A-Posteriori Bounds on Linear Functionals of Coercive 2^{nd} Order PDEs Using Discontinuous Galerkin Methods

by

Joseph S.H. Wong

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Febuary 2006

Author .....
Department of Mechanical Engineering
January 13, 2006

Certified by .....
Jaime Peraire
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
Anthony T. Patera
Professor of Mechanical Engineering

Certified by .....
David L. Darmofal
Professor of Aeronautics and Astronautics

Accepted by .....
Lallit Anand
Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students

A-Posteriori Bounds on Linear Functionals of Coercive 2^{nd} Order PDEs Using Discontinuous Galerkin Methods

by

Joseph S.H. Wong

Submitted to the Department of Mechanical Engineering
on January 13, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

In this thesis, we extend current capabilities in producing error bounds on the exact linear functionals of linear partial differential equations in a number of ways. Unlike previous approaches, we base our method on the Discontinuous Galerkin finite element method. For equations such as the convection-diffusion equation, the convection term is handled by the standard DG method for hyperbolic problems while the diffusion operator is discretized by the LDG scheme. This choice allows for the effective bounding of outputs associated with high Peclect number problems without resolving all of the details of the solution. In addition to the ability to manage convection dominated problems, we expand the scope of our error bounding algorithm beyond present capabilities to include saddle problems such as the incompressible Stokes equations. Apart from the aforementioned advantages, the DG discretization employed here also produces associated numerical fluxes, which make the complicated "equilibration" procedure that is often necessary in implicit *a-posteriori* algorithms, unnecessary.

Thesis Supervisor: Jaime Peraire
Title: Professor of Aeronautics and Astronautics

Acknowledgments

First and formost, I would like to thank my father for his many years of support and encouragement throughout my academic career, without which this work would not have been possible. In close connection, I wish to convey my gratitude to my former gaurdian, Eugene Wang, who guided my way through my early years in the United States. Obviously, the opportunities afforded me by my advisor, Professor Peraire, was instrumental to the successes I enjoyed in my doctoral studies. I would especially like to thank members of my doctoral committee, Professors Patera and Darmofal, for their time and guidance. Also critical to the successful completion of this undertaking were the help that Doctors Bethany R. Block, Michelle Massi and Elizabeth Loder provided, whose benefits go far beyond the confines of academic persuit. Finally, I thank all my friends for simply doing what friends do.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Numerical analysis has by now become a standard tool of engineering design. Given a physical problem of interest, an appropriate mathematical model is formulated and solved by numerical approximation. Indeed, with ever increasing computational resources, the numerical solution of mathematical problems that were once considered beyond reach are becoming routine. Better algorithms coupled with greater computational capabilities allow the designer to rely ever more heavily on numerical approximations to the solution of detailed mathematical models in engineering analysis. To fully exploit this tool, however, one must be confident that the numerical approximations are delivering solutions of sufficient accuracy. Assuming that the mathematical model describes the physical phenomena of interest adequately, we must ensure that the model is solved with the necessary resolution. *A-priori* error estimates provide insights concerning the asymptotic convergence behavior of the numerical solution but no guidance as to whether the requisite level of precision has been met by the numerical solution. The analyst is thus left with two choices; to either resort to "overkill" by employing a very fine discretization, which for many problems would result in prohibitive computational costs, or make critical decisions based on unreliable solutions. Clearly, the ability to assess the fidelity of the approximate solution is highly desirable. To this end, various *a-posteriori* error estimation techniques have been developed to quantify the error in the numerical approximation. Such algorithms fall under two primary categories: 1) explicit error estimators and 2) implicit error estimators. Both approaches can offer important insight concerning the fidelity of the approximate solution.

In general, there are two distinct objectives in *a-posteriori* error analysis: 1) to obtain local error indicators for use in mesh adaptation and/or increase the asymptotic rate of convergence and 2) to actually produce error bounds on the numerical solution. Explicit methods can fulfill the first objective while the second, more difficult goal usually requires the computationally more expensive implicit algorithms and can only be attained for much more restrictive classes of problems. In addition to differing approaches, *a-posteriori* error estimation algorithms also differ in the types of error estimates they provide. These algorithms produce error estimates for two distinct quantities; the error in some energy or L_2 norm of the solution and the error in certain functional outputs that are derived from the solution. We thus have a 2×2 matrix of *a-posteriori* error estimators, from explicit methods for the energy norm to implicit methods for functional outputs.

1.1 Explicit Methods

Given an approximate solution, we would like to extract adaptation and refinement indicators that can guide us in converging the numerical solution to the desired level of precision with the least amount of computational effort. Explicit error estimation algorithms can provide precisely that. Finite element analysis has, in fact, long involved explicit *a-posteriori* error estimation. Inexpensive estimators requiring only *local computations* were first proposed in [7, 6, 8] in the context of continuous Galerkin finite element discretization of elliptic problems. The method provides important insight regarding the quality of the finite element approximation by attempting to quantify the size of the numerical error in the energy norm. An estimate of the local contribution to the error in the energy norm is produced throughout the computational domain, from which mesh adaptation or refinement strategy may be based. A summary and review of this type of *a-posteriori* error estimators is given in [2]. The development of similar error estimators for hyperbolic problems has been slower. Nevertheless, the increase in popularity of discontinuous Galerkin methods in recent years has prompted significant research in this area. Work dealing with *a-posteriori* error analysis for nonlinear, hyperbolic conservation laws discretized with discontinuous finite element methods can be found, for example, in [46, 32, 29]. In [16], *a-posteriori* local L_2 error esti-

mates were derived for the Local Discontinuous Galerkin method applied to one-dimensional elliptic problems and in [14] local L_2 error estimates were derived for two-dimensional linear and nonlinear diffusion problems.

