1,1} & \cdots & A_{k-1,1}^T & A_{k,1}^T \\ \vdots & \ddots & & \vdots \\ A_{k-1,1} & & A_{k-1,k-1} & A_{k,k-1}^T \\ \hline A_{k,1} & \cdots & A_{k,k-1} & A_{k,k} \end{array} \right] \end{aligned}$$

$$\begin{aligned} & \begin{bmatrix} L_{11} & \\ \tilde{B}_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & \tilde{B}_{12} \\ & U_{22} \end{bmatrix} \begin{bmatrix} \tilde{c}_1 \\ \tilde{c}_2 \end{bmatrix} \\ &= \begin{bmatrix} L_{11} U_{11} \tilde{c}_1 + L_{11} \tilde{B}_{12} \tilde{c}_2 \\ \tilde{B}_{21} U_{11} \tilde{c}_1 + (\tilde{B}_{21} \tilde{B}_{12} + L_{22} U_{22}) \tilde{c}_2 \end{bmatrix} \\ &= \begin{bmatrix} P_1^T & \\ & P_2^T \end{bmatrix} \left[\begin{array}{c} b_1 \\ \vdots \\ b_{k-1} \\ \hline b_k \end{array} \right] \end{aligned}$$

Once again, the pre-condition asserts that $[B_{11}, P_1]$ is the LU decomposition, and c_1 the solution, associated with mesh $k-1$. The updates are again performed in-place, and the post-condition (easily verified) asserts that A and b now satisfy the pre-condition for step $k+1$. We see that this is just a blocked algorithm for computing the LU-factorization with partial pivoting, where we have restricted the pivoting

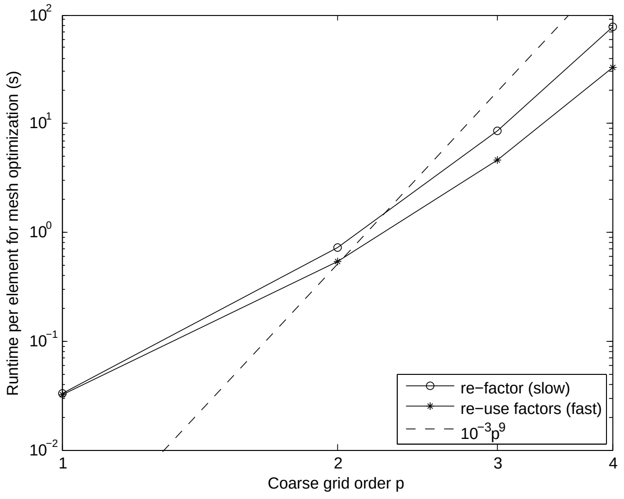


Figure 7.7: Runtime for mesh optimization as a function of coarse grid order p (symmetric indefinite case)

strategy so that only pivoting within blocks is allowed.

In Figure 7.7 we show the impact of this algorithm on the runtime of the whole mesh optimization process. The timing data was obtained by repeatedly running one step of *hp* refinement for the waveguide problem with an initial mesh of 26 elements of uniform order $p = 1, \dots, 4$. We plot the runtime (per element) when two different solvers are used for projection-based interpolation: the slow one *re-factors* the entire matrix at each step, and the fast one *re-uses* the factors from the previous step. Once again the results are pre-asymptotic since both algorithms appear to scale faster than the expected rate of p^9, and by looking at the speedup ratio (Figure 7.8) we can observe that the pre-asymptotic *rate* is improved by around 4/5.

We remark that the numerical stability of this algorithm is certainly questionable. At present, since we have only applied the algorithm to matrices arising from $\boldsymbol{H}(\text{curl})$ projection based interpolation, we simply check that the solution components corresponding to the Lagrange multiplier are close to zero at each step (and

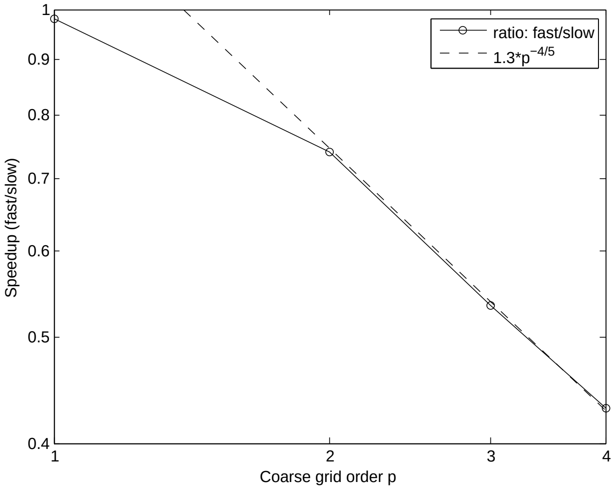


Figure 7.8: Speedup: ratio of fast runtime over slow runtime as a function of coarse grid order p (symmetric indefinite case)

of course that the diagonal block $B_{22} - B_{21}U_{11}^{-1}\tilde{B}_{12}$ is nonsingular). Should either of these tests fail, the algorithm can be restarted by combining blocks $1, \ldots, k$ into a single block. Should these restarts occur with any frequency (so that the performance degrades to that of the slow algorithm in Figure 7.7) we will investigate more robust alternatives like the incremental pivoting strategy described in [36].

