Anisotropic Adaptive Refinement For Discontinuous Galerkin Methods

A thesis submitted for the degree of

Doctor of Philosophy

at the University of Leicester

by

Edward John Cumes Hall

Department of Mathematics

University of Leicester

July 2007

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Anisotropic Adaptive Refinement For Discontinuous Galerkin Methods

Edward J. C. Hall

We consider both the *a priori* and *a posteriori* error analysis and *hp*–adaptation strategies for discontinuous Galerkin interior penalty methods for second–order partial differential equations with nonnegative characteristic form on anisotropically refined computational meshes with anisotropically enriched polynomial degrees. In particular, we discuss the question of error estimation for linear target functionals, such as the outflow flux and the local average of the solution, exploiting duality based arguments.

The *a priori* error analysis is carried out in two settings. In the first, full orientation of elements is allowed but only (possibly high-order) isotropic polynomial degrees considered; our analysis, therefore, extends previous results, where only finite element spaces comprising piecewise linear polynomials were considered, by utilizing techniques from tensor analysis. In the second case, anisotropic polynomial degrees are allowed, but the elements are assumed to be axiparallel; we thus apply previously known interpolation error results to the goal-oriented setting.

Based on our *a posteriori* error bound we first design and implement an adaptive anisotropic *h*–refinement algorithm to ensure reliable and efficient control of the error in the prescribed functional to within a given tolerance. This involves exploiting both local isotropic and anisotropic mesh refinement, chosen on a competitive basis requiring the solution of local problems. The superiority of the proposed algorithm in comparison with a standard *h*–isotropic mesh refinement algorithm and a Hessian based *h*–anisotropic adaptive procedure is illustrated by a series of numerical experiments. We then describe a fully *hp*–adaptive algorithm, once again using a competitive refinement approach, which, numerical experiments reveal, offers considerable improvements over both a standard *hp*–isotropic refinement algorithm and an *h*–anisotropic/*p*–isotropic adaptive procedure.

In memory of my grandma, Ivy Nicks, 1914-2007.

She always showed great interest in my progress and offered me much encouragement.
I wish she were able to share in my success; she will be dearly missed.

Acknowledgements

An achievement such as a PhD can never be the work of just one individual, as such, I would like to pay homage to the following people.

Firstly let me acknowledge the support of Paul Houston (my 'Real' supervisor) throughout the PhD. It is hard to imagine having a better advisor; after all, how many buy their students laptops out of their own grant money? Even during times of slow progress Paul remained positive and confident in my abilities and for this I am very grateful.

Thanks also to my 'Imaginary' supervisor, Manolis Georgoulis, for being a great source of ideas, for bending my mind and for making a mean feta pie! I must also mention my fellow postgraduate colleagues Rob Brownlee, David Hunt, Jonathan Crofts and Terhemen Aboiyar for making my time in Leicester enjoyable with interesting mathematical and non-mathematical (Big Brother, *etc*) discussions. In a similar vein, thanks to Paola Antonietti for brightening up the solemnity of the office in Nottingham.

My gratitude must also go to the EPSRC and the University of Leicester for funding my study and allowing visits to such exotic places as Chile, Belgium and .... Dundee!

Special thanks to my parents for their support and financial backing, not only through the PhD, but also for the last 24 years of education. Without their continued encouragement I wouldn't be in this position today. Thank you also to Alan and Janet Jackson for offering me a place to stay during the move from Leicester to Nottingham and for countless food pack-ups.

Last but by no means least, 'Thank you' Louise for putting up with me during the stressful writing up period, for keeping me sane and for stopping me becoming a boring 'maths geek', well, 2 out of 3 ain't bad!

Contents

Chapter 1

Introduction

The mathematical modeling of physical phenomena often leads to the formation of ordinary or partial differential equations (PDEs) equipped with appropriate boundary/initial conditions. In most cases these differential equations cannot be solved analytically, and so numerical approximations must be constructed. Science demands that these approximations be computed relatively speedily and to a high level of accuracy, for increasingly more complex problems. Improvements in hardware performance has gone a long way to meeting these demands, but enhancement of the current numerical methods still has an important role to play; as such this thesis is concerned with techniques for improving the efficiency of a certain class of numerical methods, namely *Discontinuous Galerkin Finite Element Methods*.

Finite Element Methods (FEMs) and Finite Volume Methods (FVMs) have emerged as the leading contenders for the numerical solution of PDEs, primarily due to their ability to cope with complicated geometries and the sound mathematical theory underpinning them. In both cases the domain of interest is divided into a mesh consisting of small regions (called 'elements' for FEMs and 'volumes' for FVMs); on each of these subregions an approximation to the solution is then computed. Typically, standard Galerkin FEMs use polynomials to represent the solution on each element, while maintaining continuity across element boundaries; in contrast FVMs attach a constant value of the solution to each volume, hence continuity cannot be maintained and information is passed between cells by numerical flux functions. Evidently, decreasing the size of the subregions employed is likely

to lead to improved accuracy in the solution, likewise, increasing the polynomial degree for FEMs should have the same effect, preferably this should should be done by means of an automatic adaptive strategy. In general, the construction of such an adaptive strategy involves three key steps: the derivation of a sharp *a posteriori* error bound for the finite element approximation of the partial differential equation under consideration, which is then used as a stopping criterion to terminate the adaptive algorithm once the desired level of accuracy has been achieved; the design of an appropriate refinement indicator to identify regions in the computational domain where the error is locally large; and the design of the corresponding mesh-modification/adaptive algorithm which is capable of automatically selecting the local mesh width h and/or the local degree p of the approximating polynomial in order to deliver reliable and efficient control of the discretization error. In this thesis we focus primarily on the design of the mesh-modification/adaptive algorithm.

1.1 Discontinuous Galerkin (DG) Finite Elements Methods

In recent years a certain class of FEMs, called *Discontinuous Galerkin Methods* have gained popularity. They can effectively be viewed as hybrid FEMs and FVMs, in that higher order polynomials can be used, but no inter-element continuity is imposed; in this setting, as for FVMs, numerical flux functions are used to pass information between cells.

The first DG method was proposed by Reed & Hill [112] in the 1970s for the numerical solution of the neutron-transport equation; a first-order hyperbolic problem. The first analysis of the method was undertaken by LeSaint and Raviart [97] using Fourier techniques. At the same time, but independently, Discontinuous Galerkin methods were being developed for the approximation of second-order elliptic equations, although the DG name did not appear until later. In this case they were known as *penalty methods*; the name stemming from the inclusion of certain terms aimed at penalizing the discontinuities. Babuška [14] based the first penalty method on the preceding work by Nitsche [105], where boundary conditions were enforced weakly for elliptic problems. Unfortunately, this method suffered from consistency problems and it was not until later that Wheeler [136] and Arnold [11] resolved these consistency issues. Since then a multitude of DG

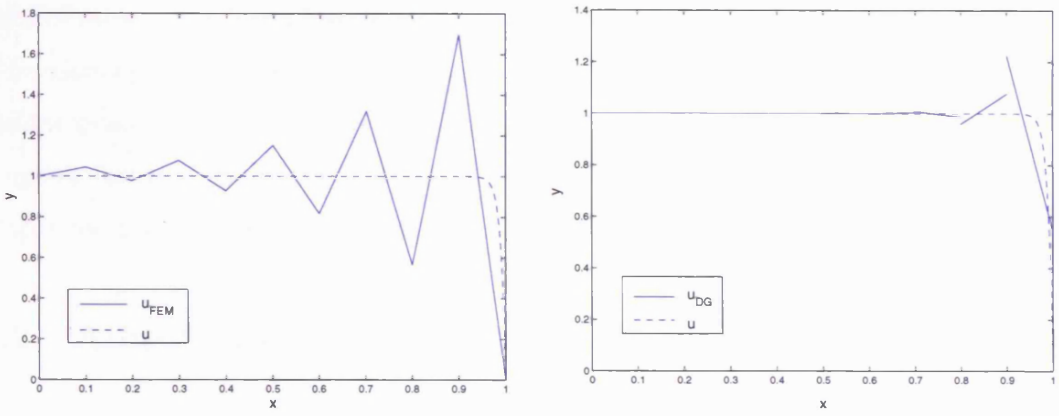
formulations have arisen for elliptic equations: see Arnold, Brezzi, Cockburn and Marini [12, 13] for a recent review. Of particular interest are the interior-penalty (IP) methods of Rivière, Wheeler, and Girault [113, 114, 115] and Houston, Schwab and Süli [76, 127], the symmetric version of which will be employed for the purposes of this thesis. A full review of the development of DG methods can be found in the review article Cockburn, Karniadakis and Shu [39].

DG methods offer a number of advantages over standard FEMs. For example, it is well documented that applying an FEM to a problem with, say, boundary or interior layers can result in a solution with non-physical oscillations when too few degrees of freedom are available to resolve the layer. To overcome this, a stabilization method has to be introduced, for example, streamline-diffusion stabilization [32, 85, 86] or bubble stabilization [29, 30, 31]. In contrast, DG methods do not suffer from this lack of stability; it appears that the discontinuous nature allows the oscillations to be damped by numerical dissipation; see [76, 127]. As an example, consider the simple one-dimensional convection-diffusion problem

$$\begin{aligned} -\varepsilon u'' + u' &= 0, \\ u(0) &= 1, \\ u(1) &= 0, \end{aligned}$$

where $\varepsilon = 0.01$, on the interval $[0, 1]$. Figures 1.1(a) and 1.1(b) show a non-stabilized FEM solution and a DG solution, respectively, computed on meshes too coarse to resolve the layer. Notice that the boundary layer causes global degradation of the FEM solution, whereas the DG solution is only locally disturbed.

DG methods also offer far greater flexibility in the mesh design compared with FEMs. FEMs require continuity to be enforced across element boundaries, hence if the mesh contains hanging nodes (vertices of one element occurring on the face of another, see Figure 1.2), special measures have to be taken to ensure the continuity holds. Possible solutions include removal of the hanging node by subdivision of neighbouring elements, or elimination of the unknowns corresponding to the hanging node by interpolation of the values at neighbouring nodes. However, for DG methods, continuity only holds weakly,



(a) (b)

Figure 1.1: (a) Instability of an FEM and (b) stability of a DG method for a boundary layer problem.

thus there can be multiple hanging nodes on a face without any computational difficulty. Similarly, varying polynomial degrees on adjacent elements is handled in a very simple manner by DG methods; for conforming FEMs the usual approach is to apply the minimal degree rule on the face and/or edges. DG methods, therefore, lend themselves particularly well to isotropic and anisotropic *hp*-adaptivity, which will be considered in this work.

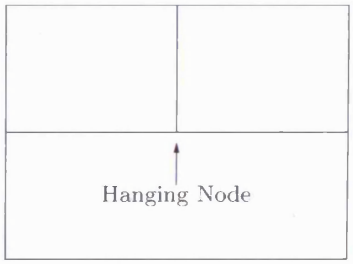


Figure 1.2: Example of a mesh with a hanging node.

Without the need for inter-element continuity a wider choice of basis functions for DG methods are available, as an example orthogonal basis functions are readily constructed, leading to diagonal mass matrices. Similarly, the lack of continuity reduces communication between elements, resulting in sparser matrices. Additionally, the method lends itself to

parallelization, which can help to negate the drawback of the increased number of degrees of freedom associated with the method, in comparison to standard conforming FEMs. For certain mixed problem, such as the Stokes problem, DG methods are stable for a wider range of function spaces than standard FEMs. DG methods also possess other useful properties, for example, unlike FEMs, many DG methods are locally conservative.

1.2 Anisotropy

Many solutions to PDEs exhibit *anisotropic* behaviour, where the solution varies rapidly in one direction, but not in orthogonal directions. Examples include boundary and interior layers in singularly perturbed convection-diffusion-reaction equations and shock waves arising in compressible fluid flows. In these cases it would seem reasonable for the mesh to display similar properties by making use of *anisotropic elements*; *i.e.*, elements which have a small mesh size in the direction where the solution changes rapidly but a larger mesh size in the orthogonal direction. For an element κ, let $\mathrm{diam}(\kappa)$ denote its diameter (*i.e.* the diameter of the element's circumcircle) and ρ_κ be the supremum of the diameters of all the balls contained within κ, then, loosely speaking, an element is anisotropic if

$$\frac{\mathrm{diam}(\kappa)}{\rho_\kappa} \gg 1,$$

the ratio of $\mathrm{diam}(\kappa)$ to ρ_κ being termed the aspect ratio. Compare this with isotropic elements, where

$$\frac{\mathrm{diam}(\kappa)}{\rho_\kappa} \approx 1.$$

Understanding the approximation properties of anisotropic elements is challenging; indeed, early interpolation results gave bounds in terms of only $\mathrm{diam}(\kappa)$ and as such the usefulness of anisotropic elements were overlooked. Sharper estimates, where the various length scales of the elements are included, are needed before the merits of anisotropic elements can be seen. The geometry of the elements needs careful consideration as well, indeed a *maximal angle condition* has often to be met if good approximations are to achieved; see, for example Babuška and Aziz [15]. A review of the techniques involved in studying anisotropic elements can be found in Thomas Apel's monograph [7]. In this thesis we

extend the anisotropic approximation techniques developed in Formaggia and Perotto [48] which precisely describe the anisotropy of the mesh. More specifically, we employ tools from tensor analysis (see De Lathauwer, Moor and Vandewalle [94]), along with local singular-value decompositions of the Jacobi matrix of the local elemental mappings, to derive directionally-sensitive bounds for arbitrary polynomial degree approximations, thus generalizing the ideas presented in [48], where only the case of approximation with conforming linear elements was considered. The advantages of this general approach are that the resulting interpolation bounds exploit the full spectral properties of the underlying (affine) element transformation, and are thereby independent of the choice of coordinate axes. Additionally, no *a priori* condition on the maximal angle of the computational mesh is required; indeed, numerical experiments presented in [48] clearly demonstrate that this approach leads to approximation bounds which show the correct asymptotic behaviour with respect to the maximal angle. We then use these interpolation estimates to carry out an *a priori* error analysis of an interior penalty DG method for second-order partial differential equations with nonnegative characteristic form on anisotropically refined computational meshes. In particular, we are concerned with the question of error estimation for certain linear target functionals of the analytical solution. Error control in this sense is particularly important in engineering applications; e.g. in fluid dynamics one may be concerned with calculating the lift and drag coefficients of a body immersed into a viscous fluid whose flow is governed by the Navier–Stokes equations. The lift and drag coefficients are defined as integrals, over the boundary of the body, of the stress tensor components normal and tangential to the flow, respectively. Similarly, in elasticity theory, the quantities of interest, such as the stress intensity factor or the moments of a shell or plate, are derived quantities. In acoustic and electromagnetic theory the quantity of interest is often the far–field pattern. Our results generalize those of Harriman *et al.* [61], where the underlying computational mesh was assumed to be shape–regular.

A posteriori estimation on anisotropic meshes is also not as straightforward as that on isotropic meshes. Indeed, the estimates for norms reviewed in [3, 135] and developed in [134] become unreliable on anisotropic meshes. Anisotropic *a posteriori* error estimates have been developed for conforming FEMs in a series of papers by Kunert [90, 91, 92]

and for DG methods in, for example, Creusé *et al.* [40]. However, these methods involve the introduction of a matching function, which determines how well the mesh is aligned with the solution, as such, the error indicators become useless if the mesh is not well matched. In the functional setting we employ the duality weighted residual approach advocated in Johnson *et al.* [46] and Becker & Rannacher [21], and further developed in, for example, [63, 77, 79]. In this case, the error indicator is actually equal to the true error in the underlying target functional, provided the analytical solution to an induced dual problem is known. Hence, the technique is applicable for both isotropic and anisotropic meshes. In general the analytical solution of the dual problem is not known and must be approximated. Numerical experimentation reveals effectivities of the error indicator approaching 1 on both isotropic and anisotropic meshes.

Isotropic mesh refinement is often carried out by simply identifying those elements in the computational mesh where the error is high, based on the *a posteriori* error estimator, and subdividing them into similarly shaped elements; see, for example, [77]. Unfortunately, the error indicators do not contain any information pertaining to the anisotropy of the solution and hence, alternate methods must be sought when attempting anisotropic refinement. In some cases the location of boundary and interior layers is known *a priori* and initial grids can be designed to resolve these features, for example, piecewise uniform grids and geometrically graded meshes; see, for example, [120] and [10], respectively. Where knowledge of the true solution is unknown beforehand, mesh refinement can be performed in the directions where an approximated solution varies rapidly. One popular method involves considering *a priori* error estimates for approximation by linear polynomials; here the estimates are based around the Hessian matrix of the solution, for example; see [22, 23, 43, 50, 83, 58, 5, 44]. To minimize these estimates, it can be shown, see Formaggia *et al.* [48, 49, 47], that an element should be orientated so that the primary left singular vector of the Jacobi matrix of the local elemental mapping is in the same direction as the eigenvector of the Hessian matrix with smallest absolute eigenvalue. The aspect ratio of the element should then be chosen to be the same as the square root of the ratio of the absolute values of the eigenvalues of the Hessian matrix. One approach to achieve well aligned meshes is to insist that all faces in the mesh have the same measure

in a metric induced from the Hessian: this can be done be means of local edge based operations, see [58], for example.

The Hessian strategy suffers from a number of drawbacks: these include assuming that the interpolation estimates are sharp and the solution has sufficient regularity. Further, the strategy has only been developed for approximation by piecewise linear polynomials. As such, in this thesis we develop an alternative strategy, based upon solving local primal and dual problems, together with a competitive refinement algorithm so that the anisotropic/isotropic subdivision of an element attaining the greatest reduction in error per degree of freedom is chosen. Numerical experiments reveal that, when applied to both elliptic and mixed elliptic/hyperbolic problems with boundary and interior layers, the new strategy can not only yield orders of magnitude reduction in the error for the same number of degrees of freedom when compared with isotropic refinement, but also offers significant improvements over a comparable Hessian based strategy. Indeed, the new strategy is seen to be robust for approximation by variable polynomial degrees, whereas the Hessian strategy typically fails for higher polynomial degrees.

Anisotropy need not only be introduced into the finite element space by means of elements with high aspect ratio. Indeed, anisotropic polynomials are also an option, *i.e.*, in directions of rapid variation of the underlying solution, a high polynomial degree can be used, while in perpendicular directions the polynomial order can be kept low. The approximation properties of such spaces has been considered by Georgoulis, see [52, 53], where the elements are assumed to be axiparallel. Using the *a priori* interpolation estimates developed in [52], we undertake the *a priori* error analysis for the target functional of interest and see that a mixture of anisotropic elements and anisotropic polynomial degrees can lead to great reductions in the computational cost required in obtaining accurate solutions. Motivated by these results we design a strategy for automatically deciding how to choose between anisotropic h–refinement and anisotropic p–enrichment and in which directions to perform these refinements/enrichments. Once again, the method is based upon the solution of local primal and dual problems and selecting the refinement/enrichment which gives rise to the largest error decrease per degree of freedom. Numerical experiments show that this new adaptive anisotropic hp–strategy is capable of yielding over

an order of magnitude reduction in the error for the same number of degrees of freedom when compared with a method utilizing adaptive anisotropic h-refinement with isotropic p-enrichment.

1.3 Outline

The aim of this work is to develop a robust adaptive strategy with the ability to produce meshes with the possibility of anisotropic elements and/or anisotropic polynomial degrees where required. In particular, we shall be concerned with designing optimal meshes for the evaluation of certain linear functionals of the solution, rather than, say, some norm of the error. In order to do this we first describe a model advection-diffusion-reaction problem in Chapter 2 and present some results concerning the existence and uniqueness of the underlying solution. We then describe the DG finite element space to which our numerical approximations shall belong, where anisotropic polynomial degrees and high aspect ratio elements are permitted. We then proceed by performing an interior penalty DG discretization of the problem and discussing the stability properties of the method.

The analysis of the method requires knowledge of the approximation properties of the finite element space; specifically, we shall need interpolation bounds for the L^2-projection operator. In Chapter 3 we state these error estimates on the reference element and describe how these bounds translate to physical anisotropic elements. In the case of isotropic polynomial degrees on non-axiparallel elements, we generalize the results of Formaggia and Perotto [48] where only piecewise linear elements were considered. To this end, we utilize some results from tensor analysis, specifically those of De Lathauwer, Moor and Vandewalle [94] concerned with tensor-matrix multiplications.

In Chapter 4 we develop anisotropic *a priori* error estimates for linear target functionals of the solution. This is achieved by first introducing an appropriate dual problem, from which we deduce an error representation formula for the error in the underlying target functional of interest. Following the analysis in Harriman *et al.* [61], we then apply the error estimates of Chapter 3 to obtain *a priori* error bounds. These bounds show that not only the anisotropy of the primal solution is important in the design of the mesh, but also

the anisotropy of the dual solution.

We turn our attention to adaptive h-refinement in Chapter 5. The chapter begins with a discussion on current *a posteriori* error techniques and how they can be used to drive an adaptive process. We then introduce duality based *a posteriori* estimates for linear target functionals, first developed by Becker and Rannacher [21]. Following this, we present a more thorough review of standard isotropic and anisotropic h-refinement strategies and show how our *a priori* error estimates of Chapter 4 can lead to a Hessian based approach and describe an algorithm to make use of these results. A discussion on the drawbacks of the Hessian approach then leads us to develop the new competitive refinement strategy.

Numerical experiments to test the performance of our new anisotropic strategy compared with the standard isotropic and Hessian based approaches are carried out in Chapter 6. Specifically, an elliptic problem with a steep boundary layer is considered in the first example and a mixed elliptic/hyperbolic problem with an interior layer is studied in the second. In both cases the new approach is seen to offer considerable improvements over the other techniques.

Chapter 7 is concerned with the development of an hp-adaptive strategy with anisotropic elements and anisotropic polynomial degrees. We first perform a further *a priori* error analysis for the case of axiparallel elements to motivate the use of anisotropic polynomial degrees, showing that, in areas where the primal and dual solutions are smooth, p-refinement is preferable to h-refinement. We discuss some techniques for determining where a solution is smooth before introducing a strategy to decide how to select when to increase the polynomial degree vector, be that isotropically or anisotropically. Once again these new methods are based around the solution of local primal and dual problems and choosing the refinement which gives rise to the largest error decrease per degree of freedom.

Some numerical experiments are then performed in Chapter 8; the first two examples are used to further motivate the need for both anisotropic h-refinement and anisotropic p-refinement, where *a priori* knowledge of the primal and dual solutions are used to design the finite element spaces. The last two examples are used to test the adaptive strategies developed in Chapter 7; once again an elliptic problem with a boundary layer and a

mixed elliptic/hyperbolic problem are used for this purpose. Comparisons with isotropic hp-adaptive and h–anisotropic/p–isotropic algorithms are very encouraging. Finally, we make some conclusions and discuss some avenues of future research in Chapter 9.

Chapter 2

Model Problem and Discontinuous Galerkin Discretization

Throughout this thesis we will consider the numerical solution of linear advection-diffusion-reaction problems. More precisely, we will concern ourselves with second-order *partial differential equations with nonnegative characteristic form*, which, under certain conditions, can be shown to be well-posed. In the following sections, after recalling some function space definitions, we define what is meant by the term *partial differential equation with nonnegative characteristic form* and give some results on the existence and uniqueness of solutions. We then proceed to construct a DG discretization of the problem and discuss its stability.

2.1 Sobolev Spaces

Before we introduce the model problem, it is first necessary to recall the definition of a Sobolev space, based around the Lebesgue space, which we define below.

Definition 2.1.1. *Let Ω be an open domain in $\mathbb{R}^d$, $d \geq 1$, $L^p(\Omega)$, $p \in [1, \infty]$ is defined as the Lebesgue space of real-valued functions with norm defined by*

$$\|u\|_{L^p(\Omega)} := \begin{cases} \left(\int_\Omega |u(x)|^p \mathrm{d}x\right)^{\frac{1}{p}}, & 1 \leq p < \infty. \\ \operatorname{ess\,sup}\{|u(x)| : x \in \Omega\}, & p = \infty. \end{cases}$$

The space $L^2(\Omega)$, replete with the standard inner product will be of particular importance, because in this case we have a Hilbert space.

For a multi-index, $\alpha = (\alpha_1, \ldots, \alpha_n)$, where the α_i, $i = 1, \ldots, n$, are nonnegative, its length $|\alpha|$ is defined as

$$|\alpha| = \sum_{i=1}^{n} \alpha_i.$$

The $|\alpha|$th order differential operator can then be concisely defined as ∂^α, where

$$\partial^\alpha = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}}.$$

We now define the standard Sobolev space of integer index.

Definition 2.1.2. *Let s be a nonnegative integer, $p \in [1, \infty]$, $\alpha = (\alpha_1, \ldots, \alpha_n)$ a multi-index and Ω an open domain in $\mathbb{R}^d$, $d \geq 1$. The Sobolev space $W_p^s(\Omega)$ on Ω is defined by*

$$W_p^s(\Omega) := \{u \in L^p(\Omega) : \partial^\alpha u \in L^p(\Omega) \text{ for } |\alpha| \leq s\},$$

with associated norm $\|\cdot\|_{W_p^s(\Omega)}$ and seminorm $|\cdot|_{W_p^s(\Omega)}$, respectively :

$$\|u\|_{W_p^s(\Omega)} := \left(\sum_{|\alpha| \leq s} \|\partial^\alpha u\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}}, \quad |u|_{W_p^s(\Omega)} := \left(\sum_{|\alpha| = s} \|\partial^\alpha u\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}},$$

for $p \in [1, \infty)$ and

$$\|u\|_{W_\infty^s(\Omega)} := \max_{|\alpha| \leq s} \|\partial^\alpha u\|_{L^\infty(\Omega)}, \quad |u|_{W_\infty^s(\Omega)} := \max_{|\alpha| = s} \|\partial^\alpha u\|_{L^\infty(\Omega)},$$

for $p = \infty$. Here, s is referred to as the Sobolev index of the function space. In the special case $p = 2$, the space $W_2^s(\Omega)$ with standard inner product is a Hilbert space and we use the notation $W_2^s(\Omega) \equiv H^s(\Omega)$.

Spaces with negative and fractional Sobolev index can also be introduced by means of duality and function-space interpolation arguments, respectively: see, for example, [1, 24, 28, 101, 103, 132] for details. Anisotropic Sobolev spaces have also been used: see for example [52], where different Sobolev indices are considered in each coordinate direction separately.

2.2 Partial Differential Equations with Nonnegative Characteristic Form

Let us consider the following advection-diffusion-reaction equation on Ω, a bounded open polyhedral domain in $\mathbb{R}^d$, $d \geq 2$:

$$\mathcal{L}u \equiv -\nabla \cdot (a\nabla u) + \nabla \cdot (\mathbf{b}u) + cu = f \quad \text{in } \Omega. \tag{2.2.1}$$

Here, $f \in L^2(\Omega)$ and $c \in L^\infty(\Omega)$ are real-valued, $\mathbf{b} = \{b_i\}_{i=1}^d$ is a vector function whose entries, b_i, are Lipschitz continuous real-valued functions on $\bar{\Omega}$ and $a = \{a_{ij}\}_{i,j=1}^d$ is a symmetric matrix whose entries a_{ij} are bounded, piecewise continuous real-valued functions defined on $\bar{\Omega}$, with

$$\zeta^T a(x)\zeta \geq 0 \quad \forall \zeta \in \mathbb{R}^d \text{ and } \forall x \in \bar{\Omega}. \tag{2.2.2}$$

Under the above assumptions (2.2.1) is referred to as a *partial differential equation with nonnegative characteristic form*. This class encompasses a large number of types of equation; they include second-order elliptic or parabolic, and first-order hyperbolic problems. As such they offer many physical applications, problems of this type occurring commonly in mathematical biology, financial mathematics and, in the non-linear case non-Newtonian fluid flow (p-Laplacian). A key feature of these types of equation is that they allow for a change of type within the computational domain, meaning they can be used as a prototype for more complicated compressible fluid flow problems, where this behaviour is common.

Writing Γ to denote the union of the $(d-1)$-dimensional faces of Ω and $\mathbf{n}(x) = \{n_i(x)\}_{i=1}^d$ the unit outward normal vector to Γ at $x \in \Gamma$, we introduce the *Fichera function*, $\mathbf{b} \cdot \mathbf{n}$, and define

$$\begin{aligned}
\Gamma_0 &= \{x \in \Gamma : \mathbf{n}(x)^\top a(x)\mathbf{n}(x) > 0\}, \\
\Gamma_- &= \{x \in \Gamma \backslash \Gamma_0 : \mathbf{b}(x) \cdot \mathbf{n}(x) < 0\}, \\
\Gamma_+ &= \{x \in \Gamma \backslash \Gamma_0 : \mathbf{b}(x) \cdot \mathbf{n}(x) \geq 0\}.
\end{aligned}$$

For obvious reasons we will refer to Γ_- and Γ_+ as the inflow and outflow boundaries, respectively. It is clear that Γ_0, Γ_-, and Γ_+ are disjoint sets and $\Gamma = \Gamma_0 \cup \Gamma_- \cup \Gamma_+$. If Γ_0 is nonempty we further split it into disjoint subsets Γ_D and Γ_N, ensuring Γ_D is nonempty.

We are now in a position to impose suitable boundary conditions on (2.2.1). On $\Gamma_D \cup \Gamma_-$ we set Dirichlet boundary conditions, while on Γ_N Neumann conditions are enforced, i.e.,

$$u = g_D \quad \text{on } \Gamma_D \cup \Gamma_-, \text{ and } (a\nabla u) \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N. \tag{2.2.3}$$

2.3 Existence and Uniqueness of the Solution

We now review some results concerning the existence and uniqueness of the analytical solution to our model problem (2.2.1), (2.2.3). For simplicity, we assume $g_D \equiv g_N \equiv 0$. Existence and uniqueness of solutions (in the weak sense) were considered by Oleĭnik and Radkevič [108] for problems with homogeneous Dirichlet boundary conditions under certain regularity assumptions; see also, Houston, Schwab and Süli [75], where analogous results were derived under weaker regularity constraints. In [76] the results were further extended to problems with mixed Dirichlet-Neumann boundary conditions; we present these results below, after some initial preparation. We define the space

$$\mathcal{V} := \{v \in H^1(\Omega) : v(x) = 0 \ \ \forall x \in \Gamma_D\},$$

and let $\mathcal{H}$ be the closure of $\mathcal{V}$ in $L^2(\Omega)$ with respect to the norm $\|\cdot\|_{\mathcal{H}} := \sqrt{(\cdot,\cdot)_{\mathcal{H}}}$, where $(\cdot,\cdot)_{\mathcal{H}}$ is the inner product defined by

$$(w,v)_{\mathcal{H}} := (a\nabla w, \nabla v) + (w,v) + (w,v)_{\Gamma_- \cup \Gamma_+ \cup \Gamma_N}.$$

Here, $(\cdot,\cdot)$ and $(\cdot,\cdot)_\gamma$ are the inner products defined, respectively, by

$$(w,v) = \int_\Omega wv\mathrm{d}x \quad \text{and} \quad (w,v)_\gamma = \int_\gamma |\mathbf{b}\cdot\mathbf{n}|wv\mathrm{d}s.$$

Thereby, $\mathcal{H}$ is a Hilbert space. We now define the bilinear form $B(\cdot,\cdot) : \mathcal{H} \times \mathcal{V} \to \mathbb{R}$ by

$$B(w,v) = (a\nabla w, \nabla v) - (w, \mathbf{b}\cdot\nabla v) + (cw, v) + (w,v)_{\Gamma_+ \cup \Gamma_N}$$

and the linear functional $\ell : \mathcal{V} \to \mathbb{R}$ by

$$\ell(v) = (f, v).$$

We shall say that $u \in \mathcal{H}$ is a weak solution to the boundary value problem (2.2.1), (2.2.3) with homogeneous boundary conditions if

$$B(u,v) = \ell(v) \quad \forall v \in \mathcal{V}. \tag{2.3.1}$$

We note that (2.3.1) stems from (2.2.1) by multiplying by $v \in \mathcal{V}$, integrating by parts and applying the homogeneous boundary conditions. We are now in a suitable position to state the following theorem.

Theorem 2.3.1. *Suppose that $a, \mathbf{b}$ and c of $\mathcal{L}$ are as in Section 2.2 and furthermore that $\mathbf{b} \cdot \mathbf{n} \geq 0$ on Γ_N and there exists a positive constant γ_0 such that*

$$c(x) + \frac{1}{2}\nabla \cdot \mathbf{b}(x) \geq \gamma_0 \quad a.e\ x \in \Omega. \tag{2.3.2}$$

Then, for every $f \in L^2(\Omega)$ there exists $u \in \mathcal{H}$ such that (2.3.1) holds. Moreover, there exists a Hilbert subspace $\mathcal{H}'$ of $\mathcal{H}$ such that $u \in \mathcal{H}'$ and u is the unique element of $\mathcal{H}'$ such that (2.3.1) holds.

Proof. See Houston and Süli [80]. ■

2.4 Meshes

We shall now describe how we construct a computational mesh over the domain Ω. Firstly, we need the following the definition.

Definition 2.4.1. *Let κ and κ' be open sets in $\mathbb{R}^d$, $d \geq 1$. A bijection $Q : \kappa \to \kappa'$ is termed a C^s-diffeomorphism for $s \geq 1$, if*

1. *Q and Q^{-1} are continuous on $\bar{\kappa}$ and $\bar{\kappa}'$, respectively.*
2. *their derivatives of order 1 through to s are bounded and continuous on $\bar{\kappa}$ and $\bar{\kappa}'$, respectively.*

Let $\mathcal{T}_h = \{\kappa\}$ be a subdivision of the (polygonal) domain Ω into disjoint open element domains κ, constructed through the use of the mapping $Q_\kappa \circ F_\kappa$, where $F_\kappa : \hat{\kappa} \to \tilde{\kappa}$ is an affine mapping from the reference element $\hat{\kappa}$ to $\tilde{\kappa}$, and $Q_\kappa : \tilde{\kappa} \to \kappa$ is a C^1–diffeomorphism from $\tilde{\kappa}$ to the physical element κ. Here, we shall assume that $\hat{\kappa}$ is either the hypercube $(-1,1)^d$ or the unit d–simplex; in the latter case Q_κ is typically the identity operator, unless curved elements are employed. The mapping F_κ defines the size and orientation of the element κ, while Q_κ defines the shape of κ, without any significant rescaling, or indeed

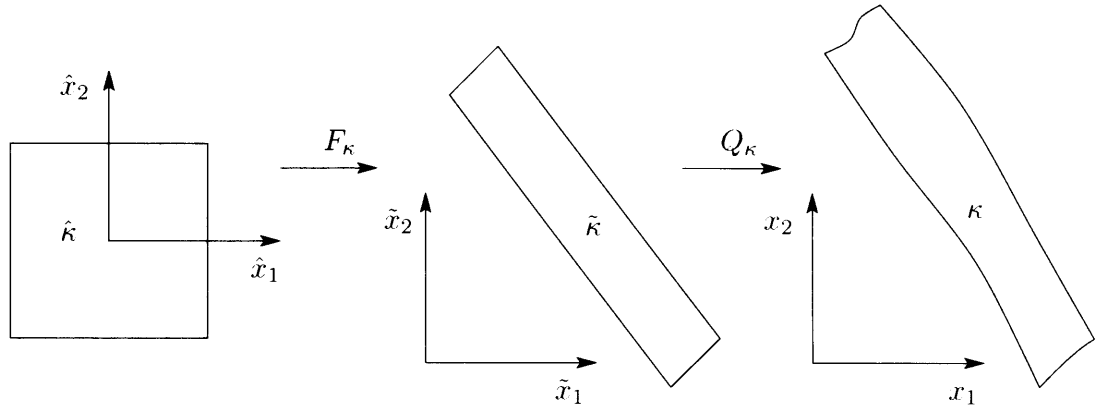


Figure 2.1: Construction of the element mapping via the composition of an affine mapping F_κ and a C^1–diffeomorphism Q_κ.

change of orientation, cf. Figure 2.1 for the case when $d = 2$ and $\hat{\kappa} = (-1, 1)^2$. With this in mind, we assume that the element mapping Q_κ is close to the identity in the following sense: the Jacobi matrix J_{Q_κ} of Q_κ satisfies

$$C_1^{-1} \leq \|\det J_{Q_\kappa}\|_{L_\infty(\kappa)} \leq C_1, \quad \|J_{Q_\kappa}^{-\top}\|_{L_\infty(\kappa)} \leq C_2, \quad \|J_{Q_\kappa}^{-\top}\|_{L_\infty(\partial\kappa)} \leq C_3 \tag{2.4.1}$$

for all κ in $\mathcal{T}_h$ uniformly throughout the mesh for some positive constants C_1, C_2, and C_3, where we denote by $\partial\kappa$ the union of $(d-1)$-dimensional open faces of κ. This assumption will be important as our error estimates will be expressed in terms of Sobolev norms over the element domains $\tilde{\kappa}$, in order to ensure that only the scaling and orientation introduced by the affine element maps F_κ are present in the analysis. Writing m_κ, $m_{\tilde{\kappa}}$, and $m_{\hat{\kappa}}$ to denote the d–dimensional measure of the elements κ, $\tilde{\kappa}$, and $\hat{\kappa}$, respectively, the above condition (2.4.1) implies that there exists a positive constant C_4 such that

$$C_4^{-1} m_{\tilde{\kappa}} \leq m_\kappa \leq C_4 m_{\tilde{\kappa}} \quad \forall \kappa \in \mathcal{T}_h. \tag{2.4.2}$$

The above maps are assumed to be constructed in such a manner as to ensure that the union of the closure of the disjoint open elements $\kappa \in \mathcal{T}_h$ forms a covering of the closure of Ω, i.e., $\bar{\Omega} = \cup_{\kappa \in \mathcal{T}_h} \bar{\kappa}$. For a function v defined on $\kappa \in \mathcal{T}_h$, we write $\tilde{v} = v \circ Q_\kappa$ and $\hat{v} = \tilde{v} \circ F_\kappa$ to denote the corresponding functions on the elements $\tilde{\kappa}$ and $\hat{\kappa}$, respectively. Thereby, we have that $\hat{v} = v \circ Q_\kappa \circ F_\kappa$.

On the elements $\hat{\kappa}$ and $\tilde{\kappa}$ we define the gradient operators $\hat{\nabla}$ and $\tilde{\nabla}$, respectively, by

$$\hat{\nabla}\hat{v} = \left(\frac{\partial \hat{v}}{\partial \hat{x}_1}, \ldots, \frac{\partial \hat{v}}{\partial \hat{x}_d}\right)^\top \text{ and } \tilde{\nabla}\tilde{v} = \left(\frac{\partial \tilde{v}}{\partial \tilde{x}_1}, \ldots, \frac{\partial \tilde{v}}{\partial \tilde{x}_d}\right)^\top.$$

Remark 2.4.2. We note that a similar construction of the element mappings for general meshes consisting of curved quadrilateral elements has also been employed for both shape-regular and anisotropic meshes in the articles [75] and [53], respectively. The key difference in the current construction to that proposed in [53] is that here the element mapping F_κ contains information about both size and orientation of κ. In contrast, in the construction developed in [53] both orientation and shape information are included in Q_κ, while F_κ only contains information relating to the size of κ.

Remark 2.4.3. Within this construction we admit meshes with possibly hanging nodes; for simplicity, we shall suppose that the mesh $\mathcal{T}_h$ is 1-irregular, that is, each element face has at most one hanging node, cf. [75].

We define an *interior face* of $\mathcal{T}_h$ to be the non-empty $(d-1)$-dimensional interior of $\partial\kappa_i \cap \partial\kappa_j$, where κ_i and κ_j are two adjacent elements of $\mathcal{T}_h$; then we define Γ_{int} to be the union of all the interior faces of $\mathcal{T}_h$. Similarly, we define a *boundary face* of $\mathcal{T}_h$ to be the non-empty $(d-1)$-dimensional interior of $\partial\kappa \cap \Gamma$.

For a face $f \in \Gamma_{\text{int}}$, shared by elements κ_i and κ_j, where $i > j$, we let $\mathbf{n}_f$ denote the unit normal vector pointing from κ_i to κ_j. If f is a boundary face then we set $\mathbf{n}_f = \mathbf{n}$.

2.4.1 Broken Sobolev Spaces

In this section we introduce the notion of a *broken* Sobolev space.

Definition 2.4.4. *For an open set Ω with corresponding subdivision $\mathcal{T}_h$, the broken Sobolev space of composite order* $\mathbf{s}$ *is defined as*

$$H^{\mathbf{s}}(\Omega, \mathcal{T}_h) = \{u \in L^2(\Omega) : u|_\kappa \in H^{s_\kappa}(\kappa)\ \forall \kappa \in \mathcal{T}_h\}.$$

where

$$\mathbf{s} := \{s_\kappa\}_{\kappa \in \mathcal{T}_h}.$$

The associated norm and seminorm are then defined, respectively, as

$$\|u\|_{\mathbf{s},\mathcal{T}_h} = \left(\sum_{\kappa \in \mathcal{T}_h} \|u\|^2_{H^{s_\kappa}(\kappa)} \right)^{\frac{1}{2}}, \quad |u|_{\mathbf{s},\mathcal{T}_h} = \left(\sum_{\kappa \in \mathcal{T}_h} |u|^2_{H^{s_\kappa}(\kappa)} \right)^{\frac{1}{2}}.$$

In a similar fashion it is necessary to give the following definition.

Definition 2.4.5. *Let $u \in H^{\mathbf{1}}(\Omega, \mathcal{T}_h)$ and $\tau \in [H^{\mathbf{1}}(\Omega, \mathcal{T}_h)]^2$, then the broken gradient $\nabla_{\mathcal{T}_h} u$ of u and the broken divergence $\nabla_{\mathcal{T}_h} \cdot \tau$ of τ are defined as*

$$(\nabla_{\mathcal{T}_h} u)|_\kappa = \nabla(u|_\kappa), \quad (\nabla_{\mathcal{T}_h} \cdot \tau)|_\kappa = \nabla \cdot (\tau|_\kappa), \quad \kappa \in \mathcal{T}_h.$$

2.4.2 Trace Operators

To proceed with the DG discretization we must now define some operators acting on functions $v \in H^1(\Omega, \mathcal{T}_h)$. For a face $f \in \Gamma_{\text{int}}$, shared by two elements κ_i and κ_j, where the indices i and j satisfy $i > j$, we define the jump of v across f and the mean value of v on f, respectively, by

$$[v] = v|_{\partial\kappa_i \cap f} - v|_{\partial\kappa_j \cap f} \quad \text{and} \quad \langle v \rangle = \frac{1}{2}(v|_{\partial\kappa_i \cap f} + v|_{\partial\kappa_j \cap f}).$$

For a boundary face, $f \subset \partial\kappa$, we set $[v] = v|_{\partial\kappa \cap f}$ and $\langle v \rangle = v|_{\partial\kappa \cap f}$. For an element $\kappa \in \mathcal{T}_h$, we let v^+_κ and v^-_κ be the interior and exterior traces of v defined on $\partial\kappa$ and $\partial\kappa \backslash \Gamma$, respectively. To ease notation, when it is be clear which element κ, the values v^+_κ and v^-_κ refer to, we shall drop the subscript κ and use just v^+ and v^-, respectively. For $x \in \partial\kappa$, we define the inflow and outflow parts of $\partial\kappa$, respectively, by

$$\partial_-\kappa = \{x \in \partial\kappa : \mathbf{b}(x) \cdot \mathbf{n}_\kappa(x) < 0\}, \quad \text{and} \quad \partial_+\kappa = \{x \in \partial\kappa : \mathbf{b}(x) \cdot \mathbf{n}_\kappa(x) \geq 0\},$$

where $\mathbf{n}_\kappa(x)$ denotes the unit outward normal vector to $\partial\kappa$ at x.

For a given face $f \subset \Gamma_{\text{int}} \cup \Gamma_{\text{D}}$, such that $f \subset \partial\kappa$, for some $\kappa \in \mathcal{T}_h$, we write $\tilde{f}$ and $\hat{f}$ to denote the respective faces of the mapped elements $\tilde{\kappa}$ and $\hat{\kappa}$, respectively, based on employing the element mappings Q_κ and F_κ. More precisely, we write $\tilde{f} = Q_\kappa^{-1}(f)$ and $\hat{f} = F_\kappa^{-1}(\tilde{f})$. Further, we define m_f, $m_{\tilde{f}}$, and $m_{\hat{f}}$ to be the $(d-1)$–dimensional measure (volume) of the faces f, $\tilde{f}$, and $\hat{f}$, respectively; for example, in two-dimensions, i.e., $d = 2$,

$m_{\hat{f}}$, the length of the corresponding face on the canonical quadrilateral element, is equal to 2. In view of (2.4.1), we note that there exists a positive constant C_5, such that

$$C_5^{-1} m_{\tilde{f}} \leq m_f \leq C_5 m_{\tilde{f}} \tag{2.4.3}$$

for every face $f \subset \Gamma_{\text{int}} \cup \Gamma_{\text{D}}$. Moreover, the surface Jacobian $S_{f,\tilde{f}}$ arising in the transformation of the face f to $\tilde{f}$ may be uniformly bounded in the following manner

$$\|S_{f,\tilde{f}}\|_{L_\infty(\tilde{f})} \leq C_6 \tag{2.4.4}$$

for all faces $f \subset \Gamma_{\text{int}} \cup \Gamma_{\text{D}}$, where C_6 is a positive constant.

2.5 Finite Element Spaces

On an element $\hat{\kappa} \in \mathcal{T}_h$ we define two polynomial spaces, one with anisotropic polynomial degree $\vec{p} := \{p_i\}_{i=1,\ldots,d}$ and one with isotropic polynomial degree p, respectively, by:

$$\mathcal{Q}_{\vec{p}} := \text{span}\{\Pi_{i=1}^d \hat{x}_i^j : 0 \leq j \leq p_i\} \text{ and } \mathcal{P}_p := \text{span}\left\{\Pi_{i=1}^d \hat{x}_i^{\alpha_i} : 0 \leq \sum_{i=1}^d \alpha_i \leq p\right\}.$$

We associate a vector $\vec{p}_\kappa$ (with entries $p_{\kappa,i} > 1, i = 1,\ldots,d$) with every $\kappa \in \mathcal{T}_h$, which is the image of the reference hypercube and a scalar p_κ with every element $\kappa \in \mathcal{T}_h$ which is the image of the unit simplex; in this case, we set $\vec{p}_\kappa \equiv p_\kappa$. Furthermore, for conciseness we let $\mathcal{P}_{\vec{p}} \equiv \mathcal{P}_p$ and introduce

$$\mathcal{R}_{\vec{p}} := \begin{cases} \mathcal{Q}_{\vec{p}} & \text{if } \hat{\kappa} \text{ is the reference hypercube,} \\ \mathcal{P}_{\vec{p}} & \text{if } \hat{\kappa} \text{ is the unit simplex;} \end{cases}$$

thereby, if $\hat{\kappa}$ is a simplex we use polynomials from $\mathcal{P}_{\vec{p}_\kappa}$, otherwise if $\hat{\kappa}$ is a hypercube we use polynomials from $\mathcal{Q}_{\vec{p}_\kappa}$. Thus, for an element which is the image of the reference hypercube, anisotropic information can be contained in the polynomial degree vector $\vec{p}_\kappa$, as well as by way of the mapping F_κ. Introducing $\vec{\mathbf{p}} = \{\vec{p}_\kappa : \kappa \in \mathcal{T}_h\}$ and $\mathbf{F} = \{Q_\kappa \circ F_\kappa : \kappa \in \mathcal{T}_h\}$, we are able to concisely define our discontinuous finite element space:

Definition 2.5.1. *We define the finite element space with respect to* Ω, $\mathcal{T}_h$, $\mathbf{F}$ *and* $\vec{\mathbf{p}}$ *by*

$$S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F}) = \{u \in L^2(\Omega) : u|_\kappa \circ Q_\kappa \circ F_\kappa \in \mathcal{R}_{\vec{p}_\kappa} \forall \kappa \in \mathcal{T}_h\}.$$

2.6 Interior Penalty Discontinuous Galerkin Discretization

We now proceed to discretize our model problem (2.2.1), (2.2.3) based on the DG method presented in Harriman *et al.* [61], for example; here the discretization of the second order elliptic term $\nabla \cdot (a \nabla u)$ is based on the work outlined in [13]. In [13] a number of different discretizations are presented, but we restrict ourselves to the derivation of the so-called *Symmetric Interior Penalty* (SIP) DG method.

In order to fully appreciate the construction of the DG method the first step is to rewrite our advection-diffusion-reaction problem (2.2.1) as an equivalent first order system of partial differential equations, which we term the auxiliary, or flux formulation:

$$\boldsymbol{\Phi} - a \nabla u = 0 \quad \text{in } \Omega, \tag{2.6.1}$$

$$-\nabla \cdot \boldsymbol{\Phi} + \nabla \cdot (\mathbf{b} u) + c u = f \quad \text{in } \Omega. \tag{2.6.2}$$

As with the standard Galerkin FEM we multiply (2.6.1) and (2.6.2) by suitable smooth test functions τ and v, respectively, though here we integrate by parts over each element κ rather than the whole of the computational domain Ω to obtain:

$$\begin{aligned} \int_\kappa \boldsymbol{\Phi} \cdot \tau \mathrm{d}x + \int_\kappa \nabla \cdot (a\tau) u \mathrm{d}x - \int_{\partial\kappa} (a\tau) \cdot \mathbf{n}_\kappa u \mathrm{d}s &= 0, \\ \int_\kappa \boldsymbol{\Phi} \cdot \nabla v \mathrm{d}x - \int_{\partial\kappa \setminus \Gamma_N} \boldsymbol{\Phi} \cdot \mathbf{n}_\kappa v \mathrm{d}s - \int_{\partial\kappa \cap \Gamma_N} g_N v \mathrm{d}s - \int_\kappa u \mathbf{b} \cdot \nabla v \mathrm{d}x & \\ + \int_{\partial\kappa} \mathbf{b} \cdot \mathbf{n}_\kappa u v \mathrm{d}s + \int_\kappa c u v \mathrm{d}x &= \int_\kappa f v \mathrm{d}x. \end{aligned}$$

In order to reduce the problem to one of finite-dimensional size we now restrict the choice of trial and test functions to subspaces based on $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$. Summing over all the elements κ, in the mesh $\mathcal{T}_h$ and introducing numerical flux functions $\widehat{u_h}$, $\mathcal{H}(u_h^+, u_h^-, n_k)$ and $\widehat{\boldsymbol{\Phi}_h \cdot \mathbf{n}_f}$, which we will define momentarily, we have the auxiliary formulation of the DG method: find $u_h \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ and $\boldsymbol{\Phi}_h \in [S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})]^d$ such that

$$\sum_{\kappa \in \mathcal{T}_h} \left(\int_\kappa \boldsymbol{\Phi}_h \cdot \tau \mathrm{d}x + \int_\kappa \nabla \cdot (a\tau) u_h \mathrm{d}x \right) - \int_{\Gamma_{\mathrm{int}} \cup \Gamma_0} [(a\tau) \cdot \mathbf{n}_f] \widehat{u_h} \mathrm{d}s = 0, \tag{2.6.3}$$

$$\begin{aligned} \sum_{\kappa \in \mathcal{T}_h} \left(\int_\kappa \boldsymbol{\Phi}_h \cdot \nabla v \mathrm{d}x - \int_\kappa (u_h \mathbf{b} \cdot \nabla v - c u_h v) \mathrm{d}x + \int_{\partial\kappa} \mathcal{H}(u_h^+, u_h^-, n_k) v^+ \mathrm{d}s \right) & \\ - \int_{\Gamma_{\mathrm{int}} \cup \Gamma_D} \widehat{\boldsymbol{\Phi}_h \cdot \mathbf{n}_f} [v] \mathrm{d}s = \sum_{\kappa \in \mathcal{T}_h} \left(\int_\kappa f v \mathrm{d}x + \int_{\partial\kappa \cap \Gamma_N} g_N v \mathrm{d}s \right) & \end{aligned} \tag{2.6.4}$$

for all $v \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ and $\tau \in [S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})]^d$.

We now discuss the numerical fluxes $\widehat{u_h}$, $\mathcal{H}(u_h^+, u_h^-, n_\kappa)$ and $\widehat{\mathbf{\Phi}_h \cdot \mathbf{n}_f}$ which represent approximations to u, $\mathbf{b} \cdot \mathbf{n}_\kappa u$, and $a\nabla u \cdot \mathbf{n}_f$, respectively, on the faces of the elements. It is essential that good choices for the numerical fluxes are selected, as they are the only means by which information is passed from one element in the mesh to another, due to the lack of continuity across inter-element boundaries. In effect we will be imposing continuity weakly across element boundaries. Two important properties that should be considered when devising numerical fluxes are consistency and conservation. The numerical fluxes are consistent if

$$\widehat{u_h}(v^+, v^-)|_f = v, \quad \mathcal{H}(v^+, v^-, n_\kappa)|_{\partial\kappa} = \mathbf{b} \cdot \mathbf{n}_\kappa v, \quad \widehat{\mathbf{\Phi}_h \cdot \mathbf{n}}_f(v^+, \nabla v^+, v^-, \nabla v^-)|_f = a\nabla v \cdot \mathbf{n}_f,$$

for any smooth function v satisfying the boundary conditions and they are conservative if they are single-valued on every face in the mesh. Should the numerical fluxes be consistent/conservative then the DG method will inherit the same properties; see Section 2.6.1.

The (hyperbolic) numerical flux function $\mathcal{H}(v_h^+, v_h^-, n_\kappa)$ is defined by

$$\mathcal{H}(u_h^+, u_h^-, n_\kappa)|_{\partial\kappa} = \begin{cases} \mathbf{b} \cdot \mathbf{n}_\kappa g_{\mathrm{D}} & \text{when } x \in \partial\kappa \cap (\Gamma_{\mathrm{D}} \cup \Gamma_-), \\ \mathbf{b} \cdot \mathbf{n}_\kappa \lim\limits_{s \to 0^+} u_h(x - s\mathbf{b}) & \text{otherwise .} \end{cases}$$

for $\kappa \in \mathcal{T}_h$. This flux is both consistent and conservative and simply selects the exterior trace of the numerical solution u_h, on an inflow face and the interior trace of u_h on an outflow face. There are many ways of choosing the numerical fluxes for the elliptic part, cf. Remark 2.6.4; here we give the numerical fluxes for the Symmetric Interior Penalty (SIP) DG method. In this case the numerical flux functions $\widehat{u_h}$ and $\widehat{\mathbf{\Phi}_h \cdot \mathbf{n}_f}$ are defined by

$$\widehat{u_h} = \begin{cases} \langle u_h \rangle, & f \subset \Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{N}}, \\ g_{\mathrm{D}}, & f \subset \Gamma_{\mathrm{D}}. \end{cases}$$

and

$$\widehat{\mathbf{\Phi}_h \cdot \mathbf{n}_f} = \begin{cases} \langle (a\nabla u_h) \cdot \mathbf{n}_e \rangle - \vartheta[u_h], & f \subset \Gamma_{\mathrm{int}}, \\ (a\nabla u_h|_f) \cdot \mathbf{n}_f - \vartheta(u_h|_f - g_{\mathrm{D}}), & f \subset \Gamma_{\mathrm{D}}. \end{cases}$$

respectively. It is trivial to show that both these fluxes are consistent and conservative. Here, ϑ is referred to as the discontinuity-penalization function and has the effect of penalizing any jump discontinuities on an interior face, and imposing the Dirichlet boundary conditions on faces which are subsets of Γ_{D}. Thereby, the larger we set ϑ the closer the numerical solution becomes to being a continuous piecewise polynomial function. The terms involving ϑ have to be added in order to ensure that the DG method is stable; in Section 2.7 we discuss how to select ϑ.

It is perfectly acceptable to leave the method in the form (2.6.3), (2.6.4) and solve this system, but instead we consider the so-called *primal formulation*. To this end, we eliminate $\mathbf{\Phi}_h$ from (2.6.3) and (2.6.4) by setting $\tau = \nabla v$ in (2.6.3) and integrating by parts. Then we combine the two equations by substituting for the term $\int_\kappa \mathbf{\Phi}_h \cdot \tau \mathrm{d}x$. Thus, the primal formulation is: find u_{DG} in $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ such that

$$B_{\mathrm{DG}}(u_{\mathrm{DG}}, v) = \ell_{\mathrm{DG}}(v) \tag{2.6.5}$$

for all $v \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$. Here, the bilinear form $B_{\mathrm{DG}}(\cdot, \cdot)$ is defined by

$$B_{\mathrm{DG}}(w, v) = B_a(w, v) + B_{\mathbf{b}}(w, v) + \theta B_f(v, w) - B_f(w, v) + B_\vartheta(w, v), \tag{2.6.6}$$

where

$$\begin{aligned}
B_a(w, v) &= \sum_{\kappa \in \mathcal{T}_h} \int_\kappa a \nabla w \cdot \nabla v \mathrm{d}x, \\
B_{\mathbf{b}}(w, v) &= \sum_{\kappa \in \mathcal{T}_h} \Bigg(- \int_\kappa (w \mathbf{b} \cdot \nabla v - cwv) \mathrm{d}x \\
&\qquad + \int_{\partial_+ \kappa} (\mathbf{b} \cdot \mathbf{n}_\kappa) w^+ v^+ \mathrm{d}s + \int_{\partial_- \kappa \setminus \Gamma} (\mathbf{b} \cdot \mathbf{n}_\kappa) w^- v^+ \mathrm{d}s \Bigg), \\
B_f(v, w) &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \langle (a \nabla w) \cdot \mathbf{n}_f \rangle [v] \mathrm{d}s, \\
B_\vartheta(w, v) &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \vartheta [w][v] \mathrm{d}s,
\end{aligned}$$

and the linear functional ℓ_{DG} is given by

$$\begin{aligned}
\ell_{\mathrm{DG}}(v) = \sum_{\kappa \in \mathcal{T}_h} \Bigg(& \int_\kappa f v \mathrm{d}x - \int_{\partial_- \kappa \cap (\Gamma_{\mathrm{D}} \cup \Gamma_-)} (\mathbf{b} \cdot \mathbf{n}_\kappa) g_{\mathrm{D}} v^+ \mathrm{d}s \\
& + \int_{\partial_- \kappa \cap \Gamma_{\mathrm{D}}} \theta g_{\mathrm{D}} ((a \nabla v^+) \cdot \mathbf{n}_\kappa) \mathrm{d}s + \int_{\partial \kappa \cap \Gamma_{\mathrm{N}}} g_{\mathrm{N}} v^+ \mathrm{d}s + \int_{\partial \kappa \cap \Gamma_{\mathrm{D}}} \vartheta g_{\mathrm{D}} v^+ \mathrm{d}s \Bigg).
\end{aligned} \tag{2.6.7}$$

Here, θ is a parameter, which in the case of the SIP method is equal to -1: the name arises from when $\mathcal{L}$ is purely elliptic, *i.e.* $\mathbf{b} \equiv \mathbf{0}$, and the bilinear form $B_{\mathrm{DG}}(\cdot,\cdot)$ becomes symmetric, that is $B_{\mathrm{DG}}(u,v) = B_{\mathrm{DG}}(v,u)$ for all u and v.

2.6.1 Consistency

Here we define the important notions of consistency and adjoint consistency.

Definition 2.6.1. *The primal formulation is consistent if*

$$B_{\mathrm{DG}}(u,v) = \ell_{\mathrm{DG}}(v) \quad \forall v \in S^{\vec{\mathbf{p}}}(\Omega,\mathcal{T}_h,\mathbf{F}).$$

where u is the analytical solution to the problem (2.2.1), (2.2.3).

Definition 2.6.2. *The primal formulation is adjoint consistent if*

$$B_{\mathrm{DG}}(v,\varphi) = \int_\Omega v g \mathrm{d}x \quad \forall v \in H^2(\Omega,\mathcal{T}_h).$$

where φ is the solution to the adjoint of (2.2.1), (2.2.3):

$$\mathcal{L}^*\varphi = g \quad in \quad \Omega,$$

subject to homogeneous boundary conditions.

If we assume that the analytical solution u of problem (2.2.1), (2.2.3) lies in $H^{3/2+\varepsilon}(\Omega,\mathcal{T}_h)$, $\varepsilon > 0$, and the functions u and $(a\nabla u)\cdot\mathbf{n}_f$ are continuous across each face $f \subset \partial\kappa\backslash\Gamma$ that intersect the subdomain of ellipticity $\Omega_a = \{x \in \bar{\Omega} : \zeta^T a(x)\zeta > 0 \quad \forall \zeta \in \mathbb{R}^d\}$, then the SIP method is consistent; this stems from the consistency of the numerical fluxes. Thereby, the Galerkin orthogonality property may be established:

$$B_{\mathrm{DG}}(u - u_{\mathrm{DG}}, v) = 0 \quad \forall v \in S^{\vec{\mathbf{p}}}(\Omega,\mathcal{T}_h,\mathbf{F}). \tag{2.6.8}$$

which is essential for the proceeding error analysis.