In many applications, however, we are less concerned with the size of the numerical error in the energy or L_2 norm at a given point in the computational domain than with the error in certain output functionals derived from the approximate solution. A simple example of this situation is given by airfoil computations; the practitioner is far more likely to be concerned with the error in the calculated values of lift and drag than the error in the L_2 norm of any quantity. An adaptation strategy based on minimizing the error in the energy or L_2 norm is not likely to produce the most efficient means of achieving a given level of precision in the output functionals of interest. This shortcoming has lead to the development of algorithms that produce error estimates on the target functionals of the solution, quantities on which actual engineering decisions will be based. Here, too, the first approaches were devised for elliptic problems solved with continuous Galerkin finite element methods. The first algorithm with such capabilities was proposed in [12], bringing the concept of *dual problems* and *dual solutions* into *a-posteriori* error analysis, which has since become standard in both explicit and implicit error estimation. Algorithms based on the same principle include those in [11, 40, 41]. In [28], an algorithm was proposed for producing *a-posteriori* error estimates on target functionals of the solution of nonlinear hyperbolic conservation laws discretized by discontinuous Galerkin finite element methods. The method also makes use of dual solutions and leads to a significantly more efficient mesh adaptation strategy than one based solely on the local L_2 error estimates. More recently, the methodology has been extended to finite-volume methods in the context of multi-dimensional compressible Euler and Navier-Stokes simulations with turbulence modeling [48, 49, 39]. The primary drawback of explicit error estimators is that while they are very useful tools for mesh adaptation and optimization and applicable to a wide range of problems, these algorithms can provide no *guarantees* of precision. All explicit error estimates contain generic unknown constants that cannot be evaluated and thus making any guarantee of absolute precision impossible.

1.2 Implicit Methods

To actually obtain absolute error bounds on the numerical solution, one would have to resort to the computationally more complex, implicit methods [1, 33, 9], to which the proposed algorithm also belongs. These methods produce error estimates that do not contain any unknown constants that render them useless as a certification tool, but instead relies on the idea of a "reference" solution. The user first chooses a conservatively refined mesh whose solution is accepted on faith as "exact". These implicit error estimators are then capable of guaranteeing that the energy norm of the discretization error as measured against the reference solution falls within the computed bounds. They are, however, much less generally applicable than the explicit methods cited earlier as they were all developed for linear, self-adjoint problems and provide bounds on only the energy norm. As pointed out earlier, we are rarely interested in the error in the energy norm but rather the error in certain output functionals upon which practical decisions will be based. We also frequently encounter nonlinear problems in practice, which these implicit algorithms cannot treat. Improvements to the cited implicit methods were first introduced in [35, 34, 37] that would allow for the bounding of general linear functional outputs derived from the numerical solution of linear coercive partial differential equations by the traditional C_0 Galerkin finite element method. The algorithm produces *uniform* error bounds on linear functional outputs with respect to that of which one would obtain from a reference mesh solution. Treatment of nonlinear and/or non-coercive problems are also possible within this new framework, although in these cases the method produces only asymptotic error bounds on output. It is important to stress that the uniform bounding property of the aforementioned implicit algorithms depends on the selection of a finite-dimensional "reference" solution and that error bounds are only guaranteed with respect to the outputs produced by this finite-dimensional solution and not the infinite-dimensional, exact solution. *True* certainty thus remains undelivered. Exploiting the complementary energy principle first proposed in the context of error estimation in [26], a further improvement to the implicit approach was made in [30, 43] by removing the need for a finite-dimensional reference solution. The new algorithm is capable of providing uniform error bounds on the linear functional outputs of linear coercive problems with respect to the

exact weak solution of the governing equations. Originally developed for scalar problems such as the Poisson equations and the advection-diffusion-reaction equation, the algorithm has since been extended to bound the linear functional outputs of multi-dimensional systems such as the governing equations of linear elasticity [36].

1.3 Proposed Algorithm

In the present work, we expand the error bounding capabilities of existing methods in a variety of ways. First, we extend the method put forth in [30, 43, 36] to cover linear functionals of linear coercive problems discretized by the Local Discontinuous Galerkin (LDG) algorithm; whose only available *a-posteriori* error estimates thus far are of the explicit L_2 energy type. We then exploit the properties of discontinuous Galerkin discretization to tackle classes of problems that have thus far eluded our grasp; namely, the high Peclet number convection-diffusion equation and saddle problems such as Stokes flow.