Chapter 8

Concluding Remarks

We have presented an algorithm for fully automatic *hp*-adaptivity for finite element approximations of elliptic and Maxwell boundary value problems in three dimensions. The algorithm automatically generates a sequence of coarse grids, and a corresponding sequence of fine grids, such that the energy norm of the error decreases exponentially with respect to the number of degrees of freedom in either sequence. At each step, a discrete optimization algorithm is employed to determine the refinements for the current coarse grid such that the projection-based interpolation error for the current fine grid solution decreases with an optimal rate with respect to the number of degrees of freedom added by the refinement. The refinements are restricted only by the requirement that the resulting mesh is at most 1-irregular, but they may be anisotropic in both element size h and order of approximation p. Exponential convergence was demonstrated numerically for a wide variety of model problems from acoustic and electromagnetic scattering and it was found that the method is particularly suited to the automatic resolution of problems truncated by a perfectly matched layer. We presented three critical aspects of the implementation, namely an efficient implementation of sum factorization, several efficient interfaces to the direct multi-frontal solver MUMPS, and some fast direct solvers for the computation of a sequence of nested projections.

We believe this work provides an ample foundation for future research on *hp*-adaptivity and would like to conclude with a short discussion of some of the possibilities. The first of these is to build directly on the presented work by completing the implementation of the presented infinite elements and perfectly matched layers for electromagnetics. This would enable a much broader range of scattering computations and (hopefully) would strengthen our claim of exponential convergence.

A more ambitious project is the extension of the implementation to handle all types of elements, including tetrahedra, prisms and pyramids. Removing the restriction to hexahedra would greatly ease the generation of initial meshes for complicated geometries. Moreover, we hope to complete the implementation for the entire exact sequence, creating a single code for *hp*-adaptivity in H^1, $\boldsymbol{H}(\text{curl})$, $\boldsymbol{H}(\text{div})$ and L^2. Such a code would enable a consistent *hp*-adaptivity for coupled problems, and extend the applicability of the method to include the multi-physics applications of ever increasing interest to scientists and engineers. We believe that for the most challenging applications, it will be critical to incorporate the material data into the routines that compute the projection-based interpolant and evaluate the projection error. Moreover, a consistent implementation of goal-oriented *hp*-adaptivity will be essential for such problems.

The meshes generated by our algorithm have a natural hierarchical structure that is currently discarded in the process of interfacing with available direct solvers. A concurrent project with R. van de Geijn and V. Eijkhout will develop a library of solvers that avoid this “flattening” of the matrix. These solvers will have full knowledge of the hierarchical structure of the mesh behind the matrix, and will respect this hierarchy during the factorization process. This should greatly reduce the cost of local refinement since the factorization will only need to be updated locally as well.

Bibliography

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions.* Dover, New York, 1972.

[2] P. R. Amestoy, I. S. Duff, J. Koster, and J. Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, 23(1):15–41, 2001.

[3] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods in Appl. Mech. Eng.*, 184:501–520, 2000.

[4] P. R. Amestoy, A. Guermouche, J. Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[5] I. Babuška and M. R. Dorr. Error estimates for the combined h and p version of the finite element method. *Numer. Math.*, 37:252–277, 1981.

[6] I. Babuška, R. B. Kellogg, and J. Pitkäranta. Direct and inverse error estimates for finite elements with mesh refinement. *Numer. Math.*, 33:447–471, 1979.

[7] I. Babuška and M. Suri. The optimal convergence rate of the p-version of the finite element method. *SIAM J. Numer. Anal.*, 24(4):750–776, 1987.

[8] I. Babuška, B. A. Szabo, and I. N. Katz. The p-version of the finite element method. *SIAM J. Numer. Anal.*, 18(3):515–545, 1981.

[9] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.

[10] J.-P. Bérenger. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 127(2):363–379, 1996.

[11] P. Bettess. *Infinite Elements*. Penshaw Press, 1992.

[12] J. Bramble and J. Pasciak. Analysis of a finite element PML approximation for the three dimensional time-harmonic Maxwell problem. *Math. Comp.*, 2007.