Remark 2.6.3. If the smoothness condition on u above is violated, then the discretization method must be modified accordingly, cf. [76].

Remark 2.6.4. The parameter, θ, from (2.6.6) and (2.6.7) can take values other than -1. Indeed, popular choices are $\theta = 1$ and $\theta = 0$ which yield the Non-Symmetric Interior Penalty (NIP) method and the Incomplete Interior Penalty (IIP) method, respectively. These alternative methods can be attained with different choices for the elliptic numerical flux functions $\widehat{u_h}$ and $\widehat{\mathbf{\Phi}_h \cdot \mathbf{n}_f}$. Indeed, there are a wide range of choices for these numerical flux functions resulting in alternative DG formulations: see [13] for a detailed review. Although, like the SIP method, both the NIP and IIP methods are consistent, only the SIP method is adjoint consistent. In general a DG formulation requires the numerical fluxes to be conservative for it to be adjoint consistent and in the case of the NIP method the numerical flux $\widehat{u_h}$ is not conservative. The lack of adjoint consistency of the NIP method has dramatic consequences when considering any duality based error analysis, such as the Aubin-Nitsche duality argument, cf. [37], or the proceeding error analysis from Chapter 4.

2.7 Stability

In Section 2.6 we derived a method for finding an approximate solution to our model problem (2.2.1), (2.2.3). A natural question to ask is the following: "How do we know that a solution to (2.6.5) actually exists and is it unique?". Before we can answer this question, we need some definitions.

Definition 2.7.1. *A bilinear form $B(\cdot,\cdot)$ on a normed linear space H, is said to be* coercive *on $H \times H$ if $\exists C_S > 0$ such that*

$$B(v,v) \geq C_S \|v\|_H^2 \quad \forall v \in H.$$

We can now state the following standard theorem.

Theorem 2.7.2. *If $B(\cdot,\cdot)$ is a coercive bilinear form on a normed, linear, finite dimensional space H, then for any linear functional $\ell(\cdot)$ there exists a unique $u \in H$ such that*

$$B(u,v) = \ell(v) \quad \forall v \in H. \tag{2.7.1}$$

Proof. Coercivity of the bilinear form $B(\cdot,\cdot)$, over $H \times H$, immediately implies that if

$$B(w, w) = 0,$$

then $w \equiv 0$. This in turn implies the uniqueness of a solution: suppose we have two different solutions of u and u^* of (2.7.1), then

$$B(u, v) - B(u^*, v) = B(u - u^*, v) = \ell(v) - \ell(v) = 0 \quad \forall v \in H.$$

Selecting $v = u - u^*$, yields $B(u - u^*, u - u^*) = 0$, hence $u \equiv u^*$. As the linear space H is finite-dimensional and (2.7.1) is a linear problem, the existence of the solution to (2.7.1) follows from the fact that its homogeneous counterpart has the unique solution $u_{\mathrm{DG}} \equiv 0$.

■

Thus, if we can prove coercivity of the bilinear from, $B_{\mathrm{DG}}(\cdot,\cdot)$ on the space $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$, we can guarantee the existence of a unique solution to (2.6.5). Before we proceed it is first necessary to define a norm on our space $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$. We define the DG-norm $|||\cdot|||_{\mathrm{DG}}$ (see [61]) by

$$\begin{aligned} |||w|||_{\mathrm{DG}}^2 &= \sum_{\kappa \in \mathcal{T}_h} (\|\sqrt{a}\nabla w\|_{L^2(\kappa)}^2 + \|c_0 w\|_{L^2(\kappa)}^2 + \frac{1}{2}\|w^+\|_{\partial_- \kappa \cap (\Gamma_{\mathrm{D}} \cup \Gamma_-)}^2 \\ &+ \frac{1}{2}\|w^+ - w^-\|_{\partial_- \kappa \setminus \Gamma}^2 + \frac{1}{2}\|w^+\|_{\partial_+ \kappa \cap \Gamma}^2 \\ &+ \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \vartheta [w]^2 \mathrm{d}s + \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \frac{1}{\vartheta} \langle (a\nabla w) \cdot \mathbf{n}_f \rangle^2 \mathrm{d}s. \end{aligned}$$

where $\|\cdot\|_\tau$, $\tau \subset \partial\kappa$, is the norm induced from the inner-product

$$(v, w)_\tau = \int_\tau |\mathbf{b} \cdot \mathbf{n}_\kappa| v w \mathrm{d}s.$$

and c_0 is defined as

$$(c_0(x))^2 = c(x) + \frac{1}{2}\nabla \cdot \mathbf{b}(x) \qquad \forall x \in \Omega.$$

cf. Assumption (2.3.2).

We now state the following lemma, which is important for the forthcoming stability arguments.

Lemma 2.7.3. *For* $w \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$

$$B_{\mathbf{b}}(w,w) = \sum_{\kappa\in\mathcal{T}_h}\left(\|c_0 w\|^2_{L^2(\kappa)} + \frac{1}{2}\|w^+\|^2_{\partial_-\kappa\cap(\Gamma_D\cup\Gamma_-)} + \frac{1}{2}\|w^+ - w^-\|^2_{\partial_-\kappa\setminus\Gamma} + \frac{1}{2}\|w^+\|^2_{\partial_+\kappa\cap\Gamma}\right). \tag{2.7.2}$$

Proof. This result follows after an application of the integration by parts formula; see [75] for details.■

To prove coercivity in the DG-norm, we assume, for simplicity, that the entries of the matrix a are constant on each element $\kappa \in \mathcal{T}_h$, *i.e.*

$$a \in [S^{\vec{\mathbf{0}}}(\Omega, \mathcal{T}_h, \mathbf{F})]^{d\times d}_{\mathrm{sym}},$$

and let $a_\kappa := a|_\kappa$. We also define $\bar{a} = |\sqrt{a}|_2^2$, where $|\cdot|_2$ denotes the matrix norm subordinate to the l_2-vector norm on $\mathbb{R}^d$, that is

$$|A|_2 := \max_{v\in\mathbb{R}^d\setminus\{0\}} \frac{\|Av\|_2}{\|v\|_2} \quad A \in \mathbb{R}^{d\times d}.$$

In this thesis we will analyze the DG method in two settings. In the first case, the polynomial degrees are restricted so that they are isotropic on each element, *i.e.* for all elements $\kappa \in \mathcal{T}_h$, $\vec{p}_\kappa \equiv p_\kappa$, where $p_\kappa \geq 1$ is an integer, but full orientation of the element is allowed, *i.e* no restriction is placed on the the mapping F_κ, except that it is affine, cf. Figure 2.1. In this case we define the finite element space to be $S^{\mathbf{p}_{\mathrm{iso}}}(\Omega, \mathcal{T}_h, \mathbf{F})$. The second setting allows anisotropic polynomial degrees, but restricts each element $\kappa \in \mathcal{T}_h$ to be an axiparallel image of the reference square (up to a C^1-diffeomorphism), in other words F_κ is an affine mapping of the form

$$F_\kappa(\hat{x}) = A_\kappa \hat{x} + \mathbf{b}_\kappa,$$

where $A_\kappa := \frac{1}{2}\mathrm{diag}(h_1^\kappa, h_2^\kappa)$, with h_1^κ and h_2^κ the lengths of the edges of $\tilde{\kappa}$ parallel to the $\tilde{x}_1$- and $\tilde{x}_2$-axes, respectively, $\mathbf{b}_\kappa$ is a two-component real-valued vector and Q_κ is a smooth diffeomorphism as before; see Figure 2.2.

In this way, our analysis will enable us to understand what effects the orientation of the elements has on the error of the method separately from the effects of employing anisotropic polynomial degrees.

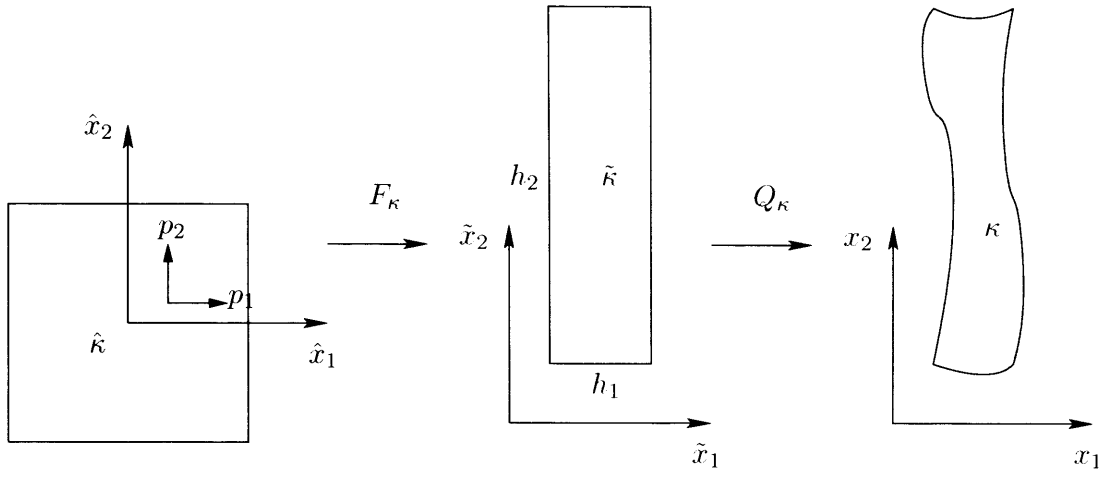


Figure 2.2: Construction of axiparallel elements via composition of affine maps and C^1-diffeomorphisms.

2.7.1 Stability with Isotropic Polynomial Degrees

For proving coercivity when isotropic polynomials degrees are used, the following inverse inequality will be required.

Lemma 2.7.4. *Let $\hat{\kappa}$ be either the reference d–hypercube or unit d–simplex, $d = 2, 3$, then for any function $\hat{v} \in \mathcal{R}_p(\hat{\kappa})$, $p \geq 1$, there exists a positive constant C'_{inv} which depends only on the dimension d, such that for $\hat{f} \subset \partial\hat{\kappa}$*

$$\|\hat{v}\|^2_{\hat{f}} \leq C'_{\text{inv}} p^2 \, \|\hat{v}\|^2_{L_2(\hat{\kappa})}. \tag{2.7.3}$$

Proof. The result follows after application of the inverse estimate

$$\|\hat{\nabla}\hat{v}\|_{L^2(\hat{\kappa})} \leq C''_{\text{inv}} p^2 \|\hat{v}\|_{L^2(\hat{\kappa})} \tag{2.7.4}$$

together with the multiplicative trace inequality

$$\|\hat{v}\|^2_{L^2(\partial\hat{\kappa})} \leq C_{\text{t}} \left(\|\hat{v}\|^2_{L^2(\hat{\kappa})} + \|\hat{v}\|_{L^2(\hat{\kappa})} \|\hat{\nabla}\hat{v}\|_{L^2(\hat{\kappa})} \right). \tag{2.7.5}$$

here both C''_{inv} and C_{t} are positive constants depending only on the dimension d. A proof of (2.7.5) for $d = 2$ can be found in, for example, Prudhomme *et al.* [109], with analogous arguments holding for $d = 3$. Schwab [119] provides a proof of (2.7.4) for $d = 2$.

while for the 3–hypercube the result can be shown by performing a tensor product of the one-dimensional result. For the reference tetrahedron a proof of (2.7.4) can be found in Georgoulis [51]. ■

We now need to scale (2.7.3) to the physical element κ: this is undertaken in the next lemma.

Lemma 2.7.5. *Let κ be an element contained in the mesh $\mathcal{T}_h$ and let f denote one of its faces. Then, the following inverse inequality holds*

$$\|v\|^2_{L_2(f)} \leq C_{\text{inv}} \frac{m_f}{m_\kappa} p^2 \|v\|^2_{L_2(\kappa)} \tag{2.7.6}$$

for all v such that $v \circ Q_\kappa \circ F_\kappa \in \mathcal{R}_p(\hat{\kappa})$, where C_{inv} is a positive constant which depends only on the dimension d.

Proof. We use (2.7.3) and rescale both the left- and right-hand sides. For the left-hand side we make use of (2.4.3) and (2.4.4) and obtain

$$\|v\|^2_{L_2(f)} \leq C_6 \|\tilde{v}\|^2_{L_2(\tilde{f})} = C_6 \frac{m_{\tilde{f}}}{m_{\hat{f}}} \|\hat{v}\|^2_{L_2(\hat{f})} \leq \frac{C_6}{C_5} \frac{m_f}{m_{\hat{f}}} \|\hat{v}\|^2_{L_2(\hat{f})}. \tag{2.7.7}$$

Similarly, for the right-hand side using (2.4.1) and (2.4.2) yields

$$\|\hat{v}\|^2_{L_2(\hat{\kappa})} = \det(F_\kappa^{-1}) \|\tilde{v}\|^2_{L_2(\tilde{\kappa})} = \frac{m_{\hat{\kappa}}}{m_{\tilde{\kappa}}} \|\tilde{v}\|^2_{L_2(\tilde{\kappa})} \leq C_4 \frac{m_{\hat{\kappa}}}{m_\kappa} \|\tilde{v}\|^2_{L_2(\tilde{\kappa})} \leq C_1 C_4 \frac{m_{\hat{\kappa}}}{m_\kappa} \|v\|^2_{L_2(\kappa)}. \tag{2.7.8}$$

Inserting (2.7.7) and (2.7.8) into (2.7.3) gives the desired result. ■

Remark 2.7.6. The inverse inequality stated in Lemma 2.7.5 is an extension of the standard result employed on isotropic finite element meshes to the case when anisotropic elements may be present. Indeed, in the isotropic setting, we have that $m_\kappa \approx h_\kappa^d$ and $m_f \approx h_\kappa^{d-1}$, where h_κ denotes the diameter of the element $\kappa \in \mathcal{T}_h$; thereby, the scaling on the right–hand side of the inequality (2.7.6) is of size $1/h_\kappa$, as expected. Moreover, this result extends the inverse inequality stated in [53] to the case when the affine mapping F_κ includes not only size, but also orientation information, cf. above.

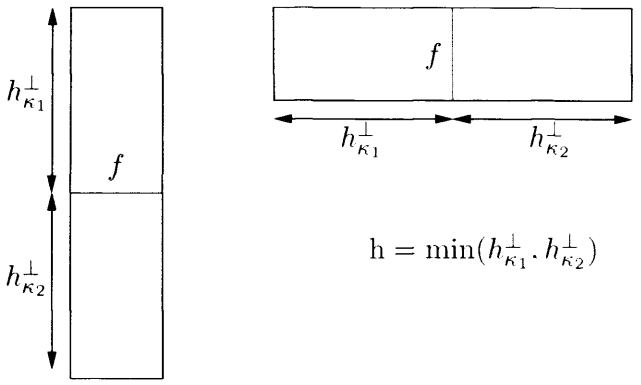


Figure 2.3: Selection of mesh function h on face f.

We now define the function $\mathrm{h} \in L_\infty(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})$, as $\mathrm{h}(x) = \min\{m_{\kappa_1}, m_{\kappa_2}\}/m_f$, if x is in the interior of $f \subset \partial\kappa_1 \cap \partial\kappa_2$ for two neighboring elements in the mesh $\mathcal{T}_h$, and $\mathrm{h}(x) = m_\kappa/m_f$, if x is in the interior of $f \subset \partial\kappa \cap \Gamma_{\mathrm{D}}$. For example, if two neighbouring elements k and k' sharing a face f are rectangular, then $\mathrm{h}|_f$ is simply the minimum of the lengths of the faces of κ and κ' in the direction orthogonal to f; see Figure 2.3. We note that in the isotropic setting we observe that $\mathrm{h} \sim h$, where h denotes the mesh local mesh size, cf. Remark 2.7.6 above. We also define the function $\mathrm{a} \in L_\infty(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})$ by $\mathrm{a}(x) = \max\{\bar{a}_{\kappa_1}, \bar{a}_{\kappa_2}\}$ if x is in the interior of $f = \partial\kappa_1 \cap \partial\kappa_2$, and $\mathrm{a}(x) = \bar{a}_\kappa$ if x is in the interior of $\partial\kappa \cap \Gamma_{\mathrm{D}}$. Similarly, we define the function $\mathrm{p}(x) \in L_\infty(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})$ by $\mathrm{p}(x) = \max\{p_{\kappa_1}, p_{\kappa_2}\}$ if x is in the interior of $f \subset \partial\kappa_1 \cap \partial\kappa_2$, and $\mathrm{p}(x) = p_\kappa$ if x is in the interior of $\partial\kappa \cap \Gamma_{\mathrm{D}}$. With this notation, we are now in a position to state and prove the following coercivity result for the bilinear form $B_{\mathrm{DG}}(\cdot,\cdot)$ over $S^{\mathbf{p}_{\mathrm{iso}}}(\Omega, \mathcal{T}, \mathbf{F}) \times S^{\mathbf{p}_{\mathrm{iso}}}(\Omega, \mathcal{T}, \mathbf{F})$.

Theorem 2.7.7. *If ϑ is defined as*

$$\vartheta|_f \equiv \vartheta_f \;=\; C_\vartheta \frac{\mathrm{ap}^2}{\mathrm{h}} \quad \textit{for} \quad f \subset \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{int}}, \tag{2.7.9}$$

then there exists a positive constant C_s, which depends only on the dimension d and the shape-regularity of $\mathcal{T}_h$, such that

$$B_{\mathrm{DG}}(v, v) \geq C_s |\!|\!| v |\!|\!|^2_{\mathrm{DG}} \quad \forall v \in S^{\mathbf{p}_{\mathrm{iso}}}(\Omega, \mathcal{T}, \mathbf{F}),$$

provided that the constant C_ϑ is chosen such that:

$$C_\vartheta > C'_\vartheta > 0.$$

where C'_ϑ is a sufficiently large positive constant.

The forthcoming proof follows the argument found in Prudhomme *et al.* [109], with minor extensions since [109] only considers the case when $\mathbf{b} \equiv \mathbf{0}$ and $a \equiv I$.

Proof. Let C_s be an arbitrary real number and pick $v \in S^{\mathbf{p}_{\text{iso}}}(\Omega, \mathcal{T}, \mathbf{F})$. Then

$$\begin{aligned} B_{\text{DG}}(v,v) - C_s |\!|\!| v |\!|\!|^2_{\text{DG}} &= (1-C_s)(B_a(v,v) + B_{\mathbf{b}}(v,v) + B_\vartheta(v,v)) \\ &\quad -2\int_{\Gamma_{\text{int}}\cup\Gamma_{\text{D}}} \langle (a\nabla v)\cdot \mathbf{n}_f\rangle [v] \mathrm{d}s \\ &\quad -C_s \int_{\Gamma_{\text{int}}\cup\Gamma_{\text{D}}} \frac{1}{\vartheta}\langle (a\nabla v)\cdot \mathbf{n}_f\rangle^2 \mathrm{d}s. \end{aligned} \tag{2.7.10}$$

Restricted to a face $f \subset \Gamma_{\text{int}}$, the interface between elements κ_i and κ_j, the last term in the above expression (2.7.10) can be bounded as follows:

$$\int_f \frac{1}{\vartheta}\langle (a\nabla v)\cdot \mathbf{n}_f\rangle^2 \mathrm{d}s \le \int_f \frac{1}{2\vartheta}\left[((a_{\kappa_i}\nabla v)\cdot \mathbf{n}_f)^2 + \left((a_{\kappa_j}\nabla v)\cdot \mathbf{n}_f\right)^2 \right] \mathrm{d}s. \tag{2.7.11}$$

Using the fact that ϑ is constant on each face and employing the inverse inequality (2.7.6) the following bound is obtained

$$\begin{aligned} \int_f \frac{1}{\vartheta}((a_{\kappa_i}\nabla v)\cdot \mathbf{n}_f)^2 \mathrm{d}s &= \frac{1}{\vartheta}\|a_{\kappa_i}\nabla v\|^2_{L_2(f)} \\ &\le \frac{C_{\text{inv}}}{\vartheta}\frac{m_f}{m_{\kappa_i}} p^2_{\kappa_i} \|a_{\kappa_i}\nabla v\|^2_{L_2(\kappa_i)} \\ &\le \frac{C_{\text{inv}}}{\vartheta}\frac{m_f \bar{a}_{\kappa_i}}{m_{\kappa_i}} p^2_{\kappa_i} \|\sqrt{a_{\kappa_i}}\nabla v\|^2_{L_2(\kappa_i)}. \end{aligned}$$

Thus, setting ϑ as

$$\vartheta|_f = C_\vartheta \frac{\mathrm{ap}^2}{\mathrm{h}}$$

yields

$$\int_f \frac{1}{\vartheta}((a_{\kappa_i}\nabla v)\cdot \mathbf{n}_f)^2 \mathrm{d}s \le \frac{C_{\text{inv}}}{C_\vartheta}\|\sqrt{a_{\kappa_i}}\nabla v\|^2_{L_2(\kappa_i)}.$$

Thereby,

$$\int_f \frac{1}{\vartheta}\langle (a\nabla v)\cdot \mathbf{n}_f\rangle^2 \mathrm{d}s \le \frac{C_{\text{inv}}}{2C_\vartheta}\left[\|\sqrt{a}\nabla v\|^2_{L_2(\kappa_i)} + \|\sqrt{a}\nabla v\|^2_{L_2(\kappa_j)} \right]. \tag{2.7.12}$$

By using (2.7.9) an analogous argument for $f \in \Gamma_{\text{D}}$ yields

$$\int_f \frac{1}{\vartheta}\langle (a\nabla v)\cdot \mathbf{n}_f\rangle^2 \mathrm{d}s \le \frac{C_{\text{inv}}}{C_\vartheta}\|\sqrt{a}\nabla v\|^2_{L_2(\kappa)}.$$

Hence,

$$\int_{\Gamma_{\rm int}\cup\Gamma_{\rm D}} \frac{1}{\vartheta}\langle(a\nabla v)\cdot\mathbf{n}_f\rangle^2 {\rm d}s \leq \frac{C_{\rm inv}C_f}{C_\vartheta} B_a(v,v). \tag{2.7.13}$$

where C_f is dependent on the maximum number of element interactions on an element boundary, that is:

$$C_f = \max_{\kappa\in\mathcal{T}_h} {\rm card}\{f\in\Gamma_{\rm int}\cup\Gamma_{\rm D} : f\subset\partial\kappa\}.$$

For a face $f\in\Gamma_{\rm int}\cup\Gamma_{\rm D}$ the following bound holds:

$$\begin{aligned} 2\int_f \langle(a\nabla v)\cdot\mathbf{n}_f\rangle[v]{\rm d}s &\leq 2\sqrt{\int_f \frac{1}{\vartheta}\langle(a\nabla v)\cdot\mathbf{n}_f\rangle^2{\rm d}s}\sqrt{\int_f \vartheta[v]^2{\rm d}s} \\ &\leq \varepsilon\int_f \frac{1}{\vartheta}\langle(a\nabla v)\cdot\mathbf{n}_f\rangle^2{\rm d}s + \frac{1}{\varepsilon}\int_f \vartheta[v]^2{\rm d}s. \end{aligned}$$

for any $\varepsilon>0$. Summing over all faces $f\in\Gamma_{\rm int}\cup\Gamma_{\rm D}$ and employing (2.7.13) we readily obtain

$$2\int_{\Gamma_{\rm int}\cup\Gamma_{\rm D}} \langle(a\nabla v)\cdot\mathbf{n}_f\rangle[v]{\rm d}s \leq \varepsilon\frac{C_{\rm inv}C_f}{C_\vartheta}B_a(v,v) + \frac{1}{\varepsilon}B_\vartheta(v,v).$$

and therefore,

$$\begin{aligned} B_{\rm DG}(v,v) - C_s|\!|\!|v|\!|\!|^2_{\rm DG} \geq & \left(1 - C_s - (C_s+\varepsilon)\frac{C_{\rm inv}C_f}{C_\vartheta}\right) B_a(v,v) \\ & + (1-C_s)\,B_{\mathbf{b}}(v,v) + \left(1 - C_s - \frac{1}{\varepsilon}\right) B_\vartheta(v,v). \end{aligned}$$

Evidently $B_a(v,v)$ and $B_\vartheta(v,v)$ are non-negative, and similarly $B_{\mathbf{b}}(v,v)\geq 0$ due to Lemma 2.7.2, hence it follows that if

$$\left(1 - C_s - (C_s+\varepsilon)\frac{C_{\rm inv}C_f}{C_\vartheta}\right) > 0, \quad (1-C_s) > 0 \quad \text{and} \quad \left(1 - C_s - \frac{1}{\varepsilon}\right) > 0 \tag{2.7.14}$$

then coercivity holds. The last inequality will only hold provided

$$\varepsilon > 1,$$

while the first implies that

$$0 < C_s < \frac{1-\varepsilon C_{\rm inv}C_f/C_\vartheta}{1+C_{\rm inv}C_f/C_\vartheta} < \frac{1-C_{\rm inv}C_f/C_\vartheta}{1+C_{\rm inv}C_f/C_\vartheta} = \frac{C_\vartheta - C_{\rm inv}C_f}{C_\vartheta + C_{\rm inv}C_f}.$$

So, in order for C_s to exist, C_ϑ must be taken sufficiently large, that is $C_\vartheta > C_{\rm inv}C_f$, in which case the second inequality from (2.7.14) automatically holds and the method is coercive. ■

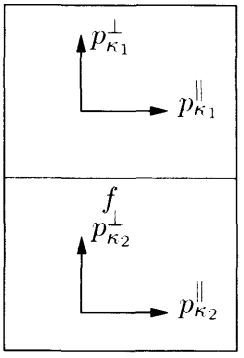


Figure 2.4: Selection of polynomial function $\bar{\mathrm{p}}$ on face f.

Thus, as coercivity has been shown, we are guaranteed the existence of a unique solution to (2.6.5).

2.7.2 Stability for Axiparallel Elements in $\mathbb{R}^2$

In this setting we make the following definition to differentiate between faces of an element

$$\partial\hat{\kappa}^1 := (-1,1) \times \{\pm 1\} \quad \text{and} \quad \partial\hat{\kappa}^2 := \{\pm 1\} \times (-1,1),$$

and similarly

$$\begin{aligned} \partial\tilde{\kappa}^i &:= F_\kappa(\partial\hat{\kappa}^i), \quad i = 1,2, \\ \partial\kappa^i &:= Q_\kappa(\partial\tilde{\kappa}^i), \quad i = 1,2. \end{aligned}$$

We define a new mesh function $\bar{\mathrm{h}}$ in $L^\infty(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})$, as $\bar{\mathrm{h}}(x) = \min\{h_i^\kappa, h_i^{\kappa'}\}$, if x is in the interior of $f = \partial\kappa \cap \partial\kappa'$ for two neighboring elements κ, κ' in the mesh $\mathcal{T}_h$, and $\tilde{f} = Q_\kappa^{-1}(f)$ is parallel to the $\tilde{x}_i$ axis: we also define $\bar{\mathrm{h}}(x) = h_i^\kappa$, if x is in the interior of $f = \partial\kappa \cap \Gamma_{\mathrm{D}}$ and $\tilde{f} = Q_\kappa^{-1}(f)$ is parallel to the $\tilde{x}_i$ axis. We remark that in the restriction to axiparallel images of the unit hypercube in $\mathbb{R}^2$ the mesh function h from Section 2.7.1 collapses to $\bar{\mathrm{h}}$. Similarly, we define $\bar{\mathrm{p}}$ in $L_\infty(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})$ by $\bar{\mathrm{p}}(x) = \max\{p_{\kappa_1,j}, p_{\kappa_2,m}\}$ if x is in the interior of $f = \partial\kappa_1^i \cap \partial\kappa_2^l$, where $i \neq j$ and $l \neq m$ and $\mathrm{p}(x) = p_{\kappa,j}$ if x is in the interior of $\partial\kappa^i \cap \Gamma_{\mathrm{D}}$, where $i \neq j$. Thus, for two elements κ and κ' sharing a face f, $\bar{\mathrm{p}}|_f$ is the maximum of the polynomial degrees of the κ and κ' in the direction orthogonal to f: see Figure 2.4. In this case, coercivity of $B_{\mathrm{DG}}(\cdot,\cdot)$ over $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F}) \times S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ can be shown.

Theorem 2.7.8. *For a mesh $\mathcal{T}_h$ consisting only of axiparallel images of the unit square (up to diffeomorphism), if ϑ is defined as*

$$\vartheta|_f \equiv \vartheta_f = C_\vartheta \frac{\bar{\mathrm{a}}\bar{\mathrm{p}}^2}{\bar{\mathrm{h}}} \quad \text{for} \quad f \subset \partial\kappa_i \cap \partial\kappa_j \in \Gamma_{\text{int}}, \tag{2.7.15}$$

where κ_i and κ_j are two neighbouring elements of $\mathcal{T}_h$, then there exists a positive constant C_s, which depends only on the dimension d and the shape-regularity of $\mathcal{T}_h$, such that

$$B_{\text{DG}}(v, v) \geq C_s |\!|\!| v |\!|\!|_{\text{DG}}^2 \quad \forall v \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F}),$$

provided that the constant C_ϑ is chosen such that:

$$C_\vartheta > C'_\vartheta > 0,$$

where C'_ϑ is a sufficiently large positive constant.

Proof. The proof follows in the same vein as for Theorem 2.7.7, but requires a modification of Lemma 2.7.4 to the case of anisotropic polynomial degrees. See Georgoulis [52] for details. ■

Remark 2.7.9. Theorem 2.7.8 implies that the direction perpendicular to the face of interest is the important one for ensuring stability. Indeed, in the case of anisotropic diffusion, it is also the case that only diffusion perpendicular to the face need be considered; see Georgoulis [52].

Remark 2.7.10. As the inverse and trace inequalities needed for the above coercivity result are proved via tensor product expansions of one–dimensional results, it is relatively simple to extend them, and hence the coercivity results, to $\mathbb{R}^3$. In this situation $\bar{h}$ and $\bar{p}$ will depend on the two length scales and two polynomial degrees perpendicular to the face of interest, respectively.

Chapter 3

Approximation Properties of Anisotropic Spaces

In this chapter we will develop some approximation properties of the anisotropic finite element spaces introduced in the previous chapter. Specifically, we shall derive interpolation estimates for the L^2-projection operator, which we will define momentarily. The standard practice involves determining bounds on the reference element, which are then transformed to the anisotropic physical element. For the case where full orientation of the physical element is allowed we shall extend the approach of Formaggia and Perotto, developed in the series of papers [48, 49, 47]. In these papers only approximation by piecewise linear polynomials was considered; here we generalize these results to include approximation by higher order polynomials, by making use of some results from tensor analysis, cf. [54]. For axiparallel quadrilaterals with anisotropic polynomial degrees, we state the results of Georgoulis [52, 53]. These interpolation results will then be used in the proceeding chapters to develop anisotropic *a priori* estimates for the DG method presented in Chapter 2.

3.1 The L^2-Projection operator

In order to obtain an optimal *a priori* error estimate for our DG scheme it is necessary to develop some error estimates for the orthogonal L^2-projector from $L^2(\hat{\kappa})$ to $\mathcal{R}_{\vec{p}}(\hat{\kappa})$. First

we define this L^2-projection operator, denoted by $\hat{\Pi}_{\vec{p}}$.

Definition 3.1.1. *For $\hat{v} \in L^2(\hat{\kappa})$, we define the L^2-projection operator*

$$\hat{\Pi}_{\vec{p}} \ : \ L^2(\hat{\kappa}) \rightarrow \mathcal{R}_{\vec{p}}$$

by

$$(\hat{\Pi}_{\vec{p}}\hat{v}, \hat{w})_{\hat{\kappa}} \ = \ (\hat{v}, \hat{w})_{\hat{\kappa}} \quad \forall \hat{w} \in \mathcal{R}_{\vec{p}}.$$

Following the definition of the L^2-projection operator on the reference element, $\hat{\kappa}$, we also define the L^2-projection operators on $\tilde{\kappa}$ and κ by means of the mappings F_κ and Q_κ introduced in Section 2.4.

Definition 3.1.2. *We define the L^2-projection operators $\tilde{\Pi}_{\vec{p}}$ and $\Pi_{\vec{p}}$ on $\tilde{\kappa}$ and κ, respectively, by the relations*

$$\begin{aligned} \tilde{\Pi}_{\vec{p}}\tilde{v} &:= (\hat{\Pi}_{\vec{p}}(\tilde{v} \circ F_\kappa)) \circ F_\kappa^{-1}, \\ \Pi_{\vec{p}} v &:= (\tilde{\Pi}_{\vec{p}}(v \circ Q_\kappa)) \circ Q_\kappa^{-1}, \end{aligned}$$

for $\tilde{v} \in L^2(\tilde{\kappa})$ and $v \in L^2(\kappa)$, respectively.

Initially we consider only uniform fixed isotropic polynomial degrees and let $\hat{\Pi}_p$, $\tilde{\Pi}_p$, and Π_p be the respective restrictions of $\hat{\Pi}_{\vec{p}}$, $\tilde{\Pi}_{\vec{p}}$ and $\Pi_{\vec{p}}$ to this case. We quote the following approximation results on the reference element $\hat{\kappa}$.

Lemma 3.1.3. *Let $\hat{\kappa}$ be the reference element, and let $\hat{f}$ denote one of its faces. Given a function $\hat{v} \in H^k(\hat{\kappa})$, the following error bounds hold for $m = 0, 1$:*

$$|\hat{v} - \hat{\Pi}_p\hat{v}|_{H^m(\hat{\kappa})} \ \leq \ C|\hat{v}|_{H^s(\hat{\kappa})}, \qquad m \leq s \leq \min(p+1, k). \tag{3.1.1}$$

$$|\hat{v} - \hat{\Pi}_p\hat{v}|_{H^m(\hat{f})} \ \leq \ C|\hat{v}|_{H^s(\hat{\kappa})}, \quad m+1 \leq s \leq \min(p+1, k). \tag{3.1.2}$$

where C is a positive constant which depends only on the dimension d and polynomial order p.

Proof. The proof of (3.1.1) is standard; see [37], for example. The approximation result (3.1.2) follows upon application of the multiplicative trace inequality (2.7.5), cf. [75]. ■

In order for the above estimates to be of use we must transform them from the reference element $\hat{\kappa}$, to a physical element κ. The following corollary provides a first step in achieving this.

Corollary 3.1.4. *Using the notation of Lemma 3.1.3, there exists a positive constant C, which depends only on the dimension d, and the polynomial order p, such that for $m = 0, 1$:*

$$|v - \Pi_p v|_{H^m(\kappa)} \leq C|\det(J_{F_\kappa})|^{1/2} \|J_{F_\kappa}^{-\top}\|_2^m |\hat{v}|_{H^s(\hat{\kappa})}, \quad m \leq s \leq \min(p+1, k). \tag{3.1.3}$$

$$|v - \Pi_p v|_{H^m(f)} \leq C|m_f|^{1/2} \|J_{F_\kappa}^{-\top}\|_2^m |\hat{v}|_{H^s(\hat{\kappa})}, \quad m+1 \leq s \leq \min(p+1, k). \tag{3.1.4}$$

Proof. Each of the inequalities (3.1.3) and (3.1.4) can be proved by employing a standard scaling argument on the left-hand sides of the respective approximation results in Lemma 3.1.3. Indeed, for (3.1.3), with $m = 0$, the use of (2.4.1) yields

$$\begin{aligned}
\|v - \Pi_p v\|_{L^2(\kappa)}^2 &= \int_{\tilde{\kappa}} \det(J_{Q_\kappa})(\tilde{v} - \tilde{\Pi}_p \tilde{v})^2 \mathrm{d}\tilde{x} \\
&\leq \|\det(J_{Q_\kappa})\|_{L^\infty(\tilde{\kappa})} \|\tilde{v} - \tilde{\Pi}_p \tilde{v}\|_{L^2(\tilde{\kappa})}^2 \\
&\leq C_1 \int_{\hat{\kappa}} \det(J_{F_\kappa})(\hat{v} - \hat{\Pi}_p \hat{v})^2 \mathrm{d}\hat{x} \\
&\leq C_1 |\det(J_{F_{\hat{\kappa}}})| \|\hat{v} - \hat{\Pi}_p \hat{v}\|_{L^2(\hat{\kappa})}^2.
\end{aligned}$$

Similarly for $m = 1$, application of the chain rule twice and employing (2.4.1) yields

$$\begin{aligned}
|v - \Pi_p v|_{H^1(\kappa)}^2 &= \int_{\tilde{\kappa}} \det(J_{Q_\kappa}) |J_{Q_\kappa}^{-T} \tilde{\nabla}(\tilde{v} - \tilde{\Pi}_p \tilde{v})|^2 \mathrm{d}\tilde{x} \\
&\leq \|J_{Q_\kappa}^{-T}\|_{L^\infty(\kappa)}^2 \|\det(J_{Q_\kappa})\|_{L^\infty(\tilde{\kappa})} |\tilde{v} - \tilde{\Pi}_p \tilde{v}|_{H^1(\tilde{\kappa})}^2 \\
&\leq C_1 (C_2)^2 \int_{\hat{\kappa}} \det(J_{F_\kappa}) |J_{F_\kappa}^{-T} \hat{\nabla}(\hat{v} - \hat{\Pi}_p \hat{v})|^2 \mathrm{d}\hat{x} \\
&\leq C_1 (C_2)^2 |\det(J_{F_{\hat{\kappa}}})| \|J_{F_\kappa}^{-T}\|_2^2 |\hat{v} - \hat{\Pi}_p \hat{v}|_{H^1(\hat{\kappa})}^2.
\end{aligned}$$

Hence, for $m = 0, 1$,

$$\|v - \Pi_p v|_{H^m(\kappa)}^2 \leq C|\det(J_{F_{\hat{\kappa}}})| \|J_{F_\kappa}^{-T}\|_2^{2m} \|\hat{v} - \hat{\Pi}_p \hat{v}\|_{H^m(\hat{\kappa})}^2. \tag{3.1.5}$$

Combining (3.1.5) with (3.1.1) gives the desired result (3.1.3).

The bound in (3.1.4) follows analogously, but this time utilizing (2.4.1), (2.4.3) and (2.4.4), results in

$$\begin{aligned}
|v - \Pi_p v|_{H^m(f)}^2 &\leq (C_3)^{2m} |\tilde{v} - \Pi_p \tilde{v}|_{H^m(\tilde{\kappa})}^2 \\
&\leq \frac{(C_3)^{2m} C_6}{C_5} \frac{m_f}{m_f} \|J_{F_\kappa}^{-T}\|_2^{2m} |\hat{v} - \Pi_p \hat{v}|_{H^m(\hat{\kappa})}^2.
\end{aligned} \tag{3.1.6}$$

for $m = 0, 1$. Upon substituting (3.1.6) into (3.1.2) we obtain (3.1.4). ■

With this corollary it now only remains to scale the $H^s(\hat{\kappa})$ semi-norm on the reference element $\hat{\kappa}$ to $\tilde{\kappa}$ in such a way as to extract the anisotropy of the mapping F_κ. To this end, we must first introduce some tensor notation and results.

3.2 Tensor Notation

We use the following definition of a tensor.

Definition 3.2.1. *An entity,*

$$\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}, \quad I_1, \ldots, I_N \in \mathbb{N},$$

is termed a real Nth–order tensor.

A tensor can therefore be regarded as a higher–order generalisation of scalars, vectors and matrices, with scalars being 0th–order tensors, vectors being 1st–order tensors and matrices being 2nd–order tensors. It is standard practice to use lower-case letters to represent scalars $(a, \ldots, \alpha, \ldots)$, bold letters to represent vectors $(\mathbf{b}, \ldots)$ and capital letters to represent matrices $(A, B, \ldots)$; for Nth–order tensors we use calligraphic letters $(\mathcal{A}, \mathcal{B}, \ldots)$. The following discussion regarding the manipulation of tensors is based on the work presented in the article [94].

Definition 3.2.2. *For an Nth order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, the matrix unfolding $A_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$, $n = 1, \ldots, N$, contains the element $a_{i_1 i_2 \ldots i_N}$ at the position with row number i_n and column number equal to*

$$(i_{n+1} - 1) I_{n+2} I_{n+3} \ldots I_N I_2 \ldots I_{n-1} + (i_{n+2} - 1) I_{n+3} I_{n+4} \ldots I_N I_1 I_2 \ldots I_{n-1} + \ldots$$
$$+ (i_N - 1) I_1 I_2 \ldots I_{n-1} + (i_1 - 1) I_2 I_3 \ldots I_{n-1} + (i_2 - 1) I_3 I_4 \ldots I_{n-1} + \ldots + i_{n-1}.$$

In essence a matrix unfolding represents a splitting of an Nth–order tensor into a vector of (N–1)th–order tensors. These (N–1)th–order tensors are then recursively unfolded until 2nd–order tensors (matrices) are realised. Figure 3.1 shows the three unfoldings possible for a 3rd–order tensor.

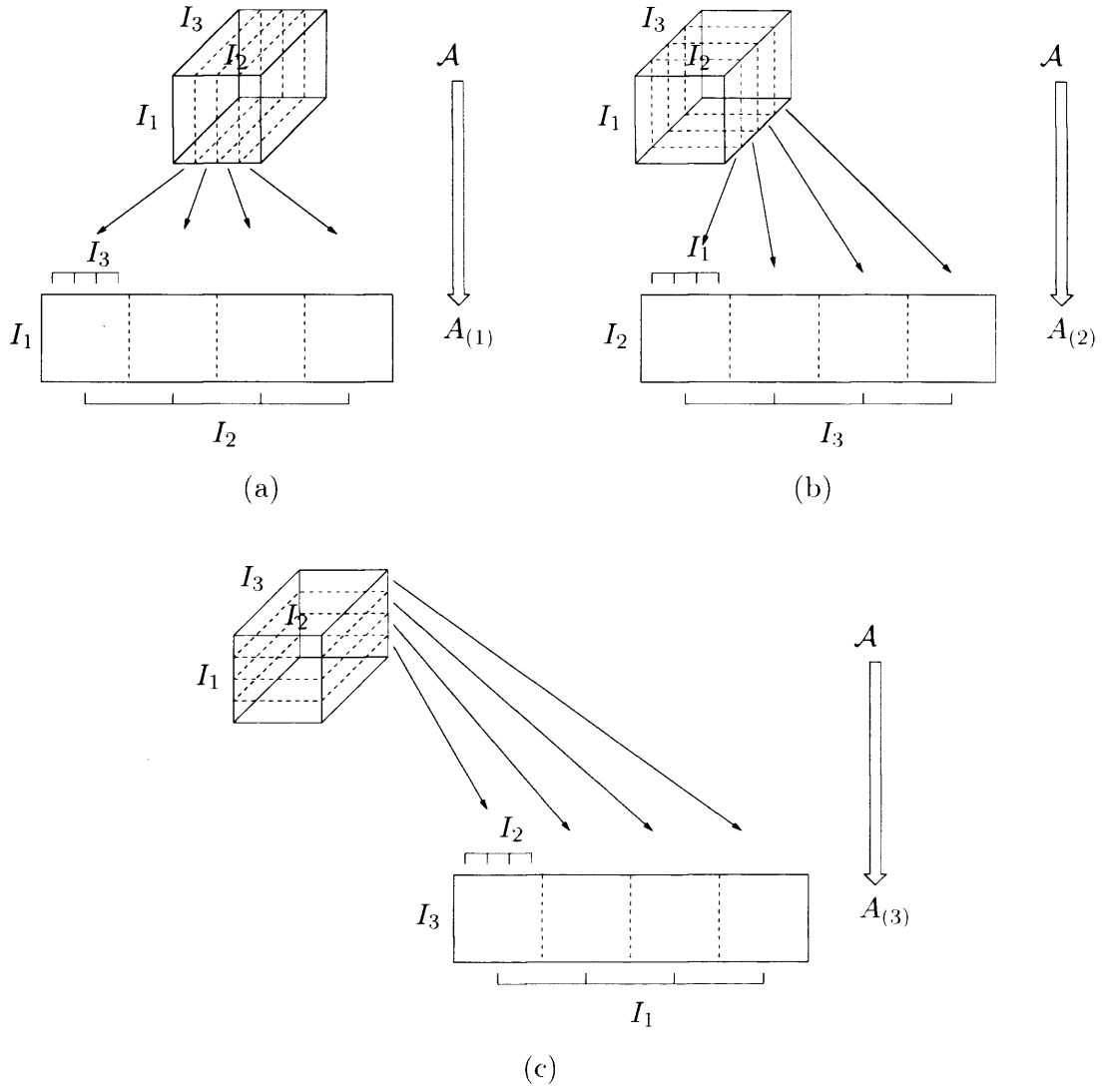


Figure 3.1: Matrix unfolding of a 3rd order tensor: (a) First unfolding (b) Second unfolding (c) Third unfolding.

The definition of a matrix unfolding prompts us to consider a way of multiplying a tensor by a matrix. Clearly, if we have a matrix $U \in \mathbb{R}^{J_n \times I_n}$ then we can pre-multiply $A_{(n)}$ by U. Forming an Nth order tensor from $UA_{(n)}$ by reversing the matrix unfolding procedure we have the product of a matrix and a tensor, giving rise to a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots I_N}$. Diagramatically this can be represented as

$$\underbrace{\mathcal{A} \xrightarrow{\text{Unfold}} A_{(n)} \xrightarrow{U\times} UA_{(n)} \xrightarrow{\text{Refold}} \mathcal{A} \times_n U.}_{\times_n U}$$

We formalize this notion in the following definition.

Definition 3.2.3. *The n-mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n U$, is an $I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots I_N$-tensor of which the entries are given by*

$$(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} := \sum_{i_n=1}^{I_n} (\mathcal{A})_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} (U)_{j_n i_n}.$$

Lemma 3.2.4. *For $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and $U \in \mathbb{R}^{J_n \times I_n}$, we have that*

$$(\mathcal{A} \times_n U)_{(n)} = U\mathcal{A}_{(n)}.$$

Proof. Consider element $(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N}$, its position in $(\mathcal{A} \times_n U)_{(n)}$ is at row number j_n and column number k, where

$$\begin{aligned} k &= (i_{n+1} - 1) I_{n+2} I_{n+3} \ldots I_N I_2 \ldots I_{n-1} + (i_{n+2} - 1) I_{n+3} I_{n+4} \ldots I_N I_1 I_2 \ldots I_{n-1} + \ldots \\ &\quad + (i_N - 1) I_1 I_2 \ldots I_{n-1} + (i_1 - 1) I_2 I_3 \ldots I_{n-1} + (i_2 - 1) I_3 I_4 \ldots I_{n-1} + \ldots + i_{n-1}. \end{aligned}$$

Now,

$$(U\mathcal{A}_{(n)})_{j_n k} = \sum_{i_n=1}^{I_n} (U)_{j_n i_n} (\mathcal{A}_{(n)})_{i_n k} = \sum_{i_n}^{I_n} (\mathcal{A})_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} (U)_{j_n i_n}.$$

Hence, $(\mathcal{A} \times_n U)_{(n)} = U\mathcal{A}_{(n)}$, as required. ■

By considering a vector $\mathbf{v}$, as an $I_n \times 1$ matrix, then an n-mode product of $\mathbf{v}^\top$ and $\mathcal{A}$ can be performed to produce an $I_1 \times I_2 \times \ldots \times I_{n-1} \times 1 \times I_{n+1} \times \ldots \times I_N$-tensor. This tensor could be viewed as an $(N-1)$-tensor, but instead we leave it as an N-tensor in order that

we can form other m-mode products without the value of m having to change. However, if we have a $1 \times 1 \times \ldots \times 1$-tensor then we simply view this as a scalar. The n-mode product satisfies the following properties.

Property 3.2.5. *For a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and the matrices $F \in \mathbb{R}^{J_n \times I_n}$ and $G \in \mathbb{R}^{J_m \times I_m}$, $n \neq m$, we have*

$$(\mathcal{A} \times_n F) \times_m G = (\mathcal{A} \times_m G) \times_n F = \mathcal{A} \times_n F \times_m G.$$

Property 3.2.6. *For a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and the matrices $F \in \mathbb{R}^{J_n \times I_n}$ and $G \in \mathbb{R}^{K_n \times J_n}$, we have*

$$(\mathcal{A} \times_n F) \times_n G = \mathcal{A} \times_n (GF).$$

We also introduce the Frobenius norm of a tensor.

Definition 3.2.7. *The Frobenius-norm $\|\cdot\|_F$ of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is given by*

$$\|\mathcal{A}\|_F^2 = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} (\mathcal{A})_{i_1 i_2 \cdots i_N}^2.$$

Lemma 3.2.8. *Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and an orthogonal matrix $F \in \mathbb{R}^{I_n \times I_n}$, the following holds*

$$\|\mathcal{A} \times_n F\|_F = \|\mathcal{A}\|_F. \tag{3.2.1}$$

Proof. For a matrix $A \in \mathbb{R}^{I_n \times m}$ we have that

$$\|FA\|_F = \|A\|_F. \tag{3.2.2}$$

Using the identity in Lemma 3.2.4, namely, $(\mathcal{A} \times_n F)_{(n)} = F\mathcal{A}_{(n)}$, we deduce that

$$\|\mathcal{A} \times_n F\|_F = \|F\mathcal{A}_{(n)}\|_F.$$

Given that $\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} \ldots I_N \ldots I_1 \ldots I_{n-1}}$, exploiting (3.2.2) gives

$$\|\mathcal{A} \times_n F\|_F = \|F\mathcal{A}_{(n)}\|_F = \|\mathcal{A}_{(n)}\|_F = \|\mathcal{A}\|_F.$$

■

3.3 Approximation Estimates On The Physical Element

We now return to the problem of rescaling the $\hat{H}^s$–seminorms. Here, we adopt the approach of Formaggia and Perotto [48]. This method has a number of attractive properties: first it incorporates all the directional information contained within the map F_κ; secondly unlike other results it is not constrained by the need for a maximal angle condition. Indeed, numerical experiments presented in [48] clearly demonstrate that this approach leads to approximation bounds which show the correct asymptotic behaviour with respect to the maximal angle. In order to perform the rescaling we first note that

$$|\hat{v}|^2_{H^s(\hat{\kappa})} = \int_{\hat{\kappa}} \|\hat{\mathcal{D}}^s(\hat{v})\|^2_F \mathrm{d}\hat{x},$$

where $\hat{\mathcal{D}}^s(\hat{v}) \in \mathbb{R}^{d\times d\times\cdots\times d}$ is the sth order tensor containing the sth order derivatives of $\hat{v}$ with respect to the coordinate system $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_d)$, i.e.,

$$(\hat{\mathcal{D}}^s(\hat{v}))_{i_1,i_2,\ldots,i_s} = \frac{\partial^s \hat{v}}{\partial \hat{x}_{i_1} \cdots \partial \hat{x}_{i_s}}, \quad i_k = 1, \ldots, d, \text{ for } k = 1, \ldots, s.$$

Thereby, for $s = 0$, $\hat{\mathcal{D}}^s(\hat{v}) = \hat{v}$, for $s = 1$, $\hat{\mathcal{D}}^s(\hat{v})$ is the gradient vector, and for $s = 2$, $\hat{\mathcal{D}}^s(\hat{v})$ is the Hessian matrix of second–order derivatives. Similarly, we write $\tilde{\mathcal{D}}^s(\tilde{v}) \in \mathbb{R}^{d\times d\times\cdots\times d}$ to denote the sth–order tensor containing the sth–order derivatives of $\tilde{v}$ with respect to the coordinate system $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_d)$. We now state the following lemma relating $|\hat{v}|^2_{H^s(\hat{\kappa})}$ to $|\tilde{v}|^2_{H^s(\tilde{\kappa})}$.

Lemma 3.3.1. *Under the foregoing assumptions, for $\tilde{v} \in H^s(\tilde{\kappa})$, $s \geq 0$, we have that*

$$|\hat{v}|^2_{H^s(\hat{\kappa})} = |\det(J^{-1}_{F_\kappa})| \int_{\tilde{\kappa}} \|\tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J^\top_{F_\kappa} \times_2 J^\top_{F_\kappa} \times_3 \ldots \times_s J^\top_{F_\kappa}\|^2_F \mathrm{d}\tilde{x}.$$

Proof. The case when $s = 0$ follows trivially. For $s \geq 1$, we first note that the entry $(\hat{\mathcal{D}}^s(\hat{v}))_{i_1 i_2 \ldots i_s}$ may be written in the form

$$\frac{\partial^s \hat{v}}{\partial \hat{x}_{i_1} \cdots \partial \hat{x}_{i_s}} = \sum_{j_1=1}^{d} \cdots \sum_{j_s=1}^{d} (J_{F_\kappa})_{j_1 i_1} \cdots (J_{F_\kappa})_{j_s i_s} \frac{\partial^s \tilde{v}}{\partial \tilde{x}_{j_1} \cdots \partial \tilde{x}_{j_s}},$$

for $i_k = 1, \ldots, d$ and $k = 1, \ldots, s$; this follows by employing an induction argument together with the chain rule. Thereby, from Definition 3.2.3 and Property 3.2.5 above, we deduce that

$$\hat{\mathcal{D}}^s(\hat{v}) = \tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J^\top_{F_\kappa} \times_2 J^\top_{F_\kappa} \times_3 \ldots \times_s J^\top_{F_\kappa}. \tag{3.3.1}$$

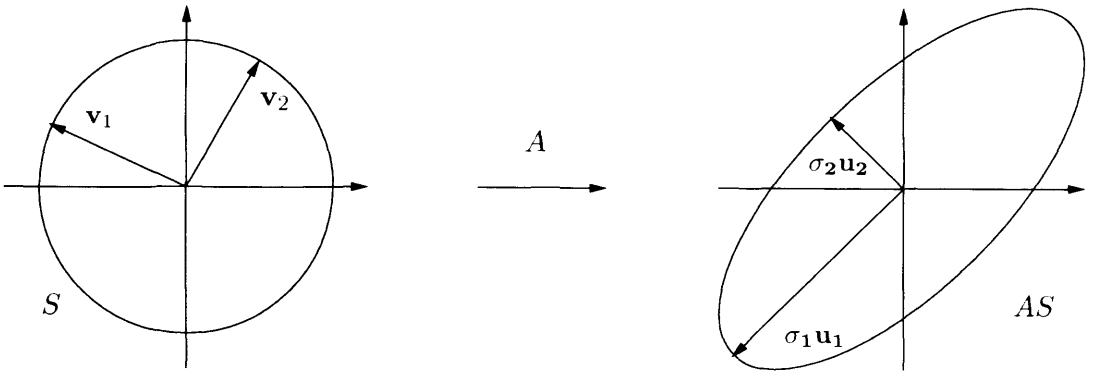


Figure 3.2: SVD of a 2×2 matrix, A.

The statement of the lemma now follows by a simple change of variables.

■

In order to describe the length scales and orientation of the element $\tilde{\kappa}$, we first need the following definition of the Singular Value Decomposition of a matrix.

Definition 3.3.2. *A matrix $A \in \mathbb{R}^{m\times n}$ can be decomposed as follows:*

$$A = U\Sigma V^{\top}.$$

where $U \in \mathbb{R}^{m\times m}$ is an orthogonal matrix termed the left singular matrix, $\Sigma \in \mathbb{R}^{m\times n}$ is a pseudo-diagonal matrix with non-negative entries called the singular values and $V \in \mathbb{R}^{n\times n}$ an orthogonal matrix termed the right singular matrix. This decomposition is called the Singular Value Decomposition (SVD).

Viewing the matrix A as a map, the left singular matrix $U = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$, is composed of orthonormal vectors $\mathbf{u}_i$, $i = 1, \ldots, m$, which are in the direction of the images of the respective orthonormal vectors $\mathbf{v}_i$ of the matrix $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$. It is convention that the singular values σ_i of Σ are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_s \geq 0$ where $s = \min(m, n)$. These singular values represent the stretching factors of the corresponding orthonormal vectors, hence the SVD provides a complete characterisation of the map A. Figure 3.2 shows the physical meaning of the SVD for a matrix $A \in \mathbb{R}^{2\times 2}$. For more information on the Singular Value Decomposition see, for example, Trefethen [131].

For higher order tensors there exists a generalisation of the Singular Value Decomposition, which is expressed in the following Theorem.

Theorem 3.3.3. *Every tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *can be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \ldots \times_N U^{(N)},$$

where

1. $U^{(n)}$ *is an orthogonal* $(I_n \times I_n)$*-matrix.*
2. $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *of which the subtensors* $\mathcal{S}_{i_n=\alpha}$*, obtained by fixing the* n*th index to* α*, have the properties of*
   (a) *all-orthogonality: two subtensors* $\mathcal{S}_{i_n=\alpha}$ *and* $S_{i_n=\beta}$ *are orthogonal for all possible values of* n*,* α *and* β *subject to* $\alpha \neq \beta$*, i.e.*

$$\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0, \quad \text{when } \alpha \neq \beta.$$

   (b) *ordering:*

$$\|\mathcal{S}_{i_n=1}\|_F \geq \|\mathcal{S}_{i_n=2}\|_F \geq \ldots \geq \|\mathcal{S}_{i_n=I_n}\|_F \geq 0$$

   for all possible values of n*.*

The Frobenius-norms $\|\mathcal{S}_{i_n=i}\|_F$ *are the* n*-mode singular values of* $\mathcal{A}$ *and the matrix* $U^{(n)}$ *is the left singular matrix of the* n*th matrix unfolding of* $\mathcal{A}$*.*

Proof. For a proof see De Lathauwer, Moor and Vandewalle, [94]. ■

Remark 3.3.4. Computation of an SVD of a tensor $\mathcal{A}$ is straight forward. The orthogonal matrices $U^{(n)}$ are computed by performing the nth matrix unfolding and then calculating the standard matrix SVD, so that:

$$\mathcal{A}_{(n)} = U^{(n)} \Sigma V.$$

With these matrices calculated, by making use of Property 3.2.5 and the orthogonal nature of each $U^{(n)}$, the core tensor $\mathcal{S}$ can be found as

$$\mathcal{S} = \mathcal{A} \times_1 (U^{(1)})^\top \times_2 (U^{(2)})^\top \times_3 \ldots \times_N (U^{(N)})^\top.$$

The following corollary can also be shown, which extends the notion of an eigenvalue decomposition to higher order tensors.

Corollary 3.3.5. *Suppose $\mathcal{A} \in \mathbb{R}^{n\times n\times\ldots\times n}$ is an Nth order tensor with the following symmetric property:*

$$a_{i_1 i_2 \ldots i_N} = a_{i_N i_1 i_2 \ldots i_{N-1}},$$

then $\mathcal{A}$ has the decomposition

$$\mathcal{A} = \mathcal{S} \times_1 U \times_2 U \times_3 \ldots \times_n U,$$

where U is an orthogonal $(n \times n)$-matrix and $\mathcal{S}$ has the same symmetry as $\mathcal{A}$.

Proof. See De Lathauwer, Moor and Vandewalle, [94]. ■

We perform an SVD decomposition of the Jacobi matrix J_{F_κ} of the affine element mapping F_κ. Thereby, we write

$$J_{F_\kappa} = U_\kappa \Sigma_\kappa V_\kappa^\top. \tag{3.3.2}$$

where U_κ and V_κ are $d \times d$ orthogonal matrices containing the left and right singular vectors of J_{F_κ}, respectively, and $\Sigma_\kappa = \mathrm{diag}(\sigma_{1,\kappa}, \sigma_{2,\kappa}, \ldots, \sigma_{d,\kappa})$ is a $d \times d$ diagonal matrix containing the singular values $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, of J_{F_κ}. Writing $U_\kappa = (\mathbf{u}_{1,\kappa} \ldots \mathbf{u}_{d,\kappa})$, we note that $\mathbf{u}_{i,\kappa}$, $i = 1, \ldots, d$, give the direction of stretching of the element κ, while $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, give the stretching lengths in the respective directions. Indeed, for axiparallel meshes, as considered in Section 2.7, for example, $\mathbf{u}_{i,\kappa}$, $i = 1, \ldots, d$ will be parallel to the coordinates axes and $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, will denote the local mesh lengths in the respective coordinate direction. With this decomposition we now give the following lemma.