One drawback of the algorithm developed in [30, 43, 36] is that when applied to the high Peclet number convection-diffusion equation with under-resolved boundary layers, very poor bounds are produced. This is true even when the output functional of interest is not sensitive to the presence of boundary layers. The problem traces back to the use of the Cauchy-Schwarz inequality and the inability of the algorithm to exploit the orthogonality between error in the primal and dual solutions. By making use of the conservation properties of discontinuous Galerkin discretization, the proposed method alleviates the difficulty presented by under-resolved boundary layers through *local* refinement of the solution space at the post-processing stage of the algorithm. Effective bounds on linear functionals of the convection-diffusion equations are produced without having to resolve all the details of the solutions in either the primal or dual solutions.

Saddle problems such as Stokes flow also poses significant difficulties for existing methods. Within the framework of the method proposed in [30, 43, 36], the incompressibility constraint makes it near impossible to produce bounds on linear functionals with respect to those calculated from the the exact solution. In the present work, we exploit LDG discretization to define the Lagrangian in such a way so as to not trigger the incompressibility condition in

a manner that would cripple the ability to produce strict error bounds on linear functional outputs. We are thus able to include symmetric-indefinite systems among those whose linear functional outputs we can bound.

One of the most complicated part of implicit error bounding methods is the "Equilibration" step in the algorithm. By using discontinuous Galerkin discretization, we can actually lessen the computational overhead and simplify the error bounding procedure. This is accomplished by exploiting the numerical fluxes produced by the discontinuous finite element approximation to eliminate the complicated equilibration step that is traditionally necessary in algorithms of this class. The work here may be seen as an extension of the implicit *a-posteriori* error bounding algorithm first developed in [35, 34, 37], as the formulation of the Lagrangian and the expression of the functional of interest as the minimum of a constrained minimization statement remain the same. The work is also in many ways an extension of LDG discretization as the *a-posteriori* error bounds are produced for the LDG scheme.

The thesis proceeds as follows; in chapter 2 we briefly review the local discontinuous Galerkin method that forms the building block of our algorithm. In chapter 3 we introduce the basic algorithm and apply it to the Poisson equation. In chapter 4 we apply the algorithm to the convection-diffusion equation and develop the necessary modifications to effectively bound outputs associated with high Peclet number problems. Chapter 5 deals with the application of the proposed method to the equations of linear elasticity. Finally, in chapter 6, we take on saddle problems by applying our method to Stokes flow.

Chapter 2

Preliminaries and Discontinuous Galerkin Discretization

In this chapter, we examine the Discontinuous Galerkin (DG) discretization for both first order hyperbolic and second order elliptic problems. The discontinuous Galerkin method is a well established technique for the solution of hyperbolic conservation laws, benefitting significantly from the knowledge derived from finite volume schemes. On the other hand, the use of DG methods for the solution of elliptic problems is more recent and several algorithms have been proposed. For the work here, we employ the Local Discontinuous Galerkin scheme developed by Cockburn and Shu [22], which has emerged as one of the most popular DG implementations for elliptic problems (see, for example, [15] for a comparison of various algorithms).

Since its initial introduction by Reed and Hill [42], the discontinuous Galerkin method has gained significant popularity in the computational fluid dynamics community for the numerical solution of hyperbolic conservation laws. The more recent interest in DG methods is sparked by the demand for an algorithm capable of systematically achieving high-order accuracy on arbitrary triangulations of complex geometries while maintaining the ability to handle solution discontinuities.

Traditional finite difference/volume methods enjoy stability and accuracy even in the presence of solution discontinuities such as shocks. This is accomplished by varying the degree of the interpolating polynomial employed in the *reconstruction* step through a pro-

cedure known as *limiting*. Generally speaking, these methods involve three distinct steps. First, appropriate *numerical fluxes* are defined such that when the interpolating polynomial is piecewise constant, the numerical solution is *monotonic*, guaranteeing stability. Second, a higher-order reconstruction is defined by using linear and higher degree interpolating polynomials. In the third step, a nonlinear procedure to limit the slope of the interpolating polynomial when in the presence of solution discontinuities is employed to achieve oscillation-free solutions; see [47, 27, 44, 45], for details. These algorithms have been successfully applied to nonlinear conservation laws discretized on structured meshes. On truly unstructured meshes involving complex geometries and boundary conditions, however, high-order reconstruction cannot be easily achieved.

In the finite element framework, on the other hand, high-order accuracy is achieved by using high degree polynomials as interpolating functions within each element and arbitrary triangulations over complicated geometries pose no difficulties in obtaining the desired accuracy. Unfortunately, traditional C^0 continuous Galerkin finite element methods lack a natural mechanism to introduce upwinding into the numerical algorithm and thus additional artificial dissipation must be explicitly applied to the numerical scheme. While such algorithms have been developed, they are significantly less robust than finite volume algorithms when applied to nonlinear hyperbolic problems whose solutions contain strong discontinuities.