[13] J. Bramble and J. Pasciak. Analysis of a finite PML approximation for the three dimensional time-harmonic Maxwell and acoustic scattering problems. *Math. Comp.*, 2007.

[14] W. Cao and L. Demkowicz. Optimal error estimate of a projection based interpolation for the p-version approximation in three dimensions. *Comput. Math. Appl.*, 50:359–366, 2005.

[15] W. Cecot, W. Rachowicz, and L. Demkowicz. An hp-adaptive finite element method for electromagnetics. Part 3: A three-dimensional infinite element for Maxwell's equations. *Int. J. Numer. Meth. Engng.*, 57:899–921, 2003.

[16] W. Chew and W. Weedon. A 3D perfectly matched medium for modified Maxwell's equations with streched coordinates. *Microwave Opt. Techno. Lett.*, 13(7):599–604, 1994.

[17] L. Demkowicz. Finite element methods for maxwell equations. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Me-*

chanics, volume 1 Fundamentals, chapter 26, pages 723–737. John Wiley & Sons, 2004.

[18] L. Demkowicz. Projection-based interpolation. In *Transactions on Structural Mechanics and Materials*, volume 302. Cracow University of Technology Publications, Cracow, 2004. see also ICES Report 04–03, UT Austin.

[19] L. Demkowicz. Fully automatic *hp*-adaptivity for Maxwell's equations. *Comput. Methods Appl. Mech. Engrg.*, 194:605–624, 2005.

[20] L. Demkowicz. Polynomial exact sequences and projection-based interpolation with application to Maxwell equations. Technical Report 06–12, ICES, UT Austin, 2006.

[21] L. Demkowicz, A. Bajer, W. Rachowicz, and K. Gerdes. 3D *hp*-adaptive finite element package. Fortran 90 implementation (`3dhp90`). Technical Report 99–29, TICAM, UT Austin, 1999.

[22] L. Demkowicz and A. Buffa. H^1, $H(\mathrm{curl})$ and $H(\mathrm{div})$-conforming projection-based interpolation in three dimensions: Quasi-optimal *p*-interpolation estimates. *Comput. Methods Appl. Mech. Engrg.*, 194:267–296, 2005.

[23] L. Demkowicz, P. Monk, L. Vardapetyan, and W. Rachowicz. De Rham diagram for *hp* finite element spaces. *Comput. Math. Appl.*, 39(7–8):29–38, 2000.

[24] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy. Toward a universal *hp* adaptive finite element strategy. Part 1: Constrained approximation and data structure. *Comput. Methods Appl. Mech. Engrg.*, 77:79–112, 1989.

[25] L. Demkowicz and M. Pal. An infinite element for Maxwell's equations. *Comput. Methods Appl. Mech. Engrg.*, 164:77–94, 1998.

[26] L. Demkowicz, D. Pardo, and W. Rachowicz. 3D *hp*-adaptive finite element package (3Dhp90). Version 2.0: The ultimate (?) data structure for three-dimensional, anisotropic *hp* refinements. Technical Report 02–24, TICAM, UT Austin, 2002.

[27] L. Demkowicz, W. Rachowicz, and Ph. Devloo. A fully automatic *hp*-adaptivity. *J. Sci. Comput.*, 17(1–4):127–155, 2002.

[28] L. Demkowicz and J. Shen. A few new (?) facts about infinite elements. *Comput. Methods Appl. Mech. Engrg.*, 195:3572–3590, 2006.

[29] L. Demkowicz and L. Vardapetyan. Modeling of electromagnetic absorption/scattering problems using *hp*-adaptive finite elements. *Comput. Methods Appl. Mech. Engrg.*, 152:103–124, 1998.

[30] Ph. Devloo, J. T. Oden, and P. Pattani. An *h-p* adaptive finite element method for the numerical simulation of compressible flow. *Comput. Methods Appl. Mech. Engrg.*, 70(2):203–235, 1988.

[31] P. Frauenfelder and Ch. Lage. Concepts - An object oriented software package for partial differential equations. *M2AN*, 36(5):937–951, 2002.

[32] W. Gui and I. Babuška. The *h*, *p* and *h-p* versions of the finite element method in 1 dimension - part i. the error analysis of the *p*-version, part ii. the error analysis of the *h*- and *h-p*-versions, part iii. the adaptive *h-p* version. *Numer. Math.*, 49:577–683, 1986.

[33] B. Guo and I. Babuška. The *h-p* version of the finite element method - part 1: The basic approximation results, part 2: General results and applications. *Comput. Mech.*, 1:21–41 and 203–220, 1986.

[34] F. Ihlenberg. *Finite Element Analysis of Acoustic Scattering*, volume 132 of *Applied Mathematical Sciences*. Springer, 1998.

[35] D. C. Jenn. *Radar and Laser Cross Section Engineering.* Education Series. American Institute of Aeronautics and Astronautics, Inc., 1995.