Lemma 3.3.6. *Under the foregoing assumptions, the following identity holds*

$$\begin{aligned}
&\|\tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J_{F_\kappa}^\top \times_2 J_{F_\kappa}^\top \times_3 \ldots \times_s J_{F_\kappa}^\top\|_F^2 \\
&= \sum_{i_1=1}^{d} \sum_{i_2=1}^{d} \cdots \sum_{i_s=1}^{d} (\sigma_{i_1,\kappa}\sigma_{i_2,\kappa} \ldots \sigma_{i_s,\kappa})^2 (\tilde{\mathcal{D}}^s(\tilde{v}) \times_1 \mathbf{u}_{i_1,\kappa}^\top \times_2 \mathbf{u}_{i_2,\kappa}^\top \times_3 \ldots \times_s \mathbf{u}_{i_s,\kappa}^\top)^2 \\
&\equiv D_\kappa^s(\tilde{v}, \Sigma_\kappa, U_\kappa).
\end{aligned} \tag{3.3.3}$$

Proof. The result follows after performing the SVD as in (3.3.2), using Lemma 3.2.8 and rearranging. ■

Remark 3.3.7. We note that, should the mapping F_κ yield a near isotropic element κ, then upon defining the standard isotropic mesh size, h_κ, (see [61], for example) by

$$h_\kappa := \operatorname{diam}(\kappa). \tag{3.3.4}$$

we have

$$\sigma_{i,\kappa} \sim h_\kappa, \quad i = 1, \ldots, d.$$

Hence, in this isotropic setting

$$\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \mathrm{d}x \leq C h_\kappa^{2s} |\tilde{v}|_{H^s(\tilde{\kappa})}.$$

We are now in a position to provide the main result from this section.

Theorem 3.3.8. *Using the notation of Lemma 3.1.3, there exists a positive constant C, which depends only on the dimension d and the polynomial order p, such that for $m = 0, 1$:*

$$\begin{aligned}
|v - \Pi_p v|_{H^m(\kappa)} &\leq C |\sigma_{d,\kappa}|^{-m} \left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right]^{1/2}, \quad m \leq s \leq \min(p+1, k), \\
\|v - \Pi_p v\|_{L^2(f)} &\leq C |\sigma_{d,\kappa}|^{-1/2} \left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right]^{1/2}, \quad 1 \leq s \leq \min(p+1, k), \\
|v - \Pi_p v|_{H^1(f)} &\leq C \left| \frac{m_f}{m_\kappa} \right|^{1/2} |\sigma_{d,\kappa}|^{-1} \left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right]^{1/2}, \quad 2 \leq s \leq \min(p+1, k).
\end{aligned}$$

Proof. Initially we substitute (3.3.3) into the result from Lemma 3.1.3, to obtain

$$|\hat{v}|^2_{H^s(\hat{\kappa})} = |\det(J_{F_\kappa}^{-1})| \int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x}. \tag{3.3.5}$$

Making the following observations

$$|\det(J_{F_\kappa})| = \Pi_{i=1}^d \sigma_{i,\kappa}, \quad \|J_{F_\kappa}^{-\top}\|_2 = 1/\sigma_{d,\kappa}, \quad m_f \leq C_7 \Pi_{i=1}^{d-1} \sigma_{i,\kappa}. \tag{3.3.6}$$

where C_7 is a positive constant independent of the element size, we use (3.3.5) in Corollary 3.1.4 to complete the proof. ■

Remark 3.3.9. For the purposes of deriving the forthcoming *a priori* error bound on the error in the computed target functional, cf. Theorem 4.1.5 in the next chapter, it is convenient to leave the statement of the third approximation result above in terms of m_f and m_κ, rather than in terms of the stretching factors $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, solely, since these quantities naturally arise within the definition of the discontinuity-penalization parameter ϑ defined in (2.7.9).

3.4 *hp*-Error Bounds On The Hypercube

For completeness we also state the following p-dependent interpolation results in the case of isotropic and anisotropic polynomial degrees on the d-hypercube.

3.4.1 Isotropic Polynomial Degrees

Lemma 3.4.1. *Let $\hat{\kappa}$ be the unit d-hypercube, and let $\hat{f}$ denote one of its faces. Given a function $\hat{v} \in H^k(\hat{\kappa})$, the following error bounds hold*

$$\|\hat{v} - \hat{\Pi}_p \hat{v}\|_{L^2(\hat{\kappa})} \leq \frac{C}{p^s} |\hat{v}|_{H^s(\hat{\kappa})}, \qquad \|\hat{v} - \hat{\Pi}_p \hat{v}\|_{L^2(\hat{f})} \leq \frac{C}{p^{s-1/2}} |\hat{v}|_{H^s(\hat{\kappa})},$$

$$|\hat{v} - \hat{\Pi}_p \hat{v}|_{H^1(\hat{\kappa})} \leq \frac{C}{p^{s-3/2}} |\hat{v}|_{H^s(\hat{\kappa})}, \qquad |\hat{v} - \hat{\Pi}_p \hat{v}|_{H^1(\hat{f})} \leq \frac{C}{p^{s-5/2}} |\hat{v}|_{H^s(\hat{\kappa})},$$

where $1 \leq s \leq \min(p+1, k)$ and in each case C is a constant dependent only on the dimension, d.

Proof. A proof can be found in Houston, Schwab and Süli [76], for example; see also Canuto and Quarteroni [33]. ■

Rescaling to the physical element we easily attain the following result.

Lemma 3.4.2. *Using the notation of Lemma 3.1.3, there exists a positive constant C,*

which depends only on the dimension d such that:

$$\|v-\Pi_p v\|_{L^2(\kappa)} \leq \frac{C}{p^s}\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\right]^{\frac{1}{2}}, \quad 0\leq s\leq \min(p+1,k). \tag{3.4.1}$$

$$|v-\Pi_p v|_{H^1(\kappa)} \leq \frac{C}{p^{s-3/2}}|\sigma_{d,\kappa}|^{-1} \times\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\right]^{\frac{1}{2}}, \quad 1\leq s\leq \min(p+1,k). \tag{3.4.2}$$

$$\|v-\Pi_p v\|_{L^2(f)} \leq \frac{C}{p^{s-1/2}}|\sigma_{d,\kappa}|^{-1/2} \times\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\right]^{\frac{1}{2}}, \quad 1\leq s\leq \min(p+1,k). \tag{3.4.3}$$

$$|v-\Pi_p v|_{H^1(f)} \leq \frac{C}{p^{s-5/2}}\left|\frac{m_f}{m_\kappa}\right|^{\frac{1}{2}}|\sigma_{d,\kappa}|^{-1} \times\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\right]^{\frac{1}{2}}, \quad 2\leq s\leq \min(p+1,k). \tag{3.4.4}$$

Proof. The same arguments from Corollary 3.1.4 and Lemma 3.3.6 can be applied to the results from Lemma 3.4.1 in order to derive the above results. ■

Remark 3.4.3. Considering once again isotropic elements and bearing in mind Remark 3.3.7 we see that Lemma 3.4.2 shares exactly the same convergence results in terms of both h_κ and p as Lemma 4.3 of [61]. Indeed, all four results from Lemma 3.4.2 show h-optimal convergence rates, with the errors in the L^2–norm exhibiting p-optimal convergence; however, both H^1–bounds are p–suboptimal, with (3.4.2) and (3.4.4) suboptimal by $p^{1/2}$ and p, respectively. Optimal hp-convergence rates have been shown for alternative projection operators; see, for example, Georgoulis [52], however, as we shall see, the L^2–projector is required for the functional *a priori* analysis in the next chapter to give h–optimal convergence rates.

3.4.2 Anisotropic Polynomial Degrees

Returning to the 2–dimensional axiparallel setting introduced in Section 2.7, where anisotropic polynomial degrees are admissible, we introduce the quantity $\Phi(p,s,h)$ by

$$\Phi_1(p,s,h) := \left(\frac{(p-(s-1))!}{(p+(s-1))!}\right)\left(\frac{h}{2}\right)^{2(s-1)}. \tag{3.4.5}$$

where p and s are integers such that $1 \leq s \leq p$. The following interpolation estimates then hold.

Lemma 3.4.4. *Let $u \in H^k(\kappa)$, for $k \geq 2$: then, for $\tilde{u} := u \circ Q_\kappa$, $\vec{p} = (p_1, p_2)$ and $p_1, p_2 \geq 1$, we have*

$$\|u - \Pi_{\vec{p}} u\|_\kappa^2 \leq C_\kappa M_\kappa^0. \tag{3.4.6}$$

where

$$M_\kappa^0 := \sum_{i=1}^{2} \Phi(p_i, s_i, h_i) \left(\frac{h_i}{2p_i}\right)^2 \|\tilde{\partial}_i^{s_i} \tilde{u}\|_{\tilde{\kappa}}^2. \tag{3.4.7}$$

and

$$\|\partial_i (u - \Pi_{\vec{p}} u)\|_\kappa^2 \leq C_\kappa^1 M_{\kappa,i}^1 + C_\kappa^2 M_{\kappa,j}^1. \tag{3.4.8}$$

with

$$\begin{aligned} M_{\kappa,i}^1 \;:=\; & p_i \Phi(p_i, s_i, h_i) \|\tilde{\partial}_i^{s_i} \tilde{u}\|_{\tilde{\kappa}}^2 \\ & + \Phi(p_j, s_j, h_j) \|\tilde{\partial}_j^{s_j - 1} \tilde{\partial}_i \tilde{u}\|_{\tilde{\kappa}}^2. \end{aligned} \tag{3.4.9}$$

where $i, j = 1, 2$, $i \neq j$, $1 \leq s_i \leq \min\{p_i + 1, k\}$, for $i = 1, 2$, and $\tilde{\partial}_i$ is the partial derivative in the $\tilde{x}_i$-direction in the $\tilde{x}_1 \tilde{x}_2$-plane.

Lemma 3.4.5. *Let $u \in H^k(\kappa)$, with $k \geq 1$: then we have*

$$\|u - \Pi_{\vec{p}} u\|_{\partial\kappa_i}^2 \leq C_\kappa M_{\partial\kappa,i}^0. \tag{3.4.10}$$

with

$$\begin{aligned} M_{\partial\kappa,i}^0 \;:=\; & \Phi(p_j, s_j, h_j) \frac{h_j}{2p_j} \|\tilde{\partial}_j^{s_j} \tilde{u}\|_{\tilde{\kappa}}^2 \\ & + \Phi(p_i, s_i, h_i) \frac{h_i}{h_j} \frac{h_i}{2p_i} \|\tilde{\partial}_i^{s_i} \tilde{u}\|_{\tilde{\kappa}}^2 \\ & + \left(\frac{p_j}{p_i} + 1\right) \Phi(p_i, s_i, h_i) \frac{h_j}{2p_j} \|\tilde{\partial}_i^{s_i - 1} \tilde{\partial}_j \tilde{u}\|_{\tilde{\kappa}}^2. \end{aligned}$$

with $i, j = 1, 2$, $i \neq j$, $1 \leq s_i \leq \min\{p_i + 1, k\}$, and $p_i \geq 1$, for $i = 1, 2$.

Lemma 3.4.6. *Let $u \in H^k(\kappa)$, with $k \geq 2$: then the following error estimates hold:*

$$\|\partial_i (u - \Pi_{\vec{p}} u)\|_{\partial\kappa_i}^2 \;\leq\; C_\kappa^1 M_{\partial\kappa,i}^1 + C_\kappa^2 M_{\partial\kappa,i}^2. \tag{3.4.11}$$

$$\|\partial_j (u - \Pi_{\vec{p}} u)\|_{\partial\kappa_i}^2 \;\leq\; C_\kappa^1 M_{\partial\kappa,i}^2 + C_\kappa^2 M_{\partial\kappa,i}^1. \tag{3.4.12}$$

with

$$\begin{aligned} M^1_{\partial\kappa,i} &:= \Phi(p_i, s_i, h_i)\frac{2p_i}{h_i}\Big(p_i\frac{h_i}{h_j}\|\tilde{\partial}_i^{s_i}\tilde{u}\|^2_{\tilde{\kappa}} \\ &\quad +\Big(1+\frac{p_i}{p_j}\Big)\frac{h_j}{h_i}\|\tilde{\partial}_i^{s_i-1}\tilde{\partial}_j\tilde{u}\|^2_{\tilde{\kappa}}\Big) \\ &\quad +\Phi(p_j, s_j, h_j)\frac{2p_j}{h_j}\|\tilde{\partial}_j^{s_j-1}\tilde{\partial}_i\tilde{u}\|^2_{\tilde{\kappa}}. \end{aligned} \tag{3.4.13}$$

for $i, j = 1, 2$, $i \neq j$, $1 \leq s_i \leq \min\{p_i + 1, k\}$, $p_i \geq 1$, $i = 1, 2$, *and*

$$\begin{aligned} M^2_{\partial\kappa,i} &:= p_j^2\Phi(p_j, s_j, h_j)\frac{2p_j}{h_j}\|\tilde{\partial}_j^{s_j}\tilde{u}\|^2_{\tilde{\kappa}} \\ &\quad +p_j\Phi(p_i, s_i, h_i)\frac{2p_j}{h_j}\|\tilde{\partial}_i^{s_i-1}\tilde{\partial}_j\tilde{u}\|^2_{\tilde{\kappa}}, \end{aligned} \tag{3.4.14}$$

for $2 \leq s_i \leq \min\{p_i + 1, k\}$.

Proof. For each of the lemmata 3.4.4-3.4.6 full proofs can be found in Georgoulis [53]. In each case, the idea is to split up the L^2-projection operator on the reference element into a tensor-product composition of one-dimensional L^2-projectors and apply one dimensional results (for example, see Schwab [119]); scaling back to the physical element then completes the proof. ■

Remark 3.4.7. By using *Stirling's formula*

$$n! \sim \sqrt{2\pi}n^{n+1/2}\mathrm{e}^{-n}, \quad n > 0, \tag{3.4.15}$$

we see that for $p \geq 1$,

$$\Phi(p, s, h) \leq C(s)p^{-2(s-1)}h^{2(s-1)}. \tag{3.4.16}$$

Thus, if we consider isotropic polynomial degrees in the results from the Lemmata 3.4.4-3.4.6 and apply (3.4.16) we return to the same asymptotic results in terms of p as for Lemma 3.4.2. Considering also isotropic h, that is $h_1^\kappa \sim h_2^\kappa$, then we recover the same approximation results from Harriman *et al.* [61].

We shall now consider the case where the functions we are approximating are analytic. In this case we shall see that the L^2 projection provides p-exponential convergence, a very desirable property, which improves on the merely algebraic convergence in p witnessed

above. In order to see this we present the following lemma, which is a slight modification of Lemma 3.42 from [52].

Lemma 3.4.8. *Let $u : \kappa \to \mathbb{R}$ have an analytic extension to an open neighbourhood of $\bar{\kappa}$. Also, let p, s, and n be positive integers such that*

$$0 \leq n \leq s := \alpha p + n \leq p.$$

with $0 < \alpha < 1$. Then the following bound holds

$$\Phi(p, s+1, h)\|\partial_i^{s+1}\partial_j^m u\|^2_{L^2(\kappa)} \leq C_u h^{2s} p^{\min\{3.n+\frac{5}{2}\}} \mathrm{e}^{-rp} m_\kappa. \tag{3.4.17}$$

where $m \in \{0, 1\}$ and r, $C_u > 0$ are constants that depend on n and u, with $i, j \in \{1, 2\}$ for $i \neq j$, and m_κ denotes the Lebesgue measure of the domain κ.

Proof. We assume that $i = 1$ as the proof is analogous for $i = 2$. Since u is analytic in a neighbourhood of $\bar{\kappa}$, its derivatives satisfy the bound

$$\|\partial_i^{s+1}\partial_j^m u\|_{L^\infty(\kappa)} \leq C(d_1)^{s+1}(d_2)^m (s+1)!m!.$$

in which case

$$\|\partial_i^{s+1}\partial_j^m u\|_{L^2(\kappa)} \leq C(d_1)^{s+1}(d_2)^m (s+1)!m! m_\kappa^{\frac{1}{2}}. \tag{3.4.18}$$

where C is a generic positive constant and $d_i > 1$, $i = 1, 2$. Using (3.4.18), utilizing Stirling's formula (3.4.15) and recalling $s = \alpha p + n$, $n \geq 0$, we obtain

$$\begin{aligned} \Phi(p, s+1, h)\|\partial_i^{s+1}\partial_j^m u\|^2_{L^2(\kappa)} \quad &\leq \quad C h^{2s} d_2^{2m} d_1^{2\alpha p + 2n + 2} (\alpha p + n + 1)^{2\alpha p + 2n + 3} \\ &\quad \times \frac{(p(1-\alpha) - n)^{p(1-\alpha) - n + \frac{1}{2}}}{(p(1+\alpha) + n)^{p(1+\alpha) + n + \frac{1}{2}}} m_\kappa. \end{aligned} \tag{3.4.19}$$

for sufficiently large p. The denominator of the fraction in (3.4.19) can be bounded from below as follows:

$$(p(1+\alpha) + n)^{p(1+\alpha)+n+\frac{1}{2}} \geq (p(1+\alpha))^{p(1+\alpha)+n+\frac{1}{2}} \geq (1+\alpha)^{p(1+\alpha)} p^{p(1+\alpha)+n+\frac{1}{2}}.$$

and assuming that $p(1-\alpha) \geq n$, the numerator can be bounded from above as such

$$(p(1-\alpha) - n)^{p(1-\alpha)-n+\frac{1}{2}} \leq (p(1-\alpha))^{p(1-\alpha)-n+\frac{1}{2}}.$$

By considering two cases we can further bound $(p(1-\alpha))^{p(1-\alpha)-n+\frac{1}{2}}$:

- if $0 \leq n \leq \frac{1}{2}$, then

$$(p(1-\alpha))^{p(1-\alpha)-n+\frac{1}{2}} \leq (1-\alpha)^{p(1-\alpha)} p^{p(1-\alpha)+\frac{1}{2}-n};$$

- if $n \geq \frac{1}{2}$, then

$$(p(1-\alpha))^{p(1-\alpha)-n+\frac{1}{2}} \leq (1-\alpha)^{p(1-\alpha)} p^{p(1-\alpha)} n^{-n+\frac{1}{2}}.$$

Thus, combining the above results, the right-hand side of (3.4.19) can be bounded as follows

$$\begin{aligned}
\Phi(p, s+1, h)\|\partial_i^{s+1}\partial_j^m u\|^2_{L^2(\kappa)} &\leq Ch^{2s} d_2^{2m} d_1^{2\alpha p+2n+2} \left(\frac{\alpha p+n+1}{p}\right)^{2\alpha p+2n} p^{\min\{3,n+\frac{5}{2}\}} \\
&\quad \times \left(\frac{(1-\alpha)^{1-\alpha}}{(1+\alpha)^{1+\alpha}}\right)^p m_\kappa \\
&\leq Ch^{2s} d_2^{2m} d_1^2 (2\alpha)^{2n} p^{\min\{3,n+\frac{5}{2}\}} \\
&\quad \times \left(\frac{(1-\alpha)^{1-\alpha}}{(1+\alpha)^{1+\alpha}} (2\alpha d_1)^{2\alpha}\right)^p m_\kappa.
\end{aligned}$$

for sufficiently large p. So, if we can find an α for which

$$F(\alpha, d_1) := \frac{(1-\alpha)^{1-\alpha}}{(1+\alpha)^{1+\alpha}} (2\alpha d_1)^{2\alpha} < 1.$$

then we have recovered the p-exponential convergence. Indeed, it can be shown that for $0 < \alpha < 1$ and $d_1 > 1$, $F(\alpha, d_1)$ attains a minimum at $\alpha_{\min} := (1+4d_1^2)^{-\frac{1}{2}}$ and in this case $F(\alpha_{\min}, d_1) < 1$. Thereby, choosing $r := \frac{1}{2}|\log(F(\alpha_{\min,d_1}))|$, we obtain the stated result. ∎

In a similar vein the following also holds.

Lemma 3.4.9. *Let $u : \kappa \to \mathbb{R}$ have an analytic extension to an open neighbourhood of $\bar{\kappa}$. Also, let p, s, and n be positive integers such that*

$$0 \leq n \leq s := \alpha p + n \leq p.$$

with $0 < \alpha < 1$. Then the following bound holds

$$\Phi(p, s+1, h)\|\partial_i^s \partial_j^m u\|^2_{L^2(\kappa)} \leq C_u h^{2s} p^{\min\{3,n+\frac{5}{2}\}} \mathrm{e}^{-rp} m_\kappa. \tag{3.4.20}$$

where $m \in \{0, 1\}$ and r, $C_u > 0$ are constants that depend on n and u, with $i, j \in \{1, 2\}$ for $i \neq j$, and m_κ denotes the Lebesgue measure of the domain κ.

We notice that the results of Lemmas 3.4.4-3.4.6 all include terms of the form

$$\Phi(p, s, h)\|\partial_i^{s-1} u\|^2_{L^2(\kappa)} \text{ or } \Phi(p, s, h)\|\partial_i^s \partial_j u\|^2_{L^2(\kappa)}.$$

and hence Lemmas 3.4.8 and 3.4.9 can be used to show that, for an analytic function u, the L^2-projector achieves p-exponential convergence in both the L^2-norm and H^1 semi-norm on the element and the element boundary.

Chapter 4

A Priori Error Analysis

In this chapter we will use the anisotropic h-interpolation estimates of the previous chapter to derive an anisotropic h-error bound for the SIP DG method. In particular, we shall focus on the estimation of certain linear functionals of the solution, rather than some norm of the solution. In this way the main result of the chapter will generalize the *a priori* estimates given in Harriman *et al.* [61] to anisotropic meshes; see also [54]. First, we discuss the importance of functional estimation and give some examples. Then, we introduce the so-called dual problem, on whose solution the *a priori* estimates will depend, a complete derivation of the estimates then follows.

4.1 Goal Oriented Anisotropic *A Priori* Error Estimates

Often it is not of practical value to have knowledge of some norm of the solution, but rather we might want to use our computed solution to calculate some other quantity of interest, the *Goal*. Error control in this sense is particularly important in engineering applications; e.g. in fluid dynamics one may be concerned with calculating the lift and drag coefficients of a body immersed into a viscous fluid whose flow is governed by the Navier–Stokes equations. The lift and drag coefficients are defined as integrals, over the boundary of the body, of the stress tensor components normal and tangential to the flow, respectively. Similarly, in elasticity theory, the quantities of interest, such as the stress intensity factor or the moments of a shell or plate, are derived quantities. In acoustic and

electromagnetic theory the quantity of interest is often the far-field pattern. Moreover, the same general theory is also applicable to the numerical computation of eigenvalue problems. In this section, we derive anisotropic h-error bounds in the case when the goal is a linear target functional of the solution.

4.1.1 The Dual Problem and Specific Linear Functionals

Suppose that we wish to control the discretisation error in a generic linear target functional $J(\cdot)$ acting on the solution. We introduce the following *dual* or *adjoint* problem: find $z \in H^2(\Omega, \mathcal{T}_h)$ such that

$$B_{\mathrm{DG}}(w, z) = J(w) \quad \forall w \in H^2(\Omega, \mathcal{T}_h). \tag{4.1.1}$$

The well-posedness of (4.1.1) depends on the choice of functional $J(\cdot)$. We will assume that a unique solution to (4.1.1) does exist: a discussion concerning the validity of this assumption can be found in Houston and Süli [79]. Three popular choices for $J(\cdot)$ covered by our hypothesis are

- Weighted Average:

$$J(u) \equiv M_\psi(u) = \int_\Omega u\psi \mathrm{d}x,$$

  where $\psi \in L^2(\Omega)$.

- Point Value:

$$J(u) = u(x_0),$$

  where $x_0 \in \Omega$.

- Weighted Boundary Flux:

$$J(u) \equiv N_\psi(u) = \int_\Gamma (a\nabla u \cdot \mathbf{n} + (\mathbf{b} \cdot \mathbf{n})u)\psi \mathrm{d}s,$$

  where $\psi \in L^2(\Gamma)$.

By performing an integration by parts of the bilinear form $B_{\mathrm{DG}}(\cdot, \cdot)$, we can see that in each case we have recovered a weak formulation of the partial differential equation

$$L^*z \equiv \nabla \cdot (a\nabla z) - \mathbf{b} \cdot \nabla z + cz = \omega_1 \text{ in } \kappa.$$

subject to boundary conditions

$$\begin{aligned} z &= \omega_2 \text{ on } \partial\kappa \cap (\Gamma_D \cup \Gamma_+). \\ \vartheta z &= 0 \text{ on } \partial\kappa \cap \Gamma_D. \\ (\mathbf{b}\cdot\mathbf{n}_\kappa)z + (a\nabla z)\cdot\mathbf{n}_\kappa &= \omega_3 \text{ on } \partial\kappa \cap \Gamma_N. \end{aligned}$$

where ω_1, ω_2 and ω_3 are functions dependent on the functional of interest.

Remark 4.1.1. When the functional of interest is a point value and $\mathcal{L}$ is purely hyperbolic, the resulting weak solution z of (4.1.1) does not belong to $L^2(\Omega)$. Thus, to avoid technical complications, J should be mollified to be a special case of the weighted average, where ψ is chosen so that $M_\psi(u)$ is some approximation to $u(x_0)$. Details can be found in Houston and Süli [79].

For the weighted boundary flux, on Γ_D a consistent reformulation of the functional is required to obtain optimal convergence rates, details can be found in Harriman, Gavaghan and Süli [60].

4.1.2 *A Priori* Error Analysis

We are now in a position to perform an *a priori* analysis of the error between the actual target functional value $J(u)$, and the estimate $J(u_{DG})$, defined as $|J(u) - J(u_{DG})|$. The following generalizes the *a priori* error estimates of Harriman *et al.* [61] to anisotropic meshes, by using the anisotropic estimates from Theorem 3.3.8. We present some preliminary lemmata and give the main result of this chapter in Theorem 4.1.5.

From now on we make the following assumption

$$\mathbf{b}\cdot\nabla_{\mathcal{T}_h} v \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F}) \quad \forall v \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F}). \tag{4.1.2}$$

and for simplicity, once again, assume the entries of the matrix a are constant on each element $\kappa \in \mathcal{T}_h$, *i.e.*,

$$a \in [S^{\mathbf{0}}(\Omega, \mathcal{T}_h, \mathbf{F})]^{d\times d}_{\text{sym}}. \tag{4.1.3}$$

and let $a_\kappa := a|_\kappa$. We also define $\bar{a} = |\sqrt{a}|_2^2$, cf. Section 2.7, and write $\bar{a}_\kappa = \bar{a}|_\kappa$ and hence

$$\bar{a}_{\tilde{\kappa}} := \max_{\kappa'}\{\bar{a}_{\kappa'}\}. \tag{4.1.4}$$

where κ' are those elements (including κ itself) which share a $(d-1)$-dimensional face with κ.

Remark 4.1.2. We note that hypothesis (4.1.2) is a standard condition assumed for the analysis of the *hp*–version of the DG method: see, for example, [53, 61, 76]. Indeed, this condition is essential for the derivation of *a priori* error bounds which are optimal in both the mesh size h and spectral order p: in the absence of this assumption, optimal h–convergence bounds may still be derived, though a loss of $p^{1/2}$ is observed in the resulting error analysis, unless the scheme (2.6.5) is supplemented by appropriate streamline–diffusion stabilization, cf. the discussion in [75].

Initially we decompose the global error $u - u_{\mathrm{DG}}$ as follows

$$u - u_{\mathrm{DG}} = (u - \Pi_p u) + (\Pi_p u - u_{\mathrm{DG}}) \equiv \eta + \xi. \tag{4.1.5}$$

where Π_p denotes the L^2-projection operator introduced in Chapter 3. Hence, we are able to estimate the error η in a number of norms following the discussion in Chapter 3, but, as yet, we know nothing about the estimation of ξ. To this end, we present the following lemma, which bounds the DG-norm of ξ in terms of η.

Lemma 4.1.3. *Assume that (2.3.2) and (4.1.2) hold and let* $\gamma_1|_\kappa = \|c/c_0\|^2_{L^\infty(\kappa)}$*: then the functions* ξ *and* η *defined by (4.1.5) satisfy the following inequality.*

$$\begin{aligned} |||\xi|||^2_{\mathrm{DG}} \leq C \Bigg[\sum_{\kappa \in \mathcal{T}_h} \Big(& \|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \gamma_1 \|\eta\|^2_{L^2(\kappa)} + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma} \\ & + 2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))} \Big) \Bigg]. \end{aligned} \tag{4.1.6}$$

where C is a positive constant that depends only on the dimension d, and the polynomial degree p.

Proof. We begin by first making use of the coercivity result (2.7.10) and Galerkin orthogonality result (2.6.8) as follows:

$$\begin{aligned} |||\xi|||^2_{\mathrm{DG}} &\leq C B_{\mathrm{DG}}(\xi, \xi) \\ &= -C B_{\mathrm{DG}}(\eta, \xi) \\ &\leq C |B_{\mathrm{DG}}(\eta, \xi)|. \end{aligned} \tag{4.1.7}$$

which holds for $\xi \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ and $\Pi_p u \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ by assumption. We now bound $B_{\mathrm{DG}}(\eta, \xi)$ from above; to this end, we write

$$B_{\mathrm{DG}}(\eta, \xi) \equiv I_1 + I_2 + I_3 + I_4 + I_5 + I_6.$$

where

$$\begin{aligned}
I_1 &= \sum_{\kappa \in \mathcal{T}_h} \int_\kappa a \nabla \eta \cdot \nabla \xi \mathrm{d}x, \\
I_2 &= \sum_{\kappa \in \mathcal{T}_h} -\int_\kappa (\eta \mathbf{b} \cdot \nabla \xi - c\eta\xi)\mathrm{d}x = -\sum_{\kappa \in \mathcal{T}_h} \int_\kappa c\eta\xi \mathrm{d}x, \\
I_3 &= \sum_{\kappa \in \mathcal{T}_h} \left(\int_{\partial_+ \kappa} (\mathbf{b} \cdot \mathbf{n}_\kappa)\eta^+ \xi^+ \mathrm{d}s + \int_{\partial_- \kappa \setminus \Gamma} (\mathbf{b} \cdot \mathbf{n}_\kappa)\eta^- \xi^+ \mathrm{d}s \right), \\
I_4 &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \langle (a \nabla \eta) \cdot \mathbf{n}_f \rangle [\xi] \mathrm{d}s, \\
I_5 &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \langle (a \nabla \xi) \cdot \mathbf{n}_f \rangle [\eta] \mathrm{d}s, \\
I_6 &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \vartheta [\eta][\xi] \mathrm{d}s.
\end{aligned}$$

For term I_2 we have used condition (4.1.2) to ensure $(\eta, \mathbf{b} \cdot \nabla_{\mathcal{T}_h} \xi) = 0$, by definiton of the L^2-projection operator. Terms I_1, I_2, I_4, I_5, and I_6 can now be bounded by use of the Cauchy-Schwarz inequality as follows

$$\begin{aligned}
|I_1| &\leq \sum_{\kappa \in \mathcal{T}_h} \|\sqrt{a} \nabla \eta\|_{L^2(\kappa)} \|\sqrt{a} \nabla \xi\|_{L^2(\kappa)}, \\
|I_2| &= |\sum_{\kappa \in \mathcal{T}_h} \int_\kappa \frac{c}{c_0} c_0 \eta \xi \mathrm{d}x| \leq \sum_{\kappa \in \mathcal{T}_h} \left\| \frac{c}{c_0} \eta \right\|_{L^2(\kappa)} \|c_0 \xi\|_{L^2(\kappa)}, \\
|I_4| &\leq \|\vartheta^{-\frac{1}{2}} \langle a \nabla \eta \rangle\|_{L^2(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})} \|\vartheta^{\frac{1}{2}} [\xi]\|_{L^2(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})}, \\
|I_5| &\leq \|\vartheta^{-\frac{1}{2}} \langle a \nabla \xi \rangle\|_{L^2(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})} \|\vartheta^{\frac{1}{2}} [\eta]\|_{L^2(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})}, \\
|I_6| &\leq \|\vartheta^{\frac{1}{2}} [\eta]\|_{L^2(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})} \|\vartheta^{\frac{1}{2}} [\xi]\|_{L^2(\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}})}.
\end{aligned}$$

For term I_3, by virtue of an interior inflow edge of one element being an outflow edge of an adjacent element, then

$$I_3 = \sum_{\kappa \in \mathcal{T}_h} \int_{\partial_+ \kappa \cap \Gamma} \mathbf{b} \cdot \mathbf{n}_\kappa \eta^+ \xi^+ \mathrm{d}s + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial_- \kappa \setminus \Gamma} \mathbf{b} \cdot \mathbf{n}_\kappa (\xi^+ - \xi^-) \eta^- \mathrm{d}s.$$

Further applications of the Cauchy-Schwarz inequality yield

$$I_3 \leq \sum_{\kappa\in\mathcal{T}_h} \left(\|\xi^+\|_{\partial_+\kappa\cap\Gamma}\|\eta^+\|_{\partial_+\kappa\cap\Gamma} + \|\xi^+ - \xi^-\|_{\partial_-\kappa\setminus\Gamma}\|\eta^-\|_{\partial_-\kappa\setminus\Gamma}\right).$$

Therefore, combining the above results and applying the discrete Schwarz inequality we obtain

$$\begin{aligned}
|B_{\mathrm{DG}}(\eta,\xi)| \leq & \left[\sum_{\kappa\in\mathcal{T}_h}\Big(\|\sqrt{a}\nabla\xi\|^2_{L^2(\kappa)} + \|c_0\xi\|^2_{L^2(\kappa)} + \|\xi^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\xi^+ - \xi^-\|_{\partial_-\kappa\setminus\Gamma}\right.\\
& \left.+2\|\vartheta^{\frac{1}{2}}[\xi]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{\frac{1}{2}}\langle a\nabla\xi\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))}\Big)\right]^{\frac{1}{2}}\\
& \times\left[\sum_{\kappa\in\mathcal{T}_h}\Big(\|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \gamma_1\|\eta\|^2_{L^2(\kappa)} + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma}\right.\\
& \left.+2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))}\Big)\right]^{\frac{1}{2}}\\
\leq & \sqrt{2}|||\xi|||_{\mathrm{DG}} \times\left[\sum_{\kappa\in\mathcal{T}_h}\Big(\|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \gamma_1\|\eta\|^2_{L^2(\kappa)} + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma}\right.\\
& \left.+\|\eta^-\|^2_{\partial_-\kappa\cap\Gamma} + 2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{\frac{1}{2}}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))}\Big)\right]^{\frac{1}{2}}.
\end{aligned}$$

Using this inequality in (4.1.7), yields the result (4.1.6). ■

We now consider the error $|J(u) - J(u_{\mathrm{DG}})|$ and prove the following lemma

Lemma 4.1.4. *Suppose $J(\cdot)$ is a linear target functional, u is the analytical solution of (2.2.1), (2.2.3) and u_{DG} is the approximate DG solution of (2.6.5), then assuming the same conditions stated in Lemma 4.1.3 hold.*

$$\begin{aligned}
|J(u) - J(u_{\mathrm{DG}})|^2 \leq & C\left(\sum_{\kappa\in\mathcal{T}_h}\Big\{\|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + (\beta_1+\gamma_1)\|\eta\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa^{-1}\|\nabla\eta\|^2_{L^2(\kappa)}\right.\\
& +\|\eta^+\|^2_{\partial_+\kappa\setminus\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma} + \|[\eta]\|^2_{\partial\kappa}\\
& \left.+\|\vartheta^{-1}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta[\eta]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))}\Big\}\right)\\
& \times\left(\sum_{\kappa\in\mathcal{T}_h}\Big\{\|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + (\beta_1+\beta_2\varepsilon_\kappa+\gamma_2)\|w\|^2_{L^2(\kappa)}\right.\\
& +\|w\|^2_{\partial\kappa} + \|w^+\|^2_{\partial_-\kappa} + \|\vartheta^{-1}\langle a\nabla w\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))}\\
& \left.+\|\vartheta[w]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}}))}\Big\}\right). \qquad (4.1.8)
\end{aligned}$$

where $\beta_1|_\kappa = \|c + \nabla \cdot \mathbf{b}\|_{L^\infty(\kappa)}$, $\beta_2|_\kappa = \|\mathbf{b}\|_{L^\infty(\kappa)}$, $\gamma_1|_\kappa = \|c/c_0\|^2_{L^\infty(\kappa)}$, $\gamma_2|_\kappa = \|(c + \nabla \cdot \mathbf{b})/c_0\|^2_{L^\infty(\kappa)}$ *and* $\varepsilon_\kappa > 0\ \forall \kappa \in \mathcal{T}_h$. *Here,* η *is as in (4.1.5) and* $w = z - \Pi_p z$, *where* $\Pi_p z$ *is the* L^2*-projection of* z *and* C *is a constant dependent only on the dimension* d.

Proof. Using (4.1.1) and Galerkin orthogonality we arrive at the error representation formula:

$$\begin{aligned} J(u) - J(u_{\mathrm{DG}}) &= B_{\mathrm{DG}}(u - u_{\mathrm{DG}}, z) \\ &= B_{\mathrm{DG}}(u - u_{\mathrm{DG}}, z - z_{h,p}). \end{aligned} \tag{4.1.9}$$

where $z_{h,p}$ can be any function from $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$. Decomposing $u - u_{\mathrm{DG}}$ as in (4.1.5) gives

$$|J(u) - J(u_{\mathrm{DG}})| \leq |I| + |II|.$$

where $I = B_{\mathrm{DG}}(\eta, z - z_{h,p})$ and $II = B_{\mathrm{DG}}(\xi, z - z_{h,p})$. Considering term I and setting $w = z - z_{h,p}$ gives

$$B_{\mathrm{DG}}(\eta, w) \equiv I_1 + I_2 + I_3 + I_4 + I_5.$$

where

$$\begin{aligned} I_1 &= \sum_{\kappa \in \mathcal{T}_h} \int_\kappa a \nabla \eta \cdot \nabla w \mathrm{d}x. \\ I_2 &= \sum_{\kappa \in \mathcal{T}_h} \left(- \int_\kappa (\eta \mathbf{b} \cdot \nabla w - c\eta w) \mathrm{d}x + \int_{\partial_+ \kappa} (\mathbf{b} \cdot \mathbf{n}_\kappa) \eta^+ w^+ \mathrm{d}s + \int_{\partial_- \kappa \setminus \Gamma} (\mathbf{b} \cdot \mathbf{n}_\kappa) \eta^- w^+ \mathrm{d}s \right). \\ I_3 &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \langle (a \nabla \eta) \cdot \mathbf{n}_f \rangle [w] \mathrm{d}s. \\ I_4 &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \langle (a \nabla w) \cdot \mathbf{n}_f \rangle [\eta] \mathrm{d}s. \\ I_5 &= \int_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{D}}} \vartheta [\eta][w] \mathrm{d}s. \end{aligned}$$

We now proceed to bound each of these five terms.

By assumption we have that $\sqrt{a}$ exists and by the Cauchy-Schwarz inequality it follows that

$$|I_1| \leq \sum_{\kappa \in \mathcal{T}_h} \|\sqrt{a} \nabla \eta\|_{L^2(\kappa)} \|\sqrt{a} \nabla w\|_{L^2(\kappa)}.$$

For term I_2, integration by parts gives

$$I_2 = \sum_{\kappa\in\mathcal{T}_h}\left(\int_\kappa (c+\nabla\cdot\mathbf{b})\eta w\mathrm{d}x + \int_\kappa \mathbf{b}\cdot\nabla\eta w\mathrm{d}x - \int_{\partial_-\kappa}\mathbf{b}\cdot\mathbf{n}_\kappa[\eta]w\mathrm{d}s\right).$$

then by application of the Cauchy-Schwarz inequality we have

$$\begin{aligned}
|I_2| &\le \sum_{\kappa\in\mathcal{T}_h}\big(\|c+\nabla\cdot\mathbf{b}\|_{L^\infty(\kappa)}\|\eta\|_{L^2(\kappa)}\|w\|_{L^2(\kappa)} + \|\mathbf{b}\|_{L^\infty(\kappa)}\|\nabla\eta\|_{L^2(\kappa)}\|w\|_{L^2(\kappa)} \\
&\quad + \|[\eta]\|_{\partial_-\kappa}\|w\|_{\partial_-\kappa}\big) \\
&\le \left(\sum_{\kappa\in\mathcal{T}_h}\left\{\beta_1\|\eta\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa^{-1}\|\nabla\eta\|^2_{L^2(\kappa)} + \|[\eta]\|^2_{\partial_-\kappa}\right\}\right)^{\frac{1}{2}}\times \\
&\quad \left(\sum_{\kappa\in\mathcal{T}_h}\left\{\beta_1\|w\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa\|w\|^2_{L^2(\kappa)} + \|w\|^2_{\partial_-\kappa}\right\}\right)^{\frac{1}{2}}.
\end{aligned}$$

where $\varepsilon_\kappa > 0$.

A simple application of the Cauchy-Schwarz inequality for each of the final three terms yields:

$$\begin{aligned}
|I_3| &\le \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}\|\vartheta^{\frac{1}{2}}[w]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}, \\
|I_4| &\le \|\vartheta^{-\frac{1}{2}}\langle a\nabla w\rangle\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}\|\vartheta^{\frac{1}{2}}[\eta]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}, \\
|I_5| &\le \|\vartheta^{\frac{1}{2}}[\eta]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}\|\vartheta^{\frac{1}{2}}[w]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}.
\end{aligned}$$

Combining the above results we obtain

$$\begin{aligned}
|B_{\mathrm{DG}}(\eta,w)| &\le \sum_{\kappa\in\mathcal{T}_h}\|\sqrt{a}\nabla\eta\|_{L^2(\kappa)}\|\sqrt{a}\nabla w\|_{L^2(\kappa)} \\
&\quad + \left(\sum_{\kappa\in\mathcal{T}_h}\left\{\beta_1\|\eta\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa^{-1}\|\nabla\eta\|^2_{L^2(\kappa)} + \|[\eta]\|^2_{\partial_-\kappa}\right\}\right)^{\frac{1}{2}}\times \\
&\quad \left(\sum_{\kappa\in\mathcal{T}_h}\left\{\beta_1\|w\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa\|w\|^2_{L^2(\kappa)} + \|w\|^2_{\partial_-\kappa}\right\}\right)^{\frac{1}{2}} \\
&\quad + \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}\|\vartheta^{\frac{1}{2}}[w]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})} \\
&\quad + \|\vartheta^{-\frac{1}{2}}\langle a\nabla w\rangle\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}\|\vartheta^{\frac{1}{2}}[\eta]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})} \\
&\quad + \|\vartheta^{\frac{1}{2}}[\eta]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}\|\vartheta^{\frac{1}{2}}[w]\|_{L^2(\Gamma_{\mathrm{int}}\cup\Gamma_{\mathrm{D}})}.
\end{aligned} \tag{4.1.10}$$

for any $\varepsilon_\kappa > 0$.

Now, for a function v, we have that

$$\begin{aligned}\|v\|^2_{L^2(\Gamma_{\text{int}}\cup\Gamma_{\text{D}})} &= \sum_{f\in\Gamma_{\text{int}}} \|v\|^2_{L^2(f)} + \sum_{f\in\Gamma_{\text{D}}} \|v\|^2_{L^2(f)} \\ &= \sum_{\kappa\in\mathcal{T}_h} \frac{1}{2}\|v\|^2_{L^2(\partial\kappa\setminus\Gamma)} + \sum_{\kappa\in\mathcal{T}_h} \|v\|^2_{L^2(\partial\kappa\cap\Gamma_{\text{D}})} \\ &\leq \sum_{\kappa\in\mathcal{T}_h} \|v\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))}.\end{aligned}$$

Thereby, substituting this inequality into (4.1.10) and applying the Cauchy inequality, we obtain

$$\begin{aligned}|B_{\text{DG}}(\eta,w)| \leq & \left(\sum_{\kappa\in\mathcal{T}_h}\left\{\|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \beta_1\|\eta\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa^{-1}\|\nabla\eta\|^2_{L^2(\kappa)} + \|[\eta]\|^2_{\partial_-\kappa}\right.\right. \\ & \left.\left. + \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))} + 2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))}\right\}\right)^{\frac{1}{2}} \times \\ & \left(\sum_{\kappa\in\mathcal{T}_h}\left\{\|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + \beta_1\|w\|^2_{L^2(\kappa)} + \beta_2\varepsilon_\kappa\|w\|^2_{L^2(\kappa)} + \|w\|^2_{\partial_-\kappa}\right.\right. \\ & \left.\left. + \|\vartheta^{-\frac{1}{2}}\langle a\nabla w\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))} + 2\|\vartheta^{\frac{1}{2}}[w]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))}\right\}\right)^{\frac{1}{2}}.\end{aligned}$$

We now turn our attention to term II. In an analogous procedure to that carried for term I we obtain

$$\begin{aligned}|B_{\text{DG}}(\xi,w)| \leq & \,|||\xi|||_{\text{DG}} \\ & \times\left[\sum_{\kappa\in\mathcal{T}_h}\left(\|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + \left\|\left(\frac{c+\nabla\cdot\mathbf{b}}{c_0}\right)w\right\|^2_{L^2(\kappa)} + \|w^+\|^2_{\partial_-\kappa}\right.\right. \\ & \left.\left. + \|\vartheta^{\frac{1}{2}}[w]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))} + \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))}\right)\right]^{\frac{1}{2}}. \quad (4.1.11)\end{aligned}$$

We are now in a position to use the result of Lemma 4.1.3: thereby, inserting (4.1.6) into (4.1.11), $B_{\text{DG}}(\xi,w)$ can be bounded in terms of η and w as follows:

$$\begin{aligned}|B_{\text{DG}}(\xi,w)| \leq & \, C\left[\sum_{\kappa\in\mathcal{T}_h}\left(\|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \left\|\frac{c}{c_0}\eta\right\|^2_{L^2(\kappa)} + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma}\right.\right. \\ & \left.\left. + 2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))} + \|\vartheta^{-\frac{1}{2}}\langle a\nabla\eta\rangle\|^2_{L^2(\partial\kappa\cap(\Gamma_{\text{int}}\cup\Gamma_{\text{D}}))}\right)\right]^{\frac{1}{2}}\end{aligned}$$

$$\times \left[\sum_{\kappa \in \mathcal{T}_h} \left(\|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + \left\| \left(\frac{c + \nabla \cdot \mathbf{b}}{c_0} \right) w \right\|^2_{L^2(\kappa)} + \|w^+\|^2_{\partial_- \kappa} \right.\right.$$
$$\left.\left. + \|\vartheta^{\frac{1}{2}}[w]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} + \|\vartheta^{-\frac{1}{2}} \langle a\nabla\eta \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right) \right]^{\frac{1}{2}}.$$

Returning to the initial problem and using our estimates for I and II, we arrive at

$$\begin{aligned}
|J(u) - J(u_{\text{DG}})| \leq\ & C \left[\left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \beta_1 \|\eta\|^2_{L^2(\kappa)} + \beta_2 \varepsilon_\kappa^{-1} \|\nabla\eta\|^2_{L^2(\kappa)} + \|[\eta]\|^2_{\partial_- \kappa} \right.\right. \right. \\
& \left. \left. + \|\vartheta^{-\frac{1}{2}} \langle a\nabla\eta \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} + 2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right\} \right)^{\frac{1}{2}} \times \\
& \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + \beta_1 \|w\|^2_{L^2(\kappa)} + \beta_2 \varepsilon_\kappa \|w\|^2_{L^2(\kappa)} + \|w\|^2_{\partial_- \kappa} \right.\right. \\
& \left. \left. + \|\vartheta^{-\frac{1}{2}} \langle a\nabla w \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} + 2\|\vartheta^{\frac{1}{2}}[w]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right\} \right)^{\frac{1}{2}} + \\
& \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + \left\| \frac{c}{c_0} \eta \right\|^2_{L^2(\kappa)} + \|\eta^+\|^2_{\partial_+ \kappa \cap \Gamma} + \|\eta^-\|^2_{\partial_- \kappa \setminus \Gamma} \right.\right. \\
& \left. \left. + 2\|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} + \|\vartheta^{-\frac{1}{2}} \langle a\nabla\eta \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right\} \right)^{\frac{1}{2}} \\
& \times \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + \left\| \left(\frac{c + \nabla \cdot \mathbf{b}}{c_0} \right) w \right\|^2_{L^2(\kappa)} + \|w^+\|^2_{\partial_- \kappa} \right.\right. \\
& \left. \left. + \|\vartheta^{\frac{1}{2}}[w]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} + \|\vartheta^{-\frac{1}{2}} \langle a\nabla\eta \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right\} \right)^{\frac{1}{2}} \Bigg].
\end{aligned}$$

and after a final application of the Cauchy inequality we obtain:

$$\begin{aligned}
|J(u) - J(u_{\text{DG}})|^2 \leq\ & C \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \|\sqrt{a}\nabla\eta\|^2_{L^2(\kappa)} + (\beta_1 + \gamma_1) \|\eta\|^2_{L^2(\kappa)} + \beta_2 \varepsilon_\kappa^{-1} \|\nabla\eta\|^2_{L^2(\kappa)} \right.\right. \\
& + \|\eta^+\|^2_{\partial_+ \kappa \cap \Gamma} + \|\eta^-\|^2_{\partial_- \kappa \setminus \Gamma} + \|[\eta]\|^2_{\partial_- \kappa} \\
& \left. \left. + \|\vartheta^{-\frac{1}{2}} \langle a\nabla\eta \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} + \|\vartheta^{\frac{1}{2}}[\eta]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right\} \right) \\
& \times \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \|\sqrt{a}\nabla w\|^2_{L^2(\kappa)} + (\beta_1 + \beta_2 \varepsilon_\kappa + \gamma_2) \|w\|^2_{L^2(\kappa)} \right.\right. \\
& + \|w\|^2_{\partial_- \kappa} + \|w^+\|^2_{\partial_- \kappa)} + \|\vartheta^{-\frac{1}{2}} \langle a\nabla w \rangle\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \\
& \left. \left. + \|\vartheta^{\frac{1}{2}}[w]\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_D))} \right\} \right).
\end{aligned}$$

which completes the proof.■

For the rest of this chapter we make the assumption that the element volumes, denoted by m_κ, for each $\kappa \in \mathcal{T}_h$ have *bounded local variation*, that is, there exists a constant $C_8 > 1$, such that, for any pair of elements κ and κ' sharing a $(d-1)$-dimensional face

$$C_8^{-1} \leq m_\kappa / m'_\kappa \leq C_8. \tag{4.1.12}$$

With this assumption, we are now ready to state and prove the main result of this chapter, which bounds the error in the target functional by using the results of Chapter 3, in the special case when the polynomial degree p, is both isotropic, uniform and fixed on the mesh; under these restrictions we define the finite element space to be $S^{p_{\text{uni}}}(\Omega, \mathcal{T}_h, \mathbf{F})$.

Theorem 4.1.5. *Let $\Omega \subset \mathbb{R}^d$ be a bounded polyhedral domain, $\mathcal{T}_h = \{\kappa\}$ a subdivision of Ω, such that the elemental volumes satisfy the bounded local variation condition (4.1.12). Then, assuming that conditions (2.3.2), (4.1.3), and (4.1.2) on the data hold, and $u \in H^k(\Omega, \mathcal{T}_h)$, $k \geq 2$, $z \in H^l(\Omega, \mathcal{T}_h)$, $l \geq 2$, then the solution $u_{\text{DG}} \in S^{p_{\text{uni}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ of (2.6.5) obeys the error bound*

$$\begin{aligned} |J(u) - J(u_{\text{DG}})|^2 \quad \leq \quad & C \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \frac{\alpha}{\sigma_{d,\kappa}^2} + \frac{\beta_2}{\sigma_{d,\kappa}} + (\beta_1 + \gamma_1) \right\} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^s(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right) \\ & \times \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \frac{\alpha}{\sigma_{d,\kappa}^2} + \frac{\beta_2}{\sigma_{d,\kappa}} + (\beta_1 + \gamma_2) \right\} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^t(\tilde{z}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right). \end{aligned} \tag{4.1.13}$$

for $2 \leq s \leq \min(p+1, k)$ and $2 \leq t \leq \min(p+1, l)$, where $\alpha|_\kappa = \bar{a}_{\tilde{\kappa}}$, $\beta_1|_\kappa = \|c + \nabla \cdot \mathbf{b}\|_{L^\infty(\kappa)}$, $\beta_2|_\kappa = \|\mathbf{b}\|_{L^\infty(\kappa)}$, $\gamma_1|_\kappa = \|c/c_0\|^2_{L^\infty(\kappa)}$, $\gamma_2|_\kappa = \|(c + \nabla \cdot \mathbf{b})/c_0\|^2_{L^\infty(\kappa)}$ for all $\kappa \in \mathcal{T}_h$. Here, C is a constant depending on the dimension d, the polynomial degree p, and the parameters C_i, $i = 1, \ldots, 8$.

Proof. We first extract the terms $\bar{a}_{\tilde{\kappa}}$ defined as in (4.1.4) and $\beta_2 := \|\mathbf{b}\|^2_{L^\infty(\kappa)}$ from the relevant norms in (4.1.9), in which case we have

$$\begin{aligned} |J(u) - J(u_{\text{DG}})|^2 \quad \leq \quad & C \Bigg(\sum_{\kappa \in \mathcal{T}_h} \Big\{ \bar{a}_{\tilde{\kappa}} \|\nabla \eta\|^2_{L^2(\kappa)} + (\beta_1 + \gamma_1) \|\eta\|^2_{L^2(\kappa)} + \beta_2 \varepsilon_\kappa^{-1} \|\nabla \eta\|^2_{L^2(\kappa)} \\ & + \beta_2 \|\eta^+\|^2_{L^2(\partial_+ \kappa \setminus \Gamma)} + \beta_2 \|\eta^-\|^2_{L^2(\partial_- \kappa \setminus \Gamma)} + \beta_2 \|[\eta]\|^2_{L^2(\partial \kappa)} \\ & + \bar{a}_{\tilde{\kappa}} \left\| \frac{\bar{a}_{\tilde{\kappa}}^{\frac{1}{2}}}{\vartheta^{\frac{1}{2}}} \langle \nabla \eta \rangle \right\|^2_{L^2(\partial \kappa \cap (\Gamma_{\text{int}} \cup \Gamma_{\text{D}}))} + \bar{a}_{\tilde{\kappa}} \left\| \frac{\vartheta^{\frac{1}{2}}}{\bar{a}_{\tilde{\kappa}}^{\frac{1}{2}}} [\eta] \right\|^2_{L^2(\partial \kappa \cap (\Gamma_{\text{int}} \cup \Gamma_{\text{D}}))} \Big\} \Bigg) \end{aligned}$$

$$\times \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \bar{a}_\kappa \|\nabla w\|^2_{L^2(\kappa)} + (\beta_1 + \beta_2 \varepsilon_\kappa + \gamma_2) \|w\|^2_{L^2(\kappa)} \right.\right.$$

$$+\beta_2 \|w\|^2_{L^2(\partial\kappa)} + \beta_2 \|w^+\|^2_{L^2(\partial_- \kappa)} + \bar{a}_\kappa \left\| \frac{\bar{a}_\kappa^{\frac{1}{2}}}{\vartheta^{\frac{1}{2}}} \langle \nabla w \rangle \right\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_{\text{D}}))}$$

$$\left.\left. + \bar{a}_\kappa \left\| \frac{\vartheta^{\frac{1}{2}}}{\bar{a}_\kappa^{\frac{1}{2}}} [w] \right\|^2_{L^2(\partial\kappa \cap (\Gamma_{\text{int}} \cup \Gamma_{\text{D}}))} \right\} \right). \tag{4.1.14}$$

Suppose we now consider $f \subset \partial\kappa$ and look at $\left\| (\bar{a}_\kappa/\vartheta)^{\frac{1}{2}} \langle \nabla \eta \rangle \right\|^2_f$. By using the arithmetic-geometric mean inequality and applying the triangle inequality we deduce that

$$\begin{aligned}
\left\| \frac{\bar{a}_\kappa^{\frac{1}{2}}}{\vartheta^{\frac{1}{2}}} \langle \nabla \eta \rangle \right\|^2_f &= \frac{1}{4} \int_f \frac{\bar{a}_\kappa}{\vartheta} |\nabla \eta_\kappa + \nabla \eta_{\kappa'}|^2 \mathrm{d}x \\
&\leq \frac{1}{2} \int_f \frac{\bar{a}_\kappa}{\vartheta} (|\nabla \eta_\kappa|^2 + |\nabla \eta_{\kappa'}|^2) \mathrm{d}x \\
&\leq C \left(\left\| \frac{\bar{a}_\kappa^{\frac{1}{2}}}{\vartheta^{\frac{1}{2}}} \nabla \eta_\kappa \right\|^2_f + \left\| \frac{\bar{a}_\kappa^{\frac{1}{2}}}{\vartheta^{\frac{1}{2}}} \nabla \eta_{\kappa'} \right\|^2_f \right) \\
&= C \frac{\bar{a}_\kappa}{\vartheta} (\|\nabla \eta_\kappa\|^2_f + \|\nabla \eta_{\kappa'}\|^2_f),
\end{aligned}$$

where κ' is the element adjacent to κ sharing face f. A similar application of the triangle inequality when applied to $\left\| \vartheta^{\frac{1}{2}} / \bar{a}_\kappa^{\frac{1}{2}} [\eta] \right\|^2_f$ yields

$$\begin{aligned}
\left\| \frac{\vartheta^{\frac{1}{2}}}{\bar{a}_\kappa^{\frac{1}{2}}} [\eta] \right\|^2_f &\leq \left\| \frac{\vartheta^{\frac{1}{2}}}{\bar{a}_\kappa^{\frac{1}{2}}} \eta_\kappa \right\|^2_f + \left\| \frac{\vartheta^{\frac{1}{2}}}{\bar{a}_\kappa^{\frac{1}{2}}} \eta_{\kappa'} \right\|^2_f \\
&= \frac{\vartheta}{\bar{a}_\kappa} (\|\eta_\kappa\|^2_f + \|\eta_{\kappa'}\|^2_f).
\end{aligned}$$

Absorbing the terms involving the element κ' into the relevant summation terms and applying Theorem 3.3.8 we obtain

$$\begin{aligned}
|J(u) - J(u_{\text{DG}})|^2 \leq\ & C \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \frac{\bar{a}_\kappa}{\sigma^2_{d,\kappa}} \left[1 + \frac{\bar{a}_\kappa}{m_\kappa} \sum_{f \subset \partial\kappa} \frac{m_f}{\vartheta_f} + \frac{\sigma_{d,\kappa} \sum_{f \subset \partial\kappa} \vartheta_f}{\bar{a}_\kappa} \right] \right.\right. \\
& \left.\left. + \frac{\beta_2}{\sigma_{d,\kappa}} \left[1 + \frac{1}{\epsilon_\kappa \sigma_{d,\kappa}} \right] + (\beta_1 + \gamma_1) \right\} \int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right) \\
& \times \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \frac{\bar{a}_\kappa}{\sigma^2_{d,\kappa}} \left[1 + \frac{\bar{a}_\kappa}{m_\kappa} \sum_{f \subset \partial\kappa} \frac{m_f}{\vartheta_f} + \frac{\sigma_{d,\kappa} \sum_{f \subset \partial\kappa} \vartheta_f}{\bar{a}_\kappa} \right] \right.\right.
\end{aligned}$$

$$+\frac{\beta_2}{\sigma_{d,\kappa}}\left[1+\epsilon_\kappa\sigma_{d,\kappa}\right]+(\beta_1+\gamma_2)\Big\}\int_{\tilde{\kappa}} D^t_{\tilde{\kappa}}(\tilde{z},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\Big).$$

We now use the definition of discontinuity penalisation parameter ϑ: however, since here we are only considering a uniform, fixed polynomial degree we first incorporate the p term into the constant C'_ϑ, *i.e.* we write

$$\vartheta|_f = C'_\vartheta(p)\frac{\mathrm{a}}{\mathrm{h}}.$$

Then, applying the local bounded variation assumption (4.1.12), using (3.3.6), choosing $\varepsilon_\kappa = 1/\sigma_{d,\kappa}$ and collecting terms we obtain the final result (4.1.13). ■

For completeness we also give the following theorem in the case where we approximate with functions now from $S^{\mathbf{p}_{\mathrm{iso}}}(\Omega,\mathcal{T}_h,\mathbf{F})$. In this case we also assume bounded local variation of the polynomial degrees, *i.e.*, there exists a constant $C_9 > 1$, such that for any pair of elements κ and κ' sharing a $(d-1)$-dimensional face

$$C_9^{-1} < p_\kappa/p_{\kappa'} < C_9. \tag{4.1.15}$$

Theorem 4.1.6. *Let $\Omega \subset \mathbb{R}^d$ be a bounded polyhedral domain, $\mathcal{T}_h = \{\kappa\}$ a subdivision of Ω, such that the elemental volumes and polynomial degrees satisfy the bounded local variation conditions (4.1.12) and (4.1.15), respectively. Then, assuming that conditions (2.3.2), (4.1.3), and (4.1.2) on the data hold, and $u \in H^k(\Omega,\mathcal{T}_h)$, $k \geq 2$, $z \in H^l(\Omega,\mathcal{T}_h)$, $l \geq 2$, then the solution $u_{\mathrm{DG}} \in S^{\mathbf{p}_{\mathrm{iso}}}(\Omega,\mathcal{T}_h,\mathbf{F})$ of (2.6.5) obeys the error bound*

$$\begin{aligned}
&|J(u)-J(u_{\mathrm{DG}})|^2 \\
&\leq C\left(\sum_{\kappa\in\mathcal{T}_h}\frac{1}{\sigma^2_{d,\kappa}}\left\{\frac{\alpha}{p_\kappa^{2(s_\kappa-3/2)}}+\frac{\beta_2\sigma_{d,\kappa}}{p_\kappa^{2(s_\kappa-1/2)}}+\frac{(\beta_1+\gamma_1)\sigma^2_{d,\kappa}}{p_\kappa^{2s_\kappa}}\right\}\int_{\tilde{\kappa}} D^{s_\kappa}_{\tilde{\kappa}}(\tilde{u},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\right) \\
&\quad\times\left(\sum_{\kappa\in\mathcal{T}_h}\frac{1}{\sigma^2_{d,\kappa}}\left\{\frac{\alpha}{p_\kappa^{2(t_\kappa-3/2)}}+\frac{\beta_2\sigma_{d,\kappa}}{p_\kappa^{2(t_\kappa-1)}}+\frac{(\beta_1+\gamma_2)\sigma^2_{d,\kappa}}{p_\kappa^{2t_\kappa}}\right\}\int_{\tilde{\kappa}} D^{t_\kappa}_{\tilde{\kappa}}(\tilde{z},\Sigma_\kappa,U_\kappa)\mathrm{d}\tilde{x}\right).
\end{aligned}$$

for $2 \leq s_\kappa \leq \min(p_\kappa+1,k)$ and $2 \leq t_\kappa \leq \min(p_\kappa+1,l)$, where $\alpha|_\kappa = \bar{a}_{\tilde{\kappa}}$, $\beta_1|_\kappa = \|c+\nabla\cdot\mathbf{b}\|_{L^\infty(\kappa)}$, $\beta_2|_\kappa = \|\mathbf{b}\|_{L^\infty(\kappa)}$, $\gamma_1|_\kappa = \|c/c_0\|^2_{L^\infty(\kappa)}$, $\gamma_2|_\kappa = \|(c+\nabla\cdot\mathbf{b})/c_0\|^2_{L^\infty(\kappa)}$, for all $\kappa \in \mathcal{T}_h$. Here, C is a constant depending on the dimension d and the parameters C_i, $i = 1,\ldots,9$.