The discontinuous Galerkin discretization combines the stability of finite volume algorithms and the accuracy of classical C^0 finite element methods, thereby obtaining both accuracy and stability. In addition to exhibiting the same accuracy of classical finite element discretizations, the DG method also inherits its compact stencil; in direct contrast to high-order finite-volume schemes whose stencil grows with increasing order of approximation. The DG method for hyperbolic problems is uniquely defined once the numerical flux, which introduces upwinding into the algorithm, is chosen. Significant work in the area of DG research has actually evolved around the selection of a suitable interface flux; see for example, [10].

The necessity of treating convection-dominated problems with non-negligible diffusive effects has prompted renewed interest in the extension of the DG concept to elliptic problems in recent years. In the 1970s, a number of interior penalty methods were developed for

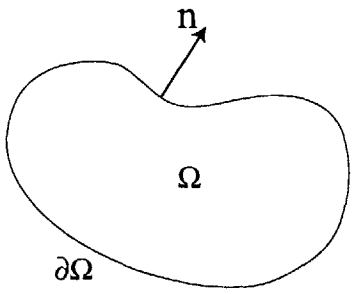
discretizing purely elliptic problems with discontinuous or nonconforming elements [3, 4]. Independent of the earlier developments, there are some recent methods designed specifically for treating the elliptic operator in convection-dominated problems which draw on the idea of *numerical fluxes* traditionally associated with purely hyperbolic equations. A popular method in this class of DG discretizations for elliptic problems is the Local Discontinuous Galerkin (LDG) method introduced in [22] and further studied in [25, 50] and [15]. A review of the available discontinuous Galerkin algorithms for elliptic problems is given in [5].

2.1 Model Problems

In this chapter, we will review the DG discretization for several model problems; namely, the linear hyperbolic equation, the Poisson equation and the convection-diffusion equation.

2.1.1 Linear Hyperbolic Equation

For the linear hyperbolic problem, we look at

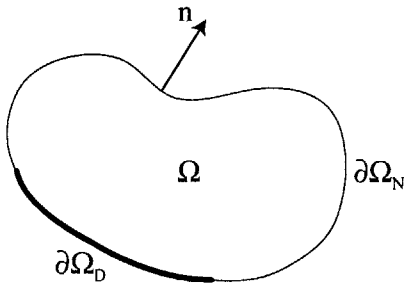


$$\begin{aligned} \nabla \cdot (\boldsymbol{a}u) - f = 0 \quad &\text{in} \quad \Omega, \\ a^- u = a^- g \quad &\text{on} \quad \partial\Omega \end{aligned} \tag{2.1}$$

where u is the solution, $g \in L_2(\partial\Omega)$ is boundary data imposed at inflow, $\boldsymbol{a} \in \Re^2$ is the velocity vector, $a^{\pm} = \frac{1}{2}(\boldsymbol{a} \cdot \mathbf{n} \pm |\boldsymbol{a} \cdot \mathbf{n}|)$ and $\mathbf{n}$ the outward unit normal; for simplicity, we also assume $\nabla \cdot \boldsymbol{a} = 0$, which eliminates any coercivity issues that may arise in the underlying PDE.

2.1.2 Poisson Equation

The Poisson equation is written as



$$\begin{aligned} -\nabla^2 u - f &= 0 \quad \text{in} \quad \Omega, \\ u &= g_D \quad \text{on} \quad \partial\Omega_D, \\ \nabla u \cdot \mathbf{n} &= g_N \quad \text{on} \quad \partial\Omega_N \end{aligned} \tag{2.2}$$

where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $f \in L_2(\Omega)$ is the given forcing and $g_D, g_N \in L_2(\partial\Omega)$ are the imposed Dirichlet and Neumann boundary data, respectively. For discontinuous Galerkin discretization, it is convenient to re-write (2.2) as a system of first order equations

$$\begin{aligned} -\nabla \mathbf{p} - f &= 0 \quad \text{in} \quad \Omega, \\ \mathbf{p} - \nabla u &= 0 \quad \text{in} \quad \Omega, \\ u &= g_D \quad \text{on} \quad \partial\Omega_D, \\ \mathbf{p} \cdot \mathbf{n} &= g_N \quad \text{on} \quad \partial\Omega_N. \end{aligned} \tag{2.3}$$

where the solution is now $\mathbf{u} = [u, \mathbf{p}]^T$.

2.1.3 Convection-Diffusion Equation

The convection-diffusion equation is written as

$$\begin{aligned} \nabla \cdot (\boldsymbol{a}u - \nabla u) - f = 0 \quad &\text{in} \quad \Omega, \\ u = g_D \quad &\text{on} \quad \partial\Omega_D, \\ \nabla u \cdot \mathbf{n} = g_N \quad &\text{on} \quad \partial\Omega_N \end{aligned} \tag{2.4}$$

with the assumptions that

$$\nabla \cdot \boldsymbol{a} = 0 \quad \text{and} \quad \boldsymbol{a} \cdot \mathbf{n}|_{\partial\Omega_N} = 0$$

which serves to eliminate any coercivity issues that may arise with the model equation. As is the case with the Poisson equation, we can re-write (2.4) as

$$\begin{aligned} \nabla \cdot (\boldsymbol{a}u - \mathbf{p}) - f = 0 \quad &\text{in} \quad \Omega, \\ \mathbf{p} - \nabla u = 0 \quad &\text{in} \quad \Omega, \\ u = g_D \quad &\text{on} \quad \partial\Omega_D, \\ \nabla u \cdot \mathbf{n} = g_N \quad &\text{on} \quad \partial\Omega_N. \end{aligned} \tag{2.5}$$

and look for the solution $\mathbf{u} = [u, \mathbf{p}]^T$.