[36] T. Joffrain, E. S. Quintana-Orti, and R. van de Geijn. Updating an LU factorization and its application to scalable out-of-core. *SIAM J. Sci. Comput.*, 2006. submitted.

[37] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods for CFD.* Oxford University Press, 1999.

[38] J. M. Melenk, K. Gerdes, and C. Schwab. Fully discrete *hp* finite elements: fast quadrature. *Comput. Meth. Appl. Mech. Engrg.*, 190:4339–4364, 2001.

[39] C. Michler, L. Demkowicz, J. Kurtz, and D. Pardo. Improving the performance of perfectly matched layers by means of *hp*-adaptivity. Technical Report 06–17, ICES, UT Austin, 2006.

[40] J. T. Oden, L. Demkowicz, W. Rachowicz, and T. A. Westermann. Toward a universal *hp* adaptive finite element strategy, Part 2: A posteriori error estimation. *Comput. Methods Appl. Mech. Engrg.*, 77:113–180, 1989.

[41] D. Pardo. *Integration of hp-adaptivity with a two grid solver: applications to electromagnetics.* PhD thesis, The University of Texas at Austin, 2004.

[42] D. Pardo and L. Demkowicz. Fully automatic goal-oriented *hp*-adaptivity for elliptic problems. Technical report, Baker-Atlas, 2004.

[43] D. Pardo and L. Demkowicz. Integration of *hp*-adaptivity and a two grid solver for elliptic problems. *Comput. Meth. Appl. Mech. Engrg.*, 195(7–8):674–710, 2006.

[44] D. Pardo, L. Demkowicz, and J. Gopalakrishnan. Integration of *hp*-adaptivity

and a two grid solver for electromagnetic problems. *Comput. Meth. Appl. Mech. Engrg.*, 195(19–22):2533–2573, 2006.

[45] M. Paszynski and L. Demkowicz. Parallel, fully automatic *hp*-adaptive 3D finite element package. Technical Report 05–33, ICES, UT Austin, 2005.

[46] M. Paszynski, J. Kurtz, and L. Demkowicz. Parallel, fully automatic *hp*-adaptive 2D finite element package. *Comput. Methods Appl. Mech. Engrg.*, 195:711–741, 2006.

[47] W. Rachowicz and L. Demkowicz. An *hp*-adaptive finite element method for electromagnetics. Part 1: Data structure and constrained approximation. *Comput. Methods Appl. Mech. Engrg.*, 187:307–337, 2000.

[48] W. Rachowicz and L. Demkowicz. An *hp*-adaptive finite element method for electromagnetics. Part 2: A 3D implementation. *Int. J. Numer. Meth. Engng.*, 53:147–180, 2002.

[49] W. Rachowicz, J. T. Oden, and L. Demkowicz. Toward a universal *hp* adaptive finite element strategy, Part 3: Design of *hp* meshes. *Comput. Methods Appl. Mech. Engrg.*, 77:181–212, 1989.

[50] W. Rachowicz, D. Pardo, and L. Demkowicz. Fully automatic *hp*-adaptivity in three dimensions. *Comput. Methods Appl. Mech. Engrg.*, 195:4816–4842, 2006.

[51] W. Rachowicz and A. Zdunek. An *hp*-adaptive finite element method for scattering problems in computational electromagnetics. *Int. J. Numer. Meth. Engng.*, 62(9):1226–1249, 2005.

[52] J. Schöberl. Start Project Y-192 on 3D *hp*-Finite Elements: Fast Solvers and Adaptivity. `http://www.hpfem.jku.at`.

[53] J. Schöberl. An advancing front 2D/3D-mesh generador based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997.

[54] Ch. Schwab. *p- and hp- Finite Element Methods.* Oxford University Press, 1998.

Vita

Jason Patrick Kurtz was born in Lancaster, PA on July 20, 1979, the son of Mervin Weaver Kurtz and Marilyn Elaine Kurtz, and younger brother of Geoffrey Frederick Kurtz. In August 1996 he enrolled in The Clarkson School, an early admission program at Clarkson University in Potsdam, NY. He received the degree of Bachelor of Science in Mathematics from Clarkson University in May 2000, and Master of Science in Mathematics in December 2002. He moved to Texas in June 2002 and enrolled in the Computational and Applied Mathematics (CAM) PhD. program in the Institute for Computational Engineering and Sciences (ICES) at the University of Texas at Austin. From August 2002 to July 2005 he was a recipient of the CAM Fellowship.

Permanent Address: 5106 N. Lamar Blvd, #244
Austin, TX 78751

This dissertation was typeset with LaTeX 2ε[1] by the author.

---

[1] LaTeX 2ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.