Proof. The proof is analogous to that for Theorem 4.1.5: however, here, we pick $\varepsilon_\kappa = p_\kappa^2/\sigma_{d,\kappa}$ and use the interpolation results from Lemma 3.4.2 together with the discontinuity penalisation term

$$\vartheta|_f = C_\vartheta \frac{\mathrm{ap}^2}{\mathrm{h}}.$$

■

Remark 4.1.7. As in Harriman *et al.* [61], we discuss some special cases of the general error bound derived in Theorem 4.1.6, where we assume uniform polynomial orders $p_\kappa = p$, $s_\kappa = s$, $t_k = t$, and uniform isotropic elements with mesh size h. In the diffusion dominated case the error bound from Theorem 4.1.6 implies the error in the computed target functional behaves like $\mathcal{O}(h^{s+t-2}/p^{k+l-2})$ as $h \to 0$ and $p \to \infty$: this is optimal with respect to h but suboptimal in p by one order. For the strictly hyperbolic case ($a \equiv 0$), the implication is that the error behaves like $\mathcal{O}(h^{s+t-1}/p^{k+l-1/2})$: once again this is optimal in h, but suboptimal in p, this time by $p^{1/2}$. In both cases we witness 'order doubling' of the method with respect to h, stemming from the duality argument.

Chapter 5

Adaptive Anisotropic Mesh Refinement

The ultimate goal of any numerical method should be to solve a problem to a high level of accuracy, while remaining as efficient as possible both in terms of CPU time and computational storage. It is also reasonable to wish that the error is equidistributed across the domain. Evidently, it is highly unlikely that our initial mesh will provide a solution with a small enough error, therefore, it is desirable to be able to automatically modify the mesh in such a way as to reduce the error without significantly increasing the computational costs. First we need to be able to establish how accurate a solution is, *i.e.*, we need some sort of error indicator; for this purpose our *a priori* estimates from Chapter 4 are useless as they require knowledge of the actual solution beforehand. Hence, we introduce the notion of an *a posteriori* error estimate, which must be computable using only the information from the approximate solution and the problem's data. Equipped with an *a posteriori* error indicator we may design an adaptive algorithm to attain a desired level of accuracy, as follows.

1. Design an initial mesh.

2. Solve the discrete system.

3. Calculate an *a posteriori* error estimate.

4. If the error estimate is less than a prescribed accuracy then stop, else goto 5.

5. Perform adaptive mesh refinement and goto 2.

A very useful property of an *a posteriori* indicator is if it can be broken down into contributions from each element in the mesh, then only those elements with a large error need be refined. Step 5 (along with the quality of the *a posteriori* error indicator) then provides the key to achieving an efficient algorithm, that is one which produces accurate results with minimal cost. Common refinement strategies include:

- h–adaptivity, where the chosen elements are split/joined in some way.
- p–adaptivity, where the polynomial degree on the chosen elements are increased/decreased.
- r–adaptivity, where mesh points are redistributed in the domain.
- hp—adaptivity, a mixture of h– and p–adaptivity.

Throughout this chapter we will concern ourselves only with h-adaptivity, returning to p- and r-adaptivity in Chapters 7 and 9, respectively. First, we look in more detail at *a posteriori* error estimation and present an *a posteriori* error estimate in the context of target functional estimation.

5.1 *A Posteriori* Error Estimation

As previously mentioned, an *a posteriori* error estimate must be computable from the numerical solution and the problem's data. Indeed, in order to ensure that the numerical solution solves the problem up to a desired level of accuracy without too much work it is essential that the *a posteriori* error estimator is sharp. To this end, we introduce the notion of reliability for an error estimator η, where, as before, we are interested in minimizing $|J(u) - J(u_{\mathrm{DG}})|$, for some functional $J(\cdot)$, where u is the analytical solution and u_{DG} is the numerical solution.

$$\text{Reliability} \quad \Leftrightarrow \quad |J(u) - J(u_{\mathrm{DG}})| \leq C_{\mathrm{r}}\eta,$$

where C_{r} is a positive constant independent of the mesh; hence, reliability guarantees that the error is below the given tolerance. Additionally, we would also wish the estimator to be efficient, in the sense that the true error can be bounded from below by the estimator, guaranteeing that not too many degrees of freedom have been employed to meet a given reliability condition. *A posteriori* error estimates are frequently 'residual' based, where the residual determines how well the numerical solution solves the underlying partial differential equation. For example, suppose we wish to solve $\mathcal{L}u = f$, where $\mathcal{L}$ is some partial differential operator, then the residual $r(u_h)$ is formally defined by

$$r(u_h) = f - \mathcal{L}u_h.$$

Similarly, where boundary conditions have been weakly imposed, such as in DG methods, for example, boundary residuals appear in the error estimate, where the boundary residuals measure how well the approximate solution matches the prescribed boundary conditions.

A posteriori error estimation on isotropic meshes for elliptic PDEs is now a very mature subject; a review of this area can be found in Szabó and Babuška [129], Ainsworth and Oden [3] or Verfürth [135], for example. For hyperbolic PDEs the *a posteriori* error estimation is at a less advanced stage; for developments in this field we refer to [38, 59, 70, 123, 124], and [62, 63, 72, 82, 89, 93] for more recent work. Specifically, in the context of energy norm *a posteriori* error estimation of elliptic problems for DG methods, we refer the reader to Becker *et al.* [19, 20], Karakashian & Pascal [87], and Houston *et al.* [74]. Further work concerning energy norm error estimation for DG methods can be found in Houston *et al.* [73] and [71] for the Stokes' problem and Maxwell equations, respectively.

In the case when anisotropic meshes are employed, energy norm *a posteriori* error estimation for a standard, conforming FEM has been considered by Kunert in the series of papers [90, 91, 92], for elliptic, convection-dominated and singularly perturbed reaction-diffusion problems, respectively. In the DG setting, Creusé *et al.* [40] extend the work in Houston *et al.* [73] to anisotropic meshes for the Stokes' problem. However, for all of the mentioned anisotropic error estimates, reliability can only be shown in the following sense:

$$\|u - u_h\| \leq m_1(u - u_h, \mathcal{T}),$$

where $\|\cdot\|$ is the norm of interest, and $m_1(\cdot,\cdot)$ is the so-called matching function, first introduced in [90]. Essentially the matching function determines how well the mesh $\mathcal{T}$ is aligned with the underlying anisotropy of the function to be approximated; if the grid is well aligned then $m_1 \sim 1$, otherwise m_1 can become arbitrarily large. Hence, for the error estimate to be reliable, the mesh must already be well aligned with anisotropic features present in the solution; this, thereby, requires exploiting *a priori* knowledge of the (unknown) analytical solution.

In the next section we present *a posteriori* error estimation in the goal oriented setting.

5.1.1 *A Posteriori* Error Estimation For Functionals

In this section we lay down the framework for *a posteriori* error estimation for general linear target functionals $J(\cdot)$ of the solution of (2.6.5). This being based on the method first developed by Becker and Rannacher [21] and extended in, for example, [63, 77, 79], where in those cases a general semi–linear form replaces $B_{\mathrm{DG}}(\cdot,\cdot)$ and $J(\cdot)$ is a non-linear functional.

We first recall the dual problem (4.1.1): find $z \in H^2(\Omega,\mathcal{T}_h)$ such that

$$B_{\mathrm{DG}}(w,z) = J(w) \quad \forall w \in H^2(\Omega,\mathcal{T}_h),$$

which, following the discussion in Section 4.1.1, we assume possesses a unique solution. We are interested in the error $J(u)-J(u_{\mathrm{DG}})$, where u is the solution to (2.2.1), (2.2.3) and u_{DG} is the solution to (2.6.5). Hence, picking $w = u - u_{\mathrm{DG}}$ and exploiting the consistency of the DG formulation, Galerkin orthogonality and the linearity of $B_{\mathrm{DG}}(\cdot,\cdot)$ and $J(\cdot)$, yields the following error representation formula

$$\begin{aligned} J(u) - J(u_{\mathrm{DG}}) &= J(u - u_{\mathrm{DG}}) \\ &= B_{\mathrm{DG}}(u - u_{\mathrm{DG}}, z) \\ &= B_{\mathrm{DG}}(u, z) - B_{\mathrm{DG}}(u_{\mathrm{DG}}, z) \\ &= \ell_{\mathrm{DG}}(z) - B_{\mathrm{DG}}(u_{\mathrm{DG}}, z - z_{h,p}), \end{aligned} \tag{5.1.1}$$

for some function $z_{h,p} \in S^{\vec{\mathbf{p}}}(\Omega,\mathcal{T}_h,\mathbf{F})$; typically $z_{h,p}$ is chosen to be a projection/interpolant of z into $S^{\vec{\mathbf{p}}}(\Omega,\mathcal{T}_h,\mathbf{F})$. It is common to write *a posteriori* error estimates in terms of

residuals which measure how much $u_{\rm DG}$ fails to satisfy the underlying partial differential equation and the boundary conditions. Therefore, we define

$$\begin{aligned} R_{\rm int}|_{\kappa} &= (f - \mathcal{L}u_{\rm DG})|_{\kappa}. \\ R_{\rm D}|_{\partial\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)} &= (g_{\rm D} - u^+_{\rm DG})|_{\partial\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)}. \\ R_{\rm N}|_{\partial\kappa\cap\Gamma_{\rm N}} &= (g_{\rm N} - (a\nabla u^+_{\rm DG})\cdot\mathbf{n})|_{\partial\kappa\cap\Gamma_{\rm N}}. \end{aligned}$$

Upon rewriting $B_f(\cdot,\cdot)$ and $B_\vartheta(\cdot,\cdot)$ in terms of sums over elements $\kappa \in \mathcal{T}_h$ and applying the divergence theorem we can rewrite (5.1.1) as follows

$$J(u) - J(u_{\rm DG}) \equiv \sum_{\kappa\in\mathcal{T}_h} \eta_\kappa, \tag{5.1.2}$$

where

$$\begin{aligned} \eta_\kappa = & \int_\kappa R_{\rm int}(z - z_{h.p})\mathrm{d}x - \int_{\partial_-\kappa\cap\Gamma} (\mathbf{b}\cdot\mathbf{n}_\kappa) R_{\rm D}(z - z_{h.p})^+ \mathrm{d}s \\ + & \int_{\partial_-\kappa\setminus\Gamma} (\mathbf{b}\cdot\mathbf{n}_\kappa)[u_{\rm DG}](z - z_{h.p})^+ \mathrm{d}s - \int_{\partial\kappa\cap\Gamma_{\rm D}} R_{\rm D}((a\nabla(z - z_{h.p})^+)\cdot\mathbf{n}_\kappa)\mathrm{d}s \\ + & \int_{\partial_\kappa\cap\Gamma_{\rm D}} \vartheta R_{\rm D}(z - z_{h.p})^+ \mathrm{d}s + \int_{\partial\kappa\cap\Gamma_{\rm N}} R_{\rm N}(z - z_{h.p})^+ \mathrm{d}s \\ + & \int_{\partial\kappa\setminus\Gamma} \left\{ \frac{1}{2}[u_{\rm DG}](a\nabla(z - z_{h.p})^+)\cdot\mathbf{n}_\kappa - \frac{1}{2}[(a\nabla u_{\rm DG})\cdot\mathbf{n}_\kappa](z - z_{h.p})^+ \right\} \mathrm{d}s \\ - & \int_{\partial\kappa\setminus\Gamma} \vartheta[u_{\rm DG}](z - z_{h.p})^+ \mathrm{d}s. \end{aligned} \tag{5.1.3}$$

After applying the triangle inequality we arrive at the following *a posteriori* error bound

$$|J(u) - J(u_{\rm DG})| \le \sum_{\kappa\in\mathcal{T}_h} |\eta_\kappa|. \tag{5.1.4}$$

where η_κ is as in (5.1.3). An error bound such as this, which includes the difference between the dual solution z and its interpolant/projection $z_{h.p}$ into $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ is referred to as a Type I *a posteriori* error bound. Using approximation theory to bound the terms involving $z - z_{h.p}$ in terms of norms of z and in turn bounding these terms by norms of the data for the dual problem, z can be completely eliminated from the error bound, this new estimate being termed a Type II *a posteriori* error bound. However, as discussed in [63, 125], the local weighting terms involving the difference between the dual solution z and its projection/interpolant $z_{h.p}$ onto $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ appearing in the Type I bound

provide invaluable information concerning the global transport of the error. Hence, we will not explore Type II error bounds further in this thesis.

The formula for η_κ may look complicated, but each term shares the common form of an integral of a residual weighted by the dual solution. Hence, we see that, even in areas where there may be a large residual, if the weighting term is small, the contribution from that element to the overall error can also be small. This fact shows us the that the dual problem is the key to deciding in which areas to refine and prevents over refinement in areas of little importance.

Returning to our Type I *a posteriori* error bound (5.1.2) we see that, unless the actual solution z to the dual solution is known, it is still not computable. However, to achieve a computable indicator it will be sufficient just to calculate an approximation z_{DG} to (4.1.1) and use this in place of z in (5.1.2) or (5.1.4). To this end, we compute the approximate solution z_{DG} to (4.1.1) on the same mesh as used for the primal problem and restrict the choice of test and trial functions to a finite dimensional space. It is not possible for our approximation z_{DG} to come from $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$, as replacing z by z_{DG} in (5.1.1) would lead to the estimate being identically zero, due to (2.6.5). Instead we pick z_{DG} from $S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\mathrm{inc}}}(\Omega, \mathcal{T}_h, \mathbf{F})$, where $p_{\mathrm{inc},i} \geq 1$ for $i = 1, \ldots, d$, that is, z_{DG} will have a higher polynomial degree than u_{DG}; for our purposes it is sufficient to set $\vec{p}_{\mathrm{inc}} = \vec{1}$. Hence, the discrete dual problem is: find $z_{\mathrm{DG}} \in S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\mathrm{inc}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ such that

$$B_{\mathrm{DG}}(w, z_{\mathrm{DG}}) = J(w) \quad \forall w \in S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\mathrm{inc}}}(\Omega, \mathcal{T}_h, \mathbf{F}). \tag{5.1.5}$$

Schneider and Jimack [118] perform some experiments to show the reliability of (5.1.4) on anisotropic triangular meshes and our experiments, see Chapter 6, confirm reliability on anisotropic quadrilateral meshes.

Remark 5.1.1. Instead of incrementing the polynomial degree as above to find the approximation z_{DG}, it is also possible to compute z_{DG} on a sequence of dual finite element meshes $\hat{\mathcal{T}}_h$, which, in general, differ from the 'primal meshes' $\mathcal{T}_h$. Alternatively, the approximate dual problem can be computed using the same mesh $\mathcal{T}_h$ and finite element space as the primal problem. The resulting approximate dual solution $z_{\mathrm{DG}} \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ is then extrapolated to $\tilde{z}_{\mathrm{DG}} \in S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\mathrm{inc}}}(\Omega, \mathcal{T}_{2h}, \mathbf{F})$: a coarser mesh with increased polynomial

degree. The latter approach is the cheapest of the three methods mentioned, and is still capable of producing adaptively refined meshes specifically tailored to the selected target functional, however the quality of the resulting approximate error representation formula may be poor. On the basis of numerical experimentation, we prefer the approach first mentioned, due to its computational simplicity of implementation.

Remark 5.1.2. The error analysis above extends naturally to the case when $B_{\mathrm{DG}}(\cdot,\cdot)$ is replaced by a semi-linear form and/or when $J(\cdot)$ is non-linear, with a linearisation about u_{DG} needing to be performed to create a computable error bound; see, for example, [63]. One application area is the approximation of eigenvalues and eigenfunctions of a differential operator; see [66].

Remark 5.1.3. In certain situations the control of more than one functional is essential; here, we refer to Hartmann and Houston [64] for the application of the theory to problems where multiple target functionals are of interest.

5.2 h-Refinement Strategies

We now turn our attention to Step 5 of the adaptive algorithm presented at the beginning of this chapter and specifically concern ourselves with h-refinement. Initially we consider widely used isotropic refinement and then move on to the problem of designing anisotropic meshes.

5.2.1 Isotropic h-Refinement

The most extensively used form of h-refinement is isotropic refinement, where cells are divided into similar shaped cells, all of roughly the same size, with no elongation in any direction. In other words all the singular values of the map F_κ, for each element κ are close to unity. Two-dimensional examples of isotropic refinement for quadrilaterals and triangles are shown in Figures 5.1 and 5.2, respectively. This type of refinement is popular because the minimum angle constraint for the standard FEM is never violated and, in the case of triangular elements, hanging nodes can easily be removed by splitting neighbouring cells, as in Figure 5.3.

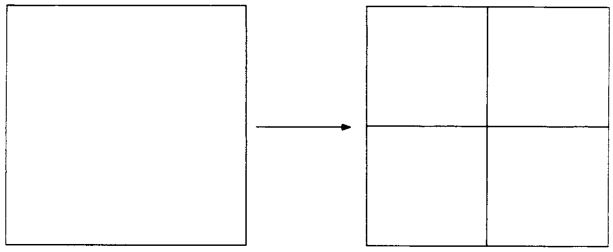


Figure 5.1: Isotropic refinement of a quadrilateral element.

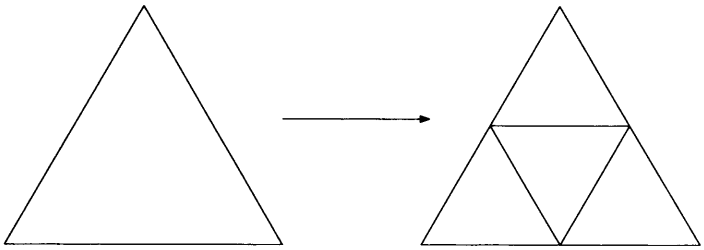


Figure 5.2: Isotropic refinement of a triangular element.

For our DG formulation, however, neither the minimum angle constraint nor the removal of hanging nodes need to be considered, but it is common to limit the number of hanging nodes to one per element face. Additionally, the following two mesh smoothing techniques are employed: (i) the removal of refined islands in the mesh; (ii) the refinement of unrefined islands. Figures 5.4 and 5.5, respectively give examples of these smoothing techniques. The use of smoothing often results in a more monotonic error convergence plot, especially in the functional estimation setting.

We now present an isotropic adaptive refinement algorithm for use in functional estimation:

1. Design an initial mesh $\mathcal{T}_h^0$.

2. (a) Solve for the primal solution u_{DG}.

   (b) Solve for the dual solution z_{DG}.

3. Compute the *a posteriori* error estimate (5.1.1).

4. If the error estimate is less than the tolerance then stop, else goto 5.

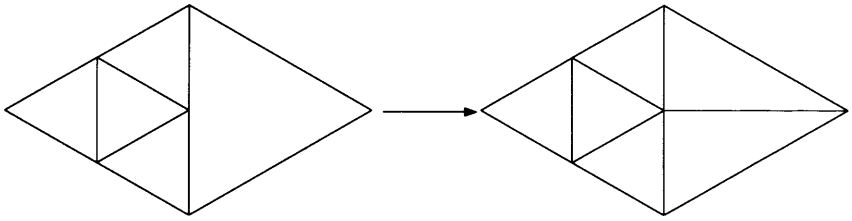


Figure 5.3: Removal of hanging nodes in a triangular mesh.

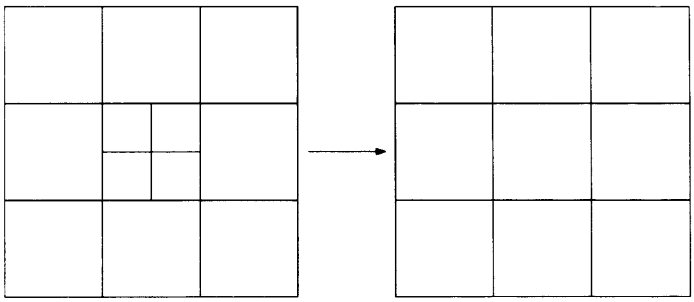


Figure 5.4: Removal of a refined island.

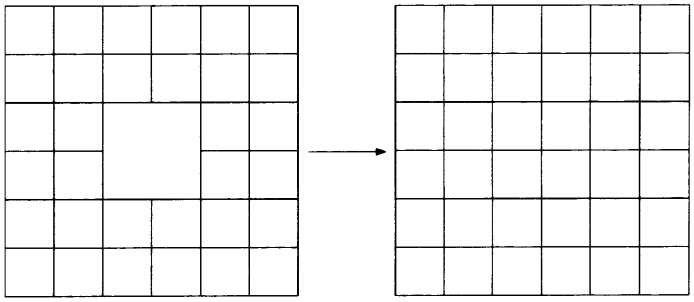


Figure 5.5: Refinement of an unrefined island.

5. Compute the individual element error indicators η_κ

6. Refine the mesh using the following:

   (a) Isotropically refine those elements where the local error estimate is high.

   (b) Derefine those elements where the local error estimate is small.

   (c) Perform mesh smoothing: removal of unrefined and refined islands and limitation of hanging nodes.

7. Goto 2.

A popular choice to decide which elements to refine is to order the elements according to the relative sizes of the local indicators η_κ and choose the top $U\%$ for refinement. In order to equidistribute the error on the mesh, derefinement of elements with small errors is also performed; similarly, as for refinement, we choose the bottom $L\%$ for derefinement. This method is called the fixed fraction strategy and we shall use it in our experiments, with $U = 20$ and $L = 10$. An alternative method would be to once again order the elements according to the relative sizes of the η_κ and select for refinement those elements with largest relative error which contribute to $T\%$ of the global error $\sum_{\kappa \in \mathcal{T}_h} \eta_\kappa$.

We shall use the above isotropic h–refinement algorithm to compare the effectiveness of the anisotropic h-refinement strategies we develop later in this chapter.

5.2.2 Anisotropic Mesh Refinement

Adaptive isotropic mesh refinement can be a powerful tool in achieving accurate numerical solutions whilst keeping computational time and storage low. However, many problems that occur in practice exhibit anisotropic behaviour, for example boundary layer phenomena in convection-dominated problems and shock formation in compressible flow. By designing anisotropic meshes which match the underlying anisotropy of the solution, significant reductions in the number of degrees of freedom required for a given accuracy, when compared with isotropic meshes, can be achieved. For a problem in two-dimensions it is very beneficial to utilize anisotropic meshes, however, for three-dimensional problems it is

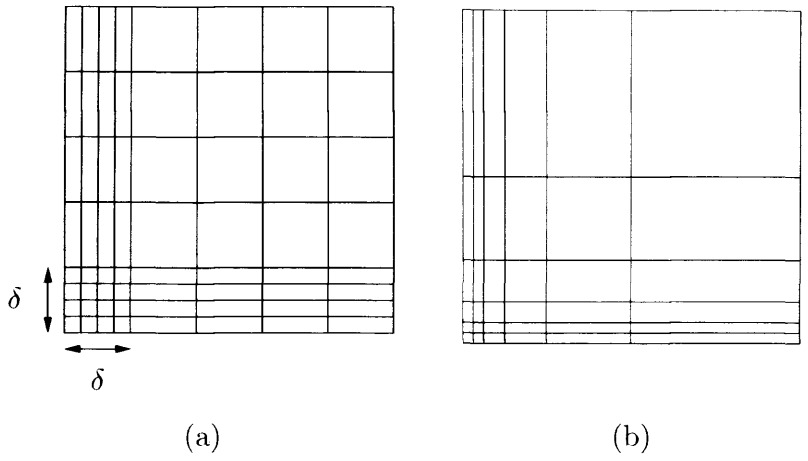


Figure 5.6: (a) A piecewise uniform mesh (b) A geometrically graded mesh.

likely to prove essential if high levels of accuracy are to be achieved while keeping computational costs relatively low. We now present a review of some of the methods employed to design anisotropic meshes.

- *Use of a priori information.* For some problems the existence and position of boundary or interior layers is known *a priori*. To capture boundary layers a common approach is to set up a piecewise uniform mesh where elements close to the boundary are stretched in the direction of the boundary but elements away from the boundary are isotropic. Meshes of this type were originally studied by Shishkin in [120]; see Figure 5.6(a) for an example of such a mesh. The problem is then how to optimally choose the length δ; Apel and Lube [9] show how to do this where a standard FEM is used to solve an advection-diffusion-reaction problem. Alternatively, meshes with elements geometrically graded toward the boundary can be used, an example of which can be seen in Figure 5.6(b). In [10], Apel and Nicaise describe how to design geometrically graded meshes, for elliptic problems with both corner and edge singularities. A further use of *a priori* information can be found in Skalický and Roos [121]; here, they exploit knowledge of the location of boundary and internal layers to design appropriate fitted meshes.

- *Location of directions of rapid change in the solution.* In Randrianarivony [111] rapidly changing directions are determined by considering the projection onto the computational domain of the unit vector normal to the surface of the solution: if this projection is long then there is large variation in that direction. A similar method is that of Kornhuber & Roitzsch, who determine anisotropy by looking at the level lines of a solution; see [88]. Jumps in the gradient are also used to drive anisotropic adaptation, for example; see Lien [99]. In [128] Sun uses a ratio of second order Taylor truncation errors to first order ones to indicate the presence of anisotropy.

- *Anisotropic mesh refinement based upon error gradients.* Apel *et al.* [8] investigate the use of local error indicators to control the error, $e_h = u - u_h$, in the norm

$$|||e_h|||^2 = \left\| \frac{\partial e_h}{\partial x} \right\|^2_{L^2(\Omega)} + \left\| \frac{\partial e_h}{\partial y} \right\|^2_{L^2(\Omega)} + \frac{1}{\varepsilon} \|e_h\|^2_{L^2(\Omega)} . \tag{5.2.1}$$

  for the problem

$$-\varepsilon \nabla^2 u + u = f, \quad (x, y) \in \Omega.$$

  By estimating each of the three terms on the right hand side of (5.2.1) the refinement is chosen which will act to equilibrate the contribution from each of them. Once equilibration has occured, isotropic mesh refinement is then utilized.

- *Hessian based anisotropic mesh refinement.* Error estimates for approximation by piecewise linear polynomials frequently bound the interpolation error in terms of the Hessian matrix of the approximated function. A direct minimization of these error bounds suggests that elements should be oriented in directions dependent on the eigenvectors of the Hessian matrix, with scale factors dependent on the ratio of corresponding eigenvalues. For a rigorous derivation of this 'Hessian strategy' we refer the reader to Formaggia and Perotto [48], where the strategy is developed simply for interpolation problems; the theory is then extended to FEMs in [49, 47]. Huang [84] investigates the case where higher order FEMs are used, advocating the use of Hessian matrices of higher order derivatives to determine anisotropy. Further work on higher order FEMs has been carried out by Cao in [34, 35]. To illustrate the popularity of this Hessian based method we give a list of some papers employing

this approach: Belhamadia *et al.* [22, 23] for the Stefan problem, Dolejší & Felcman [43] and Huang [83] for boundary layer problems, Dompierre *et al.* [58, 5, 44] and Frey & Alauzet [50] for CFD applications.

The Hessian matrix can be used to define a metric tensor $M(x) \in \mathbb{R}^{d\times d}$, which provides a way to measure distance in space. Designing a mesh which is uniform in this metric then leads to an error which is equidistributed on that mesh. A new mesh can therefore be generated at each step or, alternatively, local mesh operation such as 'Edge refinement', 'Edge coarsening', and 'Edge swapping' can be used to modify an existing mesh accordingly; see, for example, the Bidimensional Anisotropic Mesh Generator (BAMG) of Frederic Hecht [65].

- *Use of a posteriori error estimates.* As an alternative to the techniques above, which exploit *a priori* information, Schneider and Jimack [118] present an anisotropic procedure which attempts to minimize the *a posteriori* error estimate. This is done by combining elemental isotropic refinement with mesh movement. In order to create anisotropic elements aligned with the solution, a global node movement procedure is used to minimize contributions from the local error indicators. Once an anisotropic mesh has been formed, standard elemental isotropic refinement can then be used to achieve resolution.

5.3 Hessian Based Anisotropic h-refinement

Later in this chapter we shall develop a new, anisotropic, Cartesian refinement strategy based on the solution of local problems. To illustrate the effectiveness of this new strategy, we shall compare it against a Hessian based approach. Hence, in this section, motivated by the work of Formaggia *et al.* [48, 49, 47], we develop an anisotropic adaptation strategy based on the *a priori* error estimate from Theorem 4.1.5, specifically in the two-dimensional setting. The process involves writing the bound in terms of an orientation angle θ_κ and aspect ratio ς_κ for each element κ. Here, θ_κ is defined as the angle between the primary left singular vector of J_{F_κ} (*i.e.* the singular vector whose corresponding singular values has the largest value) and the x–axis. ς_κ is defined as the ratio

of the primary singular value and secondary singular value. The contribution from each element to the bound is then minimized simultaneously with respect to both θ_κ and ς_κ in order to determine an 'optimal' orientation and aspect ratio for the element. Details of this optimization can be found in Appendix A.1 for the purely diffusive case, where, due to the complexity of the problem, minimization with respect to the solutions u and z is considered independently.

For the case of approximation with bilinear elements, the results are consistent with those in [48, 49, 47], indicating that the 'optimal' scale factor should be chosen as

$$\varsigma_\kappa = \sqrt{\frac{|\mu_{1,\tilde{\kappa}}|}{|\mu_{2,\tilde{\kappa}}|}},$$

where $\mu_{1,\tilde{\kappa}}$ and $\mu_{2,\tilde{\kappa}}$ are the primary and secondary eigenvalues of the mean-valued Hessian matrix $\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}$ on the element κ, for $\tilde{v} = \tilde{u}, \tilde{z}$ and $\{\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}\}_{i,j} := (1/m_{\tilde{\kappa}}) \int_{\tilde{\kappa}} \{H^{\tilde{v}}_{\tilde{\kappa}}\}_{i,j} \mathrm{d}\tilde{x}$, for $i, j = 1, 2$. Should $\mu_{2,\tilde{\kappa}} = 0$, then a maximum scale factor S should be prescribed. Similarly, θ_κ should be chosen so that the primary singular vector of J_{F_κ} is in the same direction as the secondary eigenvector of $\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}$.

Unfortunately, a similar analysis using tensors of higher order derivatives is not possible when approximating by higher order polynomial degrees; see A.1. Nonetheless, it may be that, using numerical methods to minimize the error bound could provide the correct approach.

5.3.1 Sharpness of Anisotropic L^2-Interpolation Bound

For the optimization described above to actually yield the correct anisotropic information, an assumption on the sharpness of the interpolation error bounds of Theorem 3.3.8 has been made. Specifically, that the bounds show qualitatively the same behaviour as the true error, as the element is rotated and stretched, *i.e.*, the error bounds have, at least, local maxima and minima in the same place as the actual errors. In this subsection we shall attempt to investigate the validity of this sharpness assumption. Rather than considering all the error bounds of Theorem 3.3.8, we just examine whether the bound on the L^2–error over the element exhibits the desired property. By considering a single element, rotating and stretching it, as in Figure 5.7, and calculating the actual L^2 projection on the element,

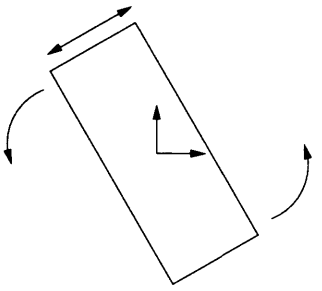


Figure 5.7: Rotation and stretching of a single element.

we can compare the L^2–error $\|v - \Pi_p v\|_{L^2(\kappa)}$ with the error bound $\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x}\right]^{1/2}$.

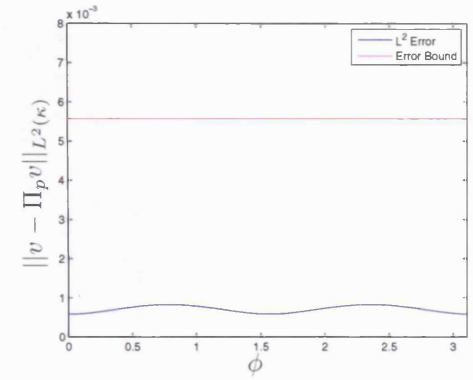
Example 1 In the first example we let

$$v = -\tanh(\sin(\pi/4)x - \cos(\pi/4)y).$$

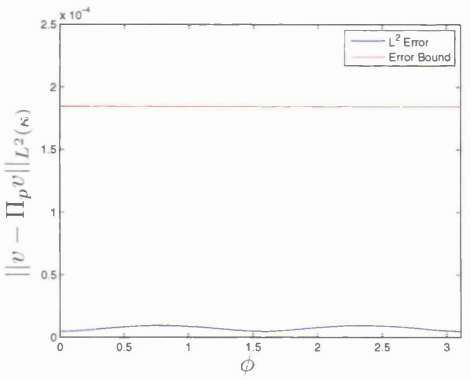
Hence, we have a function which is rapidly changing in the $[1, -1]$ direction, but does not vary at all in the $[1, 1]$ direction. Thus, we would expect elements orientated with $\phi = \pi/4$ to give the best approximation. We experiment with scale factors $\varsigma = 1$, 5, and 10 and rotate from $\phi = 0$ to $\phi = \pi$, using polynomials degrees $p = 1, 2$. Due to the Hessian being equal to zero at $[0, 0]$ we position the center of the element at $[0.2, 0]$. Figure 5.8 shows the actual errors and error bounds for the varying scale factors as the element is rotated.

We first notice that in the case when the element is isotropic, *i.e.* $\varsigma = 1$, the error bound remains constant for every value of ϕ, whereas the actual L^2 errors clearly show that the element should be orientated with $\phi = \pi/2$. For scale factors $\varsigma = 5$ and $\varsigma = 10$, we see that in the $p = 1$ case the curve of error bounds does mimic the behaviour of the actual error very well, with the minima and maxima occurring in the same place. However, we see that for $p = 2$, where we have used the tensor of third derivatives for predicting the errors, there is no longer good agreement between these estimates and the true errors. Although a local minimum of the estimate occurs in the same place as the actual error, another local minimum occurs in exactly the wrong place, that is, where the true error attains a maximum.

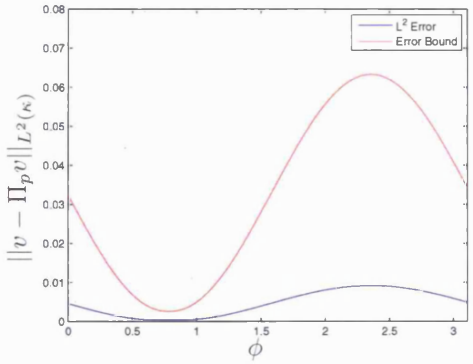
We notice, however, that the Hessian based error predictor is still very similar to the actual errors witnessed for $p = 2$, so the question arises: 'Is the Hessian always a useful



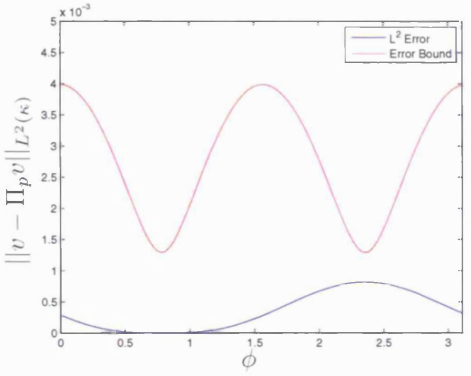
(a) $p = 1, \varsigma = 1$



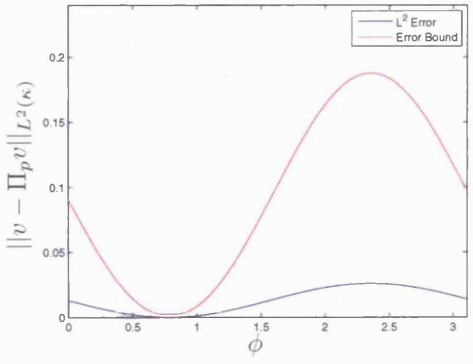
(a) $p = 2, \varsigma = 1$



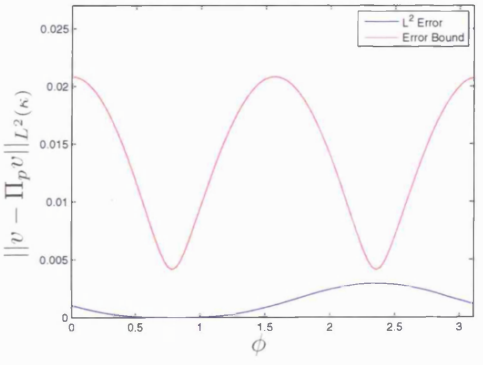
(a) $p = 1, \varsigma = 5$



(a) $p = 2, \varsigma = 5$

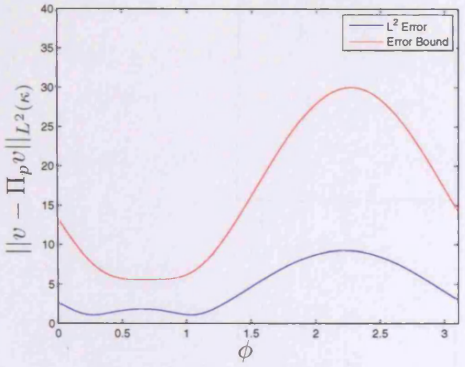


(a) $p = 1, \varsigma = 10$

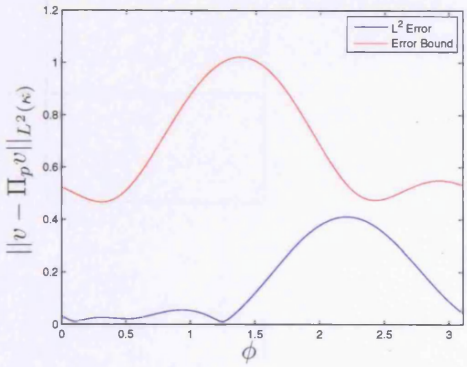


(a) $p = 2, \varsigma = 10$

Figure 5.8: Example 1: Actual L^2–errors and error bounds for varying scale factors.



(a) $p = 1$, $\varsigma = 2.18$



(a) $p = 2$, $\varsigma = 2.18$

Figure 5.9: Example 2: Actual L^2-errors and error bounds.

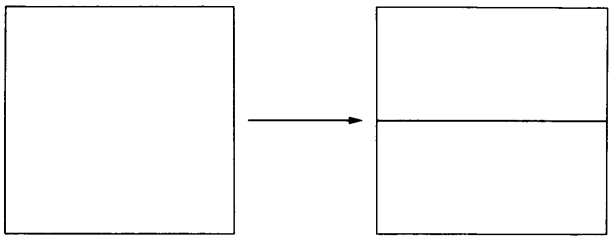
measure of the anisotropy?'. Another interesting point is that by performing a higher order eigenvalue decomposition of the tensor of third derivatives evaluated at $[0.2, 0]$, the angles inferred from the eigenvectors are $\pi/4$ and $3\pi/4$, the same as those which minimise the predicted errors. We address these issues by performing a further numerical experiment for a slightly more complicated problem.

Example 2 In this example, we consider a function v, which is a sine curve in one direction, while being exponential in the orthogonal direction. The function is rotated through an angle of $\pi/6$ so that we might expect the optimal alignment angle to be $\phi = \pi/6$. Specifically, v is given by

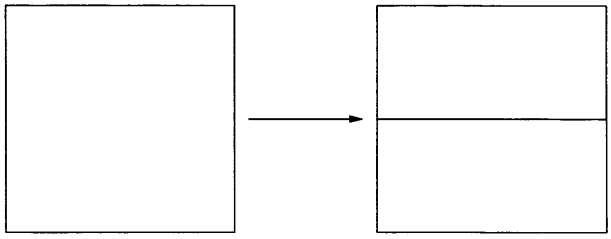
$$v = \sin(\pi(\cos(\pi/6)x + \sin(\pi/6)y))e^{10(-\sin(\pi/6)x + \cos(\pi/6)y)}.$$

We center an element about the point $[0.5, 0.5]$ and see that, based on the minimization analysis, the eigenvalue decomposition of the Hessian predicts an optimal scale factor $\varsigma \approx 2.18$ for $p = 1$. For this reason we consider only this scale factor, but rotate from $\theta = 0$ to $\theta = \pi$ as before. Figure 5.9 shows the actual errors and error bounds for $p = 1, 2$.

We see that for $p = 1$, the structure of the true errors is more complicated than for the last problem and the error bounds do not match the true errors so well. In fact the minimum of the error bounds does not occur in the same place as the true errors, but rather it coincides with a local maximum of the true errors, although the actual errors



(a)



(b)

Figure 5.10: Anisotropic Refinement of a Quadrilateral.

do have two local minima in the vicinity. For $p = 2$, again the error bound curve, based on the tensor of third derivatives, does not match the true errors and in this case there appears to be no correlation between the maxima/minima of the error bounds and the maxima/minima of the actual errors. Also, the true errors for $p = 1$ and $p = 2$ do not exhibit quite the same structure; here there are more local maxima and minima in the $p = 2$ case and the location of the global minima are different to that for $p = 1$. We also note that an eigenvalue decomposition of the tensor of third derivatives does not yield any useful information, that is, the induced angles are neither local minima/maxima of the error bounds, or of the true errors.

5.3.2 A Hessian Based Anisotropic Algorithm

In the last section we saw that the interpolation bounds of Theorem 3.3.8 may not be sufficiently sharp for the Hessian strategy to give reliable anisotropic information. Nevertheless, as it one of the most commonly used techniques for driving anisotropic refinement, new methods must be tested against it. Hence, purely for comparative purposes, we

present the following Hessian based algorithm. We consider the case where we use quadrilateral elements and only allow three refinements of the element, these being isotropic (see Figure 5.1) and anisotropic parallel to either of the reference quadrilateral's axes, as in Figure 5.10. Having marked an element for refinement based on our earlier local *a posteriori* error estimate (5.1.4), we then have to decide in which of the three ways we are going to refine.

The analysis from Appendix A.1 tells us the 'optimal' alignment and stretching factor for an element, based on information derived from either one of the primal or dual solutions. In Formaggia and Perotto [47], they base refinement on only the dual solution z, however, the error estimates clearly show some trade off between primal and dual solutions is required, hence it is important to capture information from both solutions when carrying out the refinement. To this end, we employ a method of ellipses similar to that proposed in Castro-Díaz *et al.* [36], in which the context is deciding in which direction to refine when confronted with a system of equations. On each element κ, for either the primal or dual solution, an ellipse can be associated with it by regarding the primary and secondary left eigenvectors of the appropriate Hessian matrix as the semi-minor and semi-major axes, respectively, with respective lengths $1/\sqrt{|\mu_{1,\tilde{\kappa}}|}$ and $1/\sqrt{|\mu_{2,\tilde{\kappa}}|}$. We exploit information from both solutions by computing the intersection of the ellipses, as shown in Appendix B.1. The intersected ellipse can then be used so that the ratio of the length of its semi-major axis to the length of its semi-minor axis provides the stretching factor ς_κ, while the angle between the x-axis and its semi-major axis provides the alignment angle θ_κ. Evidently the Hessian matrices must be recovered from the approximate solutions; see Appendix B.2 for details of how this is done.

The anisotropic refinements (Figure 5.10) are performed on the reference element, so we must translate the angle θ_κ to an angle $\hat{\theta}_\kappa$ on the reference cell. If $-\frac{\pi}{4} \leq \hat{\theta}_\kappa < \frac{\pi}{4}$ then we perform refinement as in Figure 5.10(a), else if $\frac{\pi}{4} \leq \hat{\theta}_\kappa < \frac{3\pi}{4}$ we perform refinement as in Figure 5.10(b). Of course, if the element has already been stretched in a direction $\bar{\theta}_\kappa$, calculated from the Jacobian of the element mapping evaluated at the element centroid such that $\bar{\theta}_\kappa - \frac{\pi}{4} \leq \theta_\kappa < \bar{\theta}_\kappa + \frac{\pi}{4}$ then we must compare the scale factor of the element $\bar{\varsigma}_\kappa$ with the optimal scale factor ς_κ. When we perform an anisotropic refinement we are

effectively doubling the scale factor, so if $\varsigma_\kappa/2 > \bar{\varsigma}_\kappa$ then we refine anisotropically, otherwise we carry out an isotropic refinement, which will retain the scale factor $\bar{\varsigma}_\kappa$. If on the other hand $\bar{\theta}_\kappa - \frac{\pi}{4} \not\leq \theta_\kappa \not< \bar{\theta}_\kappa + \frac{\pi}{4}$, then the element is oriented in the wrong direction so we carry out the anisotropic refinement regardless of the scale factor $\bar{\varsigma}_\kappa$. For every element which has been flagged for refinement we summarise the refinement process:

1. Compute the matrices defining the ellipse for both primal and dual solutions at the centroid of the element κ.
2. Compute the intersection ellipse and determine the optimal alignment angle θ_κ and scale factor ς_κ.
3. Compute the current element scale factor $\bar{\varsigma}_\kappa$ and alignment angle $\bar{\theta}_\kappa$.
4. If $\bar{\theta}_\kappa - \frac{\pi}{4} \leq \theta_\kappa < \bar{\theta}_\kappa + \frac{\pi}{4}$ then
   (a) if $\varsigma_\kappa/2 > \bar{\varsigma}_\kappa$ then calculate the local refinement angle, $\hat{\theta}_\kappa$, and anisotropically refine accordingly.
   (b) else perform isotropic refinement of the element.
5. else, calculate the local refinement angle $\bar{\theta}_\kappa - \frac{\pi}{4} \leq \theta_\kappa < \bar{\theta}_\kappa + \frac{\pi}{4}$ and anisotropically refine accordingly.

5.4 Error Optimization Approach

Although the Hessian strategy has been proven to work well in the case of approximation by linear elements we have seen that the analysis is no longer valid for approximation by higher-order polynomial degrees. The strategy is also based on *a priori* estimates, which are assumed to be sufficiently sharp, and also assume sufficient regularity of the solution, which in general may not hold. It would seem more reasonable to try and use an *a posteriori* error estimator to decide which directions to refine. Hence, we seek to find a polynomial independent refinement strategy, based purely on the *a posteriori* estimates derived in Section 5.1.1. To this end, we introduce a new local optimization strategy to

determine how an element should be refined on the basis of solving local primal and dual problems.

5.4.1 Local Problem Formulation

Suppose we have a patch of elements $\bar{\mathcal{T}}_h$, which represents some refinement of a subset of the mesh $\mathcal{T}_h$. We can solve approximate primal and dual problems locally on this patch using the global DG solutions u_{DG} and z_{DG}, respectively, to provide suitable boundary conditions. For the patch of cells $\bar{\mathcal{T}}_h$, we modify the notation from Section 2.6, so that $\bar{\Gamma}_{\mathrm{int}}$ represents the internal faces of $\bar{\mathcal{T}}_h$ and $\bar{\Gamma}$ represents those external faces of $\bar{\mathcal{T}}_h$ not contained in Γ. As for the global problem, we decompose $\bar{\Gamma}$ as

$$\begin{aligned}
\bar{\Gamma}_{\mathrm{D}} &= \{x \in \bar{\Gamma} : \mathbf{n}(x)^{\top} a(x)\mathbf{n}(x) > 0\},\\
\bar{\Gamma}_{-} &= \{x \in \bar{\Gamma}\backslash\bar{\Gamma}_{\mathrm{D}} : \mathbf{b}(x)\cdot\mathbf{n}(x) < 0\},\\
\bar{\Gamma}_{+} &= \{x \in \bar{\Gamma}\backslash\bar{\Gamma}_{\mathrm{D}} : \mathbf{b}(x)\cdot\mathbf{n}(x) \geq 0\}.
\end{aligned}$$

Hence, we shall treat the external faces of $\bar{\mathcal{T}}_h$ which are not part of the original boundary, Γ, as Dirichlet faces and use the global solutions for the boundary data. We let $\bar{u}_{\mathrm{DG}}$ be the local primal DG solution contained in $S^{\vec{\mathbf{p}}}(\bar{\mathcal{T}}_h, \mathbf{F}_{\bar{\mathcal{T}}_h})$ the restriction of $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ onto the patch $\bar{\mathcal{T}}_h$. With this notation our DG formulation for the local primal problem becomes: find $\bar{u}_{\mathrm{DG}}$ in $S^{\vec{\mathbf{p}}}(\bar{\mathcal{T}}_h, \mathbf{F}_{\bar{\mathcal{T}}_h})$ such that

$$\bar{B}_{\mathrm{DG}}(\bar{u}_{\mathrm{DG}}, v) = \bar{\ell}_{\mathrm{DG}}(v) \tag{5.4.1}$$

for all $v \in S^{\vec{\mathbf{p}}}(\bar{\mathcal{T}}_h, \mathbf{F}_{\bar{\mathcal{T}}_h})$. Here, the bilinear form $\bar{B}_{\mathrm{DG}}(\cdot,\cdot)$ is defined by

$$\bar{B}_{\mathrm{DG}}(w, v) = \bar{B}_a(w, v) + \bar{B}_{\mathbf{b}}(w, v) - \bar{B}_f(v, w) - \bar{B}_f(w, v) + \bar{B}_{\vartheta}(w, v),$$

where

$$\begin{aligned}
\bar{B}_a(w, v) &= \sum_{\kappa\in\bar{\mathcal{T}}_h}\int_{\kappa} a\nabla w\cdot\nabla v\mathrm{d}x,\\
\bar{B}_{\mathbf{b}}(w, v) &= \sum_{\kappa\in\bar{\mathcal{T}}_h}\Bigg(-\int_{\kappa}(w\mathbf{b}\cdot\nabla v - cwv)\mathrm{d}x\\
&\qquad + \int_{\partial_+\kappa}(\mathbf{b}\cdot\mathbf{n}_{\kappa})w^{+}v^{+}\mathrm{d}s + \int_{\partial_-\kappa\backslash(\Gamma\cup\bar{\Gamma})}(\mathbf{b}\cdot\mathbf{n}_{\kappa})w^{-}v^{+}\mathrm{d}s\Bigg),
\end{aligned}$$

$$\begin{aligned}\bar{B}_f(v,w) &= \int_{\bar{\Gamma}_{\rm int}\cup\Gamma_{\rm D}\cup\bar{\Gamma}_{\rm D}} \langle (a\nabla w)\cdot \mathbf{n}_f\rangle [v]\mathrm{d}s,\\ \bar{B}_\sigma(w,v) &= \int_{\bar{\Gamma}_{\rm int}\cup\Gamma_{\rm D}\cup\bar{\Gamma}_{\rm D}} \vartheta [w][v]\mathrm{d}s,\end{aligned}$$

and the linear functional $\bar{\ell}_{\rm DG}$ is given by

$$\begin{aligned}\bar{\ell}_{\rm DG}(v) = \sum_{\kappa\in\bar{\mathcal{T}}_h}\Bigg(&\int_\kappa fv\mathrm{d}x - \int_{\partial_-\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)} (\mathbf{b}\cdot\mathbf{n}_\kappa) g_{\rm D} v^+\mathrm{d}s\\ &- \int_{\partial_-\kappa\cap(\bar{\Gamma}_{\rm D}\cup\bar{\Gamma}_-)} (\mathbf{b}\cdot\mathbf{n}_\kappa) u^-_{\rm DG} v^+\mathrm{d}s - \int_{\partial_-\kappa\cap\bar{\Gamma}_{\rm D}} u^-_{\rm DG}((a\nabla v^+)\cdot\mathbf{n}_\kappa)\mathrm{d}s + \int_{\partial\kappa\cap\bar{\Gamma}_{\rm D}} \vartheta u^-_{\rm DG} v^+\mathrm{d}s\\ &- \int_{\partial_-\kappa\cap\Gamma_{\rm D}} g_{\rm D}((a\nabla v^+)\cdot\mathbf{n}_\kappa)\mathrm{d}s + \int_{\partial\kappa\cap\Gamma_{\rm N}} g_{\rm N} v^+\mathrm{d}s + \int_{\partial\kappa\cap\Gamma_{\rm D}} \vartheta g_{\rm D} v^+\mathrm{d}s\Bigg).\end{aligned}$$

For the local dual problem the formulation is slightly more complicated to establish, due to the need to impose suitable boundary conditions. The formulation becomes: find $\bar{z}\in H^2(\bar{\Omega},\bar{\mathcal{T}}_h)$ such that

$$\begin{aligned}\bar{B}_{\rm DG}(v,\bar{z}) &= \bar{J}(v) + \int_{\partial_+\kappa\cap(\bar{\Gamma}_{\rm D}\cup\bar{\Gamma}_+)} (\mathbf{b}\cdot\mathbf{n}_\kappa) z^-_{\rm DG} v^+\mathrm{d}s\\ &\quad - \int_{\partial\kappa\cap\bar{\Gamma}_{\rm D}} z^-_{\rm DG}((a\nabla v^+)\cdot\mathbf{n}_\kappa)\mathrm{d}s + \int_{\partial\kappa\cap\bar{\Gamma}_{\rm D}} \vartheta z^-_{\rm DG} v^+\mathrm{d}s, \qquad (5.4.2)\end{aligned}$$

for all $v\in H^2(\bar{\Omega},\bar{\mathcal{T}}_h)$, where $\bar{J}(\cdot)$ represents $J(\cdot)$ localized onto $\bar{\mathcal{T}}_h$.

Finally, in an analogous manner to that for the global problem, we can also define the local error estimator $\bar{\mathcal{E}}_{\bar{\mathcal{T}}_h}(\bar{u}_{\rm DG},h,p,\bar{z}-\bar{z}_{h,p})$, on the patch $\bar{\mathcal{T}}_h$ by

$$\bar{\mathcal{E}}_{\bar{\mathcal{T}}_h}(\bar{u}_{\rm DG},h,p,\bar{z}-\bar{z}_{h,p})) = \sum_{\kappa\in\bar{\mathcal{T}}_h}\bar{\eta}_\kappa$$

where

$$\begin{aligned}\bar{\eta}_\kappa = &\int_\kappa \bar{R}_{\rm int}(\bar{z}-\bar{z}_{h,p})\mathrm{d}x - \int_{\partial_-\kappa\cap(\Gamma\cup\bar{\Gamma})} (\mathbf{b}\cdot\mathbf{n}_\kappa)\bar{R}_{\rm D}(\bar{z}-\bar{z}_{h,p})^+\mathrm{d}s\\ &+ \int_{\partial_-\kappa\setminus(\Gamma\cup\bar{\Gamma})} (\mathbf{b}\cdot\mathbf{n}_\kappa)[\bar{u}_{\rm DG}](\bar{z}-\bar{z}_{h,p})^+\mathrm{d}s - \int_{\partial\kappa\cap(\Gamma_{\rm D}\cup\bar{\Gamma}_{\rm D})} \bar{R}_{\rm D}((a\nabla(\bar{z}-\bar{z}_{h,p})^+)\cdot\mathbf{n}_\kappa)\mathrm{d}s\\ &+ \int_{\partial\kappa\cap(\Gamma_{\rm D}\cup\bar{\Gamma}_{\rm D})} \vartheta\bar{R}_{\rm D}(\bar{z}-\bar{z}_{h,p})^+\mathrm{d}s + \int_{\partial\kappa\cap\Gamma_{\rm N}} \bar{R}_{\rm N}(\bar{z}-\bar{z}_{h,p})^+\mathrm{d}s\\ &+ \int_{\partial\kappa\setminus(\Gamma\cup\bar{\Gamma})} \left\{\frac{1}{2}[\bar{u}_{\rm DG}](a\nabla(\bar{z}-\bar{z}_{h,p})^+)\cdot\mathbf{n}_\kappa - \frac{1}{2}[(a\nabla\bar{u}_{\rm DG})\cdot\mathbf{n}\kappa](\bar{z}-\bar{z}_{h,p})^+\right\}\mathrm{d}s\\ &- \int_{\partial\kappa\setminus(\Gamma\cup\bar{\Gamma}_{\rm D})} \vartheta[\bar{u}_{\rm DG}](\bar{z}-\bar{z}_{h,p})^+\mathrm{d}s,\end{aligned}$$

with $\bar{z}_{h,p} \in S^{\vec{\mathbf{p}}}(\bar{\mathcal{T}}_h, \mathbf{F}_{\bar{\mathcal{T}}_h})$. As before, we have

$$\begin{aligned} \bar{R}_{\text{int}}|_\kappa &= (f - \mathcal{L}\bar{u}_{\text{DG}})|_\kappa, \\ \bar{R}_{\text{N}}|_{\partial\kappa\cap\Gamma_{\text{N}}} &= (g_{\text{N}} - (a\nabla\bar{u}^+_{\text{DG}})\cdot\mathbf{n})|_{\partial\kappa\cap\Gamma_{\text{N}}}, \end{aligned}$$

but now

$$\bar{R}_{\text{D}}|_{\partial\kappa\cap(\Gamma_{\text{D}}\cup\Gamma_-\cup\bar{\Gamma}_{\text{D}}\cup\bar{\Gamma}_-)} = \begin{cases} (g_{\text{D}} - \bar{u}^+_{\text{DG}}), & \text{for } \partial\kappa \in \Gamma_{\text{D}} \cup \Gamma_-, \\ (u^-_{\text{DG}} - \bar{u}^+_{\text{DG}}), & \text{for } \partial\kappa \in \bar{\Gamma}_{\text{D}} \cup \bar{\Gamma}_-. \end{cases} \tag{5.4.3}$$

Once again we approximate $\bar{z}$ by $\bar{z}_{\text{DG}} \in S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\text{inc}}}(\bar{\mathcal{T}}_h, \mathbf{F}_{\bar{\mathcal{T}}_h})$, where $S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\text{inc}}}(\bar{\mathcal{T}}_h, \mathbf{F}_{\bar{\mathcal{T}}_h})$, is the restriction of $S^{\vec{\mathbf{p}}+\vec{\mathbf{p}}_{\text{inc}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ onto the patch $\bar{\mathcal{T}}_h$, and use this to calculate the error estimate.