2.2 Domain Decomposition and Function Spaces

We consider a partition T of the domain, Ω, into N_e non-overlapping subdomains such that

$$\Omega = \sum_{j=1}^{N_e} \Omega_j, \qquad \Gamma = \sum_{j=1}^{N_e} \partial\Omega_j \setminus \partial\Omega \tag{2.6}$$

where Γ is the set of all internal subdomain interfaces. We define the space V and Q and a generic function $\mathbf{v} = [v, \mathbf{q}]^T = [v, q_1, q_2, q_3]^T \in \mathbf{X}$, where

$$\begin{aligned} V &= \{v \in L_2(\Omega), V|_{\Omega_j} \in H^1(\Omega_j), \forall \Omega_j \in T\} \\ Q &= \{\mathbf{q} \in L_2(\Omega)^d, Q|_{\Omega_j} \in H(div, \Omega_j), \forall \Omega_j \in T\} \end{aligned} \tag{2.7}$$

and $\mathbf{X} = V \times Q$. We also introduce the finite-dimensional counterpart, $\mathbf{X}_h = V_h \times Q_h$, where

$$\begin{aligned} V_h &= \{v \in L_2(\Omega), V|_{\Omega_j} \in \mathcal{P}_k(\Omega_j), \forall \Omega_j \in T\} \\ Q_h &= \{\mathbf{q} \in L_2(\Omega)^d, Q|_{\Omega_j} \in \mathcal{P}_k(\Omega_j)^d, \forall \Omega_j \in T\} \end{aligned} \tag{2.8}$$

and $\mathcal{P}_k$ denotes the space of polynomials of degree k.

2.3 Notation and Operators

Given two adjacent subdomains Ω_K^+ and Ω_K^- sharing an interface $\partial\Omega_K^\pm$, we define the following interface quantities for an arbitrary scalar valued function v

$$\{v\} = \frac{v^+ + v^-}{2}, \quad [v] = v^+\mathbf{n}^+ + v^-\mathbf{n}^- \tag{2.9}$$

where $\mathbf{n}^\pm$ are the respective outward unit normals to $\partial\Omega_K^\pm$ at an arbitrary point $\mathbf{x}$ on $\partial\Omega_K^\pm$. Here $v^\pm$ are the traces of v on $\partial\Omega_K^\pm$ from the interiors of $\Omega_K^\pm$. For arbitrary vector valued functions $\mathbf{q}$, we define

$$\{\mathbf{q}\} = \frac{\mathbf{q}^+ + \mathbf{q}^-}{2}, \quad [\mathbf{q} \cdot \mathbf{n}] = \mathbf{q}^+ \cdot \mathbf{n}^+ + \mathbf{q}^- \cdot \mathbf{n}^-, \quad [[\mathbf{q}]] = \mathbf{q}^+ \otimes \mathbf{n}^+ + \mathbf{q}^- \otimes \mathbf{n}^-. \tag{2.10}$$

As before, $\mathbf{q}^\pm$ are the traces $\mathbf{q}$ on $\partial\Omega_K^\pm$ from the interiors of $\Omega_K^\pm$ and $\mathbf{q} \otimes \mathbf{n}$ denotes the matrix whose ij^{th} entry is $q_i n_j$. Note that in our definition, $[v]$ is a vector while $[\mathbf{q} \cdot \mathbf{n}]$ is a scalar.

2.4 DG for Linear Hyperbolic Equations

A description of the discontinuous Galerkin discretization for first-order hyperbolic problems can be found in many references, see for example, [21, 20, 17, 23]. Here, we briefly review it for completeness. After multiplying (2.1) with a test function $v \in V$, integrating by parts over each subdomain Ω_j and replacing the multi-valued subdomain interface flux with a single valued numerical flux we write

$$\sum_{j=1}^{N_e} \Big\{ \int_{\Omega_j} (-\nabla v \cdot \boldsymbol{a} u - v f) d\mathbf{x} + \int_{\partial\Omega_j \setminus \partial\Omega} v \hat{h} ds + \int_{\partial\Omega_j \cap \partial\Omega} v(a^+ u + a^- g) ds \Big\} = 0, \quad \forall v \in V. \tag{2.11}$$