Remark 5.4.1. We notice that, for $\bar{\mathcal{T}}_h$ a subset of $\mathcal{T}_h$ with no additional refinement, for a pure advection-reaction equation the local solutions $\bar{u}_{\text{DG}}$ and $\bar{z}_{\text{DG}}$ will be identical to the global solutions u_{DG} and z_{DG} on $\bar{\mathcal{T}}_h$, respectively. For a problem with diffusion this is no longer the case as boundary conditions on $\bar{\Gamma}_{\text{D}}$ have no longer been imposed in quite the same fashion.

5.4.2 Error Optimisation Algorithm

We now present a novel use of local problems to determine in which directions to perform anisotropic refinement. As for the Hessian based approach, we consider a simple Cartesian refinement strategy, where elements can be subdivided isotropically, as in Figure 5.1, or anisotropically, as in Figure 5.10. In order to determine the optimal refinement, stimulated by the articles [110, 118], we propose the following two strategies based on choosing the most competitive subdivision of κ from a series of trial refinements, whereby an approximate local error indicator on each trial patch is determined.

`Algorithm 1`: Given an element κ in the computational mesh $\mathcal{T}_h$ (which has been marked for refinement), we first construct the mesh patches $\bar{\mathcal{T}}_{h,i}$, $i = 1, 2, 3$, based on refining κ according to Figures 5.10(a), (b) and Figure 5.1, respectively. On each mesh patch, $\bar{\mathcal{T}}_{h,i}$, $i = 1, 2, 3$, we compute the approximate error estimators

$$\bar{\mathcal{E}}_{\kappa,i}(\bar{u}_{\text{DG},i}, \bar{z}_{\text{DG},i} - \bar{z}_{h,p}) = \sum_{\kappa\in\bar{\mathcal{T}}_{h,i}} \bar{\eta}_{\kappa,i},$$

for $i = 1, 2, 3$, respectively. Here, $\bar{u}_{\mathrm{DG},i}$, $i = 1, 2, 3$, is the discontinuous Galerkin approximation to (2.2.1), (2.2.3) computed on the mesh patch $\bar{\mathcal{T}}_{h,i}$, $i = 1, 2, 3$, as discussed above. Similarly, $\bar{z}_{\mathrm{DG},i}$ denotes the discontinuous Galerkin approximation to $\bar{z}$ computed on the local mesh patch $\bar{\mathcal{T}}_{h,i}$, $i = 1, 2, 3$, respectively, with polynomials of degree $\hat{p}$.

The element κ is then refined according to the subdivision of κ which satisfies

$$\min_{i=1,2,3} \frac{|\eta_\kappa| - |\bar{\mathcal{E}}_{\kappa,i}(\bar{u}_{\mathrm{DG},i}, \bar{z}_{\mathrm{DG},i} - \bar{z}_{h,p})|}{\#\mathrm{dofs}(\bar{\mathcal{T}}_{h,i}) - \#\mathrm{dofs}(\kappa)},$$

where $\#\mathrm{dofs}(\kappa)$ and $\#\mathrm{dofs}(\bar{\mathcal{T}}_{h,i})$, $i = 1, 2, 3$, denote the number of degrees of freedom associated with κ and $\bar{\mathcal{T}}_{h,i}$, $i = 1, 2, 3$, respectively.

Algorithm 2: This is very similar to **Algorithm 1**; however, here we only construct the mesh patches $\bar{\mathcal{T}}_{h,i}$, $i = 1, 2$, and compute the approximate local primal and dual solutions on these meshes only. Given an anisotropy parameter $\theta \geq 1$, isotropic refinement is selected when

$$\frac{\max_{i=1,2} |\bar{\mathcal{E}}_{\kappa,i}(u_{\mathrm{DG},i}, \hat{z}_i - z_{h,p})|}{\min_{i=1,2} |\bar{\mathcal{E}}_{\kappa,i}(u_{\mathrm{DG},i}, \bar{z}_{\mathrm{DG}i} - \bar{z}_{h,p})|} < \theta;$$

otherwise an anisotropic refinement is performed based on which refinement gives rise to the smallest predicted error indicator, i.e., the subdivision for which $|\bar{\mathcal{E}}_{\kappa,i}(\bar{u}_{\mathrm{DG},i}, \bar{z}_{\mathrm{DG},i} - \bar{z}_{h,p})|$, $i = 1, 2$, is minimal. Based on computational experience, we select θ in the range $[1, 3]$.

Remark 5.4.2. The solution of the local problems described in the above algorithms are, computationally, relatively inexpensive, yet further reductions in cost can be achieved as the algorithms can be easily parallelized. Evidently, **Algorithm 2** will require less computational effort than **Algorithm 1** and will therefore be the most favourable to use, provided both algorithms give quantitatively similar errors for the same number of degrees of freedom.

We perform numerical experiments to test the effectiveness of the Hessian and local error estimation strategies in comparison with the standard isotropic technique in the following chapter.

Chapter 6

h-Adaptivity Numerical Experiments

In this chapter we present a number of experiments to numerically demonstrate the performance of the anisotropic adaptive algorithms proposed in Sections 5.3 and 5.4; see also [54]. All calculations were carried out using the MADNESS software package; for details; see Appendix C and Houston & Hall [69].

6.1 Example 1

In this first example we consider a linear singularly perturbed advection-diffusion problem on the (unit) square domain $\Omega = (0,1)^2$, where $a = \varepsilon I$, $0 < \varepsilon \ll 1$, $\mathbf{b} = (1,1)^\top$, $c = 0$, and f is chosen so that

$$u(x,y) = x + y(1-x) + [\mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-(1-x)(1-y)/\varepsilon}]\,[1 - \mathrm{e}^{-1/\varepsilon}]^{-1}, \tag{6.1.1}$$

cf. [76]. For $0 < \varepsilon \ll 1$, solution (6.1.1) has boundary layers along $x = 1$ and $y = 1$; throughout this section we set $\varepsilon = 10^{-2}$ and Figure 6.1 (a) shows the primal solution in this case.

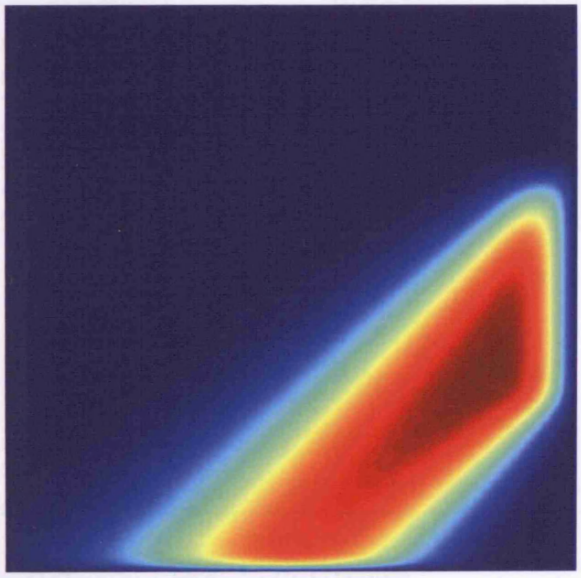
Here, we suppose that the aim of the computation is to calculate the (weighted) mean value of u over Ω, i.e., $J(u) = \int_\Omega u\psi \mathrm{d}x$, where $\psi = 100(1 - \tanh(100(r_1 - 0.01)(r_1 + 0.01)))(1 - \tanh(100(r_2 - 0.2)(r_2 + 0.2)))$, $r_1 = x - 1.0$ and $r_2 = y - 0.5$; thereby, $J(u) =$

4.409917162888037. Figure 6.1 (b) shows the dual solution when, again, $\varepsilon = 10^{-2}$. Hence the dual solution also exhibits boundary layers, but this time along $x = 1$ and $y = 0$.

To demonstrate the versatility of **Algorithms 1** and **2** of Section 5.4, here we employ bi-linear, bi-quadratic, and bi-cubic elements, i.e., $p = 1$, $p = 2$, and $p = 3$, respectively. To this end, in Figures 6.2 and 6.3 we plot the error in the computed target functional $J(\cdot)$ using both an isotropic (only) mesh refinement algorithm, together with the three anisotropic refinement strategies described in Sections 5.3 and 5.4. Additionally, in Tables 6.1 and 6.2 information is provided on the number of elements, degrees of freedom (DOF), the true functional error $|J(u)-J(u_{\mathrm{DG}})|$, the computed *a posteriori* error indicator $|\sum_{\kappa}\eta_{\kappa}|$ and the corresponding effectivity index $k = |\sum_{\kappa}\eta_{\kappa}|\,/|J(u)-J(u_{\mathrm{DG}})|$, for $p = 1$ and $p = 2$, respectively, comparing isotropic refinement with anisotropic refinement.

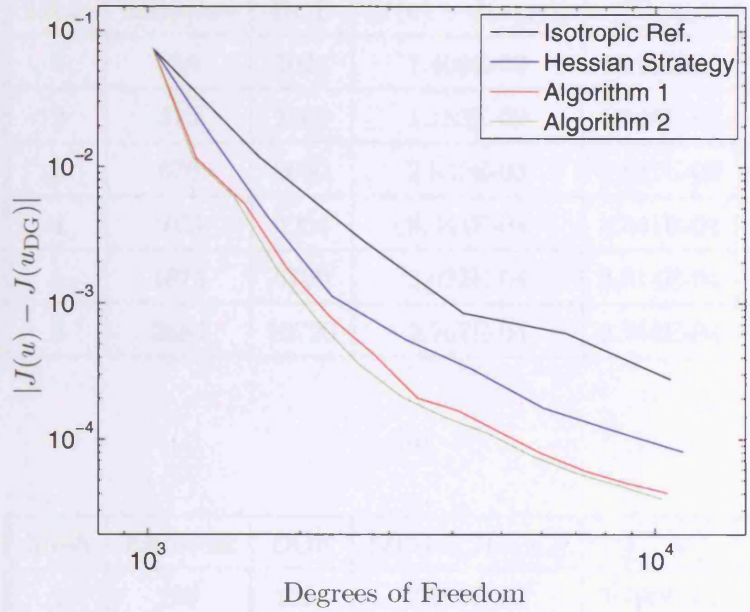
Firstly, for each polynomial degree employed, we clearly observe the superiority of employing the anisotropic mesh refinement **Algorithms 1** and **2** in comparison with standard isotropic subdivision of the elements. Indeed, the error $|J(u) - J(u_{\mathrm{DG}})|$ computed on the series of anisotropically refined meshes designed using the two proposed algorithms outlined in Section 5.4 is always less than the corresponding quantity computed on the isotropic grids. Here, we observe that there is an initial transient whereby the error in the computed target functional decays rapidly using the former refinement algorithms, in comparison with the latter, after which the gradient of the convergence curves become very similar. This type of behavior is indeed expected, since for a fixed order method, i.e. h–version, we can only expect to improve the convergence of the error by a fixed constant, as the mesh is refined. Notwithstanding this, we note that, for each polynomial degree employed, the true error between $J(u)$ and $J(u_{\mathrm{DG}})$ using anisotropic refinement is around an order of magnitude smaller than the corresponding quantity when isotropic refinement is employed alone. Secondly, we observe that for all polynomial degrees employed, the Hessian strategy is inferior to **Algorithms 1** and **2**, in the sense that the error in the target functional computed using the either of the two latter strategies is always smaller that the corresponding quantity computed using the former strategy, for a fixed number of degrees of freedom. Indeed, even for bi-linear elements, for which the Hessian strategy has been proposed on the basis of interpolation theory, **Algorithms 1** and **2** lead to a 35%

(a)

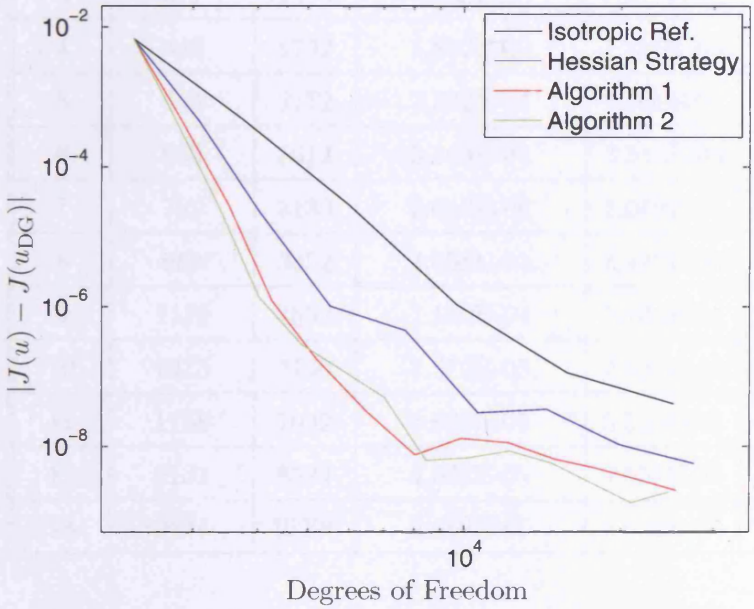


(b)

Figure 6.1: Example 1: $\varepsilon = 10^{-2}$ (a) Primal solution (b) Dual solution.



(a)



(b)

Figure 6.2: Example 1: Comparison between adaptive isotropic and anisotropic mesh refinement. (a) $p = 1$; (b) $p = 2$.

Mesh	Elements	DOF	$\lvert J(u) - J(u_{\mathrm{DG}})\rvert$	$\lvert\sum_\kappa \eta_\kappa\rvert$	k
1	256	1024	7.400E-02	7.790E-02	1.05
2	415	1660	1.183E-02	1.205E-02	1.02
3	670	2680	2.832E-03	2.847E-03	1.01
4	1051	4204	8.444E-04	8.441E-04	1.00
5	1675	6700	6.022E-04	6.014E-04	1.00
6	2680	10720	2.767E-04	2.764E-04	1.00

(a)

Mesh	Elements	DOF	$\lvert J(u) - J(u_{\mathrm{DG}})\rvert$	$\lvert\sum_\kappa \eta_\kappa\rvert$	k
1	256	1024	7.400E-02	7.790E-02	1.05
2	308	1232	1.158E-02	1.179E-02	1.02
3	371	1484	5.980E-03	5.976E-03	1.00
4	448	1792	1.812E-03	1.806E-03	1.00
5	543	2172	7.362E-04	7.301E-04	0.99
6	653	2612	3.563E-04	3.513E-04	0.99
7	795	3180	2.042E-04	2.019E-04	0.99
8	963	3852	1.458E-04	1.447E-04	0.99
9	1173	4692	1.110E-04	1.103E-04	0.99
10	1435	5740	7.572E-05	7.524E-05	0.99
11	1758	7032	5.628E-05	5.599E-05	0.99
12	2131	8524	4.572E-05	4.551E-05	1.00
13	2574	10296	3.659E-05	3.634E-05	0.99

(b)

Table 6.1: Example 1: Adaptive results for $p = 1$, (a) isotropic refinement (b) anisotropic refinement by `Algorithm 2`.

Mesh	Elements	DOF	$\lvert J(u) - J(u_{\mathrm{DG}})\rvert$	$\lvert\sum_\kappa \eta_\kappa\rvert$	k
1	256	2304	6.826E-03	6.705E-03	0.98
2	415	3735	4.786E-04	4.777E-04	1.00
3	667	6003	2.833E-05	2.833E-05	1.00
4	1075	9675	1.071E-06	1.118E-06	1.04
6	2797	25173	4.055E-08	4.022E-08	0.99

(a)

Mesh	Elements	DOF	$\lvert J(u) - J(u_{\mathrm{DG}})\rvert$	$\lvert\sum_\kappa \eta_\kappa\rvert$	k
1	256	2304	6.826E-03	6.705E-03	0.98
2	316	2844	4.792E-04	4.781E-04	1.00
3	390	3510	2.856E-05	2.834E-05	0.99
4	473	4257	1.225E-06	1.068E-06	0.87
5	577	5193	1.777E-07	1.153E-07	0.65
6	719	6471	3.209E-08	5.281E-08	1.65
7	895	8055	7.596E-09	8.215E-09	1.08
8	1098	9882	1.300E-08	1.950E-08	1.50
9	1356	12204	1.129E-08	1.170E-08	1.04
10	1662	14958	6.595E-09	7.084E-09	1.07
11	1987	17883	4.920E-09	5.508E-09	1.12
12	2388	21492	3.389E-09	3.975E-09	1.17
13	2831	25479	2.353E-09	2.946E-09	1.25

(b)

Table 6.2: Example 1: Adaptive results for $p = 2$, (a) isotropic refinement (b) anisotropic refinement by **Algorithm 1**.

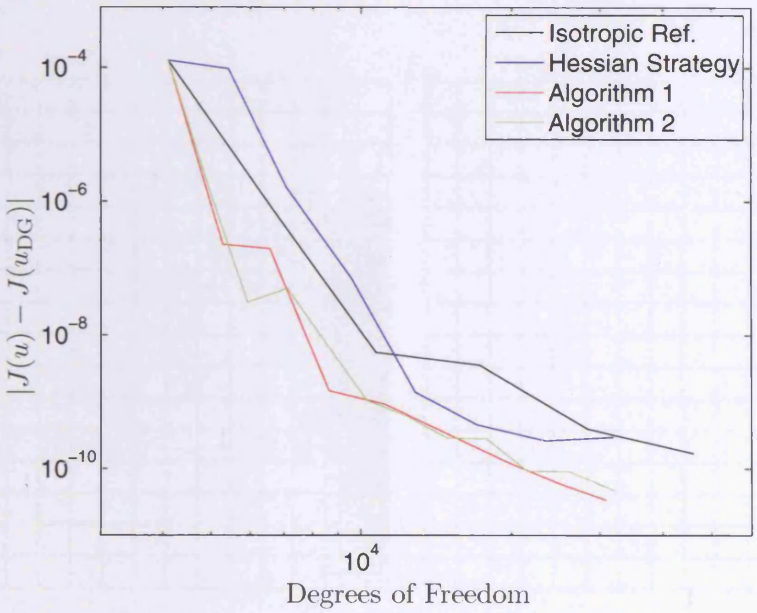


Figure 6.3: Example 1: Comparison between adaptive isotropic and anisotropic mesh refinement, $p = 3$.

reduction in the error on the final mesh in comparison with the corresponding quantity computed using the former strategy. Similar behavior is also observed for bi-quadratic and bi-cubic elements, though in the latter case, the Hessian strategy actually generates meshes which in many cases are inferior to their isotropic counterparts. Finally, we note that, despite the additional work involved in the implementation of `Algorithm 1` in comparison to `Algorithm 2`, we see that both approaches lead to quantitatively very similar reductions in the error in the computed target functional.

We also notice that on both isotropically and anisotropically refined meshes the *a posteriori* error indicators are performing extremely well. Indeed, on all the isotropic meshes the effective indices are close to 1. The same can be said for anisotropic meshes with $p = 1$, however a slight degradation of the quality occurs on highly refined anisotropic meshes with $p = 2$, nonetheless, even in this case, the error indicators are still very reliable. For $p = 3$ similar effectivities are witnessed as for $p = 2$; for brevity, these numerics have been omitted.

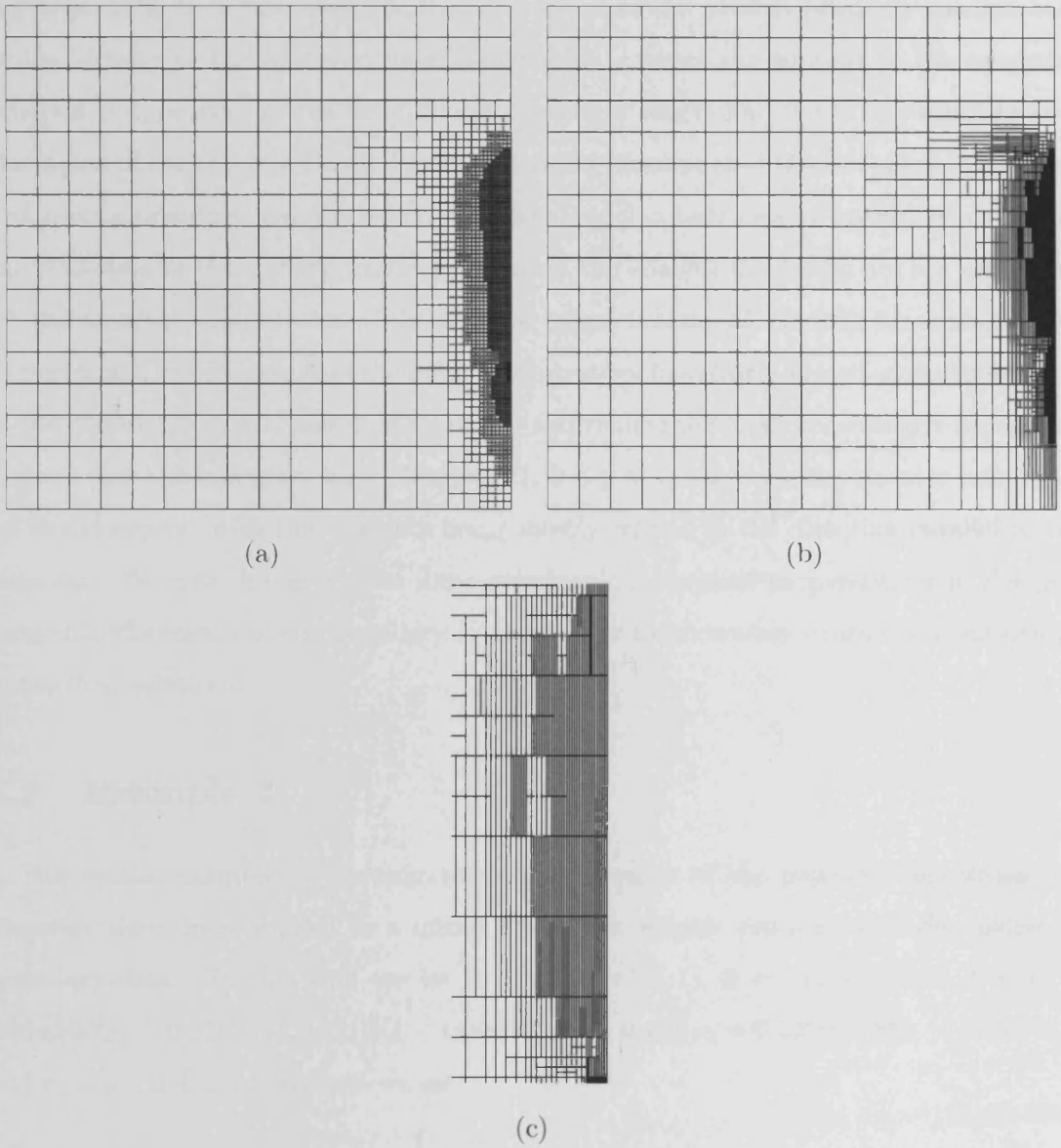


Figure 6.4: Example 1: Adaptively refined meshes for $p = 1$. (a) Isotropic mesh after 5 adaptive refinements, with 2680 elements; (b) Anisotropic mesh designed using `Algorithm 2` after 7 adaptive refinements, with 963 elements (c) Anisotropic mesh detail at (0.9,0.5).

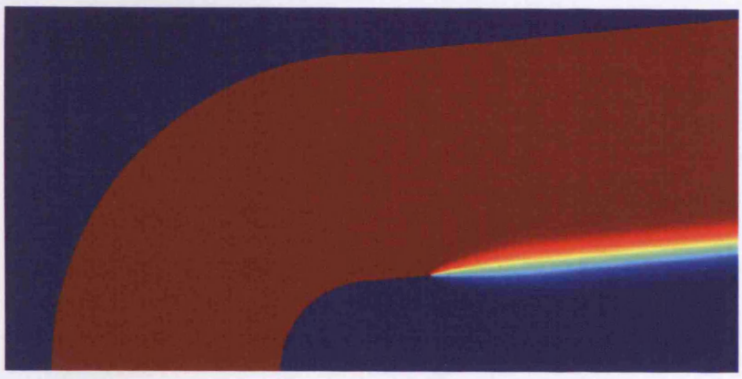
In Figure 6.4 we show the meshes generated using both isotropic and anisotropic mesh adaptation. For brevity, we only show the meshes for $p = 1$, and in the latter case employing **Algorithm 2**. Firstly, we note that in both cases the mesh is primarily concentrated in the vicinity of the boundary layer along $x = 1$, where the support of the weighting function ψ appearing in the definition of the target functional $J(\cdot)$ is non-zero. Indeed, the region of the computational domain where the remainder of the boundary layer along $x = 1$ and moreover where the boundary layer along $y = 1$ are located are not refined, since the resolution of these sharp features present in the analytical solution are not important for the accurate computation of the selected target functional, cf. [63], for example. For **Algorithm 2**, we observe that the refinement strategy has clearly identified the anisotropy in the underlying primal and dual solutions, and refined the mesh accordingly. Indeed, we observe that the boundary layer along $x = 1$, $0 \leq y \leq 1$, has been significantly refined, as we would expect, with the elements being mostly refined in the direction parallel to the boundary. We note, however, that some anisotropic refinement perpendicular to Γ is performed in the region of the boundary layer in order to accurately capture the anisotropy of the dual solution z.

6.2 Example 2

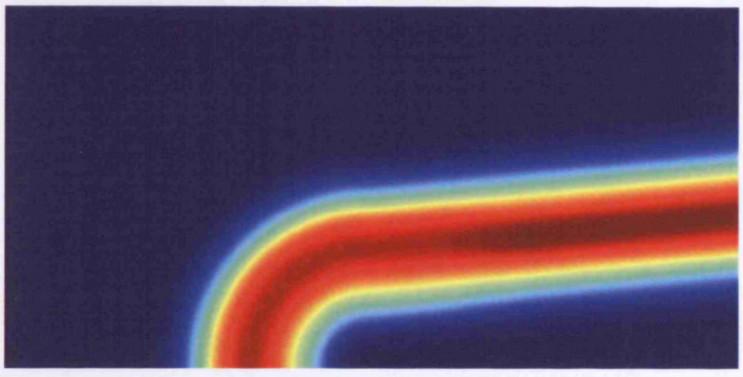
In this second example we investigate the performance of the proposed anisotropic refinement algorithms applied to a mixed hyperbolic–elliptic problem with discontinuous boundary data. To this end, we let $\Omega = (0,2) \times (0,1)$, $a = \varepsilon(x)I$, where $\varepsilon = (1 - \tanh(100(r_1 - 0.12)(r_1 + 0.12)))(1 - \tanh(100(r_2 - 0.12)(r_2 + 0.12)))/1000$, $r_1 = x - 1.3$ and $r_2 = y - 0.3$. Furthermore, we set

$$\mathbf{b} = \begin{cases} (y, 1-x)^\top & \text{if } x < 1, \\ (1, 1/10)^\top & \text{if } x \geq 1, \end{cases}$$

$c = 0$, and $f = 0$. On the inflow boundary Γ_-, we select $u(x,y) = 1$ along $y = 0$, $1/8 < x < 3/4$ and $u(x,y) = 0$, elsewhere. This is a variant of the test problem presented in [68]. We note that the diffusion parameter ε will be approximately equal to 3.6×10^{-3} in the square region $(1.18, 1.42) \times (0.18, 0.42)$, where the underlying partial differential



(a)

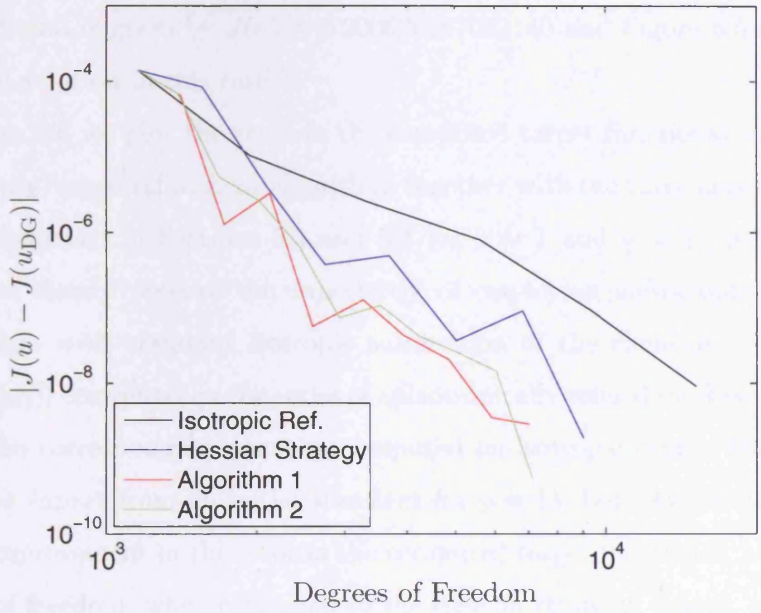


(b)

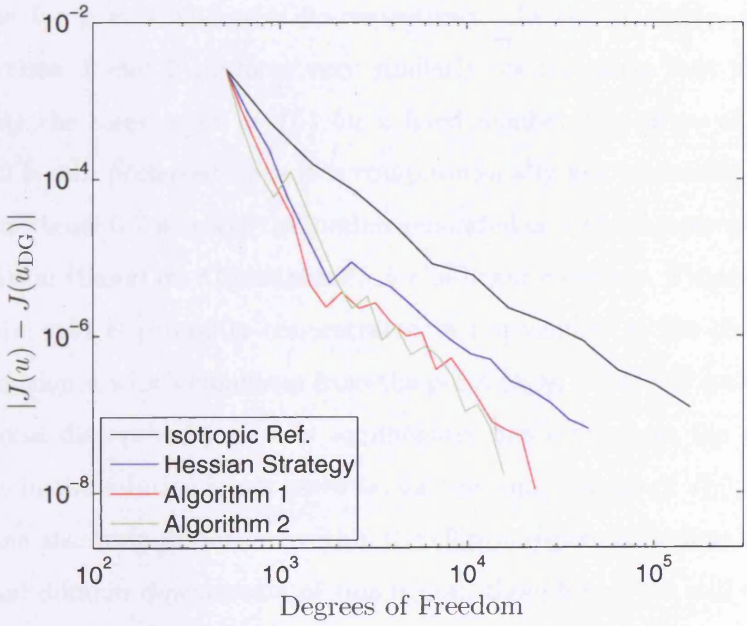
Figure 6.5: Example 2:(a) Primal solution (b) Dual solution.

equation is uniformly elliptic. As (x, y) moves outside of this region, ε rapidly decreases through a layer of width $\mathcal{O}(0.1)$; for example, when $x = 1.3$ and $y > 0.7$ we have $\varepsilon < 10^{-15}$, so from the computational point of view ε is zero to within rounding error; in this region, the partial differential equation undergoes a change of type becoming, in effect, hyperbolic. Thus, we shall refer to the part of Ω containing this square region (including a strip of size $\mathcal{O}(0.1)$) as the *elliptic region*, while the remainder of the computational domain will be referred to as the *hyperbolic region*. [Strictly speaking, the partial differential equation is elliptic in the whole of $\bar{\Omega}$.] Figure 6.5 (a) shows the exact primal solution.

Here, we suppose that the aim of the computation is to calculate the value of the (weighted) outflow advective flux along $x = 2, 0 \leq y \leq 1$, i.e., $J(u) = \int_0^1 (\mathbf{b}\cdot\mathbf{n})u(2, y)\psi(y)\mathrm{d}y$,



(a)



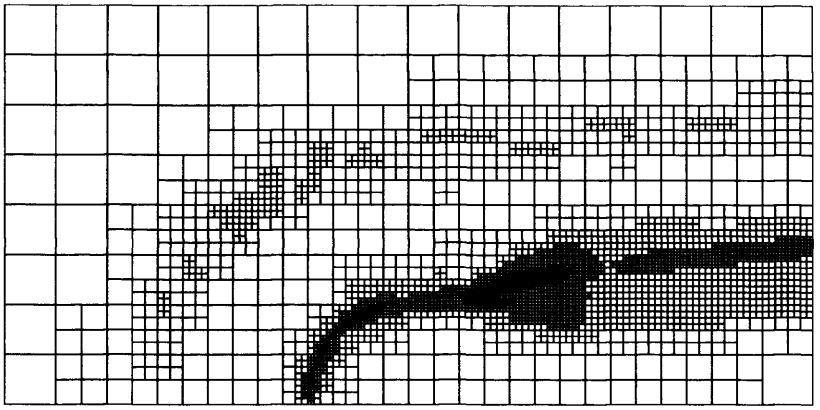
(b)

Figure 6.6: Example 2: Comparison between adaptive isotropic and anisotropic mesh refinement. (a) $p = 1$; (b) $p = 2$.

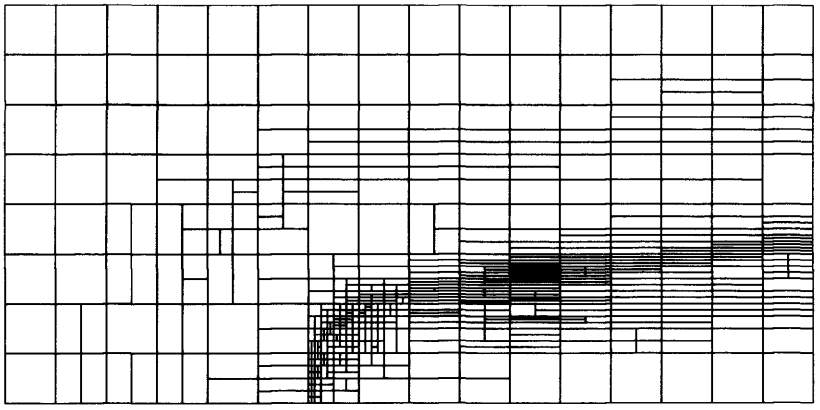
where the weight function $\psi(y) = e^{(3/8)^{-2}-((y-17/40)^2-3/8)^{-2}}$. The (approximate) true value of the functional is given by $J(u) = 0.200620167062140$ and Figure 6.5 (b) shows the resultant dual solution in this case.

In Figure 6.6 we plot the error in the computed target functional $J(\cdot)$ using both an isotropic (only) mesh refinement algorithm, together with the three anisotropic refinement strategies described in Sections 5.3 and 5.4 for $p = 1$ and $p = 2$. As for the previous example, we clearly observe the superiority of employing anisotropic mesh refinement in comparison with standard isotropic subdivision of the elements. Indeed, the error $|J(u)-J(u_{\mathrm{DG}})|$ computed on the series of anisotropically refined meshes is (almost) always less than the corresponding quantity computed on isotropic grids. Moreover, we again observe that (apart from an initial transient for $p = 1$), both `Algorithms 1` and `2` give rise to an improvement in the error in the computed target functional, for a given number of degrees of freedom, when compared to the Hessian strategy; indeed, on the final mesh, `Algorithm 2` leads an improvement in $|J(u) - J(u_{\mathrm{DG}})|$ of around one and two orders of magnitude for $p = 1$ and $p = 2$, respectively. In this example, we again observe that `Algorithms 1` and `2` perform very similarly, in the sense that they both lead to approximately the same error in $J(\cdot)$ for a fixed number of degrees of freedom, though `Algorithm 2` is still preferred since it is computationally less expensive.

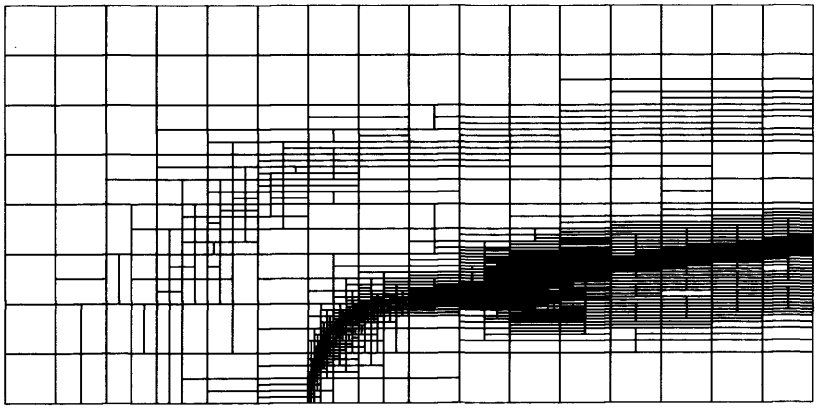
Finally, in Figure 6.7 we show the meshes generated using both isotropic and anisotropic mesh adaptation (based on `Algorithm 2`), for bi-linear elements. Firstly, we note that in both cases the grid is primarily concentrated in the vicinity of the discontinuity of the analytical solution u which emanates from the point $(x, y) = (3/4, 0)$ on the inflow boundary; the second discontinuity in u is significantly less refined, as the resolution of this sharp feature in the solution is not essential for the computation of $J(\cdot)$. Additional mesh refinement has also been performed within the elliptic region, as well as the portion of the computational domain downstream of this region, though here we still observe a general concentration of elements within the 'smoothed' discontinuity of the analytical solution. Secondly, we observe that the anisotropic refinement algorithm has clearly identified the anisotropy in the underlying primal and dual solutions, and refined the mesh accordingly. Indeed, here we observe that in regions where the discontinuities/layers in u are well



(a)

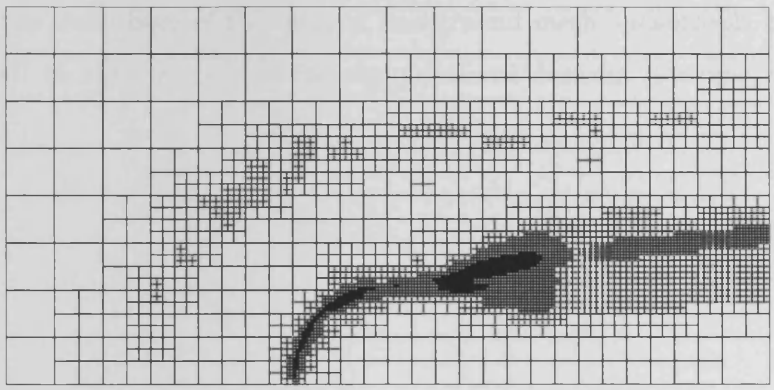


(b)

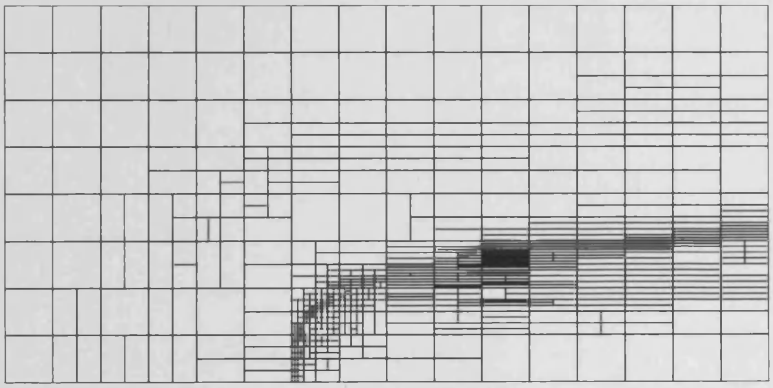


(c)

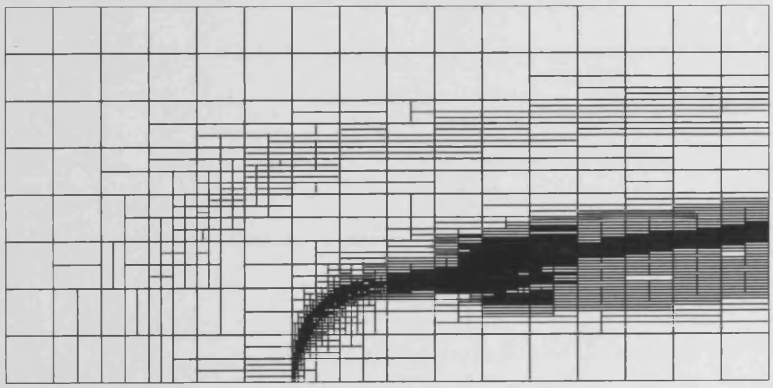
Figure 6.7: Example 2: Adaptively refined meshes for $p = 1$. (a) Isotropic mesh after 8 adaptive refinements, with 6539 elements; (b) & (c) Anisotropic meshes designed using `Algorithm 2` after: 8 adaptive refinements, with 606 elements, and 14 adaptive refinements, with 1762 elements, respectively.



(a)



(b)



(c)

Figure 6.7: Example 2: Adaptively refined meshes for $p = 1$. (a) Isotropic mesh after 8 adaptive refinements, with 6539 elements; (b) & (c) Anisotropic meshes designed using `Algorithm 2` after: 8 adaptive refinements, with 606 elements, and 14 adaptive refinements, with 1762 elements, respectively.

aligned with the mesh lines of the original background mesh, anisotropic refinement has been employed; in other regions of the computational domain, isotropic refinement has been utilized.

Chapter 7

Anisotropic *hp*-Adaptive Refinement

So far in this thesis we have only considered isotropic and anisotropic h-refinement strategies, however, Theorem 4.1.6 indicates that some balance between h-refinement and p-refinement can lead to a better reduction in error than pure h-refinement. Indeed, hp-adaptive methods, as they are known, have become increasingly popular over the years since their first analysis by Babuška and Dorr [16]. In this chapter we shall move away from the standard isotropic p-refinement techniques and attempt to develop a fully adaptive anisotropic hp-strategy for goal-oriented error estimation. First, we shall perform an analysis of the DG method in the case when anisotropic polynomials have been used, to support the need for anisotropic p. Then we shall discuss some of the methods currently used to determine whether h- or p- refinement should be carried out, before finally presenting the full anisotropic hp-algorithm.

7.1 Anisotropic *hp*–Error Analysis for Functionals

Once again we return to the case of axiparallel elements and perform a similar error analysis to that in Chapter 4 and obtain the proceeding theorem. In a slight variation to the assumptions required for the Theorem 4.1.6, new bounded local variation conditions on the element sizes and polynomial degrees are required: we suppose that there exist ρ_i

and δ_i, for $i = 1, 2$ such that

$$\rho_i^{-1} \leq p_i^{\kappa}/p_i^{\kappa'} \leq \rho_i, \tag{7.1.1}$$

$$\delta_i^{-1} \leq h_i^{\kappa}/h_i^{\kappa'} \leq \delta_i, \tag{7.1.2}$$

$i = 1, 2$, for all pairs of neighbouring elements κ and κ'.

Theorem 7.1.1. *Let $\Omega \subset \mathbb{R}^2$ be a an axiparallel polygonal domain, $\mathcal{T}_h = \{\kappa\}$ a subdivision of Ω into axiparallel images of the 2-hypercube, such that the bounded local variation conditions, (7.1.1) and (7.1.2), hold. Then, assuming that conditions (2.3.2), (4.1.3), and (4.1.2) on the data hold, and $u|_\kappa \in H^{k_\kappa}(\kappa)$, $k_\kappa \geq 2$, for $\kappa \in \mathcal{T}_h$ and $z \in H^{l_\kappa}(\kappa)$, $l_\kappa \geq 2$, for $\kappa \in \mathcal{T}_h$, then the solution $u_{\mathrm{DG}} \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ of (2.6.5) obeys the error bound*

$$\begin{aligned}
|J(u) - J(u_{\mathrm{DG}})|^2 \leq\ & C\Big(\sum_{\kappa \in \mathcal{T}_h} \sum_{i=1}^{2} \Phi(p_i^{\kappa}, s_i^{\kappa}, h_i^{\kappa}) \max_{(m,n)\in A} \Big\{ \Big(\frac{p_j^{\kappa}}{p_i^{\kappa}}\Big)^m \Big(\frac{h_i^{\kappa}}{h_j^{\kappa}}\Big)^n \Big\} \\
& \times \Big(\alpha_\kappa p_i^{\kappa} + \frac{h_i^{\kappa}}{p_i^{\kappa}}\beta_2 + \Big(\frac{h_i^{\kappa}}{p_i^{\kappa}}\Big)^2 (\beta_1 + \gamma_1)) |u|^2_{s_i^{\kappa},\kappa,i} \Big) \\
& \times \Big(\sum_{\kappa \in \mathcal{T}_h} \sum_{i=1}^{2} \Phi(p_i^{\kappa}, t_i^{\kappa}, h_i^{\kappa}) \max_{(m,n)\in A} \Big\{ \Big(\frac{p_j^{\kappa}}{p_i^{\kappa}}\Big)^m \Big(\frac{h_i^{\kappa}}{h_j^{\kappa}}\Big)^n \Big\} \\
& \times \Big(\alpha_\kappa p_i^{\kappa} + h_i^{\kappa}\beta_2 + \Big(\frac{h_i^{\kappa}}{p_i^{\kappa}}\Big)^2 (\beta_1 + \gamma_2)) |z|^2_{t_i^{\kappa},\kappa,i} \Big),
\end{aligned}$$

with $A = \{(0,0), (0,1), (0,2), (-1,0), (-1,1)(1,2), (2,1), (2,2)\}$, and

$$|w|_{r,\kappa,i} := \Big(\|\tilde{\partial}_i^r w\|^2_{\tilde{\kappa}} + \Big(\frac{h_j^{\kappa}}{h_i^{\kappa}}\Big)^2 \|\tilde{\partial}_i^{r-1} \tilde{\partial}_j w\|^2_{\tilde{\kappa}} \Big)^{1/2}, \quad i, j = 1, 2, \quad i \neq j,$$

for $2 \leq s_i^{\kappa} \leq \min(p_\kappa + 1, k_\kappa)$ and $2 \leq t_i^{\kappa} \leq \min(p_\kappa + 1, l_\kappa)$, where $\alpha|_\kappa = \bar{a}_{\tilde{\kappa}}$, $\beta_1|_\kappa = \|c + \nabla \cdot \mathbf{b}\|_{L_\infty(\kappa)}$, $\beta_2|_\kappa = \|\mathbf{b}\|_{L_\infty(\kappa)}$, $\gamma_1|_\kappa = \|c/c_0\|^2_{L_\infty(\kappa)}$, $\gamma_2|_\kappa = \|(c + \nabla \cdot \mathbf{b})/c_0\|^2_{L_\infty(\kappa)}$, for all $\kappa \in \mathcal{T}_h$. Here, C is a constant depending on the parameters δ_i, ρ_i, $i = 1, 2$.

Proof. Inequality (4.1.14) is also applicable in this case, by rearranging the terms we

obtain

$$
\begin{aligned}
|J(u) - J(u_{\mathrm{DG}})|^2 \;\leq\; & C \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \bar{a}_{\bar{\kappa}} \left(\|\nabla \eta\|^2_{L^2(\kappa)} + \frac{\bar{a}_{\bar{\kappa}}}{\vartheta} \|\nabla \eta\|^2_{L^2(\partial\kappa)} + \frac{\vartheta}{\bar{a}_{\bar{\kappa}}} \|\eta\|^2_{L^2(\partial\kappa)} \right) \right. \right. \\
& \left. \left. + \beta_2 \left(\varepsilon_\kappa^{-1} \|\nabla \eta\|^2_{L^2(\kappa)} + \|\eta\|^2_{L^2(\partial\kappa)} \right) \right) + (\beta_1 + \gamma_1) \|\eta\|^2_{L^2(\kappa)} \right\} \Bigg) \\
& \times \left(\sum_{\kappa \in \mathcal{T}_h} \left\{ \bar{a}_{\bar{\kappa}} \left(\|\nabla w\|^2_{L^2(\kappa)} + \frac{\bar{a}_{\bar{\kappa}}}{\vartheta} \|\nabla w\|^2_{L^2(\partial\kappa)} + \frac{\vartheta}{\bar{a}_{\bar{\kappa}}} \|w\|^2_{L^2(\partial\kappa)} \right) \right. \right. \\
& \left. \left. \beta_2 \left(\varepsilon_\kappa \|w\|^2_{L^2(\kappa)} + \|w\|^2_{L^2(\partial\kappa)} \right) + (\beta_1 + \gamma_2) \|w\|^2_{L^2(\kappa)} \right\} \right) . \\
\equiv\; & C \left(\sum_{\kappa \in \mathcal{T}_h} I^\kappa_{1,\eta} + I^\kappa_2 + I^\kappa_3 \right) \times \left(\sum_{\kappa \in \mathcal{T}_h} I^\kappa_{1,w} + I^\kappa_4 + I^\kappa_5 \right) .
\end{aligned}
$$

For term $I^\kappa_{1,\eta}$ (similarly $I^\kappa_{1,w}$) we first split into contributions from the faces $\partial\kappa_i$ and $\partial\kappa_j$, such that

$$
I^\kappa_{1,\eta} \;\leq\; \bar{a}_{\bar{\kappa}} \left(\|\nabla \eta\|^2_{L^2(\kappa)} + \sum_{i=1}^{2} \frac{\bar{a}_{\bar{\kappa}}}{\vartheta} \|\nabla \eta\|^2_{L^2(\partial\kappa_i)} + \frac{\vartheta}{\bar{a}_{\bar{\kappa}}} \|\eta\|^2_{L^2(\partial\kappa_i)} \right) .
$$

Then, employing the interpolation result from Lemma 3.4.4 we obtain

$$
\begin{aligned}
\|\nabla \eta\|^2_{L^2(\kappa)} \;&\leq\; C \sum_{i=1}^{2} \left(p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \|\tilde{\partial}_i^{s_i^\kappa} \tilde{u}\|^2 + \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \|\tilde{\partial}_i^{s_i^\kappa - 1} \tilde{\partial}_j \tilde{u}\|^2 \right) \\
&\leq\; C \sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \left[1 + \left(\frac{h_i^\kappa}{h_j^\kappa} \right)^2 \right] |u|^2_{s_i^\kappa, \kappa, i} \\
&\leq\; C \sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{n=\{0,2\}} \left(\frac{h_i^\kappa}{h_j^\kappa} \right)^n |u|^2_{s_i^\kappa, \kappa, i} .
\end{aligned}
$$

By using the definition of the discontinuity penalization term ϑ from (2.7.15), the results of Lemma 3.4.6 and utilizing the bounded local variation conditions, we also see that

$$
\begin{aligned}
\sum_{i=1}^{2} \frac{\bar{a}_{\bar{\kappa}}}{\vartheta} \|\nabla \eta\|^2_{L^2(\partial\kappa_i)} \;&\leq\; C \sum_{i=1}^{2} \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \\
&\quad \times \left(\frac{h_j^\kappa}{(p_j^\kappa)^2} \left[\left(\frac{p_i^\kappa h_j^\kappa}{h_i^2} \left(1 + \frac{p_i^\kappa}{p_j^\kappa} \right) + \frac{(p_j^\kappa)^2}{h_j^\kappa} \right) \|\tilde{\partial}_i^{s_i^\kappa - 1} \tilde{\partial}_j \tilde{u}\|^2 \right. \right. \\
&\quad \left. \left. + \frac{2(p_i^\kappa)^2}{h_j^\kappa} \|\tilde{\partial}_i^{s_i^\kappa} \tilde{u}\|^2 \right] + \frac{h_i^\kappa}{(p_i^\kappa)^2} \left[\frac{(p_i^\kappa)^3}{h_i^\kappa} \|\tilde{\partial}_i^{s_i^\kappa} \tilde{u}\|^2 + \frac{p_i^\kappa}{h_i^\kappa} \|\tilde{\partial}_i^{s_i^\kappa - 1} \tilde{\partial}_j \tilde{u}\|^2 \right] \right) \\
&\leq\; C \sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \left[\left(\frac{p_j^\kappa}{p_i^\kappa} \right)^{-1} + 1 + \left(\frac{h_i^\kappa}{h_j^\kappa} \right)^2 \right] |u|^2_{s_i^\kappa, \kappa, i}
\end{aligned}
$$

$$\leq C\sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{(m,n)\in A_1} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |u|^2_{s_i^\kappa,\kappa,i},$$

where $A_1 = \{(0,0),(0,2),(-1,0)\}$.

Similarly, by using Lemma 3.4.5 we obtain

$$\sum_{i=1}^{2} \frac{\vartheta}{\bar{a}_\kappa} \|\eta\|_{L^2(\partial\kappa_i)} \leq C\sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{(m,n)\in A_2} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |u|^2_{s_i^\kappa,\kappa,i},$$

where $A_2 = \{(1,2),(2,2)\}$. Hence, it follows that

$$I_{1,\eta}^\kappa \leq C\sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{(m,n)\in A} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |u|^2_{s_i^\kappa,\kappa,i},$$

and

$$I_{1,w}^\kappa \leq C\sum_{i=1}^{2} p_i^\kappa \Phi(p_i^\kappa, t_i, h_i^\kappa) \max_{(m,n)\in A} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |z|^2_{t_i^\kappa,\kappa,i}.$$

For terms I_2^κ and I_4^κ we make the selection $\varepsilon_\kappa = \max_{i=1,2}((p_i^\kappa)^2/h_i^\kappa)$ and using the same techniques as above we deduce that

$$\begin{aligned} I_2^\kappa &\leq \beta_2 \sum_{i=1}^{2} \frac{h_i^\kappa}{p_i^\kappa} \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{(m,n)\in A_3} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |u|^2_{s_i^\kappa,\kappa,i}, \\ I_4^\kappa &\leq \beta_2 \sum_{i=1}^{2} h_i^\kappa \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) \max_{(m,n)\in A_4} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |z|^2_{t_i^\kappa,\kappa,i}. \end{aligned}$$

where $A_3 = \{(0,0),(0,1),(-1,1)\}$ and $A_4 = \{(0,0),(0,1),(2,1),(2,2)\}$.

A simple use of Lemma 3.4.4 also yields

$$\begin{aligned} I_3^\kappa &\leq (\beta_1+\gamma_1) \sum_{i=1}^{2} \left(\frac{h_i^\kappa}{p_i^\kappa}\right)^2 \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) |u|^2_{s_i^\kappa,\kappa,i}, \\ I_5^\kappa &\leq (\beta_1+\gamma_2) \sum_{i=1}^{2} \left(\frac{h_i^\kappa}{p_i^\kappa}\right)^2 \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) |w|^2_{t_i^\kappa,\kappa,i}. \end{aligned}$$

Combining the results for terms I_1-I_5 completes the proof. ■

Remark 7.1.2. Upon application of Stirling's formula for the factorials arising in the definition of Φ, as in Remark 3.4.7, it can be shown that the error estimate stated in Theorem 4.1.6 is h-optimal and slightly p-suboptimal (by one order of p). This is in complete agreement with the results presented for the isotropic case in [61].

When the analytical solution of both the primal and the dual problems are sufficiently smooth, then it can be shown that the error converges to zero at an exponential rate with respect to the local (directional) polynomial degrees. More precisely, we state the following result.

Corollary 7.1.3. *Let $\Omega \subset \mathbb{R}^2$ be a bounded polyhedral domain, $\mathcal{T} = \{\kappa\}$ a 1-irregular subdivision of Ω, such that the mesh parameters satisfy the bounded local variation conditions (7.1.1) and (7.1.2). Then, assuming that conditions (2.3.2), (4.1.3), and (4.1.2) hold, and that u, z are analytic functions on a neighbourhood of Ω, the solution $u_{\mathrm{DG}} \in S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ of (2.6.5) obeys the error bound*

$$|J(u) - J(u_{DG})|^2 \leq C(\alpha, \beta_1, \beta_2, \gamma_1, \gamma_2) \times \Big(\sum_{\kappa \in \mathcal{T}} \sum_{i=1}^{2} \mathrm{e}^{-r_i p_i^\kappa} N_i^\kappa \Big) \Big(\sum_{\kappa \in \mathcal{T}} \sum_{i=1}^{2} \mathrm{e}^{-q_i p_i^\kappa} N_i^\kappa \Big).$$

where

$$N_i^\kappa := \left(h_i^\kappa\right)^{2s_i^\kappa} m_{\tilde{\kappa}} \max_{(m,n) \in A} \left\{ \left(p_i^\kappa\right)^{4-m} \left(p_j^\kappa\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n \right\},$$

r_i, q_i are positive constants depending on the domain of analyticity of u and z, respectively, and $m_{\tilde{\kappa}}$ is the Lebesgue measure of $\tilde{\kappa}$; the set A and the data-related constants $\alpha, \beta_1, \beta_2, \gamma_1$, and γ_2 are as in the statement of Theorem 7.1.1.

Proof. The result follows simply after applying Lemmas 3.4.8 and 3.4.9 to

$$\Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) |u|^2_{s_i^\kappa, \kappa, i} = \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \Big(\|\tilde{\partial}_i^{s_i} u\|^2_{\tilde{\kappa}} + \Big(\frac{h_j^\kappa}{h_i^\kappa}\Big)^2 \|\tilde{\partial}_i^{s_i - 1} \tilde{\partial}_j u\|^2_{\tilde{\kappa}} \Big),$$

and similarly to

$$\Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) |z|^2_{t_i^\kappa, \kappa, i} = \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) \Big(\|\tilde{\partial}_i^{t_i} z\|^2_{\tilde{\kappa}} + \Big(\frac{h_j^\kappa}{h_i^\kappa}\Big)^2 \|\tilde{\partial}_i^{t_i - 1} \tilde{\partial}_j z\|^2_{\tilde{\kappa}} \Big).$$

■

The results of Theorem 4.1.6 clearly show that in the case where the Sobolev regularity of the primal solution u or the dual solution z exceed the polynomial degree of the approximating solutions it will be more beneficial to increase the polynomial degree rather than decreasing the size of the mesh. Indeed, in the case where u and z are real analytic functions polynomial enrichment can lead to exponential convergence. Results have even

been shown that for problems with corner singularities a careful choice of mesh refinement into the corner together with increasing polynomial degree away from the corner can still lead to exponential convergence; see, for example, [119, Section 4.5].

Further, the convergence estimates from Theorem 7.1.1 and Corollary 7.1.3 indicate that it might be advantageous to use anisotropic polynomial degrees as well as anisotropic elements when constructing a finite element space. In Sections 8.1 and 8.2 we perform simple numerical experiments to motivate the use of anisotropic polynomial degrees, where anisotropic elements and anisotropic polynomial degrees are chosen based on *a priori* knowledge of the primal and dual solutions. We see that a very simple *a priori* anisotropic strategy can produce a surprisingly large improvement over isotropic polynomial degrees. Driven on by these results we now proceed to describe an automatic anisotropic *hp*-adaptive strategy.

7.2 Adaptive Strategy

In the goal-oriented setting the target error is highly dependent on both the primal and dual solutions, hence the smoothness of both must be taken into account. Indeed, based on the *a priori* error analysis if either the primal or dual is 'smooth' then it is natural to perform p-enrichment, while if both are 'non-smooth' h-refinement would seem the logical choice. With this in mind we propose the following adaptive algorithm for anisotropic *hp*-adaptation:

1. Select an element κ for refinement or derefinement.
2. Estimate the 'smoothness' of both u and z on the element κ.
3. If the element has been selected for refinement.
   (a) If u or z is smooth perform anisotropic p-refinement, else
   (b) Perform anisotropic h-refinement,

   else
4. If the element κ has been selected for derefinement.

(a) If u or z is smooth on κ perform h-derefinement, else

(b) Perform p-derefinement.

5. Stop.

Anisotropic h-refinement has been investigated in depth in Chapter 5, thus the two important issues are how to identify whether a function is smooth or not and how to perform anisotropic p-refinement. We consider the former in the next section and the latter in Section 7.4.

7.3 Smoothness Estimation

It is essential that we have some way of assessing the regularity of the solution on an element which has been flagged for refinement/derefinement. Much research has gone into this and we present a brief review of some of these techniques, together with our preferred approach, in this section.

- *Use of a priori information.* For some problem types it is possible to know beforehand where the solution is regular and where it is not. Consider, for example, a linear elliptic boundary value problem, with piecewise analytic coefficients, forcing functions and boundary data on a computational domain with a piecewise analytic boundary surface. In this case the solution will have singularities in the neighbourhood of corners of the domain and singularities in the boundary data. Thus, elements in the neighbourhood of these singularities can simply be chosen for subdivision, whilst all others can be selected for p-refinement. Examples of this method can be found in [25, 133]. Of course for more complicated problems, the location of singularities is not known *a priori*, so this method has limited appeal.

- *Type-Parameter.* Suppose that on each element κ in the computational mesh, an error indicator $\eta_\kappa(u_{h,p}, h_\kappa, p_\kappa)$ dependent on the numerical solution $u_{h,p}$, the element size h_κ and the polynomial degree p_κ can be calculated. Suppose also that one has

access to $\eta_\kappa(u_{h,p-1}, h_\kappa, p_\kappa - 1)$. Defining ζ_κ, by

$$\zeta_\kappa = \begin{cases} \eta_\kappa(u_{h,p}, h_\kappa, p_\kappa)/\eta_\kappa(u_{h,p-1}, h_\kappa, p_\kappa - 1), & \eta_\kappa(u_{h,p-1}, h_\kappa, p_\kappa - 1) \neq 0, \\ 0, & \text{otherwise} \end{cases}$$

and selecting the parameter, $0 < \gamma < 1$, if $\zeta_\kappa \leq \gamma$ then the element is said to be of *p-type* and a p-refinement is carried out, otherwise the element is said to be of *h-type* and the element subdivided. Here, γ is known as the *type parameter*; see [57].

A number of possibilities then exist. For example:

1. On an element with polynomial degree p_κ, a local problem is solved with new polynomial degree $p_\kappa+1$ and the error indicator $\eta_\kappa(u_{h,p+1}, h_\kappa, p_\kappa+1)$ calculated. The ratio of the computed error indicators then yields the type. For example, see [57].
2. Alternatively, global solutions with polynomial degrees p and $p-1$ can be solved at the outset and hence ζ_κ is immediately available for each κ. This method has been considered in, for example, [2].

- *Mesh optimization procedure.* This method is similar to `Algorithms 1` and `2` introduced in Section 5.4 and to `Algorithms 3` and `4` which will be introduced later in this chapter, in that it is based on *competitive refinements*. Suppose first that a reference solution has been calculated on some fine mesh. Now suppose that an element in the computational mesh has been chosen for refinement, where the refinement to be performed is to be taken from a finite list of possibilities. By projecting the reference solution onto the local space formed by each of the possible refinements an indication of the expected reduction in error per degree of freedom increase can be calculated. Hence the refinement which maximizes the error reduction per degree of freedom is chosen. Evidently setting up the reference solution is computationally expensive, but the mesh after refinement should be 'optimized' and this cost negated. This method was first proposed in [110], with extensions in [42] and [122], where in the latter, goal-oriented adaptation is considered.

- *'Texas 3 Step'.* The 'Texas 3 Step' as it has become known was first introduced by Oden *et al.* [107] with further work undertaken in [26, 106]. It is not so much a means

to determine local regularity of a solution, but rather seeks to optimally choose the element size and polynomial degrees across the domain to achieve a given level of accuracy. Two error tolerance are chosen, Tol_{I} and Tol_{F}, standing for Intermediate and Final, respectively. The eponymous steps then follow as:

1. Setup an initialize mesh, which lies in the h-asymptotic range. Then, by equating *a priori* and *a posteriori* error estimates, determine properties of the solution so that it is possible to:
2. Perform adaptive h-refinement to ensure the error is below Tol_{I} and is equidistributed across the mesh. Recompute the *a posteriori* error estimates and again equate with the *a priori* error estimate in order to:
3. Calculate the polynomial distribution which achieves equidistribution of the error across the mesh, subject to the error begin less than the Tol_{F}. Enrich the mesh and check that the tolerance is actually satisfied.

- *Predicted error reduction.* The idea of predicting the new error indicator on a refined element, based on *a priori* error estimates was first introduced by Melenk and Wohlmuth [104]. Suppose that an element has been chosen for either h- or p-refinement, then the *a priori* estimates allow, assuming the solution is locally smooth, a predicted error indicator $\eta_{\kappa_i}^{\mathrm{pred}}$ to be calculated on the resultant elements κ_i, based on the current error indicator, η_κ. Evidently, as p-refinement results in exponential convergence $\eta_{\kappa_i}^{\mathrm{pred}}$ in this case will be different to the case when h-refinement is used and only algebraic convergence is witnessed. Thus, the next time one of the resultant elements is chosen for refinement, the actual error indicator η_{κ_i} can be compared with the predicted error indicator $\eta_{\kappa_i}^{\mathrm{pred}}$. If it happens that $\eta_{\kappa_i} > \eta_{\kappa_i}^{\mathrm{pred}}$ then the solution is not smooth and h-refinement will be chosen, otherwise the solution is smooth and p-refinement is picked. On the initial mesh, the choice must therefore be made either to perform h-refinement of any selected elements and set $\eta_{\kappa_i}^{\mathrm{pred}} = 0$ for all κ, or set $\eta_{\kappa_i}^{\mathrm{pred}} = \infty$ and perform p-refinement, cf., also, [67].

- *Rate of decay of Legendre expansion coefficients.* First proposed in [102] is the idea of using the decay rate of the Legendre coefficients of a function to determine

whether it is locally smooth or non-smooth. In one-dimension, for an approximate solution of polynomial order p, the Legendre coefficients a_i, for $i = 0, \ldots, p$, are readily obtainable. Real analytic functions exhibit exponential decay of the Legendre coefficients, that is $a_i \sim \mathrm{e}^{-\sigma i}$. By using the available Legendre coefficients, an estimate of σ can be found by means of a least-squares fit. In [102] the choice was made that if $\sigma < 1$ the function was not smooth enough to warrant p-refinement and an h-refinement carried out, otherwise p-refinement was utilized. Further work can also be found in Houston & Süli [81], see below. Melenk and Eibner [45] extend the theory which this technique is based upon to the case of triangular and tetrahedral elements.

- *Local regularity estimation.* By attempting to explicitly approximate the local Sobolev regularity k_κ of a solution, p-refinement can be performed on those elements where $k_\kappa > p_\kappa + 1$, and h-refinement performed otherwise. Techniques to approximate the Sobolev regularity were considered in [4], where they came about as a direct result of the *a posteriori* error estimation of elliptic boundary problems. Suppose the analytical solution is denoted by u and the numerical solution u_h. The idea is to solve local problems for the error $u - u_h$ in order to calculate *a posteriori* estimates, which can then be corrected using *a priori* estimates of the local problems. By solving the local problem for multiple polynomial degrees, the terms involved in the *a priori* error estimates can be computed and the Sobolev regularity of $u - u_h$ estimated. Noticing that the Sobolev index of the error $(u - u_h)|_\kappa$ restricted to the element is the same as that of $u|_\kappa$, the local Sobolev regularity of the solution has also been found. Extensions of this method to linear and non-linear hyperbolic problems have been considered in [77, 79, 126].

  The assessment of local Sobolev regularity has also been considered in [78], where in this case the coefficients of the Legendre polynomials are analyzed to directly reveal the regularity.

Due to the simplicity and economy of its implementation and the robustness witnessed in numerical experiments, the method we choose to use is based on the penultimate algo-

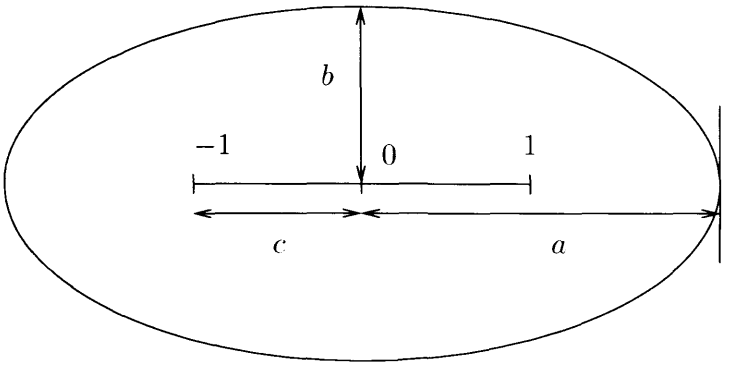


Figure 7.1: Bernstein ellipse on the interval $\hat{I} = (-1, 1)$.

rithm discussed above and involves estimating the domain of analyticity of a solution by considering the decay rates of the Legendre expansion coefficients; it was first introduced by Houston *et al.* in [81]. Consider a function v, defined on the one-dimensional reference domain $\hat{I} = (-1, 1)$ and associate with it the Bernstein ellipse $\hat{\mathcal{E}}_\rho$, defining the complex domain of analyticity of v, with foci $x = \pm 1$ and radius $\rho = (a + b)/c \geq 1$, where a and b are the lengths of the semi–major and semi–minor axes, respectively, and c is equal to half the length of the interval $\hat{I}$, cf. Figure 7.1. In the case $\rho = 1$, then we have the degenerate situation $a = 1$, $b = 0$ and $\hat{\mathcal{E}}_\rho = [-1, 1]$; thus v is singular in $\hat{I}$. The following result then holds.

Theorem 7.3.1. *Let $z \mapsto v(z)$ be analytic in the interior of $\hat{\mathcal{E}}_\rho$, $\rho > 1$, but not in the interior of any other $\hat{\mathcal{E}}_{\rho'}$ with $\rho' > \rho$. Then the Legendre series*

$$v(z) = \sum_{i=0}^{\infty} b_i L_i(z), \quad b_i = \frac{2i+1}{2} \int_{-1}^{1} v(z) L_i(z) \mathrm{d}z \tag{7.3.1}$$

converges absolutely and uniformly on any closed set in the interior of $\hat{\mathcal{E}}_\rho$ and diverges in the exterior of $\hat{\mathcal{E}}_\rho$. Moreover,

$$\frac{1}{\rho} = \limsup_{i \to \infty} |b_i|^{1/i}. \tag{7.3.2}$$

Conversely, if $(b_i)_{i \geq 0}$ is a sequence satisfying (7.3.2) with some $\rho > 1$, then the Legendre series (7.3.1) converges absolutely and uniformly on any closed set inside $\hat{\mathcal{E}}_\rho$ to an analytic function $z \mapsto v(z)$ satisfying (7.3.1)-(7.3.2). The series diverges in the exterior of $\hat{\mathcal{E}}_\rho$.

Proof. See Davis [41], Theorem 12.4.7, for details. ■

We now extend the above theorem to the case where the interval is no longer the standard $\hat{I} = [-1, 1]$, but rather $I = [x_1, x_2]$, with mesh size $h_I = (x_2 - x_1)/2$. In order to do this we follow the discussion presented in [81] and introduce the family of $L^2(I)$-orthogonal polynomials $\{L_i^I(x)\}_{i=0}^{\infty}$ and we deduce that

$$L_i^I = (1/h_I)^{1/2} L_i((x - m_I)/h_I),$$

where m_I is the midpoint of the interval I, hence $m_I = (x_1 + x_2)/2$. By the completeness of $\{L_i^I(x)\}_{i=0}^{\infty}$ in $L^2(I)$, for a function $v \in L^2(I)$ we can write

$$v(x)|_I = \sum_{i=0}^{\infty} a_i^I L_i^I(x), \quad \text{where } a_i^I = \frac{2i+1}{2} \int_I v(x) L_i^I(x) \mathrm{d}x. \tag{7.3.3}$$

With this notation, the analogue of Theorem 7.3.1 on an interval I holds, in which case the new local Bernstein ellipse $\hat{\mathcal{E}}_{\rho_I}$ has foci at x_1, x_2 and radius $\rho_I = (a_I + b_I)/h$, where $a_I \geq h_I$ and b_I are the lengths of the semi-major and semi-minor axes, respectively. Indeed, with the Legendre coefficients of v being defined as in (7.3.3), if v is analytic in the interior of $\hat{\mathcal{E}}_{\rho_I}$, but not in the interior of any $\hat{\mathcal{E}}_{\rho_I'}$ with $\rho_I' > \rho_I$ the elemental Bernstein radius satisfies

$$\frac{1}{\rho_I} = \limsup_{i \to \infty} |a_i^I|^{1/i}, \tag{7.3.4}$$

for some $\rho_I > 1$. Hence,

$$\theta_I = \frac{1}{\rho_I} \tag{7.3.5}$$

is some measure of the domain of analyticity of v relative to the size of the interval I. By considering the local analogue of Theorem 7.3.1 we deduce that $0 \leq \theta_I \leq 1$: with $\theta_I = 0$ indicating an entire analytic function, in contrast $\theta_I = 1$ corresponds to functions with singular support in I. We also remark that for a fixed, real analytic function v an h-refinement of the interval I will always yield an increase in the relative size of the domain of analyticity of v and, therefore, a decrease of θ_I.

The discussion above only considers functions defined in one-dimension and assumes a complete knowledge of the Legendre coefficients; however, we need to be able to determine smoothness based on our approximate solutions u_{DG} and z_{DG} and deal with higher spatial

dimensions. To extend the result to higher dimensions we simply choose the centroid of the reference element and evaluate the Legendre coefficients in each coordinate direction separately, a value θ^j can then be obtained for each $j = 1, \ldots, d$. Slightly more complicated is how to use the approximate solution to obtain the analyticity estimate. Suppose that (for the d-hypercube) we have used a polynomial degree vector $\{p_j\}_{j=1}^d$ to approximate the solution, then approximations to the first $p_j + 1$ Legendre coefficients are readily obtained, bearing in mind our basis functions are no more than the tensor products of Legendre polynomials; see Section C.5. One solution, motivated by (7.3.4), would be to approximate θ^j by $\hat{\theta}^j = |a_{p_j}|^{1/p_j}$, for $j = 1, \ldots, d$. However, for many functions $a_{p_i} = 0$; consider, for example, functions whose Legendre coefficients have repeating patterns of zero coefficients, such as functions which are symmetric or antisymmetric about the midpoint and hence $\hat{\theta}_i$ is liable to give a poor approximation to θ_i.

Instead we attempt to use information from all of the available Legendre coefficients. Once again, using (7.3.4) we see that if v is analytic on I, then $|a_i| \sim (1/\rho_I)^i$, as $i \to \infty$, which in turn implies $\log |a_i^I| \sim i \log(1/\rho_I)$, as $i \to \infty$. So, by performing linear regression to estimate the slope, μ_I of $|\log |a_i^I|| = i\mu_I + b_I$, for the available a_i^I, $i = 0, \ldots p_j$, the estimate $\hat{\theta^j}$ becomes

$$\hat{\theta}^j = \mathrm{e}^{-\mu_j}, \qquad i = 1, \ldots d. \tag{7.3.6}$$

We notice that the closed form for μ_j is given by

$$\mu_j = 6 \frac{2 \sum_{i=0}^{p_j} i y_i - p_j \sum_{i=0}^{p_j} y_i}{(p_j + 1)((p_j + 1)^2 - 1)},$$

with $y_i = |\log |a_i^j||$, for $i = 0, \ldots, p_j$. For $p_j \geq 1$, $j = 1, \ldots, d$, we will have enough data to calculate an approximation $\hat{\theta}^j$, although, evidently, the higher the polynomial degree used the better the approximation is likely to be. For this reason, we shall start computations on grids with uniform polynomial degrees, with $p_j = 2$, $j = 1, \ldots, d$.

Equipped with our approximations for $\hat{\theta}^j$, $j = 1, \ldots, d$, we must decide how this translates into whether a function is 'smooth' or not. We make the decision that, if for any $j = 1, \ldots, d$, $\theta^j \leq \theta$, where θ is a user defined parameter (selected to be 0.25 in our experiments), the function is smooth enough that p-refinement be carried out, otherwise h-refinement must suffice.

Having decided upon whether a function is smooth or not we now turn our attention to how to perform anisotropic p-refinement.

7.4 Anisotropic p-Refinement Strategies

Although we obtain smoothness estimates in each direction, these do not provide us with sufficient information to decide in which directions we would wish to increase the polynomial degree, because they give no information about the potential error reduction. We are, therefore, motivated to use a similar strategy to that presented for anisotropic h-refinement, that is, competitive refinement based on the error indicators calculated on trial patches. Once again we consider cartesian meshes and solve local problems based on increasing the polynomial degree anisotropically in one direction at a time by one degree, or isotropically by one degree. Figure 7.2 provides a visualisation of the local mesh patches in two-dimensions, where the original polynomial degree vector on the element of interest is $\vec{p}_\kappa = [p_1, p_2]$. The formulations presented in Section 5.4 for the local solution of the primal and dual problems are used and once again we present two possible anisotropic algorithms; see also [56].

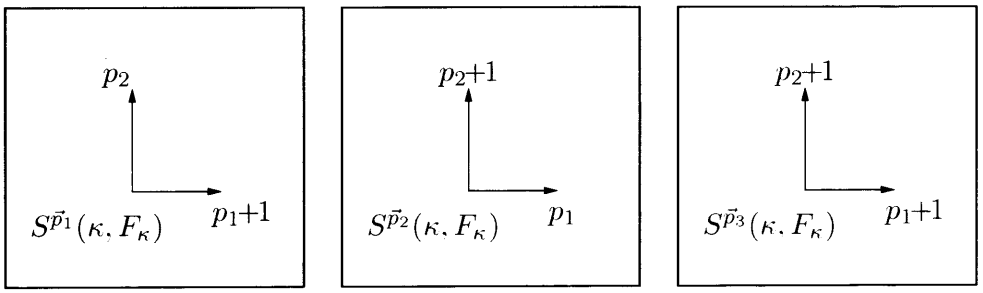


Figure 7.2: Polynomial Enrichment in 2D: (a) & (b) Anisotropic Enrichment: (c) Isotropic Enrichment.

`Algorithm 3`: This algorithm is completely analogous to the `Algorithm 1` of Section 5.4. Given an element κ in the computational mesh $\mathcal{T}_h$ (which has been marked for refinement), we first construct the finite element spaces $S^{\vec{p}_i}(\kappa, F_\kappa)$, $i = 1, 2, 3$, based on enriching $\vec{p}_\kappa$ according to Figure 7.2, respectively. On each finite element space $S^{\vec{p}_i}(\kappa, F_\kappa)$,

$i = 1, 2, 3$, we compute the approximate error estimators

$$\bar{\mathcal{E}}_{\kappa,i}(\bar{u}_{\mathrm{DG},i}, \bar{z}_{\mathrm{DG},i} - \bar{z}_{h,p}) = \bar{\eta}_{\kappa,i},$$

for $i = 1, 2, 3$, respectively. Here, $\bar{u}_{\mathrm{DG},i}$, $i = 1, 2, 3$, is the discontinuous Galerkin approximation to (2.2.1), (2.2.3) computed on the space $S^{\vec{p}_i}(\kappa, F_\kappa)$, $i = 1, 2, 3$, as discussed in Section 5.4. Similarly, $\bar{z}_{\mathrm{DG},i}$ denotes the discontinuous Galerkin approximation to $\bar{z}$ computed on the local space $S^{\vec{p}_i + \vec{p}_{\mathrm{inc}}}(\kappa, F_\kappa))$, $i = 1, 2, 3$, respectively.

The element κ is then refined according to the subdivision of κ which satisfies

$$\min_{i=1,2,3} \frac{|\eta_\kappa| - |\bar{\mathcal{E}}_{\kappa,i}(\bar{u}_{\mathrm{DG},i}, \bar{z}_{\mathrm{DG},i} - \bar{z}_{h,p})|}{\#\mathrm{dofs}(S^{\vec{p}_i}(\kappa, F_\kappa)) - \#\mathrm{dofs}(S^{\vec{p}}(\kappa, F_\kappa))},$$

where $\#\mathrm{dofs}(S^{\vec{p}}(\kappa, F_\kappa))$ and $\#\mathrm{dofs}(S^{\vec{p}_i}(\kappa, F_\kappa))$, $i = 1, 2, 3$, denote the number of degrees of freedom associated with $S^{\vec{p}}(\kappa, F_\kappa)$ and $S^{\vec{p}_i}(\kappa, F_\kappa)$, $i = 1, 2, 3$, respectively.

`Algorithm 4`: This is very similar to `Algorithm 2` of Section 5.4 where only the mesh patches $\bar{T}_{h,i}$, $i = 1, 2$, corresponding to the anisotropic refinements are carried out. However, as the original element may have had anisotropic polynomial degree, the number of degrees of freedom on the refined mesh patches are no longer directly comparable, hence a slight modification is needed. Thus, given an anisotropy parameter $\omega \geq 1$, isotropic refinement is selected when

$$\frac{\max_{i=1,2}(|\eta_\kappa| - \bar{\mathcal{E}}_{\kappa,i})/(\#\mathrm{dofs}(S^{\vec{p}_i}(\kappa, F_\kappa)) - \#\mathrm{dofs}(S^{\vec{p}}(\kappa, F_\kappa)))}{\min_{i=1,2}(|\eta_\kappa| - \bar{\mathcal{E}}_{\kappa,i})/(\#\mathrm{dofs}(S^{\vec{p}_i}(\kappa, F_\kappa)) - \#\mathrm{dofs}(S^{\vec{p}}(\kappa, F_\kappa)))} < \omega,$$

otherwise an anisotropic refinement is performed based on which enrichment gives rise to the smallest predicted error indicator, i.e., the subdivision for which $(|\eta_\kappa| - \bar{\mathcal{E}}_{\kappa,i})/(\#\mathrm{dofs}(S^{\vec{p}_i}(\kappa, F_\kappa)) - \#\mathrm{dofs}(S^{\vec{p}}(\kappa, F_\kappa)))$, $i = 1, 2$, is minimal. Here, for brevity we have written $\bar{\mathcal{E}}_{\kappa,i}$ in lieu of $\bar{\mathcal{E}}_{\kappa,i}(\bar{u}_{\mathrm{DG},i}, \bar{z}_{\mathrm{DG},i} - \bar{z}_{h,p})$. Based on computational experience, we select ω in the range $[2, 3]$. Once again we observe that both of the above algorithms are fully parallelizable.

Remark 7.4.1. We remark that we could have combined `Algorithm 1` or `2` with either `Algorithm 3` or `Algorithm 4` to decide which is the best refinement to perform, h- or p-, without needing to estimate the smoothness of u and z, in a technique very similar to the *mesh optimization strategy* of [110]. Indeed, numerical experiments have revealed

very similar convergence plots as those using the smoothness test. However, compared to solving the local problems, computational expense to ascertain smoothness is minimal and is therefore preferred.

In the case when both u and z are non-smooth, we wish to perform a derefinement of the polynomial degree vector. To avoid the unnecessary solving of local problems, we simply perform an isotropic p-derefinement, that is, we reduce the polynomial by one degree in each coordinate direction.

7.4.1 Smoothing

For isotropic p-refinement, a smoothing of the resultant polynomials across the domain has proven to be essential in achieving a smooth reduction in the error. Standard techniques involve ensuring that the polynomial jump across an element face does not exceed more than one degree, cf. Figure 7.3(a). Other techniques are the removal of unrefined islands, an analogue of the h-refinement case; see Figure 7.3(b). We simply extend these smoothing methods to the anisotropic setting by regarding the polynomial degree in each coordinate direction separately.

7.5 Full Anisotropic hp-Adaptive Algorithm

We are now in a position to describe a fully anisotropic hp-adaptive algorithm for a goal oriented problem, where the error needs to be below a prescribed tolerance Tol. Figure 7.4 shows this algorithm as a flow chart.

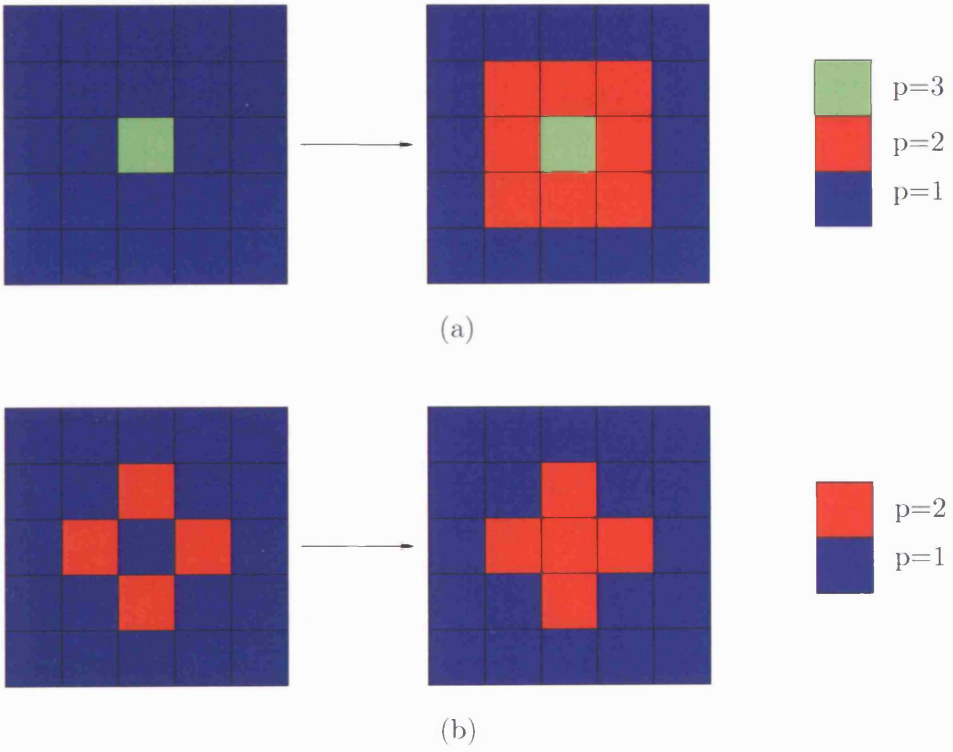


Figure 7.3: *p*-smoothing techniques.

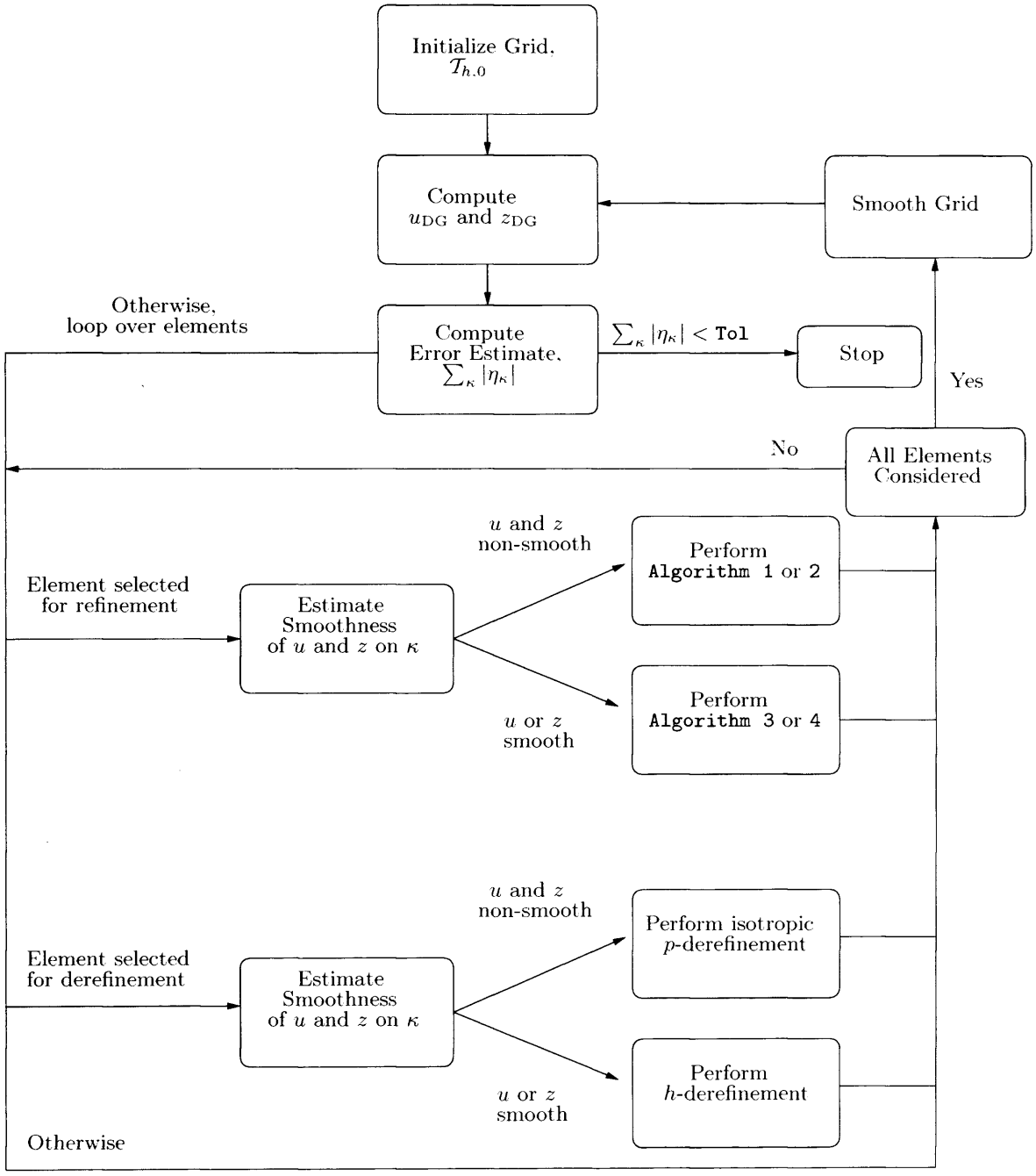


Figure 7.4: Anisotropic hp-adaptive algorithm.

Chapter 8

hp-Adaptivity Numerical Experiments

In this chapter we present the results of four numerical experiments: the first two are aimed motivate the use of isotropic and anisotropic polynomial degree enrichment, respectively, cf. [55], while the final two show the performance of the anisotropic hp-strategies developed in Section 7.4, cf. [56].

8.1 Example 1

In this first example, we consider the following singularly perturbed advection–diffusion–reaction problem proposed in [100]:

$$-\varepsilon\Delta u + 2u_x + 3u_y + u = f,$$

for $(x, y) \in (0, 1)^2$, where $0 < \varepsilon \ll 1$ and f is selected so that

$$u(x, y) = 2\sin x \left(1 - \mathrm{e}^{-2(1-x)/\varepsilon}\right) y^2 \left(1 - \mathrm{e}^{-(1-y)/\varepsilon}\right). \tag{8.1.1}$$

Throughout this section we set $\varepsilon = 10^{-3}$; in this case, the analytical solution (8.1.1) has boundary layers along $x = 1$ and $y = 1$, cf. Figure 8.1(a). Here, we suppose that the aim of the computation is to calculate the (unweighted) mean-value of u over the entire

computational domain Ω, i.e.,

$$J(u) = \int_{\Omega} u \mathrm{d}\mathbf{x};$$

thereby, the true value of the functional is given by $J(u) = 0.320534488112846$. This functional leads to the development of boundary layers in the analytical solution z to the corresponding dual problem (4.1.1) along $x = 0$ and $y = 0$, cf. Figure 8.1(b).

In this section we investigate the effectiveness of employing local anisotropic h–refinement in comparison to enriching the polynomial degree within (boundary) layer regions for the accurate computation of above target functional of interest. To this end, starting from a uniform 5×5 square mesh, we first perform n, $n \geq 0$, anisotropic refinements of this initial mesh within boundary layer regions. Here, we consider two types of refined computational meshes: the first, referred to as Type I, is constructed to resolve the boundary layers present in u by refining only elements which lie on the right-hand side or top boundaries of Ω, cf. Figure 8.2(a) for the case when $n = 5$. Secondly, in view of the boundary layers present in the dual problem z, we also consider a sequence of computational meshes, referred to as Type II, based on refining all elements which lie on the boundary of Ω, cf. Figure 8.2(b) for the case when $n = 5$. The design of this latter set of meshes is inspired by the fact that the accurate computation of target functionals typically requires the resolution of important features present in both the primal and dual solutions, cf. [21, 63], for example.

Once the mesh has been constructed, this is kept *fixed*, while the polynomial degree is uniformly (and isotropically) increased, starting from $\vec{p} = [1, 1]$. In Figures 8.3 and 8.4 we plot the (square root of) the degrees of freedom employed in the finite element space $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ against the error in the computed target functional $J(\cdot)$ using mesh Type I and mesh Type II, respectively. Firstly, we note that, after an initial transient, the convergence lines are (on average) straight, indicating exponential rates of convergence as the polynomial degree is increased, cf. Corollary 7.1.3. Moreover, we observe that in both cases, as the resolution of the mesh in the boundary layer is initially increased, i.e., as n becomes larger, the effectiveness of enriching the polynomial degree (uniformly) increases, in the sense that the error in the computed target functional $J(\cdot)$ decreases for a fixed number of degrees of freedom. However, as n is increased further, the error in $J(\cdot)$, for a

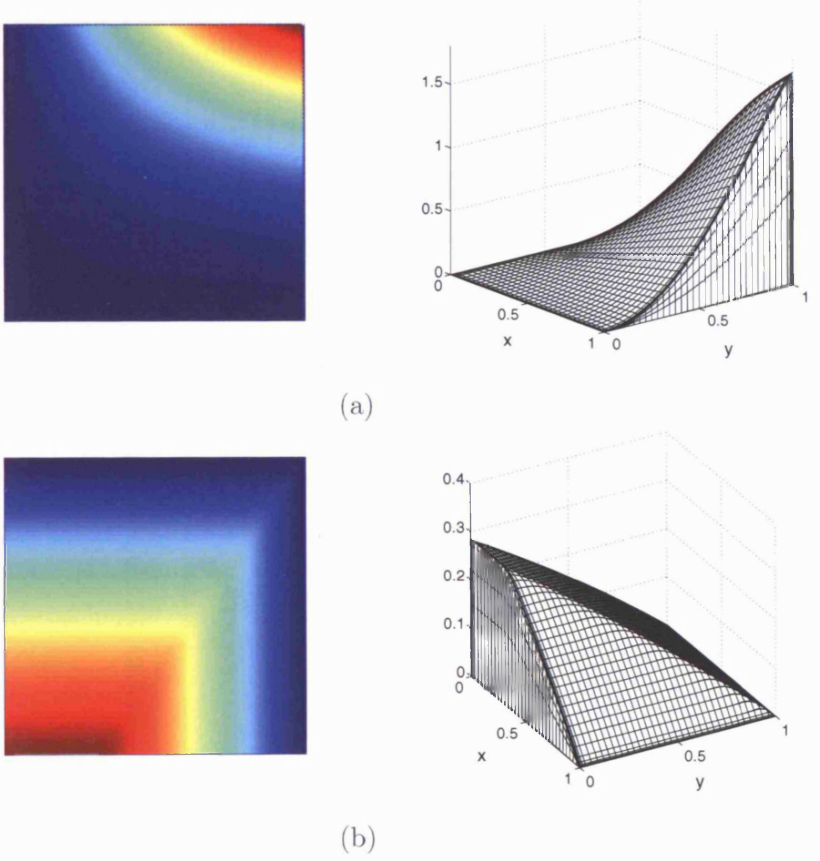
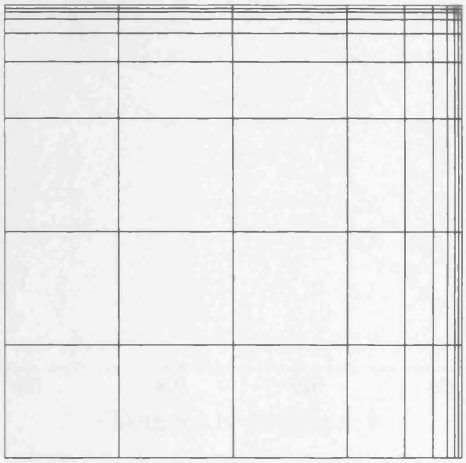
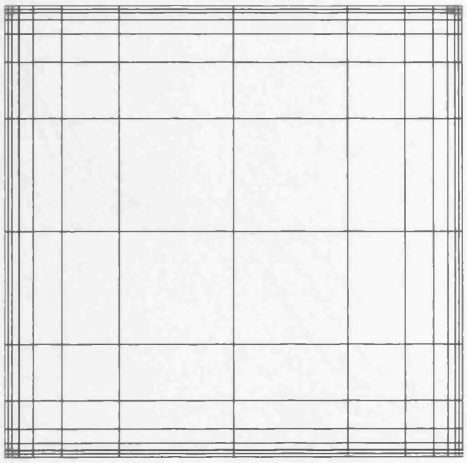


Figure 8.1: Example 1. (a) & (b) Analytical primal and dual solutions, respectively, for $\varepsilon = 10^{-3}$.



(a)



(b)

Figure 8.2: Example 1. (a) & (b) Anisotropically refined meshes of Type I and Type II, respectively, with $n = 5$.

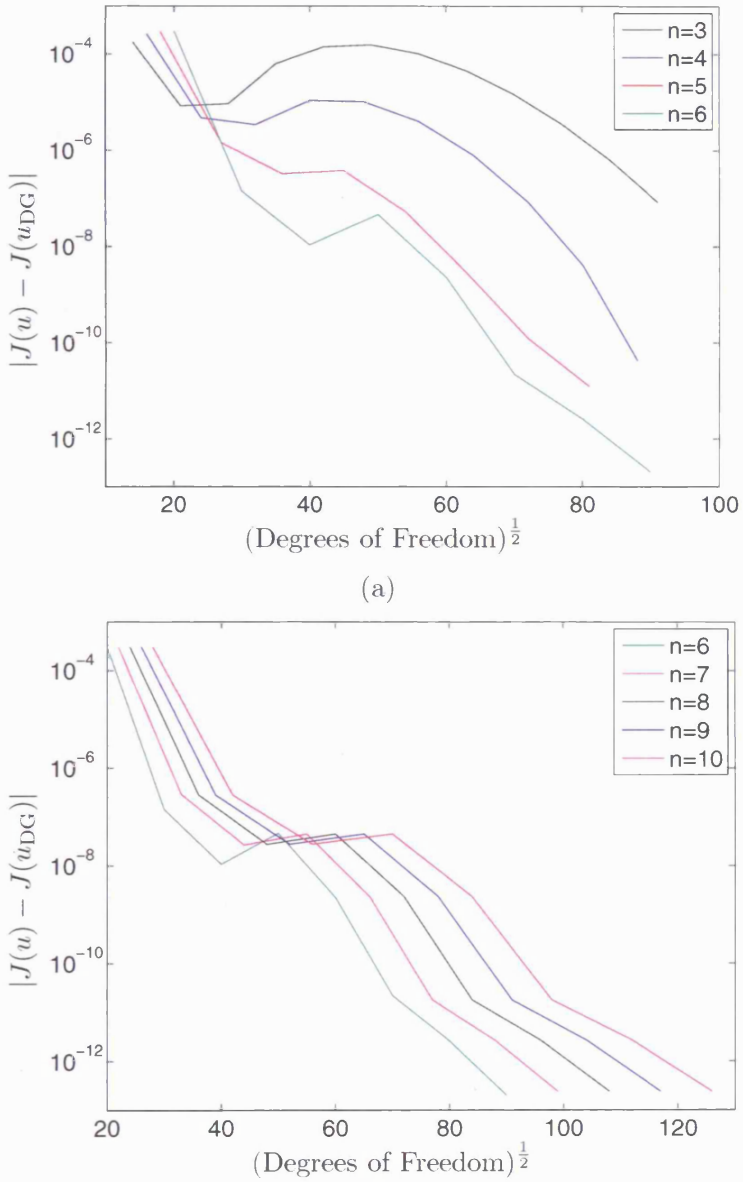
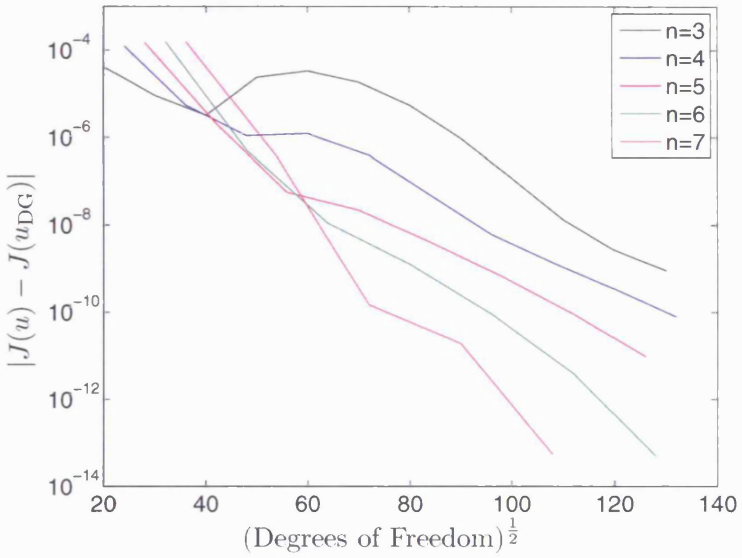
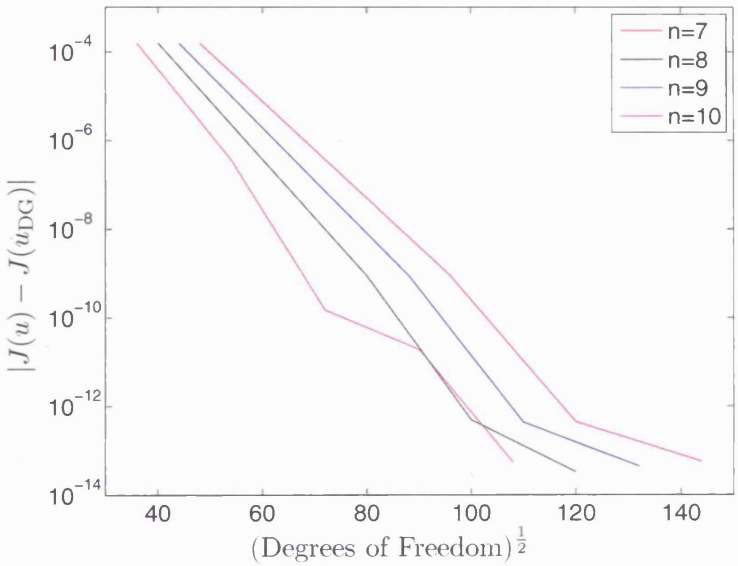


Figure 8.3: Example 1. Comparison of the error in the computed target functional with respect to the (square root of the) number of degrees of freedom employing meshes of Type I for: (a) $n = 3, 4, \ldots, 6$; (b) $n = 6, 7, \ldots, 10$.



(a)



(b)

Figure 8.4: Example 1. Comparison of the error in the computed target functional with respect to the (square root of the) number of degrees of freedom employing meshes of Type II for: (a) $n = 3, 4, \ldots, 7$; (b) $n = 7, 8, \ldots, 10$.

fixed number of degrees of freedom, starts to deteriorate, which indicates that there is an optimal balance between resolving the boundary layers using h– and p–refinement. On the basis of the meshes employed here, an optimal value of n for mesh Type I is 6, while $n = 7$ gives rise to the smallest error in $J(\cdot)$, for a fixed number of degrees of freedom, when mesh Type II is employed. This indicates that an optimal mesh adaptation algorithm will first anisotropically refine the mesh within unresolved boundary layer regions in the computational mesh, before automatically deciding to increase the local polynomial degree distribution.

Finally, in Figure 8.5 we compare the optimal refinement strategy for mesh Type I (with $n = 6$) to the corresponding optimal approach for mesh Type II (with $n = 7$). Here, we observe that the first approach where only the boundary layers present in the analytical solution u are resolved using anisotropic h–refinement leads to the smallest error in the computed target functional $J(\cdot)$, for a fixed number of degrees of freedom, in comparison to the latter strategy where both the boundary layers present in u and the dual solution z have been refined.

8.2 Example 2

This numerical example is designed to highlight the practical performance of the DGFEM on a sequence of *a priori* designed anisotropic hp–refined computational meshes. To this end, we consider the following singularly perturbed advection–diffusion problem equation $-\varepsilon\Delta u + u_x + u_y = f$, for $(x, y) \in (0, 1)^2$, where $0 < \varepsilon \ll 1$ and f is chosen so that

$$u(x, y) = x + y(1 - x) + [\mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-(1-x)(1-y)/\varepsilon}]\,[1 - \mathrm{e}^{-1/\varepsilon}]^{-1}. \tag{8.2.1}$$

cf. [76]. For $0 < \varepsilon \ll 1$ the solution (8.2.1) has boundary layers along $x = 1$ and $y = 1$. Here, we suppose that the aim of the computation is to calculate the value of the (weighted) mean-value of u over the computational domain Ω, i.e.,

$$J(u) = \int_{\Omega} u\psi \mathrm{d}\mathbf{x},$$

where the weight function ψ is chosen so that

$$z(x, y) = 4y(1 - y)(1 - \mathrm{e}^{-\alpha(1-x)} - (1 - \mathrm{e}^{-\alpha})(1 - x)):$$

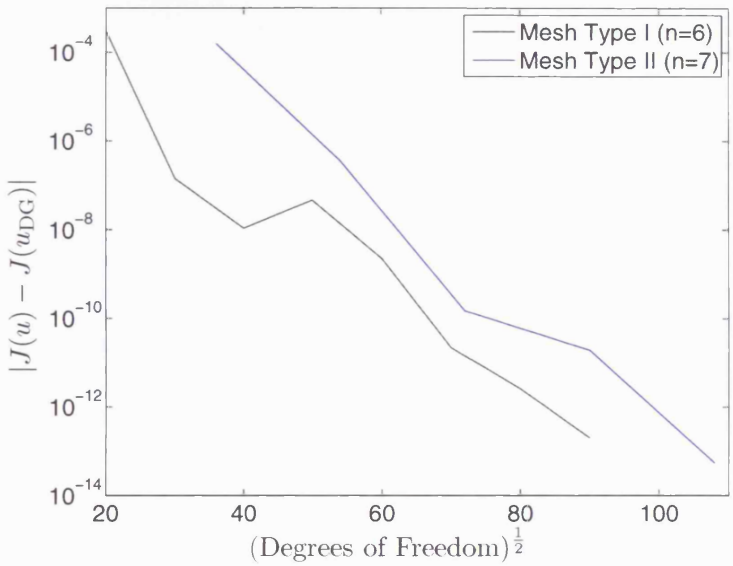
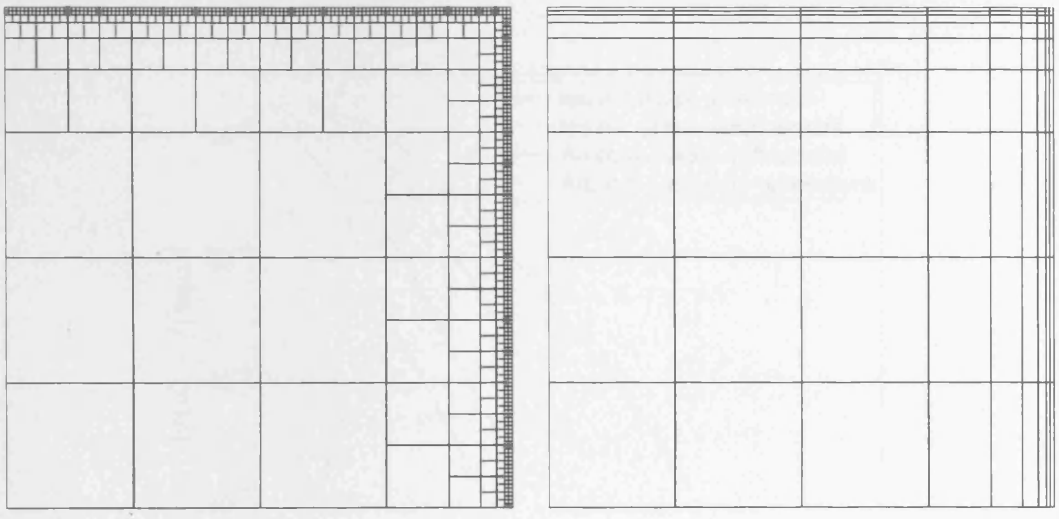


Figure 8.5: Example 1. Comparison of error in the computed target functional with respect to the (square root of the) number of degrees of freedom employing mesh Type I with $n = 6$ and mesh Type II with $n = 7$.

setting $\alpha = 100$ gives rise to a strong boundary layer along the boundary $x = 1$, $0 \leq y \leq 1$, cf. [48].

Here, we consider a sequence of *hp*-finite element spaces employing a combination of isotropically/anisotropically refined computational meshes with isotropic/anisotropic polynomial degrees. More precisely, starting from a uniform 5×5 square mesh, we first perform n, $n \geq 0$, isotropic or anisotropic refinements of this initial mesh in order to capture the boundary layers present within the underlying primal solution u. Here, only elements which lie on the right-hand side or top boundaries of Ω are refined; Figure 8.6 shows the two types of meshes generated by this algorithm with $n = 5$. Once the mesh has been refined, this is then kept fixed, and the polynomial degrees are either uniformly (isotropically) increased, or anisotropically refined using the following strategy: at each step of the adaptive algorithm, the polynomial degrees in the y-direction are increased by 1, while those in the x-direction are increased by 2. This latter strategy is motivated by the fact that the dual solution z only has anisotropy in the x-direction. We remark

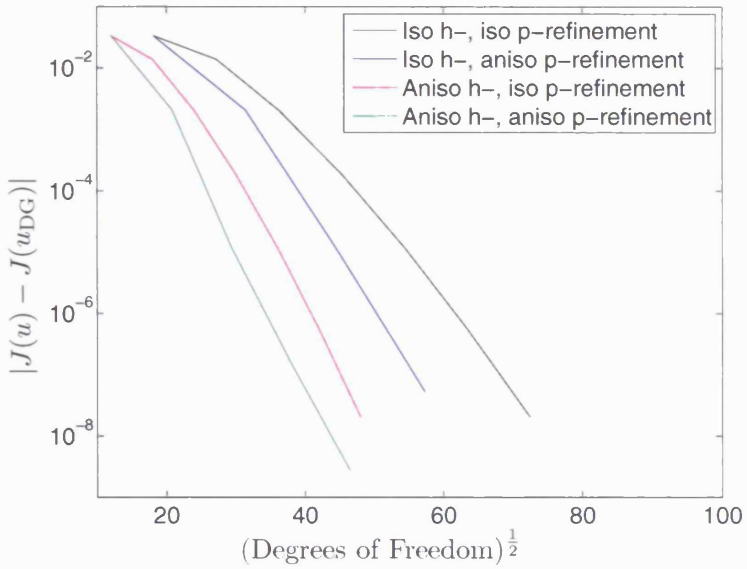


(a) (b)

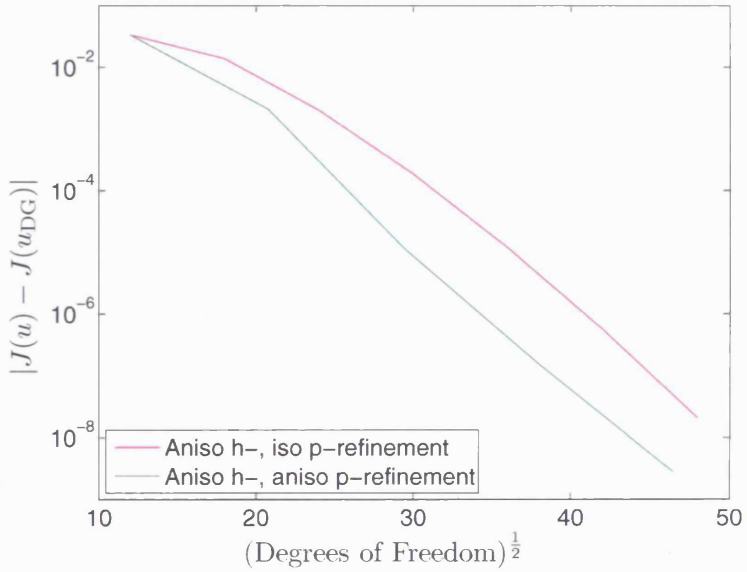
Figure 8.6: Example 2. (a) Isotropically refined mesh employed for $\varepsilon = 10^{-3}$, with 745 elements; (b) Anisotropically refined mesh employed for $\varepsilon = 10^{-3}$, with 81 elements.

that these *hp*–meshes are designed purely on the basis of *a priori* considerations, are not expected to be optimal, but are constructed merely to demonstrate the potential benefits of employing anisotropic *hp*–mesh refinement. Indeed, given the structure of the dual solution z, one may well expect that the computational mesh may be less refined in the region containing the boundary layer along $y = 1$ present in the primal solution u.

In Figures 8.7, 8.8, & 8.9 we plot the (square root) of the degrees of freedom employed in the finite element space $S^{\vec{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$ against the error in the computed target functional $J(\cdot)$, for $\varepsilon = 10^{-2}, 10^{-3}, 10^{-4}$, respectively, using each of the four refined *hp*–mesh distributions defined above, namely: isotropic h and isotropic p, isotropic h and anisotropic p, anisotropic h and isotropic p, anisotropic h and anisotropic p. Here, we have selected $n = 2, 5, 8$ for $\varepsilon = 10^{-2}, 10^{-3}, 10^{-4}$, respectively. Firstly, we note that in all cases, the convergence lines are (on average) straight, indicating exponential rates of convergence have been achieved using all four refinement strategies for each ε, which is in agreement with Corollary 7.1.3. Secondly, for each ε we observe that the computed error, for a given number of degrees of freedom, employing the isotropic h and isotropic

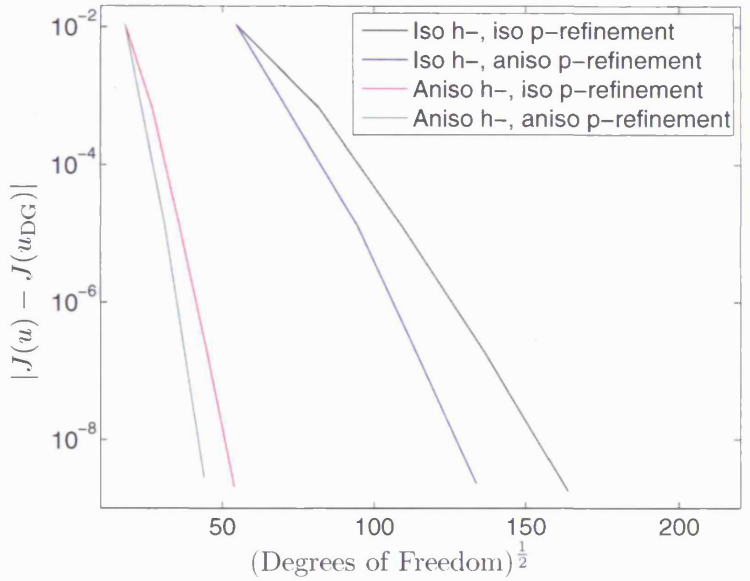


(a)

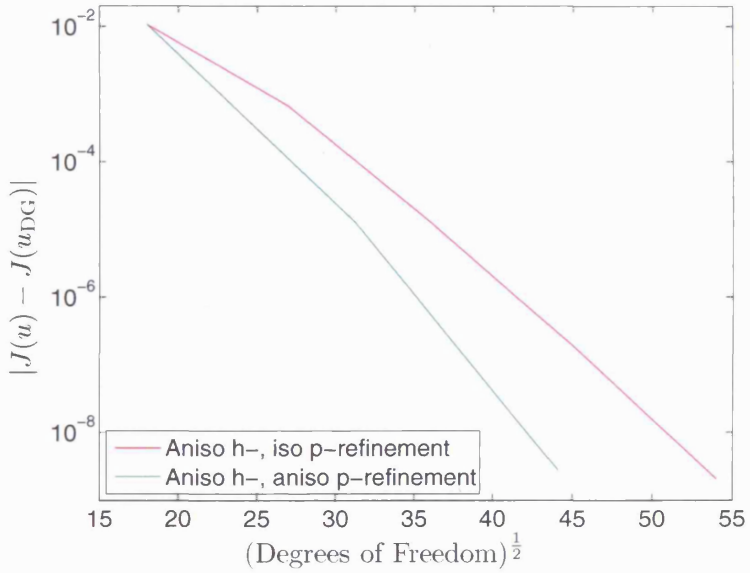


(b)

Figure 8.7: Example 2. (a) Comparison between adaptive isotropic and anisotropic hp-mesh refinement algorithms for $\varepsilon = 10^{-2}$; (b) Zoom of (a) comparing only the adaptive algorithms based on employing anisotropic h–refinement.

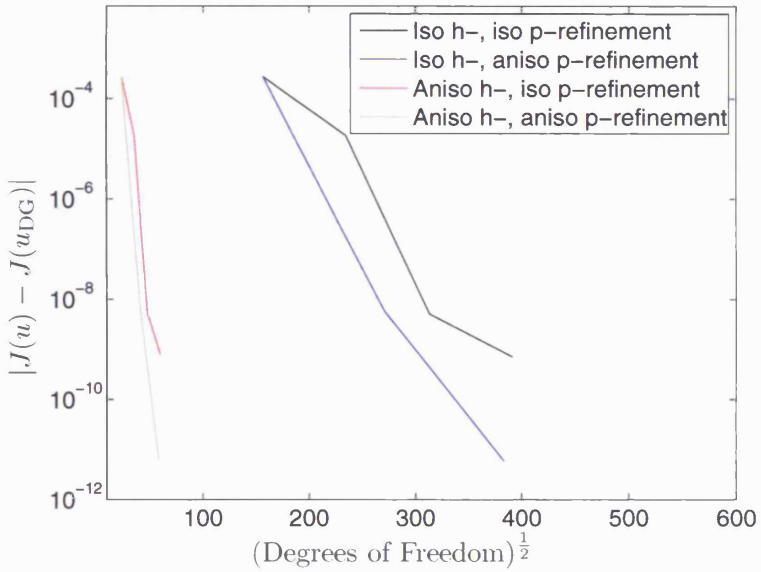


(a)

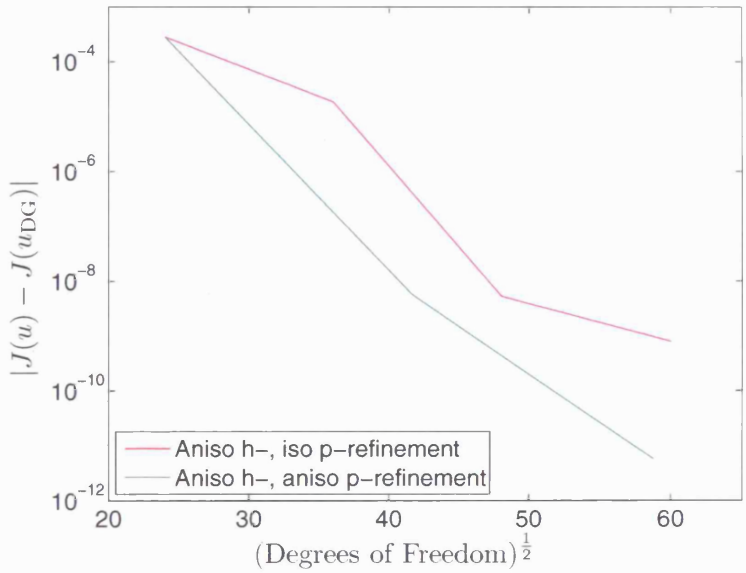


(b)

Figure 8.8: Example 2. (a) Comparison between adaptive isotropic and anisotropic hp-mesh refinement algorithms for $\varepsilon = 10^{-3}$; (b) Zoom of (a) comparing only the adaptive algorithms based on employing anisotropic h–refinement.



(a)



(b)

Figure 8.9: Example 2. (a) Comparison between adaptive isotropic and anisotropic hp-mesh refinement algorithms for $\varepsilon = 10^{-4}$; (b) Zoom of (a) comparing only the adaptive algorithms based on employing anisotropic h–refinement.

p strategy is always inferior to the algorithm employing isotropic h and anisotropic p. Similarly, this latter strategy is inferior to exploiting anisotropic h and isotropic p, which is in turn inferior to the use of anisotropic h and anisotropic p-refinement. Indeed, here we observe that the use of an anisotropically refined starting mesh yields a vast improvement in the computed error, for a given number of degrees of freedom, in comparison to standard isotropic refinement, since the mesh resolution needed to adequately capture the boundary layers can be achieved with significantly less elements when the former strategy is employed. This behaviour becomes increasingly more evident the smaller ε is chosen. For a given mesh, we see that even employing a simple-minded anisotropic polynomial distribution still yields significant improvements in comparison to isotropic refinement of p; indeed, here we observe that the former strategy leads to between one and two orders of magnitude improvement in the computed error in $J(\cdot)$, for a given number of degrees of freedom, compared with the latter approach.

8.3 Example 3

For this example we consider the same problem as in Example 1 of Section 6.1, where the primal solution has boundary layers of order $\mathcal{O}(\varepsilon)$ along $x = 1$ and $y = 1$ and the dual solution also has a boundary layer along $x = 1$, as well as one along $y = 0$; see Figure 6.1. As such, it is a good problem to try our Cartesian based hp–algorithm on. As `Algorithm 2` proved itself to be the most effective algorithm to use for h-refinement in Examples 1 and 2 of Chapter 6, we limit ourselves to only using this when h-refinement is needed. To illustrate the robustness of the algorithm, in this case we consider both $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$. In both cases we begin with a uniform mesh with 17 points in each coordinate direction and assign a uniform polynomial degree vector $\vec{p} = [2, 2]$ on each element. We compare our hp-adaptive strategy using both `Algorithms 3` and `4` for p-refinement against standard isotropic hp-refinement and anisotropic h-refinement with isotropic p-refinement. Figures 8.10(a) and (b) plot the logarithm of error in the target functional, $|J(u) - J(u_{\mathrm{DG}})|$, against the square root of the number of degrees of freedom, for the cases $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$, respectively.