The numerical interface flux, $\hat{h}$, is given by

$$\hat{h} = \Big\{ \boldsymbol{a}\{u\} + \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|[u] \Big\} \cdot \mathbf{n} \tag{2.12}$$

where $\mathbf{n}$ is the outward unit normal from Ω_j. We point out that with this definition, full upwind is achieved in the numerical interface flux. In compact notation, we can then write: Find $u \in V$ such that

$$a(v,u) = l(v), \quad \forall v \in V \tag{2.13}$$

where $a : X \times X \mapsto \Re$ and $l : X \mapsto \Re$ are given by

$$a(v,w) = -\int_\Omega \nabla v \cdot \boldsymbol{a} w d\mathbf{x} + \int_\Gamma [v] \cdot (\boldsymbol{a}\{w\} + \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|[w]) ds + \int_{\partial\Omega} v a^+ w ds \tag{2.14}$$

$$l(v) = \int_\Omega v f d\mathbf{x} - \int_{\partial\Omega} v a^- g ds. \tag{2.15}$$

Note that this is a variational formulation of the infinite-dimensional continuous problem. We point out that in this formulation, boundary conditions are naturally incorporated into the subdomain interface fluxes; the boundary interface is treated no differently than internal subdomain interfaces with relevant boundary data incorporated into the righthand side of

the equation. From (2.14), we can set $w = v$ to obtain the following expression for $a(v, v)$

$$a(v,v) = -\int_\Omega \nabla v \cdot \boldsymbol{a} v d\mathbf{x} + \int_\Gamma [v] \cdot (\boldsymbol{a}\{v\} + \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|[v])ds + \int_{\partial\Omega} (a^+ v^2 - \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|v^2)ds \quad (2.16)$$

which after some simplification results in

$$a(v,v) = \int_\Gamma \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|[v]^2 ds + \int_{\partial\Omega} \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|v^2 ds \quad (2.17)$$

It is clear that for stability we require the $a(v, v)$ to be strictly positive ($a(v, v) \geq 0, \forall v$), which ensures coercivity. We point out that if we substitute test function v in equation (2.13) by the exact solution u, we obtain the following energy equality

$$\begin{aligned} a(u,u) - l(u) &= -\int_\Omega u f d\mathbf{x} + \int_\Gamma \frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|[u]^2 ds \\ &\quad + \int_{\partial\Omega} (\frac{1}{2}|\boldsymbol{a} \cdot \mathbf{n}|u^2 + u a^- g)ds = 0. \end{aligned} \quad (2.18)$$

This expression contains linear and quadratic terms in u, with the latter being strictly positive, a property that we will exploit later on. We note that for the exact solution, u, the interface jump, $[u]$ would be zero.

2.4.1 Discrete Problem

Statement (2.13) together with (2.14) and (2.15) is the point of departure for DG discretization. We formulate the following discrete problem: Find $u_h \in V_h$ such that

$$a(v, u_h) = l(v), \quad \forall v \in V_h \quad (2.19)$$

or

$$\begin{aligned} &\sum_{j=1}^{N_e} \Big\{ \int_{\Omega_j} (-\nabla v \cdot \boldsymbol{a} u_h - v f)d\mathbf{x} + \int_{\partial\Omega_j \setminus \partial\Omega} v \hat{h} ds \\ &+ \int_{\partial\Omega_j \cap \partial\Omega} v(a^+ u_h + a^- g)ds \Big\} = 0, \quad \forall v \in V_h. \end{aligned} \quad (2.20)$$

2.4.2 Convergence

We now examine the convergence behavior of the DG algorithm for linear first-order equations. The order of convergence for the discontinuous Galerkin method for scalar hyperbolic equations for general triangulations was shown in [31] to be of order $k + 1/2$ in the L_2 norm when polynomials of degree k are used in the numerical approximation. The corresponding *a-priori* result is given by

$$\|e\|_{L_2(\Omega)} \leq C_1 h^{k+\frac{1}{2}} |\boldsymbol{a}| |u|_{H^{k+1}(\Omega)} \tag{2.21}$$

for $e = u - u_h$ and constant C_1 depending on k but independent of u. This result was proven to be sharp in [38]; however, (2.21) must be seen as a conservative estimate since in practice, one routinely obtains order $k + 1$ convergence in most applications. Indeed, for sufficiently smooth solutions, Cockburn proved in [24] that one obtains order $k + 1$ convergence with the following error estimate

$$\|e\|_{L_2(\Omega)} \leq C_2 h^{k+1} |\boldsymbol{a}| |u|_{H^{k+2}(\Omega)} \tag{2.22}$$

where again, C_2 depends on k but is independent of u.

To illustrate this point, we now test the convergence behavior of the DG discretization for linear hyperbolic equations in two dimensions with a simple example. We solve

$$\nabla \cdot (\boldsymbol{a}u) - f = 0 \quad \text{in} \quad \Omega = [0,1] \times [0,1]$$

using P_1 elements with $a_1 = 1, a_2 = 0$ and $f = -a_1 x_2 (x_2 - 1)$ such that the exact solution is $u = (1 - x_1) x_2 (x_2 - 1)$. The inflow boundary condition is handled weakly through the boundary flux. The computational mesh and solution contours for the $h = 1/32$ computation are shown in figures (2-1) and (2-2). Table (2.1) shows the L_2 grid convergence results for this test problem, verifying that the optimal rate of h^{k+1} is indeed obtained.