Firstly, we notice that for $\varepsilon = 10^{-2}$ in every case we have obtained exponential convergence, as predicted, and secondly our anisotropic h-refinement strategy is once again far superior to the standard isotropic refinement. Further, we observe that both anisotropic p–algorithms are always better than the isotropic p– cases. Indeed, on the final grids, for the same number of degrees of freedom, the anisotropic hp–strategies yield over 2 orders of magnitude improvement than the h–anisotropic p–isotropic case and nearly 2 orders of magnitude improvement over the hp–isotropic method. Alternatively, for the same functional error we see that the new hp-anisotropic strategy yields around a 14% reduction in the number of degrees of freedom when compared with the h–anisotropic p–isotropic case and more than a 50% reduction when compared with the hp–isotropic case. Another point of interest is that the hp–anisotropic strategies show an immediate improvement over the other two strategies, implying that the method is useful even in the relatively low accuracy region. Figure 8.11 shows the resultant mesh when `Algorithm 4` is used after 4 refinement steps, with Figures 8.11(a) and (b) showing the polynomial degrees used in the x- and y-directions respectively. We notice that anisotropic h-refinement has been employed in order to resolve the right hand boundary layer and anisotropic p-refinement has been utilized inside the domain. Indeed, the polynomial enrichment in the x–direction has only been used slightly more than in the y-direction, however, we witness a vast reduction in error compared with isotropic p-refinement.

For $\varepsilon = 10^{-3}$ we once again see a great improvement over the isotropic hp-refinement for every anisotropic strategy used. The hp–anisotropic strategies do not show the immediate gain over the h–anisotropic/p–isotropic method witnessed for $\varepsilon = 10^{-2}$, nevertheless, on the final grids around two orders of magnitude improvement in error for the same number of degrees of freedom is seen. We remark that the three anisotropic strategies performing comparably on the first few meshes is most likely due to anisotropic h-refinement being required initially so that the boundary layer can be resolved and may not be the case if a finer initial mesh had been used to start with. Comparing the hp–anisotropic results with the standard hp–isotropic results we notice on the final grids a decrease of around seven orders of magnitude in functional error for the same numbers of degrees of freedom.

Finally, we mention that both `Algorithms 3 and 4` compare favourably, indicating

that **Algorithm 4** may be preferable as it is more computationally efficient.

8.4 Example 4

In this final example we investigate the performance of the proposed hp–anisotropic refinement algorithms applied to the mixed hyperbolic–elliptic problem with discontinuous boundary data first introduced in Example 2 of Section 6.2. Although, here, we suppose that the aim of the computation is to calculate the value of the (weighted) outflow advective flux along $x = 2$, $0 \leq y \leq 1$, i.e., $J(u) = \int_0^1 (\mathbf{b} \cdot \mathbf{n}) u(2, y) \psi(y) \mathrm{d}y$, where the weight function, in a modification to Example 2 of Section 6.2, is

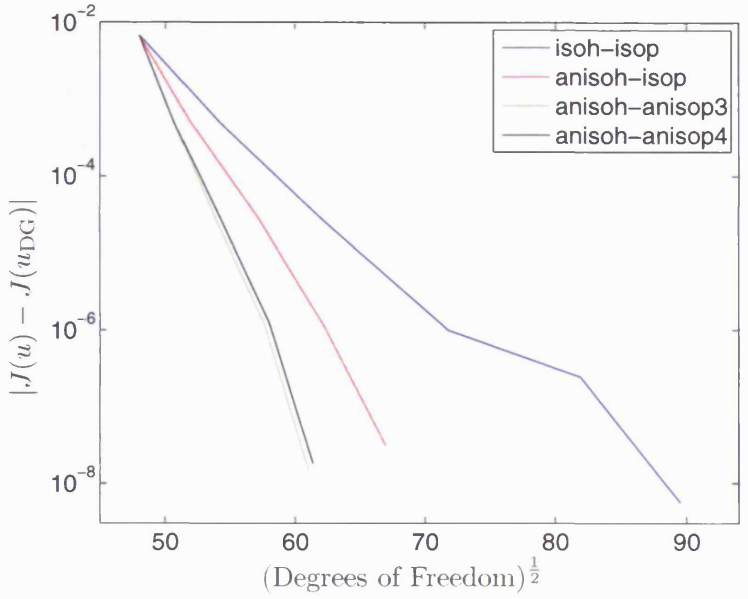
$$\psi(y) = \begin{cases} (\tanh(50(y - 7/40)) + 1)/2 & y < 17/40, \\ (\tanh(-50(y - 27/40)) + 1)/2 & y \geq 17/40. \end{cases}$$

The dual solution is shown in Figure 8.12, we notice that here the internal layers are much steeper than in Example 2 of Section 6.2. The true value of the functional is given by $J(u) = 0.324999805677598$.

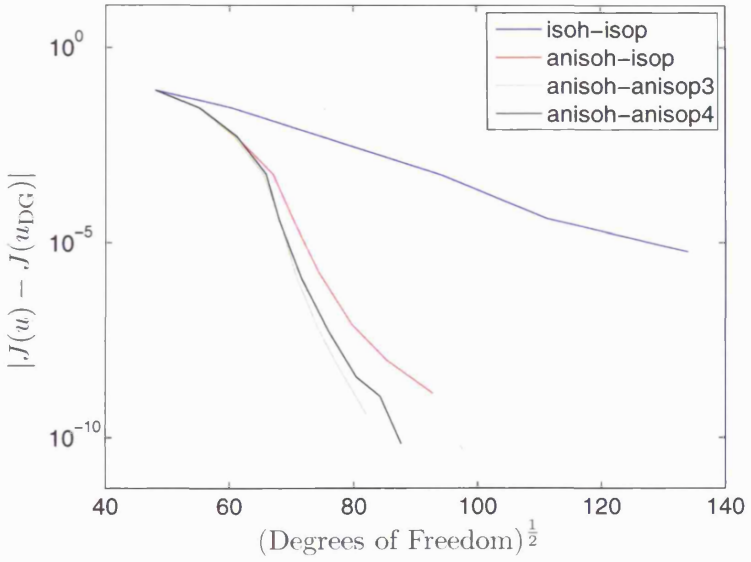
Given the qualitatively comparable results shown in the previous example, between exploiting the p–anisotropic algorithms **Algorithm 3** and **Algorithm 4**, in conjunction with the h–anisotropic algorithm **Algorithm 2**, in this section we shall only consider the latter approach, *i.e.*, the hp–anisotropic algorithm exploiting **Algorithm 2** and **Algorithm 4**, as it is more computationally efficient. Once again we compare this hp–anisotropic strategy with both hp–isotropic and h–anisotropic/p–isotropic refinement algorithms. In all cases the starting hp–mesh distribution is a 17×9 grid, consisting of uniform square elements, with the uniform polynomial degree distribution $\vec{p} = [2, 2]$ on each element.

Figure 8.13 shows a plot of the target functional errors, $|J(u) - J(u_{\mathrm{DG}})|$, against the (square root of the) number of degrees of freedom in the finite element space $S^{\tilde{\mathbf{p}}}(\Omega, \mathcal{T}_h, \mathbf{F})$, for each of the three hp–mesh refinement algorithms described above, while Figures 8.14(a) and (b) show the resultant grid and polynomial degrees in the x– and y–directions respectively, after 8 steps of the hp–anisotropic refinement strategy.

Concentrating on Figure 8.13 we first notice that, after an initial transient, once again, the convergence lines are (on average) straight, indicating exponential rates of convergence



(a)



(b)

Figure 8.10: Example 3: Comparison between adaptive *hp*–isotropic and anisotropic mesh refinement. (a) $\varepsilon = 10^{-2}$; (b) $\varepsilon = 10^{-3}$.

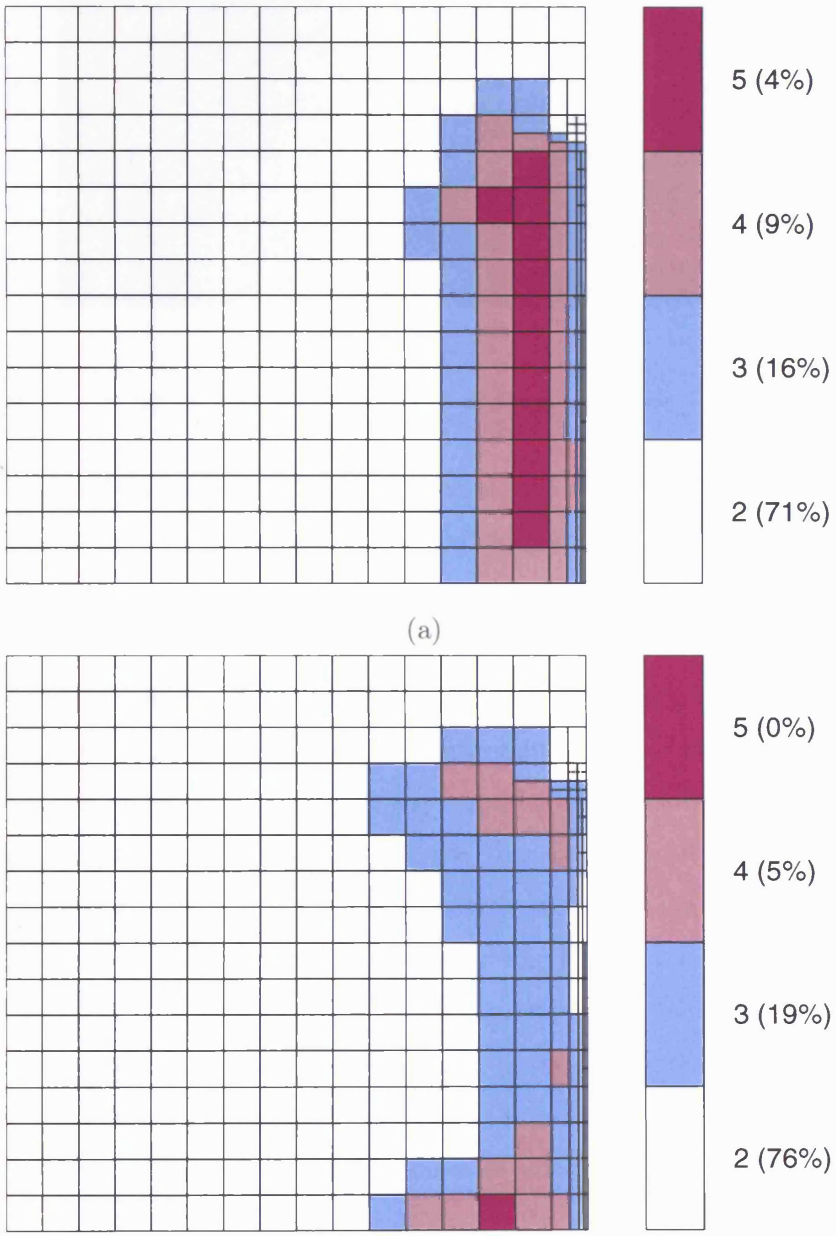


Figure 8.11: Example 3: Anisotropic *hp*-meshes after 4 refinement steps for `Algorithm 4`, with 316 elements and 3767 degrees of freedom, (a) p_x and (b) p_y, for $\epsilon = 10^{-2}$.

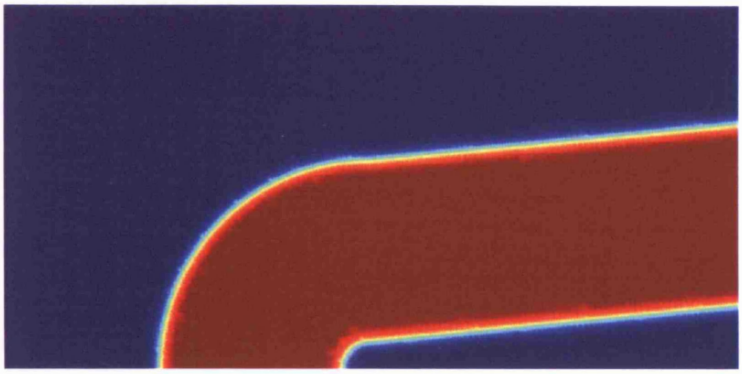


Figure 8.12: Example 4: Dual solution.

have been achieved for each of the three strategies. We also notice that at each stage the *hp*–isotropic strategy is performing the least efficiently, whilst the *hp*–anisotropic is always superior to both isotropic-p algorithms. Evidently the majority of improvement over the *hp*–isotropic strategy is due to the h–anisotropic algorithm, cf. the previous example when $\varepsilon = 10^{-3}$, yet in the asymptotic regime the *hp*–anisotropic strategy consistently shows around an order of magnitude improvement in the error for the same number of degrees of freedom, when compared with the h–anisotropic/p–isotropic strategy.

Examining Figures 8.14(a) and (b) we see that the majority of h–refinement has taken place primarily along the layer of the analytic solution u emanating from the point $(x, y) = (3/4, 0)$. In other regions p–enrichment has been favoured; indeed, there is a marked difference between the polynomial degrees used in the x– and y–directions, with the majority of elements having had no p–enrichment in the x–direction, while most elements have had some p–enrichment in the y–direction. The p–enrichment in the x–direction has been concentrated in the left half of the domain as this is where layers in the primal and dual solutions run parallel to the y–axis, while for the same reason p–enrichment in the y–direction is concentrated in the right portion of the domain.

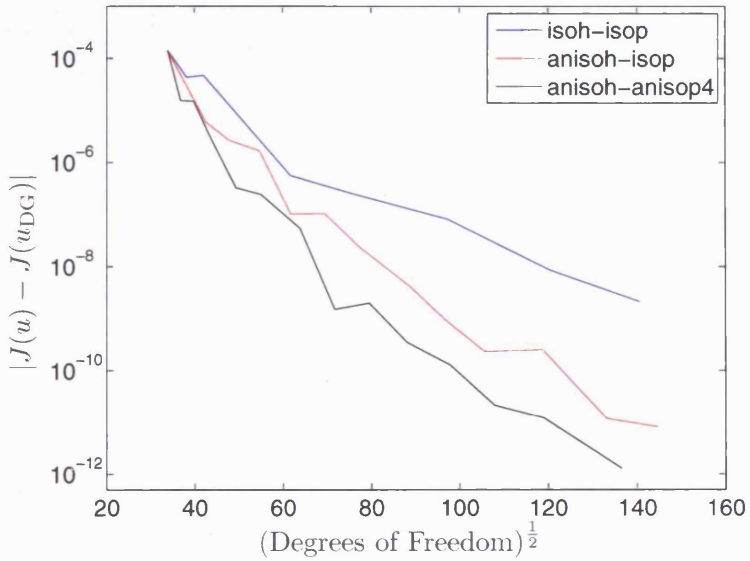


Figure 8.13: Example 4: Comparison between adaptive *hp*–isotropic and anisotropic refinement strategies.

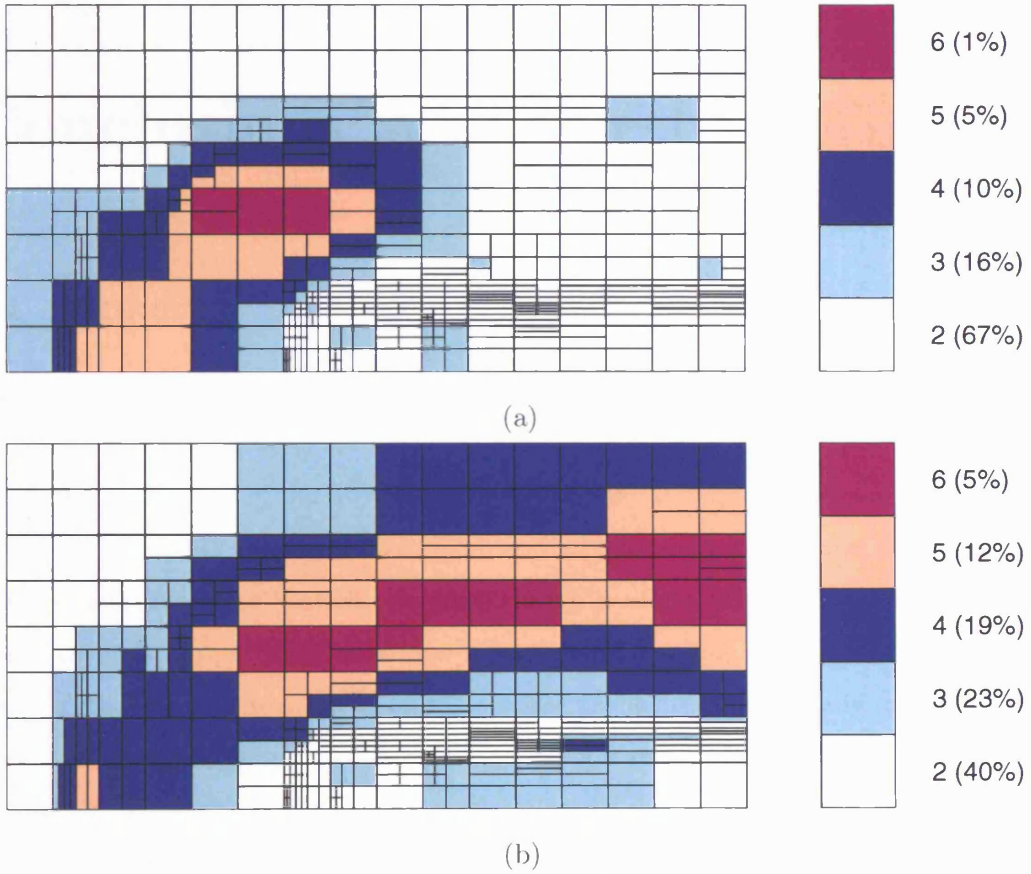


Figure 8.14: Example 4: Anisotropic hp–meshes after 8 refinement steps, with 410 elements and 6338 degrees of freedom, (a) p_x and (b) p_y.

Chapter 9

Conclusions And Further Work

9.1 Summary

This thesis is concerned with the development of adaptive anisotropic refinement strategies for the solution of second-order partial differential equations with nonnegative characteristic form, specifically in the case when control of a linear target functional of the solution is required. For this purpose a discontinuous Galerkin method was used to discretize the PDE, primarily for the flexibility in mesh design it offers. The PDE which was studied was a relatively simple linear advection-diffusion-reaction one, but which was able to offer many of the features present in more complex problems. In this way, the focus could remain on the development of automatic anisotropic adaptivity algorithms.

Chapter 2 presented the definition of the model problem and the formulation of the Symmetric Interior Penalty (SIP) DG method used to discretize it. This included the setting up of appropriate anisotropic meshes and anisotropic function spaces. A discussion then followed concerning the stability of the method.

In order to analyze the DG method presented, L^2-interpolation results were needed. Following the work of Formaggia *et* al. [48], L^2-interpolation results on an element and its faces were presented, with new generalizations to the case when higher order isotropic polynomial degrees are utilized for the approximation. These results followed by considering higher-order tensor manipulations as presented in De Lathauwer, Moor and Vandewalle [94]. L^2-interpolation results were also stated in the case of anisotropic axiparallel ele-

ments with anisotropic polynomial degrees, as first described by Georgoulis [52]. These anisotropic interpolation results were then used to extend the *a priori* error analysis for target functionals of the solution, previously considered in the work of Harriman *et* al. [61], where only isotropic elements with isotropic polynomial degrees were used. It was also noted that the new bounds collapsed back to the previous bounds upon the use of shape regular meshes with isotropic polynomial degrees.

A duality based *a posteriori* error indicator for use with target functionals was then presented and standard strategies for isotropic mesh refinement were discussed, plus some current techniques for anisotropic mesh refinement, the most popular of which being the use of the Hessian matrix of the solution to drive the adaptive process. Our error analysis implied that in the case when high order polynomial degrees were being used the Hessian would no longer provide optimal anisotropic information and higher order derivatives might need to be considered. However, numerical experiments indicated that these higher order tensors did not provide sharp enough information to control anisotropic refinement and hence we developed a new competitive method, based on the solution of local problems, by selecting the refinement which gave the greatest reduction in error per degree of freedom. Some numerical experiments were then carried out, comparing the new algorithms with a Hessian based approach and standard isotropic refinement. These experiments showed an improvement over both the isotropic and Hessian strategies for a number of different polynomials degrees; the experiments also showed the limitations of the Hessian strategy when higher order polynomial degrees are employed.

The *a priori* error analysis on axiparallel meshes with anisotropic polynomials hinted that great savings would be possible compared with isotropic polynomial degrees. Some numerical experiments were performed to confirm this, simply by designing finite element spaces using *a priori* information about the solution. In light of this, a fully anisotropic *hp*-adaptive algorithm was then proposed, again based on competitive refinements. Numerical experiments were then carried out comparing the *hp*-anisotropic algorithm with a standard isotropic *hp*-strategy and an *h*-anisotropic/*p*-isotropic method. The results were very impressive, showing that orders of magnitude improvement in the error for the same number of degrees of freedom are possible.

In conclusion, we have seen that both anisotropic h- and anisotropic p-refinement offer distinct advantages over their isotropic counterparts, when performed correctly. Indeed, automatic adaptive anisotropic hp-algorithms are possible and should be utilized in the future if the growing demands of science are to be met.

9.2 Future Work

9.2.1 Limitations of Using Local Problems

The proposed means of determining in which direction to refine all rely on the solution of local problems, where appropriate boundary conditions are obtained from the global solution. Hence, the anisotropy of the primal and dual solutions can only be reliably extracted when there is sufficient mesh resolution. For example, a mesh may be sufficiently fine for the solution of only one of the primal or dual solutions and the resultant refinements be in an incorrect direction. Thus, further work needs to be undertaken to ensure a robust selection of the refinement directions. Of course, this resolution problem will be common to any approach which attempts to extract anisotropy from the computed solutions.

9.2.2 Mesh Alignment

In this work only a relatively simple cartesian mesh refinement strategy has been considered, which, although has proved itself very useful for the problems considered herein, is unlikely to be very efficient in those cases where anisotropies occur in non mesh-aligned directions. Initially, it was thought that offering more flexibility in the subdivision of elements may be the way forward, as such refinement by cutting through the centroid of the reference element at an angle $\theta_{\hat{\kappa}}$, as depicted in Figure 9.1, has been considered.

The Hessian strategy considered in Section 5.3 provides the simplest way of determining the angle at which to refine (a global angle is computed, which must be transformed back to the reference element). Initial trials, using the exact Hessian, provided convincing meshes. For example, consider the following simple advection problem

$$\nabla \cdot (\mathbf{b}u) = 0,$$

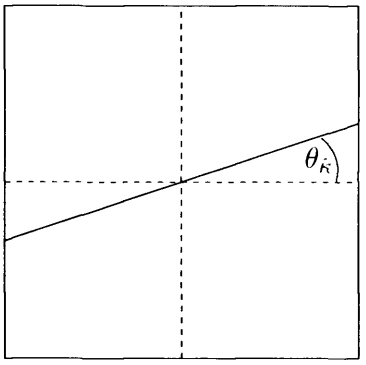


Figure 9.1: Anisotropic h-refinement at an angle $\theta_{\hat{\kappa}}$.

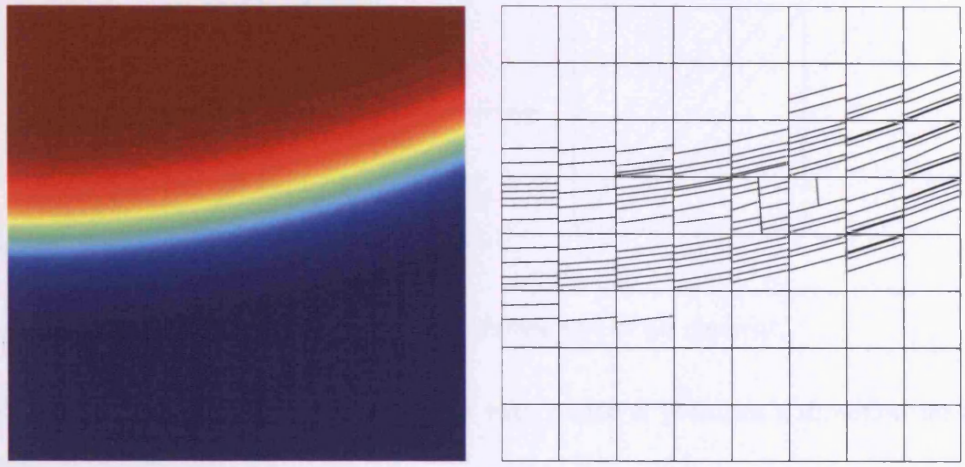
on the domain $\Omega = [0,1]^T$, where $\mathbf{b} = (1, 0.4x)^T$ and the exact solution is chosen to be

$$u(x,y) = \tanh(10(y - 0.2x^2 - 0.5)).$$

Hence, there is a layer in the solution which follows the quadratic curve $y = 0.2x^2 + 0.5$. The exact solution is shown in Figure 9.2(a) and the grid produced after a number of adaptive steps is shown in Figure 9.2(b). We note the presence of a number of incorrectly refined elements, and remark that this has occurred due to the Hessian becoming identically zero at the centroid of the parent elements. The rates of convergence of L^2-error were also seen to be better than in the case when standard isotropic h-refinement was employed.

Unfortunately, testing on more complex problems and using approximated Hessians has not proved so easy. As such, a number of issues have arisen which could be looked at in the future. Figure 9.2(b) alludes to one of these problems: suppose an element and its children are repeatedly chosen for refinement in the same direction, then if the angle reduces the first quadrilateral element to nearly triangular elements, the successive refinements can lead to closely bunched narrow elements next to much larger elements. In such cases it is highly likely that the anisotropy in the solution could be missed entirely. Figure 9.3 gives an example off this phenomenon. Performing additional refinement as in Figure 9.4 could be the answer to this, but unfortunately this increases the number of elements after a refinement and also introduces triangular elements, which if the goal is to use anisotropic p-enrichment causes problems.

Another issue with this type of refinement is that it is liable to produce a highly mesh



(a) (b)

Figure 9.2: (a) Exact Solution (b) Mesh refined by Hessian strategy.

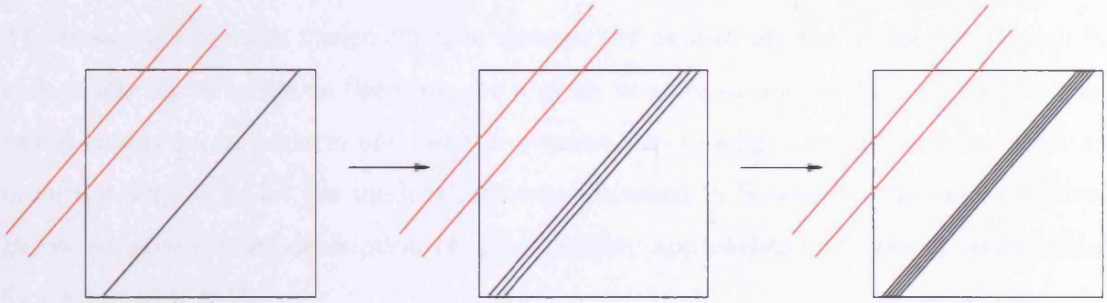


Figure 9.3: Failure to resolve an anisotropy (red) by refinement.

dependent solution if the refinements are chosen incorrectly to begin with, something which is likely on coarse initial grids. Thus, further refinements also choose the wrong angles and a correctly aligned mesh appears out of the question. Other points which must be considered are that the proposed refinements can often lead to highly distorted elements, that is, the mapping Q_κ discussed in Section 2.4 is no longer close to the identity and cells possessing a very large internal angle can appear, in which case the DG solution will no longer represent the true solution well. Furthermore, controlling the number of hanging nodes on an element face with this type of refinement also proves problematic. It may well be that introducing some specialized mesh smoothing could solve the above problems,

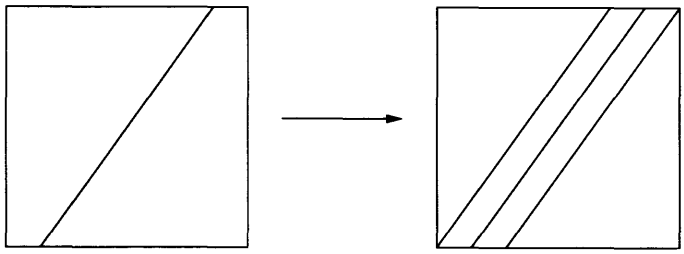


Figure 9.4: Additional refinement of an element.

but it is not yet clear how to proceed with this, hence it provides a direction for future research.

An alternative strategy to the mesh refinement proposed above could be mesh movement (r-refinement), which has proved popular and indeed many anisotropic strategies involve mesh movement in some way, for example, the work of Dompierre *et* al. [58, 5, 44]. The basic idea of mesh movement is to arrange (by movement) the nodes of the mesh in such a way as to optimize the error for a given number of nodes. In our case the goal would simply be to perform one mesh movement step to align the grid with the solution in such a way as to let the mesh refinements discussed in Section 5.4 be more effective. Below we give a brief description of some possible approaches and present some initial forays into this field.

Ideally we would like the mesh movement to be driven by the *a posteriori* analysis developed in this thesis. As such a simple first step can be based around a modification of the technique presented in [17]. A loop over all the nodes in the mesh is performed and for a node n, with coordinates x_n, the node is moved to a weighted average of the positions of the centroids of the N neighbouring elements, by the formula

$$x_n^{\text{new}} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i},$$

where the x_i are the coordinates of the centroids of κ_i and w_i are the weights. The weights can simply be chosen as η_{κ_i}, which has the effect of pushing the node towards the areas of high error.

Alternatively, the method proposed by Schneider and Jimack [118] can be modified to achieve mesh movement. In [118] a strategy to relocate all of the nodes of a mesh

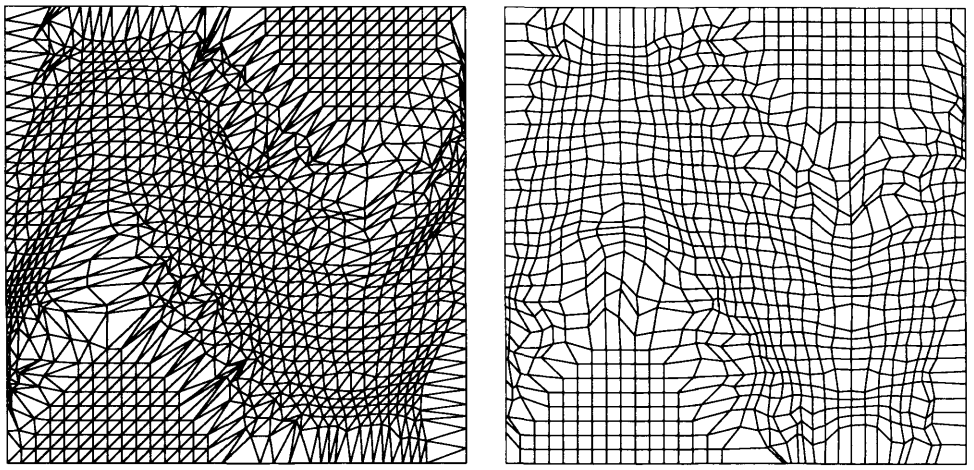
simultaneously by means of error optimization was described. This falls in very well with the optimization algorithms presented in this thesis and it can be modified to allow for independent node movement by local error optimizations.

Of course, there is the possibility of using the metric based techniques for mesh movement already used extensively in the literature. This process involves generating a mesh which is quasi-uniform in some metric, commonly based around the Hessian of the solution; see, for example, [58]. Rather than basing the metric around the Hessian of a solution or higher order derivatives it could be possible to use the *a posteriori* error estimates, for example the spatial derivatives of the error indicator might provide some useful anisotropic information. In any case, there is plenty of scope for future work in this direction.

We performed a number of experiments on quadrilateral and triangular grids to see how effective the three mesh movement strategies mentioned above are. For simplicity no PDE was actually solved, instead interpolation of the function

$$u = \tanh\left(-100\left(y - \frac{1}{2} - \frac{1}{4}\sin(2\pi x)\right)^2\right), \tag{9.2.1}$$

which had previously been considered in [84], was chosen for investigation. Figure 9.5(a) shows the resultant mesh for triangular elements and Figure 9.5(b) shows the mesh for quadrilateral elements, for a metric based movement strategy. For brevity, meshes for the two other strategies are omitted, as they exhibit many of the features present for the metric case. We first notice that, for both quadrilateral and triangular meshes, the general form of the solution has been recovered. However, in the quadrilateral case the nodes have been moved in such a way as to align the anisotropy in the solution along the diagonals of the elements. Hence, performing anisotropic refinement on these elements will not prove effective as the incorrect direction will still be picked out. Ideally we would wish a quadrilateral element to remain, roughly, as a parallelogram with the anisotropy along the direction of one pair of faces in order that an anisotropic refinement will maintain the implied direction of that element. Also, in each of the three cases considered above, highly distorted elements appear in the mesh and some correction to the elements is going to be required if any of the methods are to prove viable. A possible cure could be only allowing internal angles up to a prescribed limit, although this restriction may not allow sufficient



(a) (b)

Figure 9.5: Metric based mesh movement (a) triangular grid and (b) quadrilateral grid.

movement of the mesh to achieve alignment. Huang [83] proposed including a measure of the quality of the mesh into the metric tensor to achieve non-distorted meshes, this presents some interesting avenues for future work. Alternatively, applying mesh smoothing techniques to an already moved mesh could prove the way forward, indeed work in this direction has already been undertaken, by Lee and Lee [95], for example.

It may well be that the node movement algorithms proposed above are not sufficient for our needs and another method for achieving well aligned grids has to be sought. A possible contender is moving the mesh by way of harmonic maps, first suggested by Winslow [137]. Here the idea is to create a mapping from a uniform computational domain Ω_c to the physical domain Ω. By elliptic regularity of the solutions to harmonic equations, the resultant meshes are generally smooth and, therefore, may be suitable for our purposes. Indeed, the results of Tang [130] show very well aligned quadrilateral meshes, which seem ideal for anisotropic refinement. This moving mesh method has also been developed for DG schemes; see, for example, [98].

9.2.3 Other Directions for Future Research

- **Extension of anisotropic *hp*–error analysis.** The anisotropic *hp*–error analysis in this thesis was carried out only for axiparallel meshes. Further work could be

carried out to extend the results to the case when no restriction is placed on the mappings F_κ.

- **Extensions to energy norm control.** This thesis concentrated on just control of the error in some target functional. Instead, the approximation results of Chapter 3 could be applied to the case of energy norm error analysis. It would also be very interesting to see how applicable the competitive refinement anisotropic algorithms are in this case.

- **Extension to three-Dimensions.** Anisotropic mesh refinement is likely to prove even more useful when three-dimensional problems are considered. A completely analogous set of refinements are possible (at least for hexahedra) and so local problems again present an opportunity for determining the optimal refinement; it would be very interesting to see what gains could be made.

- **Extensions to More Complex Problems.** The problems considered within this thesis were only model problems with which to develop the anisotropic refinement strategies. It is now time to test these methods on more complicated problems, such as the Euler and Navier-Stokes equations for fluid flow with more complex geometries. An interesting avenue of research is also to apply the techniques to eigenvalue problems arising in the field of bifurcation analysis. Indeed, funding has been secured to study the transition to turbulence for fluid flows in a pipe with a sudden expansion.

Appendix A

Technical Results

A.1 Minimisation of Error Bounds

Here we directly minimise the right hand side of the *a priori* estimate (4.1.13) in order to determine possible orientations and scale factors for the elements of the mesh. We remark that this requires sharpness of the (4.1.13) estimate, cf. the discussion in Section 5.3.1. In the following analysis we assume a two–dimensional problem and consider two case separately, the first where bilinear elements are employed, the second where uniform higher polynomial degrees are used.

Case 2, $p = 1$

To make things a little simpler we consider a purely diffusive problem, in which case the estimate (4.1.13) reduces to:

$$|J(u) - J(u_{\mathrm{DG}})|^2 \leq C \left(\sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{d,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^{s_\kappa}(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right) \times \left(\sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{d,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^{t_\kappa}(\tilde{z}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right), \tag{A.1.1}$$

where $s_\kappa = 2$ and $t_\kappa = 2$.

To begin with we assume that we have a maximum number of elements to cover the computational domain and that we have control over the orientation and aspect ratio of

elements, but not the position of the centroid of each element nor the length of the longest edge of the element. With this assumption we cannot guarantee the computational domain will be completely covered, but it will suffice for us to make decisions on the 'optimal' orientation and aspect ratio of elements situated at certain points in the domain. The goal, therefore, is to minimise the error $|J(u) - J(u_{\mathrm{DG}})|$, in which case we are looking also to minimise

$$I := C \left(\sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{2,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^2(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right) \left(\sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{2,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^2(\tilde{z}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right).$$

We consider the easier problem of minimising with respect to one of the solution variables, u or z, that is minimising either

$$I_1 := \sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{2,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^2(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x}$$

or

$$I_2 := \sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{2,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^2(\tilde{z}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x},$$

as both procedures are completely analogous we now only consider I_1. Expanding $\int_{\tilde{\kappa}} D_{\tilde{\kappa}}^2(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x}$ using (3.3.3) we obtain

$$\begin{aligned} I_1 &= \sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{2,\kappa}^2} \int_{\tilde{\kappa}} \sum_{i_1=1}^{2} \sum_{i_2=1}^{2} (\sigma_{i_1,\kappa} \sigma_{i_2,\kappa})^2 (\tilde{\mathcal{D}}^2(\tilde{v}) \times_1 \mathbf{u}_{i_1,\kappa}^\top \times_2 \mathbf{u}_{i_2,\kappa}^\top)^2 \mathrm{d}\tilde{x} \\ &= \sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{2,\kappa}^2} \int_{\tilde{\kappa}} \sum_{i_1=1}^{2} \sum_{i_2=1}^{2} (\sigma_{i_1,\kappa} \sigma_{i_2,\kappa})^2 (\mathbf{u}_{i_1,\kappa}^\top H^{\tilde{v}} \mathbf{u}_{i_1,\kappa})^2 \mathrm{d}\tilde{x} \\ &= \sum_{\kappa \in \mathcal{T}_h} \alpha \left(\frac{\sigma_{1,\kappa}^4}{\sigma_{2,\kappa}^2} L_{1,1}^{\tilde{v},\tilde{\kappa}} + \sigma_{2,\kappa}^2 L_{2,2}^{\tilde{v},\tilde{\kappa}} + 2\sigma_{1,\kappa}^2 L_{1,2}^{\tilde{v},\tilde{\kappa}} \right), \end{aligned}$$

where

$$L_{i,j}^{\tilde{v},\tilde{\kappa}} := \int_{\tilde{\kappa}} (\mathbf{u}_{i_1,\kappa}^\top H^{\tilde{v}} \mathbf{u}_{i_1,\kappa})^2 \mathrm{d}\tilde{x}.$$

We introduce the aspect ratio variable of an element by $\varsigma_\kappa := \sigma_{1,\kappa}/\sigma_{2,\kappa}$, thus

$$I_1 = \sum_{\kappa \in \mathcal{T}_h} \alpha \left(\varsigma_{1,\tilde{\kappa}}^2 \left(s_\kappa^2 L_{1,1}^{\tilde{v},\tilde{\kappa}} + \frac{1}{\varsigma_\kappa^2} L_{2,2}^{\tilde{v},\tilde{\kappa}} + 2 L_{1,2}^{\tilde{v},\tilde{\kappa}} \right) \right). \tag{A.1.2}$$

For ease of exposition we make the assumption that $\tilde{H}^{\tilde{u}}(\tilde{\mathbf{x}})$ remains fairly constant over the element $\tilde{\kappa}$ with a mean value $\tilde{H}^{\tilde{u}}_{\tilde{\kappa}}$, in which case the following holds

$$\begin{aligned} L^{\tilde{v},\tilde{\kappa}}_{i,j} &= \int_{\tilde{\kappa}} (\mathbf{u}^T_{i,\tilde{\kappa}} \tilde{H}^{\tilde{v}} \mathbf{u}_{j,\tilde{\kappa}})^2 \mathrm{d}\tilde{x} \\ &\approx m_{\tilde{\kappa}} (\mathbf{u}^T_{i,\tilde{\kappa}} \tilde{H}^{\tilde{v}}_{\tilde{\kappa}} \mathbf{u}_{j,\tilde{\kappa}})^2 \quad \text{for} \quad i,j = 1,2, \end{aligned}$$

where $m_{\tilde{\kappa}}$ is the Lebesgue measure of element $\tilde{\kappa}$.

We now seek to introduce an orientation angle into term I_1. To this end, we perform an eigenvalue decomposition of $\tilde{H}^{\tilde{u}}_{\tilde{\kappa}}$ as follows

$$\tilde{H}^{\tilde{v}}_{\tilde{\kappa}} = [\mathbf{x}_{1,\tilde{\kappa}}, \mathbf{x}_{2,\tilde{\kappa}}] \begin{bmatrix} \mu_{1,\tilde{\kappa}} & 0 \\ 0 & \mu_{2,\tilde{\kappa}} \end{bmatrix} \begin{bmatrix} \mathbf{x}^T_{1,\tilde{\kappa}} \\ \mathbf{x}^T_{2,\tilde{\kappa}} \end{bmatrix},$$

where $\mu_{1,\tilde{\kappa}} > \mu_{2,\tilde{\kappa}}$ are the eigenvalues of $\tilde{H}^{\tilde{u}}_{\tilde{\kappa}}$ with respective right eigenvectors $\mathbf{x}_{1,\tilde{\kappa}}$ and $\mathbf{x}_{2,\tilde{\kappa}}$. Due to the symmetry of $\tilde{H}^{\tilde{u}}_{\tilde{\kappa}}$ we can be sure that $\mathbf{x}_{1,\tilde{\kappa}}$ and $\mathbf{x}_{2,\tilde{\kappa}}$ are orthogonal and we prescribe that they are both unit vectors. Without loss of generality we assume that $|\mu_{1,\tilde{\kappa}}| \geq |\mu_{2,\tilde{\kappa}}|$. In this situation we can rewrite $\mathbf{u}_{1,\tilde{\kappa}}$ and $\mathbf{u}_{2,\tilde{\kappa}}$ as

$$\begin{aligned} \mathbf{u}_{1,\tilde{\kappa}} &= \cos\phi_\kappa \mathbf{x}_{1,\tilde{\kappa}} + \sin\phi_\kappa \mathbf{x}_{2,\tilde{\kappa}}, \\ \mathbf{u}_{2,\tilde{\kappa}} &= -\sin\phi_\kappa \mathbf{x}_{1,\tilde{\kappa}} + \cos\phi_\kappa \mathbf{x}_{2,\tilde{\kappa}}, \end{aligned}$$

where ϕ_κ represents the angle between the primary singular vector $\mathbf{u}_{1,\tilde{\kappa}}$ and primary eigenvector $\mathbf{x}_{1,\tilde{\kappa}}$. Substituting these expressions into (A.1.2) gives

$$\begin{aligned} L^{\tilde{v},\tilde{\kappa}}_{1,1} &\approx m_{\tilde{\kappa}} (\mu_{1,\tilde{\kappa}} \cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}} \sin^2\phi_\kappa)^2, \\ L^{\tilde{v},\tilde{\kappa}}_{1,1} &\approx m_{\tilde{\kappa}} (\mu_{1,\tilde{\kappa}} \sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}} \cos^2\phi_\kappa)^2, \\ L^{\tilde{v},\tilde{\kappa}}_{1,1} &\approx m_{\tilde{\kappa}} (\mu_{1,\tilde{\kappa}} \cos\phi_\kappa \sin\phi_\kappa + \mu_{2,\tilde{\kappa}} \cos\phi_\kappa \sin\phi_\kappa)^2 \\ &= m_{\tilde{\kappa}} \frac{\sin^2(2\phi_\kappa)}{4} (\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2. \end{aligned}$$

Hence,

$$\begin{aligned} I_1 \approx \sigma^2_{1,\tilde{\kappa}} m_{\tilde{\kappa}} \Bigg(& \varsigma^2_\kappa (\mu_{1,\tilde{\kappa}} \cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}} \sin^2\phi_\kappa)^2 + \frac{1}{\varsigma^2_\kappa} (\mu_{1,\tilde{\kappa}} \sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}} \cos^2\phi_\kappa)^2 \\ & + \frac{\sin^2(2\phi_\kappa)}{2} (\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2 \Bigg). \end{aligned}$$

By prescribing a fixed value for each $\sigma_{1,\tilde{\kappa}}$ we notice that full control over the shape and orientation of the element can still be achieved, although the element measure $m_{\tilde{\kappa}}$, will not remain fixed. To compensate for this, rather than seeking to minimize I_1 we minimize $\bar{I}_1 := I_1/(m_{\tilde{\kappa}}\sigma_{1,\kappa})$, indeed this makes sense because we are then effectively finding the minimum error for a given element size. Hence, we attempt to minimize $\bar{I}_1$ with respect to σ_κ and ϕ_κ simultaneously with conditions $\sigma_\kappa \geq 1$ and $\phi_\kappa \in [0, \pi)$. Thereby,

$$\begin{aligned}\bar{I}_1 &= \varsigma_\kappa^2(\mu_{1,\tilde{\kappa}}\cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\sin^2\phi_\kappa)^2 + \frac{1}{\varsigma_\kappa^2}(\mu_{1,\tilde{\kappa}}\sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\cos^2\phi_\kappa)^2 \\ &\quad + \frac{\sin^2(2\phi_\kappa)}{2}(\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2.\end{aligned}$$

Differentiating with respect to both ς_κ and ϕ_κ and setting both results equal to 0 yields the following simultaneous equations

$$0 = \varsigma_\kappa(\mu_{1,\tilde{\kappa}}\cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\sin^2\phi_\kappa)^2 - \frac{1}{\varsigma_\kappa^3}(\mu_{1,\tilde{\kappa}}\sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\cos^2\phi_\kappa)^2, \quad \text{(A.1.3)}$$

$$\begin{aligned}0 &= 2\varsigma_\kappa^2(\mu_{1,\tilde{\kappa}}\cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\sin^2\phi_\kappa)(\sin(2\phi_\kappa)(\mu_{2,\tilde{\kappa}} - \mu_{1,\tilde{\kappa}})) \\ &\quad + \frac{2}{\varsigma_\kappa^2}(\mu_{1,\tilde{\kappa}}\sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\cos^2\phi_\kappa)(\sin(2\phi_\kappa)(\mu_{1,\tilde{\kappa}} - \mu_{2,\tilde{\kappa}})) \\ &\quad + 4(\sin\phi_\kappa\cos\phi_\kappa(\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}}))(\cos(2\phi_\kappa)(\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})) \\ &= 2\sin(2\phi_\kappa)\Big[(\mu_{1,\tilde{\kappa}} - \mu_{2,\tilde{\kappa}})\Big(\frac{\mu_{1,\tilde{\kappa}}\sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\cos^2\phi_\kappa}{\varsigma_\kappa^2} \\ &\quad - \varsigma_\kappa^2(\mu_{1,\tilde{\kappa}}\cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\sin^2\phi_\kappa)\Big) + 2(\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2\cos(2\phi_\kappa)\Big]. \quad \text{(A.1.4)}\end{aligned}$$

We also differentiate with respect to ς_κ once more to obtain

$$\frac{\partial^2 I}{\partial\varsigma_\kappa^2} = (\mu_{1,\tilde{\kappa}}\cos^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\sin^2\phi_\kappa)^2 + \frac{3}{\varsigma_\kappa^4}(\mu_{1,\tilde{\kappa}}\sin^2\phi_\kappa + \mu_{2,\tilde{\kappa}}\cos^2\phi_\kappa)^2 \geq 0. \quad \text{(A.1.5)}$$

Hence, for a fixed ϕ_κ, if ς_κ satisfies (A.1.3) then ς_κ is a minimum. We consider a number of cases separately:

1. $\mu_{1,\tilde{\kappa}} = \mu_{2,\tilde{\kappa}} = 0$

   We see that both of the simultaneous equations are immediately satisfied, so any value of ς_κ and ϕ_κ will be applicable to provide the minimum $\bar{I}_1 = 0$.

2. $\mu_{1,\tilde{\kappa}} > \mu_{2,\tilde{\kappa}} > 0$ or $\mu_{1,\tilde{\kappa}} < \mu_{2,\tilde{\kappa}} < 0$

Rearranging the equation, (A.1.3), gives:

$$s_\kappa^2 = \frac{\mu_{1,\tilde{\kappa}} \sin^2 \phi_\kappa + \mu_{2,\tilde{\kappa}} \cos^2 \phi_\kappa}{\mu_{1,\tilde{\kappa}} \cos^2 \phi_\kappa + \mu_{2,\tilde{\kappa}} \sin^2 \phi_\kappa},$$

and substituting this into (A.1.4) yields

$$\begin{aligned} 0 &= 2\sin(2\phi_\kappa)\left[(\mu_{1,\tilde{\kappa}} - \mu_{2,\tilde{\kappa}})\left(\mu_{1,\tilde{\kappa}} \cos^2 \phi_\kappa + \mu_{2,\tilde{\kappa}} \sin^2 \phi_\kappa \right.\right. \\ &\quad \left.\left. -(\mu_{1,\tilde{\kappa}} \sin^2 \phi_\kappa + \mu_{2,\tilde{\kappa}} \cos^2 \phi_\kappa)\right) + 2(\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2 \cos(2\phi_\kappa)\right] \\ &= 2\sin(2\phi_\kappa)\cos(2\phi_\kappa)\left[(\mu_{1,\tilde{\kappa}} - \mu_{2,\tilde{\kappa}})^2 + 2(\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2\right]. \end{aligned}$$

In which case we see that there are 4 possibilities for ϕ_κ, these being $\phi_\kappa = 0$, $\pi/2$, $\pi/4$ and $3\pi/4$. Now, if $\phi_\kappa = 0$ then $\varsigma_\kappa^2 = \mu_{2,\tilde{\kappa}}/\mu_{1,\tilde{\kappa}}$, which implies $\varsigma_\kappa < 1$, which cannot hold, so this case can be neglected. In the case of $\phi_\kappa = \pi/2$, then $\varsigma_\kappa^2 = \mu_{1,\tilde{\kappa}}/\mu_{2,\tilde{\kappa}}$ and $\bar{I}_1 = 2\mu_{1,\tilde{\kappa}}\mu_{2,\tilde{\kappa}}$. For $\phi_\kappa = \pi/4$ or $\phi_\kappa = 3\pi/4$ we see that $\bar{I}_1 = (\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2$. Now, $2\mu_{1,\tilde{\kappa}}\mu_{2,\tilde{\kappa}} \leq (\mu_{1,\tilde{\kappa}} + \mu_{2,\tilde{\kappa}})^2$, so bearing in mind (A.1.5), we see that the minimum occurs with $(\phi_\kappa, \varsigma_\kappa^2) = (\pi/2, \mu_{1,\tilde{\kappa}}/\mu_{2,\tilde{\kappa}})$.

3. $\mu_{1,\tilde{\kappa}} = \mu_{2,\tilde{\kappa}} \neq 0$

   We can use the simultaneous equations (A.1.3) and (A.1.4) in this case also. Similarly, we obtain 4 possible values of ϕ_κ: $\phi_\kappa = 0$, $\pi/2$, $\pi/4$ and $3\pi/4$. $\phi_\kappa = 0$ is an option in this case as then $\varsigma_\kappa^2 = \mu_{2,\tilde{\kappa}}/\mu_{1,\tilde{\kappa}} = 1$. With both $\phi_\kappa = 0$ and $\phi_\kappa = \pi/2$ we have $\bar{I}_1 = 2\mu_{1,\tilde{\kappa}}^2$ and with $\phi_\kappa = \pi/4$ and $\phi_\kappa = 3\pi/4$ then $\bar{I}_1 = 4\mu_{1,\tilde{\kappa}}^2$. Thus, with (A.1.5) the minimum occurs with either $\phi_\kappa = 0$ or $\phi_\kappa = \pi/2$ and $\varsigma_\kappa = 1$.

4. $|\mu_{1,\tilde{\kappa}}| \geq \mu_{2,\tilde{\kappa}} = 0$

   In this case there are no local minima, but performing the same manipulation as for case 2, we see that as $\varsigma_\kappa \to \infty$ then with $\phi_\kappa = \pi/2$, $\bar{I}_1$ is a decreasing function. Thus, the element should be stretched as much possible but still orientated such that $\phi_\kappa = \pi/2$.

5. $\mu_{1,\tilde{\kappa}}$ and $\mu_{2,\tilde{\kappa}}$ have opposing sign.

   In this case a local minimum occurs with $(s_\kappa^2, \phi_\kappa) = (|\mu_{1,\tilde{\kappa}}|/|\mu_{2,\tilde{\kappa}}|, \pi/2)$. However, as $\varsigma_\kappa \to \infty$ then if ϕ_κ is such that $\tan^2(\phi_\kappa) = -\mu_{1,\tilde{\kappa}}/\mu_{2,\tilde{\kappa}}$, $\bar{I}_1$ is a decreasing function,

so the local minimum stated need not be a global minimum. For simplicity though, we use the local minimum.

In practice this means that, whatever the values of $\mu_{1,\tilde{\kappa}}$ and $\mu_{2,\tilde{\kappa}}$, an element κ should be orientated such that the left singular vector associated with the largest absolute singular value of J_{F_κ} be in the same direction as the eigenvector of $\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}$ whose corresponding eigenvalue has the smallest absolute value. In those cases when neither eigenvalues of $\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}$ are zero we choose $\varsigma_\kappa = \sqrt{|\mu_{1,\tilde{\kappa}}|/|\mu_{2,\tilde{\kappa}}|}$. When only one of the eigenvalues of $\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}$ are zero we stretch the elements up to a maximum prescribed aspect ratio, S, and when both eigenvalues are zero we let $\varsigma_\kappa = 1$.

Remark A.1.1. In the case where convection and/or reaction terms are present a similar analysis reveals that the elements should be orientated as for the purely diffusive case. However, the aspect ratio of the elements is no longer the same, but rather depends on the relative sizes of the diffusion, convection and reaction terms. For simplicity in experiments we always set $\varsigma_\kappa = \sqrt{|\mu_{1,\tilde{\kappa}}|/|\mu_{2,\tilde{\kappa}}|}$.

Case 2, $p > 1$

Unfortunately, it does not appear possible to directly minimize the error bound

$$|J(u) - J(u_{\mathrm{DG}})|^2 \leq C \left(\sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{d,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^{s_\kappa}(\tilde{u}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right) \times \left(\sum_{\kappa \in \mathcal{T}_h} \frac{\alpha}{\sigma_{d,\kappa}^2} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^{t_\kappa}(\tilde{z}, \Sigma_\kappa, U_\kappa) \mathrm{d}\tilde{x} \right),$$

when $p > 1$, $s_\kappa > 2$ and $t_\kappa > 2$. For Case 1, we saw that the important directions for determining anisotropy are the eigenvectors of the Hessian matrices $\tilde{H}^{\tilde{v}}_{\tilde{\kappa}}$, $\tilde{v} = \tilde{u}, \tilde{z}$, however, in this case there appears to be no link between the eigenvalues and eigenvectors (see Theorem 3.3.3) of the higher order tensor of derivatives and the optimal element scale factor and orientation angle, respectively. Indeed, the experiments of Section 5.3.1 reveal that for $p > 1$ the interpolation bounds of Chapter 3 are not sharp enough for this minimization approach to be effective.

Appendix B

Computational Methods

B.1 Ellipse Intersection Algorithm

The Hessian based refinement strategy laid out in Section 5.3.2 requires finding the intersection of two ellipses, one derived from the primal solution u the other from the dual solution z. Finding this intersection is non-trivial and so we approximate it by a modification of the method in Castro Díaz *et al.* [36]. The procedure is as follows:

1. Let the two ellipses be denoted $\mathcal{E}_1$ and $\mathcal{E}_2$.

2. If the ellipses do not intersect (*i.e.* when one is contained within the other) then the smallest ellipse is the one used.

3. Otherwise, for the two solutions define the following: $\mathbf{x}_1^i$, $\mathbf{x}_2^i$ the semi-major and semi-minor axes of the ellipse $\mathcal{E}_i$ with respective lengths λ_1^i and λ_2^i. Then let

$$\begin{aligned} r_1^i &= \text{the radius of } \mathcal{E}_i \text{ in the direction } \mathbf{x}_1^j \quad i,j=1,2,\ i\neq j, \\ r_2^i &= \text{the radius of } \mathcal{E}_i \text{ in the direction } \mathbf{x}_2^j \quad i,j=1,2,\ i\neq j. \end{aligned}$$

   Hence we form two more ellipses with corresponding matrices:

$$J_{F_\kappa}^1 = [\mathbf{x}_1^1, \mathbf{x}_2^1] \begin{bmatrix} \bar{\lambda}_1^1 & 0 \\ 0 & \bar{\lambda}_2^1 \end{bmatrix}$$

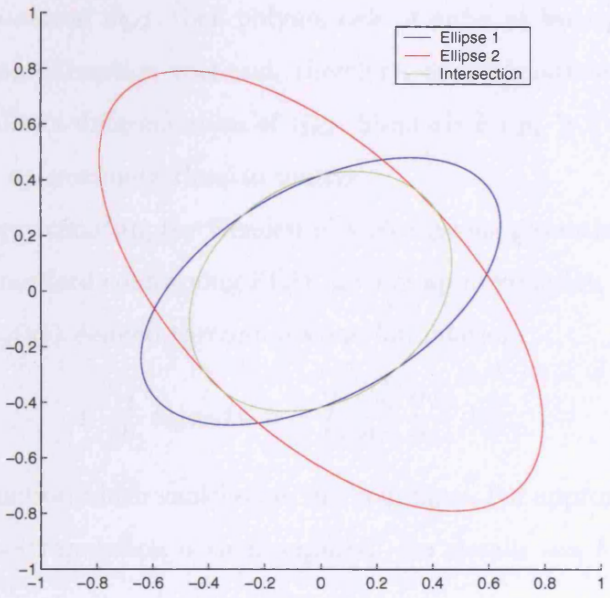


Figure B.1: Approximate Intersection of two Ellipses.

and

$$J^2_{F_{\tilde{\kappa}}} = [\mathbf{x}^2_1, \mathbf{x}^2_2] \begin{bmatrix} \bar{\lambda}^2_1 & 0 \\ 0 & \bar{\lambda}^2_2 \end{bmatrix},$$

where

$$\bar{\lambda}^i_1 = \max(\mu^i_1, r^i_1) \quad \text{and} \quad \bar{\lambda}^i_2 = \max(\mu^i_2, r^i_2) \quad i = 1, 2.$$

4. The intersection ellipse is then derived from the left singular vectors and singular values of the matrix:

$$J_{F_\kappa} = \frac{J^1_{F_{\tilde{\kappa}}} + J^2_{F_{\tilde{\kappa}}}}{2}.$$

We give an example to show this in practice; see Figure B.1.

B.2 Higher Order Derivative Recovery

The Hessian based refinement algorithm introduced in Section 5.3.2 requires an approximation of the Hessian matrices of the true solutions of both the primal and dual problems, u and z, respectively. To do this information must be recovered from the computed solutions u_{DG} and z_{DG}, respectively. Suppose that a polynomial of order $p_\kappa \geq 1$ has been

used for the approximation u_{DG}, then polynomials of order at least $p_\kappa + 1 \geq 2$ will have been used for the approximation z_{DG} and, therefore, approximations of the Hessian of z can be obtained by direct differentiation of z_{DG}. Similarly for $p_\kappa \geq 2$ direct differentiation of u_{DG} will yield an approximate Hessian matrix.

In the case of approximating the Hessian of a continuous piecewise linear function u_h, techniques, for the standard conforming FEM, involve approximation by a piecewise linear function $\mathbf{H}(\mathbf{x}) = (h_{ij}(\mathbf{x}))$ defined through a weak formulation,

$$\int_\Omega h_{ij}\phi_h \mathrm{d}x = -\int_\Omega \frac{\partial u_h}{\partial x_i}\frac{\partial \phi_h}{\partial x_j}\mathrm{d}x,$$

where ϕ_h is a test function which vanishes on the boundary. For approximating the Hessian near the boundary, extrapolation is then required. For details see, for example, Habashi *et al.* [58].