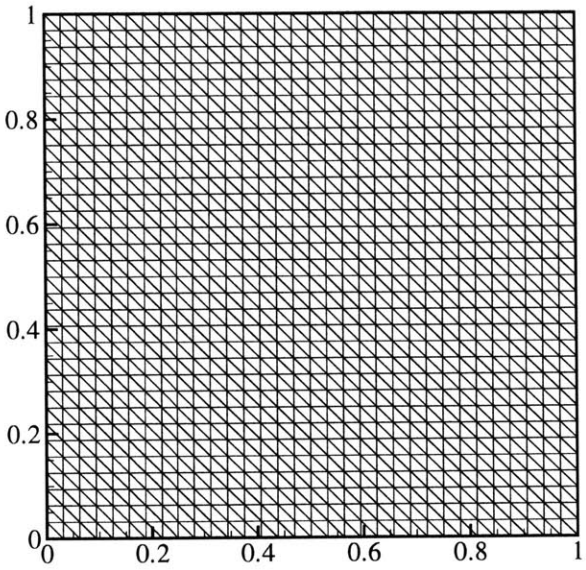


Figure 2-1: $h = 1/32$ Computational Mesh

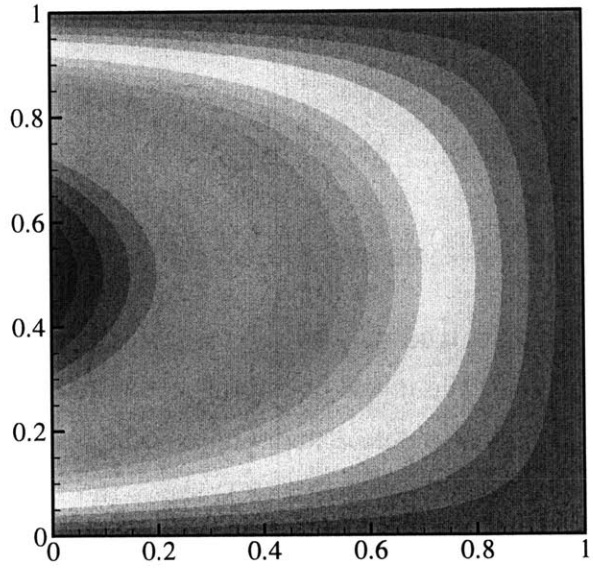


Figure 2-2: $h = 1/32$ LDG solution

h	$\|u - u_h\|_2$	Order
1/8	3.472^{-3}	-
1/16	8.631^{-4}	2.00
1/32	2.151^{-4}	2.00
1/64	5.369^{-5}	2.00

Table 2.1: L_2 Errors and Orders of Convergence

2.5 DG Discretization for Elliptic Problems: The LDG Algorithm

We now review the implementation of discontinuous Galerkin discretization for second-order, elliptic problems. As pointed out earlier, there is a plethora of proposed algorithms for the DG discretization of elliptic equations. For the work here, we have selected the LDG scheme based on its favorable stability and accuracy properties [15, 25]. We consider discretizing the Poisson equation, given by (2.2). The LDG discretization involves first introducing auxiliary variables for the solution gradient and re-writing (2.2) as a system of first-order equations such that we arrive at (2.3). We then multiply (2.3) by arbitrary test functions $\mathbf{v} \in \mathbf{X} \equiv V \times Q$ and integrate by parts over each subdomain Ω_j. Replacing the multi-valued inter-subdomain fluxes with appropriate numerical interface fluxes we obtain

$$\begin{aligned}
&\sum_{j=1}^{N_e}\left\{\int_{\Omega_j} \nabla v \cdot \mathbf{p} - vf d\mathbf{x} - \int_{\partial\Omega_j} v\hat{p} ds\right\} = 0 \\
&\sum_{j=1}^{N_e}\left\{\int_{\Omega_j} (\mathbf{q}\cdot\mathbf{p} + \nabla\cdot\mathbf{q}u) d\mathbf{x} - \int_{\partial\Omega_j} \mathbf{q}\cdot\mathbf{n}\hat{u} ds\right\} = 0, \quad \forall \mathbf{v} \in \mathbf{X}
\end{aligned} \tag{2.23}$$

Here $\mathbf{v} = [v, \mathbf{q}]^T, \mathbf{q} = [q_1, q_2, q_3]^T, \mathbf{u} = [u, \mathbf{p}]^T, \mathbf{p} = [p_1, p_2, p_3]^T$. The numerical fluxes $\hat{u}$ and $\hat{p}$ are given by

$$\begin{aligned}
\hat{p} &= (\{\mathbf{p}\} + \boldsymbol{C}_{11}\cdot[u] + \boldsymbol{C}_{12}\cdot[[\mathbf{p}]])\cdot\mathbf{n} \\
\hat{u} &= \{u\} - \boldsymbol{C}_{21}\cdot[u] + C_{22}[\mathbf{p}\cdot\mathbf{n}].
\end{aligned} \tag{2.24}$$