Rather than approximating the Hessian by means of a weak formulation, we modify the method used in the DEAL (Differential Equation Analysis Library) II package of Bangerth, Hartmann and Kanschat [18]. Rather than using finite differences to calculate the Hessian at the centroid of each element as in [18], we instead calculate the Hessian at the nodes of the element, as this leads to improved convergence rates on triangular elements. For ease of exposition we look first at approximation of the gradient. For every regular node $\mathcal{N}_l$, with coordinate $\mathbf{x}_l$ from $\mathcal{T}_h$, *i.e.* those which are not hanging, consider those elements κ_m, $m = 1, \ldots, n$, which have $\mathcal{N}_l$ as a vertex, thus u_{DG} is multivalued at all regular interior nodes. We can define a continuous bilinear representation of u_{DG}, which we call $\bar{u}_{\mathrm{DG}}$ such that

$$\bar{u}_{\mathrm{DG}}(\mathbf{x}_l) := \frac{1}{n}\sum_{m=1}^{n} u_{\mathrm{DG}}^{\kappa_m}(\mathbf{x}_l),$$

where $u_{\mathrm{DG}}^{\kappa} := u_{\mathrm{DG}}|_\kappa$. Values of $\bar{u}_{\mathrm{DG}}$ at hanging nodes can then be found via interpolation. Let a regular node $\mathcal{N}_l$ be connected by a face to the regular or irregular node $\mathcal{N}_{l'}$ and define $\mathbf{y}_{ll'} := \mathbf{x}_{l'} - \mathbf{x}_l$ then

Let a regular node $\mathcal{N}_l$ be connected by a face to the regular or irregular node $\mathcal{N}_{l'}$ and define $\mathbf{y}_{ll'} := \mathbf{x}_{l'} - \mathbf{x}_l$ then

$$\bar{u}_{\mathrm{DG}}(\mathbf{x}_{l'}) - \bar{u}_{\mathrm{DG}}(\mathbf{x}_l)$$

er

all neighbouring nodes then yields:

$$\sum_{l'} \frac{\mathbf{y}_{ll'}}{\|\mathbf{y}_{ll'}\|} \frac{\mathbf{y}_{ll'}^T}{\|\mathbf{y}_{ll'}\|} \nabla u = \left(\sum_{l'} \frac{\mathbf{y}_{ll'}}{\|\mathbf{y}_{ll'}\|} \frac{\mathbf{y}_{ll'}^T}{\|\mathbf{y}_{ll'}\|} \right) \nabla u \approx \sum_{l'} \frac{\mathbf{y}_{ll'}}{\|\mathbf{y}_{ll'}\|} \frac{\bar{u}_{\mathrm{DG}}(\mathbf{x}_l') - \bar{u}_{\mathrm{DG}}(\mathbf{x}_l)}{\|\mathbf{y}_{ll'}\|}.$$

Thus, as long as $\sum_{l'} \frac{\mathbf{y}_{ll'}}{\|\mathbf{y}_{ll'}\|} \frac{\mathbf{y}_{ll'}^T}{\|\mathbf{y}_{ll'}\|} = Y_l$ is non-singular, multiplication from the left by Y_l^{-1} gives:

$$\nabla u \approx Y_l^{-1} \sum_{l'} \frac{\mathbf{y}_{ll'}}{\|\mathbf{y}_{ll'}\|} \frac{\bar{u}_{\mathrm{DG}}(\mathbf{x}_l') - \bar{u}_{\mathrm{DG}}(\mathbf{x}_l)}{\|\mathbf{y}_{ll'}\|}.$$

We can be sure that Y_l does indeed have an inverse as long as the directions $\{\mathbf{y}_{ll'}\}_{l'}$ span the whole space $\mathbb{R}^d$, which in our case will always be true. The gradient can now be interpolated at any point in the mesh.

For higher order derivatives, of order n, then as long as we have a solution of order $n-1$ then the above method can be used to approximate the higher order derivatives. For example for the Hessian, we create a continuous bilinear representation $\bar{u}_{\mathrm{DG},x_i}$, for $i = 1, \ldots, d$, in the same way as we did for $\bar{u}_{\mathrm{DG}}$. Thus,

$$\nabla u_{x_i}(\mathbf{x}_l) \approx Y_l^{-1} \sum_{l'} \frac{\mathbf{y}_{ll'}}{\|\mathbf{y}_{ll'}\|} \frac{\bar{u}_{DG,x_i}(\mathbf{x}_{l'}) - \bar{u}_{DG,x_i}(\mathbf{x}_l)}{\|\mathbf{y}_{ll'}\|},$$

for $i = 1, \ldots, d$.

Of course, $u_{x_i x_j} \equiv u_{x_j x_i}$, but the method above does not guarantee that our two approximations are equal. To compensate for this we use the arithmetic mean of the two approximations to ensure the Hessian is symmetric. We denote our approximate derivative by $H(u_{\mathrm{DG}})$. For higher order derivatives the method is completely analogous.

One advantage of this method is that the calculation of the Hessian within an element can be computed locally by considering only those nodes which are vertices of the element, thus, much computational expense is saved when compared with globally approximating the Hessian.

Remark B.2.1. Our method for computing the Hessian results in a continuous linear approximation, which can prove useful when performing r-refinement in a metric setting; see Section 9.2.2 for details.

We now look at an example of this method in practice. We solve the following problem

$$-\nabla^2 u = f,$$

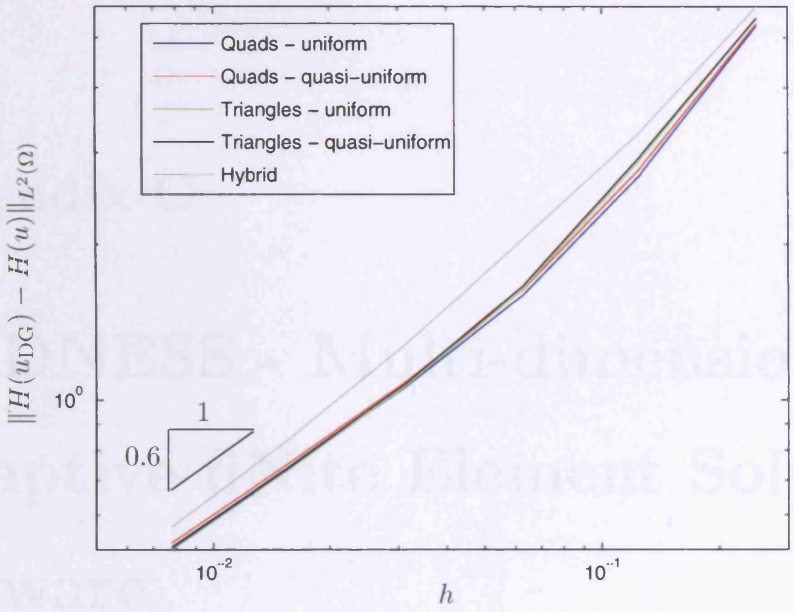


Figure B.2: Convergence of recovered Hessians.

on $\Omega = [0,1]^2$, where boundary conditions and f are chosen such that

$$u = cos(\pi x)sin(\pi y).$$

We compute approximations of the Hessian on a series of uniform and quasi-uniform quadrilateral, triangular and hybrid grids and look at convergence of the error of the approximated Hessian in the L^2-norm, as the mesh size h is decreased; see Figure B.2. We see that convergence of order roughly $\mathcal{O}(h^{0.6})$ is observed in each case.

Appendix C

MADNESS - Multi-dimensional ADaptive fiNite Element Solver Software

A main part of the project was spent designing and implementing a new finite element software package. Although other well established packages are available, for example the DEAL II package of Bangerth, Hartmann and Kanschat [18], none of them offer the flexibility needed to perform the mesh refinements considered in this thesis. As such, a new software package, written in Fortran 95 and christened MADNESS - Multi-dimensional ADaptive fiNite Element Solver Software, was born, with the following objectives in mind:

- Dimensionally independent data structures.
- Admissibility of hybrid grids - *i.e.* triangular and quadrilateral meshes in 2D and tetrahedral, hexahedral, pyramidal, and prismatic meshes in 3D.
- Applicability to a wide range of problems.
- Availability of a wide range of refinement types.
- Ease of use.

These constraints meant that the code had to be developed in as general a way as possible.

In this appendix we shall discuss some of the important decisions made when designing the MADNESS code and although not meant to be an instruction manual, the reader should gain some insight into the use of the code. A more thorough review of the package can be found in [69].

C.1 General Design of a Finite Element Code

Any adaptive finite element code can be broken down into the following procedures:

1. The setting up of a mesh and a finite element space.
2. The assembly of a discrete system of linear equations.
3. The solution of this system of equations.
4. The adaptive refinement/derefinement process.
5. Solution evaluation and visualization/post-processing.

Figure C.1 shows the flow through the program and further breaks down each of the above points into smaller building blocks. These points are discussed in detail in the following sections.

C.2 Source Code and Problem Setup

The main body of code should offer many of the subroutines needed for a user to create their own solver for whichever application they desire. Upon compilation of the source code, there should be little need to modify its contents again, except for major changes such as a switch of the arithmetic precision. In certain circumstances the user may wish to use, say, a custom quadrature or custom finite element space not included within the package, the modular nature of the code facilitates this.

A number of template programs are provided with the code, specifically for the solution of advection-diffusion-reaction problems, the Stokes' problem and compressible flow problems. The files for each of these model problems are stored in separate directories

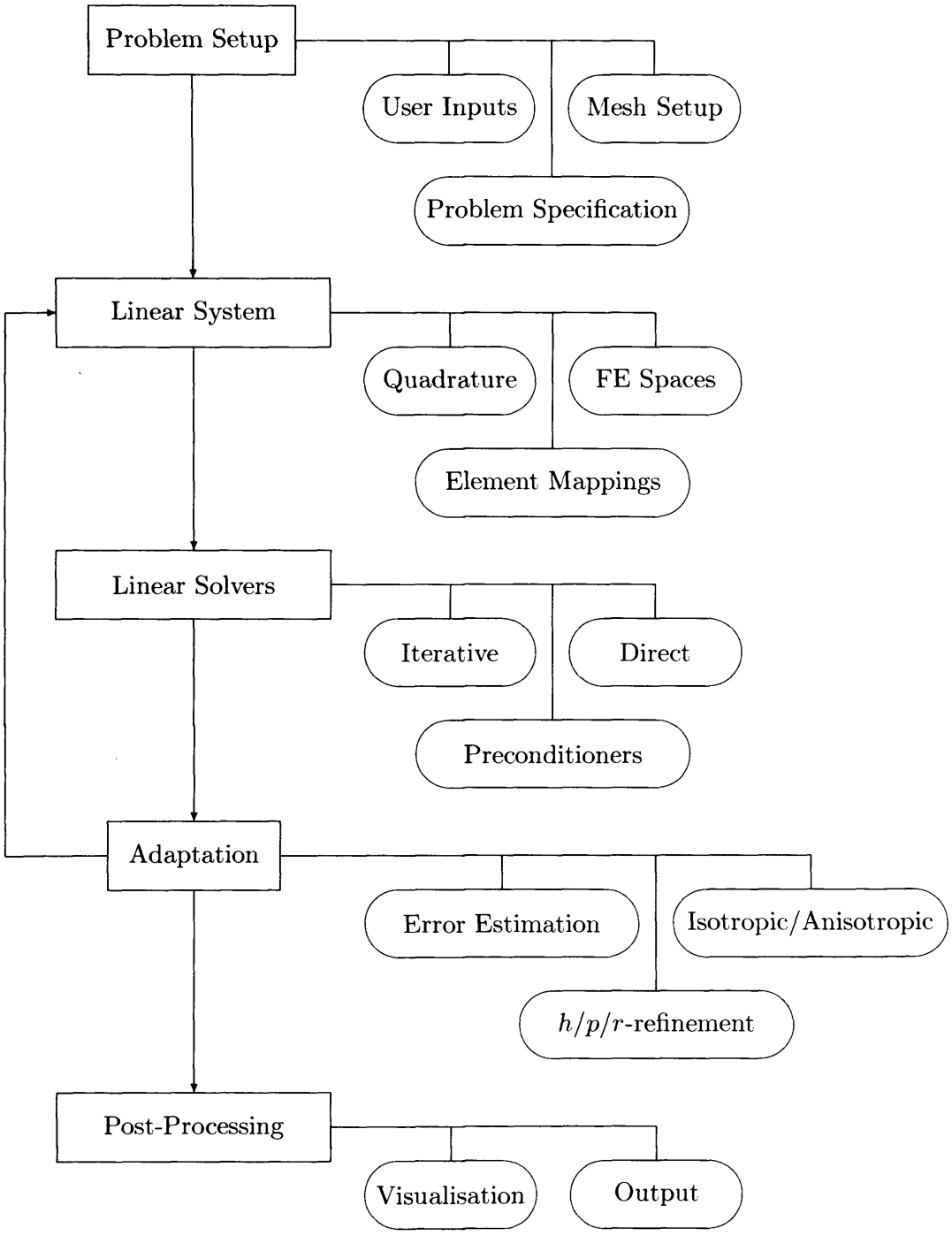


Figure C.1: Flow through a MADNESS program.

and individual Makefiles provide the means to link with the source code. Otherwise, the files and documentation should provide enough insight into the running of the code that little difficulty be required in setting up other applications.

C.3 Inputs and Output

To offer the greatest flexibility to the user and prevent endless recompilation, all of the parameters, except the arithmetic precision, can be provided at runtime. For the template programs, these options are passed in by way of the file `femdata.dat`.

On completion of solving the problem to the desired level of accuracy, the results need to be output in some way. Solutions and meshes can be output in a format suitable either to be read in by the code again, or for visualisation, either by Matlab or by HiVision (See Section C.8). For the output of information such as convergence results, functional values, *etc*, there are subroutines which output to `.dat` and `.tex` files, where the user has full control over the information exported.

C.4 Meshes

C.4.1 Computational Mesh

The three most important parts of a DG discretization are the cells and faces of the triangulation and numerical integration. Indeed, in order to be able to set up the system of linear equations rapidly, another requirement is that the cells, which border a face, need to be established quickly. For this reason a mesh data structure has been utilized, where such entities as element connectivity and face information are stored as arrays, in contrast to the mesh tree (see below), where information is stored in linked lists, by way of pointers.

At the heart of the computational mesh data structure is the derived type `mesh`, which contains general information about the mesh, plus arrays of derived type `element` and `face`, Figure C.2 shows these derived types. No information concerning the finite element space is included in the mesh definition, hence this data structure can be used multiple

times within the program, for example for both the primal and dual problems, and also means there is no restriction to only a DG discretization.

C.4.2 Mesh Tree

For purposes of mesh refinement/derefinement it was decided that a tree structure was the best way to proceed. More precisely, since a refinement not only results in the subdivision of elements, but also the subdivision of faces (and edges in 3D), two (three in 3D) trees are setup, one consisting of elements, the other of the faces. This way, a full history of all the elements and faces is stored and hence derefinement is simply a matter of returning to a previous element/face in the trees. Initially the computational mesh is setup, based on either input from a data file, or by the simple hypercube mesh generator included, then prior to the first adaptive step the trees are setup. Hence, the first level of the trees consists of the elements and faces in the initial mesh, as such, derefinement to a mesh more coarse than the original is forbidden. In the case of faces, a refinement, as well as creating subfaces of existing faces, also creates entirely new faces, these are tagged on to the first level of the tree. Pointers between the elements and their corresponding faces are then required to maintain order. Additional pointers between the computational mesh's elements/faces and the corresponding elements/faces of the tree are also setup. Figure C.3 shows a square grid after one isotropic and one anisotropic refinement step, together with the tree representation of its elements and faces. Thus, terminal nodes in the tree are active in the computational mesh, while non-terminal nodes represent the now redundant ancestor cells and faces.

Extraction of a computational mesh from the tree is carried out by means of a 'tree walk'. The process begins at the first node of the first level and looks at the children of that node. If no children are present then this node must be part of the computational mesh and will be included, otherwise the children are considered sequentially and levels descended until there are no children and the nodes are terminal.

```
type mesh
   real(kind=db), dimension(:,:), pointer :: coords,face_normals
   real(kind=db), dimension(:), pointer :: face_jacobian
   type(face), dimension(:), pointer :: mesh_faces
   type(element), dimension(:), pointer :: mesh_ele
   type(linkedlistint), dimension(:),pointer :: node_eles
   integer :: no_eles,no_nodes,no_dofs,no_faces,which_face_representation
   integer :: problem_dim
end type mesh
```

(a)

```
type element
   type(linkedlistint), pointer :: ele_nodes
   type(linkedlistint), dimension(:), pointer :: faces_subfaces
   integer, dimension(:), pointer :: face_representation
   integer :: no_faces
   integer :: element_type
   type(tree_node), pointer :: tree_location
end type element
```

(b)

```
type face
   type(linkedlistint), pointer :: ele_nodes
   integer, dimension(2) :: neighbours,loc_face_no
end type face
```

(c)

Figure C.2: Mesh structure derived types (a) mesh, (b) element and (c) face.

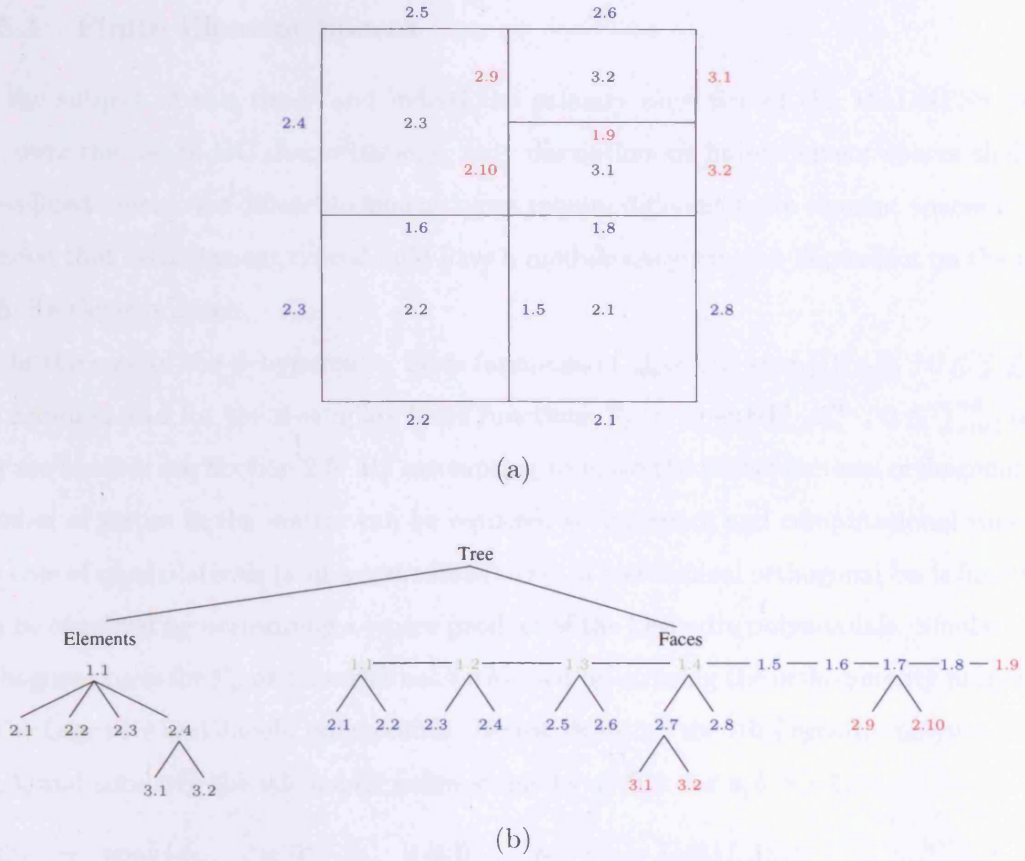


Figure C.3: (a) Square mesh after one isotropic refinement and one anisotropic refinement and (b) the resultant tree structure.

C.5 Linear System

Equipped with a computational mesh, the next step is to setup the linear system which discretizes the differential equation. In order to do this, the key features required of the code are a finite element space defined on the reference element, quadratures defined on the reference element and faces, element mappings to transform the finite element space and quadrature points to the physical element and a data structure to store the resultant matrix. The discussion below deals with these points separately.

C.5.1 Finite Element Spaces

As the subject of this thesis and indeed the primary objective of the MADNESS package were the use of DG discretizations, only discontinuous finite element spaces shall be considered herein. As different element types require different finite element spaces it was decided that each element type should have a module assigned to it dependent on the type of finite element space.

In the case of the d–hypercube, basis functions of $\mathcal{Q}_{\vec{p}}(\hat{\kappa}) := \text{span}\{\Pi_{i=1}^{d}\hat{x}_i^j : 0 \leq j \leq p_i\}$ are required and for the d–simplex basis functions $\mathcal{P}_p := \text{span}\{\Pi_{i=1}^{d}\hat{x}_i^{\alpha_i} : 0 \leq \sum_{i=1}^{d}\alpha_i \leq p\}\}$ are needed; see Section 2.5. By attempting to make the basis functions orthogonal the number of zeroes in the matrix can be reduced, saving space and computational time. In the case of quadrilaterals (and hexahedrons) a set of hierarchical orthogonal basis functions can be obtained by performing a tensor product of the Legendre polynomials. Similarly, an orthogonal basis for $\mathcal{P}_p$ on triangles can be formed by utilizing the orthogonality properties of the Legendre and Jacobi polynomials. Hence, denoting the ith Legendre polynomial by $L_i(\cdot)$, and similarly the ith Jacobi polynomials by $J_i^{a,b}(\cdot)$, for $a, b > -1$,

$$\begin{aligned}
\mathcal{Q}_{\vec{p}} &= \text{span}\{\phi_{ij},\ \ i = 0 \ldots p_1,\ \ j = 0, \ldots, p_2 : \phi_{ij} = L_i(\hat{x})L_j(\hat{y})\} \\
\mathcal{P}_p &= \text{span}\left\{\phi_{ij},\ \ i = 0 \ldots p,\ \ j = 0, \ldots, p - i : \phi_{ij} = \left(\frac{1-\hat{y}}{2}\right)^i L_i(\hat{\eta})J_j^{2i+1,0}(\hat{y})\right\},
\end{aligned}$$

where $\hat{\eta} = 2(1+\hat{x})/(1-\hat{y})$. Basis functions are nearly always computed at the quadrature points and, for a polynomial degree p in one coordinate direction, $p + npinc$ quadrature points are used so that integration of the basis functions is exact. Hence, to save computational time the basis functions are computed the first time for a given degree p and then stored for quick access in the future. Similarly, the Legendre and Jacobi polynomials are also computed and stored in this way.

C.5.2 Quadrature

Similarly to the finite element space, each type of element needs its own quadrature defined upon it, that is a set of points, $\mathbf{x}_q$, and weights w_q such that for a function f,

$$\int_\kappa f\mathrm{d}x \approx \sum_q f(\mathbf{x}_q)w_q\mathrm{d}x.$$

As such, quadrature routines for each element are all based around performing tensor product manipulations of the one-dimensional quadrature on the interval $[-1, 1]$, which are stored once they have been computed for future use. Suppose the one-dimensional quadrature points are ρ_i, $i = 1, \ldots, n$, with corresponding weights w_i, $i = 1, \ldots, n$, then for the quadrilateral element the set of points and weights are given by

$$\{\mathbf{x}, w\}_{ij} = \{(\rho_i, \rho_j), w_i w_j\}.$$

For the triangle the quadrature points and weights are found by mapping from the reference quadrilateral to the reference triangle. The mapping used is

$$\begin{aligned} x^T &= \frac{(1 + x^Q)(1 - y^Q)}{2} - 1, \\ y^T &= y^Q, \end{aligned}$$

where (x^Q, y^Q) are the coordinates in the quadrilateral reference frame and (x^T, y^T) are in the triangle reference frame. The new weights are then found by multiplication of the old weights by the Jacobian of the mapping at the corresponding coordinates. Evidently this mapping collapses the line $y^T = 1$ to a single point and is therefore singular. However, as long as the one-dimensional quadrature does not include the end-points of the interval, quadrature on a triangle is well defined. Analogous operations can be performed for three dimensional elements to formulate quadratures. Currently only Gauss quadrature is coded, where the quadrature points are the roots of the Legendre polynomials.

C.5.3 Element Mappings

Computations are not performed on the reference element or reference element face, but rather on the physical element. Suppose the physical element is transformed from the reference element by way of the mapping F_κ, cf Figure C.4 (not to be confused with the F_κ introduced in Chapter 2) and consider, for example, the bilinear form of the Laplacian operator

$$\sum_{\kappa \in \mathcal{T}_h} \int_\kappa \nabla u_h \cdot \nabla u_h \mathrm{d}x,$$

which results in the need to calculate, for every $\kappa \in \mathcal{T}_h$,

$$\int_\kappa \nabla \phi_i \cdot \nabla \phi_j \mathrm{d}x \qquad 1 \leq i, j \leq n_\kappa,$$

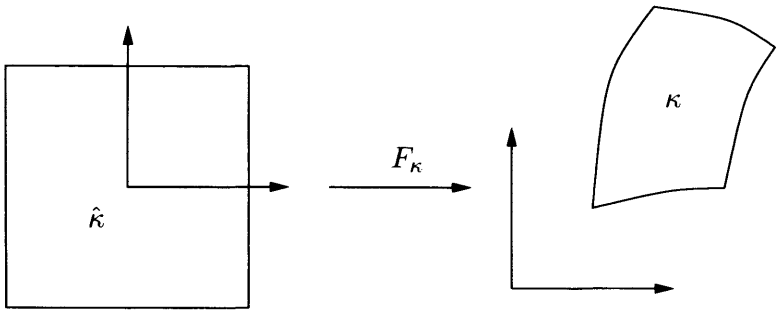


Figure C.4: Element Mappings.

where n_κ are the number of degrees of freedom on element κ and ϕ_i, $i = 1, \ldots, n_\kappa$ are the basis functions associated with the element. In this case numerical quadrature gives

$$I_{ij} := \int_\kappa \nabla u_h \cdot \nabla u_h \mathrm{d}x \approx \sum_q \nabla \phi_i(\mathbf{x}_q) \cdot \nabla \phi_j(\mathbf{x}_q) w_q,$$

where $\mathbf{x}_q$ are a set of quadrature points and w_q are the corresponding set of weights. Transforming from the reference element, under the mapping F_κ, using the chain rule gives

$$I_{ij} = \sum_q J_{F_\kappa}^{-1} \hat{\nabla} \hat{\phi}_i(\hat{\mathbf{x}}_q) \cdot J_{F_\kappa}^{-1} \hat{\nabla} \hat{\phi}_j(\hat{\mathbf{x}}_q) |\det(J_{F_\kappa}(\mathbf{x}_q))| \hat{w}_q,$$

where J_{F_κ} represents the Jacobian of the mapping F_κ, $\hat{\phi}_i$ represent the local basis functions on the reference element and $\mathbf{x}_q$ and w_q are the quadrature points and weights, respectively, defined on the reference element.

Therefore, it is essential to able to map from the reference element to the physical element and vice versa and to evaluate the Jacobian of the mapping (and its inverse) at the quadrature points; subroutines which do this are included. In the case of affine element mappings the Jacobian remains constant and hence can be stored easily for quick access. For non-affine mappings, it is not practical to store the Jacobian at any coordinates and it must be calculated as and when it is needed.

C.5.4 `dg_volume_integration_info` and `dg_face_integration_info`

When computing volume integrals and face integrals three pieces of information are always needed: quadrature points and weights, basis function values and element mappings,

```
type dg_soln
  real(kind=db), dimension(:), pointer :: solnvalues
  integer, dimension(:,:), pointer :: istart
  integer, dimension(:,:), pointer :: basis
  integer, dimension(:,:,:), pointer :: poly_vec
  integer :: no_dofs
  integer :: nv
end type dg_soln
```

Figure C.5: Derived type `dg_soln`.

both of which are evaluated at the quadrature points. Hence it seems logical to provide a single function which returns these entities. MADNESS provides this by way of the subroutine `dg_volume_integration_info` when a volume integral is about to be calculated and `dg_face_integration_info` in the case of face integrals. Of course, the objects are also still available separately, should the need arise.

C.5.5 Degrees of Freedom and Solution and Matrix Setup

For a particular solution, on every element in the mesh, there is an associated number of degrees of freedom, which represent the coefficients of the basis functions discussed above. These number of degrees of freedom can be calculated using the polynomial degree vector and the type of basis. As mentioned before it is not practical to keep the information concerning finite element space with the mesh, but rather it is stored with the `dg_soln` derived type, for which the code is shown in Figure C.5. The `istart` pointer is of dimension (`nv`,`no_eles`) and denotes where the degrees of freedom start in `solnvalues` for the element and variable numbers. `basis` has dimension (`nv`,`no_eles`) and `poly_vec` is of dimension (`nv`,`problem_dim`,`no_eles`) and give the basis type and polynomial vector for the element and variable respectively.

DG discretizations are liable to produce a very large system of linear equations and hence could potentially lead to matrices requiring huge amounts of storage. However, the matrices tend to be very sparse. This sparsity can be utilized by providing sparse matrix data structures. Included in the MADNESS package is the *Compressed Sparse Row* format,

```
type csr_sp_matrix
  real(kind=db), dimension(:), pointer :: matrix_entries
  integer, dimension(:), pointer :: column_no, row_start
end type sparse_matrix
```

Figure C.6: Derived type `csr_sp_matrix`.

contained in module `csr_sparse_matrix`. the code for the derived type is shown in Figure C.6. Here, the ith entry of `column_no` vector gives the column number in the full matrix of the ith entry in `matrix_entries`. The entries in both `column_no` and `matrix_entries` are stored in row order and the jth entry of `row_start` gives the position in `column_no` and `matrix_entries` where the jth row begins. An example provides a clearer insight into how this works.

$$A = \begin{bmatrix} 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \rightarrow \quad \begin{cases} \texttt{matrix\_entries} = \begin{bmatrix} 2 & 1 & 4 & 5 & 1 & 7 & 1 \end{bmatrix}. \\ \texttt{column\_no} = \begin{bmatrix} 1 & 4 & 3 & 2 & 3 & 5 & 1 \end{bmatrix}. \\ \texttt{rowstart} = \begin{bmatrix} 1 & 3 & 4 & 6 & 7 & 8 \end{bmatrix}. \end{cases}$$

Coupled with the `csr_sparse_matrix` are routines which can input/retrieve entries given the row and column locations in the full matrix and routines for providing matrix-vector multiplications.

Alternatively, there is the *MUMPS sparse matrix*. for use when the direct MUMPS solver is to be employed; see Section C.6. The same routines for inputting/retrieving entries are also available in this case.

In the case when local solutions need to be found (for example for purposes of anisotropic refinement) the size of the matrix formed is relatively small. combining this with the overheads involved in setting up a sparse matrix. it seems more logical just to use the standard array type provided by Fortran 95.

C.6 Linear Solvers

After computation of the matrix and corresponding, the (often very large) matrix problem

$$A\mathbf{x} = \mathbf{b}$$

has to be solved, where $\mathbf{x}$ represents the solution vector. Means of doing this fall into two categories: *I*: direct solvers, and *II*: iterative solvers. Although direct solvers lead to exact (at least to machine precision) solutions, they are often highly memory intensive and as such iterative solvers have generally been preferred.

C.6.1 Iterative Solvers

A particularly common class of iterative solver are the so called *Krylov subspace methods*. The nth Krylov subspace of a matrix, $A \in \mathbb{R}^{m\times m}$, with respect to a vector, $\mathbf{v} \in \mathbb{R}^m$, is defined as

$$\mathcal{K}_n(A, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{n-1}\mathbf{v}\},$$

and the associated nth Krylov matrix, $K_n \in \mathbb{R}^{m\times n}$, is then defined as

$$K_n = \left[\begin{array}{c|c|c|c} \mathbf{v} & A\mathbf{v} & \cdots & A^{n-1}\mathbf{v} \end{array}\right].$$

If the wish is to solve the matrix equation

$$A\mathbf{x} = \mathbf{b}.$$

then a sequence of iterates, $\mathbf{x}_n \in \mathbf{x}_0 + \mathbf{v}_n$, can be found which approximate $\mathbf{x}$ in some way, with increasing accuracy as n is increased. Here, $\mathbf{x}_0 \in \mathbb{R}^m$ is an initial guess to the solution and $\mathbf{v}_n \in \mathcal{K}_n(A, \mathbf{v}_0)$, for some $\mathbf{v}_0$, is a correctional term. The question is then how to choose the correctional terms. A popular Krylov subspace method is the Generalized Minimal RESidual solver (GMRES) suitable for general non–symmetric matrices; see [117], hence MADNESS comes equipped with an implementation of GMRES.

C.6.2 Preconditioning

For a non-singular matrix, the GMRES can be shown to be convergent, with the rate of convergence dependent on the condition number of the matrix. Generally, it is not practical to use an iterative solver without some preconditioning. Suppose, as usual, the system of equations to solve is

$$A\mathbf{x} = \mathbf{b}, \tag{C.6.1}$$

where $A \in \mathbb{R}^{m\times m}$. For a non-singular matrix $M \in \mathbb{R}^{m\times m}$, the system

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b} \tag{C.6.2}$$

shares the same solution as (C.6.1). Hence, if the inverse of the *preconditioner* M approximates the inverse of A the condition number of (C.6.2) is likely to be smaller than that of (C.6.1) and will require fewer iterations to solve. Of course, a good preconditioner must approximate A well, but also be easy to compute, else the computational effort will just be switched from the iteration method to the calculation of M.

MADNESS provides a number of preconditioners, which utilize the structure of the matrix, these are discussed below.

Block Preconditioners

The matrices formed by a DG discretization have a distinct block structure, with each block on the diagonal corresponding to the volume integrals on each element. Blocks below and above the diagonal then emanate from the interactions between elements. Consider now the block Jacobi, block Gauss-Seidel and block Symmetric Succesive Over Relaxation (SSOR) iterations, which represent the extensions to block systems of their non-block counterparts:

$$\begin{aligned}
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + D^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}). \quad \text{Jacobi} \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + (D+L)^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}). \quad \text{Gauss-Seidel} \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + (D+U)^{-1}D(D+L)^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}). \quad \text{SSOR.}
\end{aligned}$$

where $A = D + L + U$, with D the block diagonal component, and L and U the lower and upper block components respectively. Hence, in each case A^{-1} has been approximated by

D^{-1}, $(D+L)^{-1}$ and $(D+U)^{-1}D(D+L)^{-1}$, respectively and thus

$$\begin{aligned} M &= D \\ M &= (D+L) \\ M &= (D+U)D^{-1}(D+L) \end{aligned}$$

are potential candidates for preconditioners. For each case, the matrix D^{-1} can be computed and stored for further use, and the matrix-vector multiplications by $(D+L)^{-1}$ and $(D+U)^{-1}$ can be achieved by back and forward substitutions respectively.

Incomplete LU (ILU) Factorisation

Suppose A is factored into the product of a lower triangular matrix L, and an upper triangular matrix U, as $A = LU$. Then, evidently LU is the perfect preconditioner, although of course by performing the LU factorization the matrix equation (C.6.1) has effectively been solved already at some cost. Suppose rather that the sparsity structure of the matrix A is made use of and an incomplete LU factorization carried out to create lower and upper triangular matrices, $\hat{L}$ and $\hat{U}$, where non-zero elements in L and U are set to zero if the corresponding elements of A are zero. Of course, this is not the way the incomplete LU factorization is actually computed, but rather it is done with far fewer operations than computing the actual LU factorization. The preconditioner

$$M = \hat{L}\hat{U}$$

can then be used and matrix multiplications can, once again, be done using successive back and forward substitutions.

C.6.3 Direct Solvers

The MADNESS package provides support for the serial version of the direct solver MUMPS (MUltifrontal Massively Parallel sparse direct Solver), which has proven to be very effective for solving systems which are not too large. The techniques used in the MUMPS solver are beyond the scope of this thesis, but can be found in [6].

C.7 Adaptive Refinement

The majority of this thesis has been concerned with isotropic and anisotropic refinement techniques, so there is little to be said on that subject in this section, except to discuss how these are actually implemented.

C.7.1 *A Posteriori* Error Estimation

The calculation of an *a posteriori* error estimate for functionals requires both the solution of the primal problem and the dual problem. The same mesh structure can be used in each case, but the polynomial degree must be increased by *pinc* for the dual problem. In the linear functional case the same routines can be used to setup the matrix and right hand side for the dual problem, as for the primal problem, however the solution step will involve transposing the resultant matrix in the dual case.

C.7.2 Obtaining Refinement Indicators

When determining anisotropy using the solution of local problems, a mesh subset is required, which picks out the element that is a candidate for refinement. In some instances more than one element may be required. This mesh subset must have pointers to the original mesh, so that boundary conditions can be enforced using the global solution. By storing the mesh subset as a new tree, the adaptive refinements described above can again be used to form the local patch on which the local problem is solved, extraction to a computational mesh is then required. The routines used to create the global matrix can then be called with a local argument, so that they can be used with the local mesh. As noted before, the local matrices are stored as simple arrays, rather than in sparse format.

C.8 External Packages

For improved performance, MADNESS provides interfaces to the following packages:

- GotoBLAS - a highly optimized Basic Linear Algebra Subprograms (BLAS) package.
- MUMPS - a direct linear solver package, [6].

- SparseKit - a package for the manipulation of sparse matrices, including iterative solvers, [116].
- ARPACK - a package for the solution of eigenvalue/eigenvector problems, [96].

Visualization of solutions is available either through Matlab or through the HiVision package; see Bönisch and Heuveline [27]. The HiVision package is based around the C++ Visualization Toolkit (VTK) and offers excellent graphical output, especially for three-dimensional geometries.

C.9 Future Development

Currently it is planned to develop the code in the following directions.

- Three-dimensional adaptive refinement.
- Parallelization.
- Extension to eigenvalue/eigenvector problems.
- Incorporation of a graphical user interface (GUI).
- Implementation of conforming FEMs in three-dimensions.

Bibliography

[1] R. A. Adams. *Sobolev spaces.* Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1975. Pure and Applied Mathematics, Vol. 65.

[2] S. Adjerid, M. Aiffa, and J. E. Flaherty. Computational methods for singularly perturbed systems. In J. Cronin and R.E. O'Malley, editors, *Analyzing multiscale phenomena using singular perturbation methods (Baltimore, MD, 1998)*, volume 56 of *Proc. Sympos. Appl. Math.*, pages 47–83. Amer. Math. Soc., Providence, RI, 1999.

[3] M. Ainsworth and J. T. Oden. A posteriori error estimation in finite element analysis. *Comput. Methods Appl. Mech. Engrg.*, 142(1-2):1–88, 1997.

[4] M. Ainsworth and B. Senior. An adaptive refinement strategy for *hp*-finite element computations. *Appl. Numer. Math.*, 26(1-2):165–178, 1998.

[5] D. Ait-Ali-Yahia, G. Baruzzi, W. G. Habashi, M. Fortin, J. Dompierre, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. II. Structured grids. *Internat. J. Numer. Methods Fluids*, 39(8):657–673, 2002.

[6] 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, 200.

[7] T. Apel. *Anisotropic finite elements: Local estimates and applications.* Advances in Numerical Mathematics. Teubner, Stuttgart, 1999.

[8] T. Apel, S. Grosman, P.K. Jimack, and A. Meyer. A new methodology for anisotropic mesh refinement based upon error gradients. *Appl. Numer. Math.*, 50:329–341, 2004.

[9] T. Apel and G. Lube. Anisotropic mesh refinement for a singularly perturbed reaction diffusion model problem. *Appl. Numer. Math.*, 26(4):415–433, 1998.

[10] T. Apel and S. Nicaise. The finite element method with anisotropic mesh grading for elliptic problems in domains with corners and edges. *Math. Methods Appl. Sci.*, 21(6):519–549, 1998.

[11] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[12] D. N. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Discontinuous Galerkin methods for elliptic problems. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 89–101. Springer, Berlin, 2000.

[13] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02.

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

[15] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM J. Numer. Anal.*, 13(2):214–226, 1976.

[16] I. Babuška and M. R. Dorr. Error estimates for the combined h and p versions of the finite element method. *Numer. Math.*, 37(2):257–277, 1981.

[17] M. J. Baines and M. E. Hubbard. Multidimensional upwinding with grid adaptation. In *Numerical methods for wave propagation (Manchester, 1995)*, volume 47 of *Fluid Mech. Appl.*, pages 33–54. Kluwer Acad. Publ., Dordrecht, 1998.

[18] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II — a general purpose object oriented finite element library. *ACM Trans. Math. Software*. to appear.

[19] R. Becker, P. Hansbo, and M.G. Larson. Energy norm a posteriori error estimation for discontinuous Galerkin methods. *Comput. Methods Appl. Mech. Engrg.*, 192:723–733, 2003.

[20] R. Becker, P. Hansbo, and R. Stenberg. A finite element method for domain decomposition with non-matching grids. *Modél. Math. Anal. Numér.*, 37:209–225, 2003.

[21] R. Becker and R. Rannacher. Weighted a posteriori error control in FE methods. Technical report. Preprint 1, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Universität Heidelberg, Heidelberg, Germany, 1996.

[22] Y. Belhamadia, A. Fortin, and É. Chamberland. Anisotropic mesh adaptation for the solution of the Stefan problem. *J. Comput. Phys.*, 194(1):233–255, 2004.

[23] Y. Belhamadia, A. Fortin, and É. Chamberland. Three-dimensional anisotropic mesh adaptation for phase change problems. *J. Comput. Phys.*, 201(2):753–770, 2004.

[24] J. Bergh and J. Löfström. *Interpolation spaces. An introduction.* Springer-Verlag, Berlin, 1976. Grundlehren der Mathematischen Wissenschaften, No. 223.

[25] C. Bernardi, N. Fiétier, and R. G. Owens. An error indicator for mortar element solutions to the Stokes problem. *IMA J. Numer. Anal.*, 21(4):857–886, 2001.

[26] K.S. Bey, A. Patra, and J. T. Oden. *hp*-version discontinuous Galerkin methods for hyperbolic conservation laws: a parallel adaptive strategy. *Internat. J. Numer. Methods Engrg.*, 38(22):3889–3908, 1995.

[27] S. Bönisch and V. Heuveline. HiVision, an advanced framework for flow visualization. Technical report, Institute of Applied Mathematics, University of Heidelberg, 2004.

[28] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[29] F. Brezzi, L. P. Franca, T. J. R. Hughes, and A. Russo. $b = \int g$. *Comput. Methods Appl. Mech. Engrg.*, 145(3-4):329–339, 1997.

[30] F. Brezzi, T. J. R. Hughes, L. D. Marini, A. Russo, and E. Süli. A priori error analysis of residual-free bubbles for advection-diffusion problems. *SIAM Journal on Numerical Analysis*, 36(6):1933–1948, 1999.

[31] F. Brezzi, D. Marini, and E. Süli. Residual-free bubbles for advection-diffusion problems: the general error analysis. *Numerische Mathematik*, 85(1):31–47, 2000.

[32] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982.

[33] C. Canuto and A. Quarteroni. Approximation results for orthogonal polynomials in Sobolev spaces. *Math. Comp.*, 38(157):67–86, 1982.

[34] W. Cao. An interpolation error estimate in R^2 based on the anisotropic measures of higher order derivatives. *Math. Comp., to appear.*

[35] W. Cao. Anisotropic measures of third order derivatives and the quadratic interpolation error on triangular elements. *SIAM J. Sci. Comput.*, 29(2):756–781, 2007.

[36] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *Internat. J. Numer. Methods Fluids*, 25:475–491, 1997.

[37] P. G. Ciarlet. *The finite element method for elliptic problems.* North-Holland Publishing Co., Amsterdam, 1978. Studies in Mathematics and its Applications, Vol. 4.

[38] B. Cockburn and H. Gau. A posteriori error estimates for general numerical methods for scalar conservation laws. *Comput. Appl. Math.*, 14:37–47, 1995.

[39] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. The development of discontinuous Galerkin methods. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 3–50. Springer, Berlin, 2000.

[40] E. Creusé and S. Nicaise. Anisotropic a posteriori error estimation for the mixed discontinuous Galerkin approximation of the Stokes problem. *Numer. Methods Partial Differential Equations*, 22(2):449–483, 2006.

[41] P. J. Davis. *Interpolation and approximation.* Blaisdell Publishing Co. Ginn and Co. New York-Toronto-London, 1963.

[42] L. Demkowicz, W. Rachowicz, and Ph. Devloo. A fully automatic *hp*-adaptivity. In C.-W. Shu, editor, *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala)*, volume 17, pages 117–142, 2002.

[43] V. Dolejší and J. Felcman. Anisotropic mesh adaptation for numerical solution of boundary value problems. *Numer. Methods Partial Differential Equations*, 20(4):576–608, 2004.

[44] J. Dompierre, M.-G. Vallet, Y. Bourgault, M. Fortin, and W. G. Habashi. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. III. Unstructured meshes. *Internat. J. Numer. Methods Fluids*, 39(8):675–702, 2002.

[45] T. Eibner and J. M. Melenk. An adaptive strategy for *hp*-fem based on testing for analyticity. Technical Report 12/2004, University of Reading, Department of Mathematics, 2004.

[46] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. In A. Iserles, editor, *Acta Numerica*, pages 105–158. Cambridge University Press, 1995.

[47] L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaption in computational fluid dynamics: application to the advection-diffusion-reaction and the Stokes problems. *Appl. Numer. Math.*, 51(4):511–533, 2004.

[48] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numer. Math.*, 89:641–667, 2001.

[49] L. Formaggia and S. Perotto. Anisotropic error estimates for elliptic problems. *Numer. Math.*, 94(1):67–92, 2003.

[50] P. J. Frey and F. Alauzet. Anisotropic mesh adaptation for CFD computations. *Comput. Methods Appl. Mech. Engrg.*, 194(48-49):5068–5082, 2005.

[51] E. H. Georgoulis. Inverse-type estimates on *hp*-finite element spaces and applications. *Math. Comp., to appear.*

[52] E. H. Georgoulis. Discontinuous Galerkin methods on shape-regular and anisotropic meshes. *D.Phil. Thesis*, University of Oxford, 2003.

[53] E. H. Georgoulis. *hp*–version interior penalty discontinuous Galerkin finite element methods on anisotropic meshes. *Int. J. Numer. Anal. Model.*, 3:52–79, 2006.

[54] E. H. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods for advection–diffusion–reaction problems on anisotropically refined meshes. *SIAM J. Sci. Comput., to appear.*

[55] E. H. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*-anisotropic meshes I: A priori error analysis. *International Journal of Computing Science and Mathematics, to appear.*

[56] E. H. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*-anisotropic meshes II: A posteriori error analysis and adaptivity. *Submitted for publication.*

[57] W. Gui and I. Babuška. The h, p and h–p versions of the finite element method in 1 dimension. Part III. The adaptive h–p version. *Numer. Math.*, 49:659–683, 1986.

[58] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. I. General principles. *Internat. J. Numer. Methods Fluids*, 32(6):725–744, 2000.

[59] P. Hansbo and C. Johnson. Adaptive streamline diffusion finite element methods for compressible flow using conservative variables. *Comput. Methods Appl. Mech. Engrg.*, 87:267–280, 1991.

[60] K. Harriman, D Gavaghan, and E Süli. The importance of adjoint consistency in the approximation of linear functionals using the discontinuous Galerkin finite element method. Technical Report 04/18, University of Oxford, 2004.

[61] K. Harriman, P. Houston, B. Senior, and E. Süli. *hp*-version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form. In S.Y. Cheng, C.-W. Shu, and T. Tang, editors, *Recent advances in scientific computing and partial differential equations (Hong Kong, 2002)*, volume 330 of *Contemp. Math.*, pages 89–119. Amer. Math. Soc., Providence, RI, 2003.

[62] R. Hartmann. Adaptive FE Methods for Conservation Equations. In Heinrich Freistühler and Gerald Warnecke, editors, *Hyperbolic Problems: theory, numerics, applications: eighth international conference in Magdeburg, February, March 2000*, volume 141 of *International series of numerical mathematics*, pages 495–503. Birkhäuser, Basel, 2001.

[63] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 24(3):979–1004, 2002.

[64] R. Hartmann and P. Houston. Goal-oriented a posteriori error estimation for multiple target functionals. In T. Y. Hou and E. Tadmor, editors, *Hyperbolic problems: theory, numerics, applications*, pages 579–588. Springer, Berlin, 2003.

[65] F. Hecht. BAMG: Bidimensional anisotropic mesh generator. Technical report, INRIA, France, 1998.

[66] V. Heuveline and R. Rannacher. A posteriori error control for finite approximations of elliptic eigenvalue problems. *Adv. Comput. Math.*, 15(1-4):107–138 (2002). 2001.

[67] V. Heuveline and R. Rannacher. Duality-based adaptivity in the *hp*-finite element method. *J. Numer. Math.*, 11(2):95–113, 2003.

[68] P. Houston, E. H. Georgoulis, and E. Hall. Adaptivity and a posteriori error estimation for DG methods on anisotropic meshes. In G. Lube and G. Rapin, editors, *Proceedings of the International Conference on Boundary and Interior Layers (BAIL) - Computational and Asymptotic Methods*. 2006.

[69] P. Houston and E. Hall. Madness. Technical report, University of Nottingham. In Preparation.

[70] P. Houston, J.A. Mackenzie, E. Süli, and G. Warnecke. A posteriori error analysis for numerical approximations of Friedrichs systems. *Numer. Math.*, 82:433–470, 1999.

[71] P. Houston, I. Perugia, and D. Schötzau. Energy norm a posteriori error estimation for mixed discontinuous Galerkin approximations of the Maxwell operator. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):499–510, 2005.

[72] P. Houston, R. Rannacher, and E. Süli. A posteriori error analysis of stabilised finite element approximations of transport problems. *Comput. Methods Appl. Mech. Engrg.*, 190(11-12):1483–1508, 2000.

[73] P. Houston, D. Schötzau, and T. Wihler. Energy norm a posteriori error estimation for mixed discontinuous Galerkin approximations of the Stokes problem. *J. Sci. Comput.*, 22(1):357–380, 2005.

[74] P. Houston, D. Schötzau, and T. P. Wihler. Energy norm a posteriori error estimation of *hp*-adaptive discontinuous Galerkin methods for elliptic problems. *Math. Models Methods Appl. Sci.* In press.

[75] P. Houston, Ch. Schwab, and E. Süli. Stabilized *hp*-finite element methods for first-order hyperbolic problems. *SIAM J. Numer. Anal.*, 37(5):1618–1643, 2000.

[76] P. Houston, Ch. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39(6):2133–2163, 2002.

[77] P. Houston, B. Senior, and E. Süli. *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity. *Internat. J. Numer. Methods Fluids*, 40(1-2):153–169, 2002.

[78] P. Houston, B. Senior, and E. Süli. Sobolev regularity estimation for *hp*–adaptive finite element methods. In F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2001*, pages 631–656. Springer, 2003.

[79] P. Houston and E. Süli. *hp*–Adaptive discontinuous Galerkin finite element methods for hyperbolic problems. *SIAM J. Sci. Comput.*, 23:1225–1251, 2001.

[80] P. Houston and E. Süli. Stabilized *hp*-finite element approximation of partial differential equations with non-negative characteristic form. *Computing*, 66:99–119, 2001.

[81] P. Houston and E. Süli. A note on the design of *hp*–adaptive finite element methods for elliptic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):229–243, 2005.

[82] Paul Houston, Ralf Hartmann, and Andre Süli. Adaptive discontinuous Galerkin finite element methods for compressible fluid flows. In M. Baines, editor, *Numerical methods for Fluid Dynamics VII. ICFD*, pages 347–353, 2001.

[83] W. Huang. Metric tensors for anisotropic mesh generation. *J. Comput. Phys.*, 204:633–665, 2005.

[84] W. Huang. Mathematical principles of anisotropic mesh adaptation. *Commun. Comput. Phys.*, 1(2):276–310, 2006.

[85] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Comput. Methods Appl. Mech. Engrg.*, 45(1-3):285–312, 1984.

[86] C. Johnson and J. Saranen. Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Math. Comp.*, 47(175):1–18, 1986.

[87] O.A. Karakashian and F. Pascal. A posteriori error estimation for a discontinuous Galerkin approximation of second order elliptic problems. *SIAM J. Numer. Anal.*, 41:2374–2399, 2003.

[88] R. Kornhuber and R. Roitzch. On adaptive grid refinement in the presence of internal and boundary layers. *Impact Comput. Sci. Engrg.*, 2:40–72, 1990.

[89] D. Kröner and M. Ohlberger. A-posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions. *Math. Comp.*, 69:25–39, 2000.

[90] G. Kunert. A local problem error estimator for anisotropic tetrahedral finite element meshes. *SIAM J. Numer. Anal.*, 39(2):668–689, 2001.

[91] G. Kunert. A posteriori error estimation for convection dominated problems on anisotropic meshes. *Math. Methods Appl. Sci.*, 26(7):589–617, 2003.

[92] G. Kunert. A posterior H^1 error estimation for a singularly perturbed reaction diffusion problem on anisotropic meshes. *IMA J. Numer. Anal.*, 25(2):408–428, 2005.

[93] M.G. Larson and T.J. Barth. A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications, Lecture Notes in Computational Science and Engineering, Vol. 11*. Springer, 2000.

[94] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000.

[95] Y. K. Lee and C. K. Lee. A new indirect ansiotropic quadrilateral mesh generation scheme with enhanced local mesh smoothing pocedures. *Internat. J. Numer. Methods Engrg.*, 58(2):277–300, 2003.

[96] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods.* SIAM, 1997.

[97] P. LeSaint and P.-A. Raviart. On a finite element method for solving the neutron transport equation. In *Mathematical aspects of finite elements in partial differential equations (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1974)*, pages 89–123. Publication No. 33. Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York, 1974.

[98] R. Li and T. Tang. Moving mesh discontinuous Galerkin method for hyperbolic conservation laws. *J. Sci. Comput.*, 27(1-3):347–363, 2006.

[99] F.-S. Lien. A pressure-based unstructured grid method for all-speed flows. *Internat. J. Numer. Methods Fluids*, 33:355–374, June 2000.

[100] T. Linßand M. Stynes. Numerical methods on Shishkin meshes for linear convection–diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 190:3527–3542, 2001.

[101] J.-L. Lions and E. Magenes. *Non-homogeneous boundary value problems and applications. Vol. III.* Springer-Verlag, New York, 1973. Translated from the French by P. Kenneth, Die Grundlehren der mathematischen Wissenschaften, Band 183.

[102] C. Mavriplis. Adaptive mesh strategies for the spectral element method. *Comput. Methods Appl. Mech. Engrg.*, 116(1-4):77–86, 1994.

[103] V. G. Maz'ja. *Sobolev spaces.* Springer Series in Soviet Mathematics. Springer-Verlag, Berlin, 1985. Translated from the Russian by T. O. Shaposhnikova.

[104] J. M. Melenk and B. I. Wohlmuth. On residual-based a posteriori error estimation in *hp*-FEM. *Adv. Comp. Math.*, 15:311–331, 2001.

[105] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*, 36:9–15, 1971. Collection of articles dedicated to Lothar Collatz on his sixtieth birthday.

[106] J. T. Oden and A. Patra. A parallel adaptive strategy for *hp* finite element computations. *Comput. Methods Appl. Mech. Engrg.*, 121(1-4):449–470, 1995.

[107] J. T. Oden, A. Patra, and Y. S. Feng. An *hp*-adaptive strategy. In A.K. Noor, editor, *Adaptive, Multilevel and Hierarchical Computational Strategies*, pages 23–46. ASME Publications, 1993.

[108] O. A. Oleĭnik and E. V. Radkevič. *Second order equations with nonnegative characteristic form.* Plenum Press, New York, 1973. Translated from the Russian by Paul C. Fife.

[109] S. Prudhomme, F. Pascal, J. T. Oden, and A. Romkes. Review of a priori error estimation for discontinuous Galerkin methods. Technical report, TICAM Report 00–27, Texas Institute for Computational and Applied Mathematics, 2000.

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

[111] M. Randrianarivony. Anisotropic finite elements for the Stokes problem: a posteriori error estimator and adaptive mesh. *J. Comput. Appl. Math.*, 169(2):255–275, 2004.

[112] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[113] B. Riviére and M. F. Wheeler. Optimal error estimates for discontinuous galerkin methods applied to linear elasticity problems. Technical report, TICAM Report 00–30, Texas Institute for Computational and Applied Mathematics, 2000.

[114] B. Rivière, M. F. Wheeler, and V. Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems, Part I. *Computational Geosciences*, 3:337–360, 1999.

[115] B. Rivière, M. F. Wheeler, and V. Girault. A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM J. Numer. Anal.*, 39(3):902–931, 2001.

[116] Y. Saad. Sparsekit: a basic tool kit for sparse matrix computations. Technical report, Computer Science Department, University of Minnesota, 1994.

[117] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.

[118] R. Schneider and P. Jimack. Toward anisotropic mesh adaptation based upon sensitivity of a posteriori estimates. Technical Report 2005.03, School of Computing, University of Leeds, 2005.

[119] Ch. Schwab. *p- and hp-finite element methods*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 1998. Theory and applications in solid and fluid mechanics.

[120] G. I. Shishkin. Approximation of solutions of singularly perturbed boundary value problems with a corner boundary layer. *Zh. Vychisl. Mat. i Mat. Fiz.*, 27(9):1360–1374, 1987 (in Russian).

[121] T. Skalický and H.-G. Roos. Anisotropic mesh refinement for problems with internal and boundary layers. *Internat. J. Numer. Methods Engrg.*, 46(11):1933–1953, 1999.

[122] P. Šolín and L. Demkowicz. Goal-oriented *hp*-adaptivity for elliptic problems. *Comput. Methods Appl. Mech. Engrg.*, 193(6-8):449–468, 2004.

[123] T. Sonar and E. Süli. A dual graph-norm refinement indicator for finite volume approximations of the Euler equations. *Numer. Math.*, 78:619–658, 1998.

[124] E. Süli. A posteriori error analysis and adaptivity for finite element approxiamtions of hyperbolic problems. In D. Kröner, M. Ohlberger, and C. Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics of Conservation Laws*, volume 5 of *Lecture Notes in Computational Science and Engineering*, pages 23–194. Springer, Berlin Heidelberg, 1998.

[125] E. Süli and P. Houston. Adaptive finite element approximation of hyperbolic problems. In T. Barth and H. Deconinck, editors, *Error Estimation and Adaptive Dis-*

cretization Methods in Computational Fluid Dynamics. Lect. Notes Comput. Sci. Engrg., volume 25, pages 269–344. Springer, 2002.

[126] E. Süli, P. Houston, and Ch. Schwab. *hp*-Finite element methods for hyperbolic problems. In J.R. Whiteman, editor, *The Mathematics of Finite Elements and Applications X*, pages 143–162. Elsevier, 2000.

[127] E. Süli, Ch. Schwab, and P. Houston. *hp*–DGFEM for partial differential equations with nonnegative characteristic form. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications, Lecture Notes in Computational Science and Engineering, Vol. 11*, pages 221–230. Springer, 2000.

[128] M. Sun. Numerical and experimental studies of shock wave interaction. *Ph.D. Thesis*, Tohoku University, 1998.

[129] B. Szabó and I. Babuška. *Finite Element Analysis*. J. Wiley & Sons, New York, 1991.

[130] T. Tang. Moving mesh methods for computational fluid dynamics. In Z.-C. Shi, Z. Chen, T. Tang, and D. Yu, editors, *Recent advances in adaptive computation*, volume 383 of *Contemp. Math.*, pages 141–173. Amer. Math. Soc., Providence, RI, 2005.

[131] L. N. Trefethen and D. Bau, III. *Numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[132] H. Triebel. *Interpolation theory, function spaces, differential operators*. Johann Ambrosius Barth, Heidelberg, second edition, 1995.

[133] J. Valenciano and R. G. Owens. An *h-p* adaptive spectral element method for Stokes flow. In J.S. Hesthaven, D. Gottlieb, and E. Turkel, editors, *Proceedings of the Fourth International Conference on Spectral and High Order Methods (ICOSAHOM 1998) (Herzliya)*, volume 33, pages 365–371, 2000.

[134] R. Verfürth. A posteriori error estimators for convection-diffusion equations. *Numer. Math.*, 80(4):641–663, 1998.

[135] R. Verfürth. A review of a posteriori error estimation techniques for elasticity problems. *Comput. Methods Appl. Mech. Engrg.*, 176(1-4):419–440, 1999.

[136] M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM J. Numer. Anal.*, 15(1):152–161, 1978.

[137] A. M. Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J. Computational Phys.*, 1:149–172, 1967.