For stability [22], it is desirable for the matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & C_{22} \end{bmatrix} \tag{2.25}$$

to be symmetric and positive definite. This would require C_{22} to be nonzero and we would lose the ability to eliminate the auxiliary variables $\mathbf{p}$ at the elemental level in the resulting dicretized scheme which results in a much more costly algorithm. We therefore only demand $\boldsymbol{C}$ to be positive semi-definite with C_{22} set to zero and take it to be

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ -\boldsymbol{C}_{12} & 0 \end{bmatrix} \tag{2.26}$$

with $C_{11} \geq 0$. For interior interfaces, we follow the choice given in [22] and take

$$\boldsymbol{C}_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{C}_{12} = \beta \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \quad \boldsymbol{C}_{21} = \boldsymbol{C}_{12}^T \tag{2.27}$$

where

$$\beta = \frac{1}{2}\text{sign}(\mathbf{b} \cdot \mathbf{n}). \tag{2.28}$$

Here $\mathbf{b}$ is a fixed arbitrary non-zero vector. This choice for β is motivated by the desire to avoid selecting all interface fluxes associated with a given subdomain from the interior trace (see [25]).

The above choices of numerical fluxes are not always compatible with the imposed boundary data at boundary interfaces. To properly account for the imposed boundary conditions, we use

$$\begin{aligned} \hat{p} &= \{\mathbf{p} - \alpha(u - g_D)\mathbf{n}\} \cdot \mathbf{n} \\ \hat{u} &= g_D \end{aligned} \tag{2.29}$$

for Dirichlet boundaries, where α is a penalization parameter for the enforcement of Dirichlet

boundary conditions. For Neumann boundaries, we use

$$\begin{aligned} \hat{p} &= g_N \\ \hat{u} &= u. \end{aligned} \tag{2.30}$$

In compact notation, we write the following problem statement: Find $\mathbf{u} \in \mathbf{X}$ such that

$$a(\mathbf{v}, \mathbf{u}) = l(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{X} \tag{2.31}$$

where $a : \mathbf{X} \times \mathbf{X} \mapsto \Re$ and $l : \mathbf{X} \mapsto \Re$ are given by

$$\begin{aligned} a(\mathbf{v}, \mathbf{w}) &= \int_\Omega \left\{ \boxed{\nabla v \cdot \mathbf{r}} + \boxed{\boxed{\mathbf{q} \cdot \mathbf{r} + \nabla \cdot \mathbf{q} w}} \right\} d\mathbf{x} \\ &\quad - \int_\Gamma \left\{ \boxed{[v] \cdot (\{\mathbf{r}\} + \boldsymbol{C}_{12} \cdot [[\mathbf{r}]])} + \boxed{\boxed{[\mathbf{q} \cdot \mathbf{n}](\{w\} - \boldsymbol{C}_{21} \cdot [w])}} \right\} ds \\ &\quad - \int_{\partial\Omega_D} \boxed{v(\mathbf{r} - \alpha w \mathbf{n}) \cdot \mathbf{n}} ds - \int_{\partial\Omega_N} \boxed{\boxed{\mathbf{q} \cdot \mathbf{n} w}} ds \qquad (2.32) \\ l(\mathbf{v}) &= \int_\Omega \boxed{vf} d\mathbf{x} + \int_{\partial\Omega_D} (\boxed{\alpha v} + \boxed{\boxed{\mathbf{q} \cdot \mathbf{n}}}) g_D ds + \int_{\partial\Omega_N} \boxed{v g_N} ds. \qquad (2.33) \end{aligned}$$

Here $\mathbf{v} = [v, \mathbf{q}]^T$ and $\mathbf{w} = [w, \mathbf{r}]^T \in \mathbf{X}$. In the above expression, the boxed-quantities are associated with the conservation law while the double-boxed quantities are connected with the definition of the auxiliary variables. Equation (2.31) together with (2.32) and (2.33) defines the variational formulation of the infinite-dimensional problem given in (2.2). From (2.32), setting $\mathbf{w} = \mathbf{v}$ results in the following expression for $a(\mathbf{v}, \mathbf{v})$

$$\begin{aligned} a(\mathbf{v}, \mathbf{v}) &= \int_\Omega \left\{ \boxed{\nabla v \cdot \mathbf{q}} + \boxed{\boxed{\mathbf{q} \cdot \mathbf{q} + \nabla \cdot \mathbf{q} v}} \right\} d\mathbf{x} + \int_{\partial\Omega_D} \boxed{\alpha v^2} ds \\ &\quad - \int_\Gamma \left\{ \boxed{[v] \cdot (\{\mathbf{q}\} + \boldsymbol{C}_{12} \cdot [[\mathbf{q}]])} + \boxed{\boxed{[\mathbf{q} \cdot \mathbf{n}](\{v\} - \boldsymbol{C}_{21} \cdot [v])}} \right\} ds \qquad (2.34) \end{aligned}$$

which after some simplification may be written as

$$a(\mathbf{v}, \mathbf{v}) = \int_\Omega \mathbf{q} \cdot \mathbf{q} d\mathbf{x} + \int_{\partial\Omega_D} \alpha v^2 ds. \tag{2.35}$$

We see that $(a(\mathbf{v},\mathbf{v}) \geq 0, \forall \mathbf{v} \in \mathbf{X})$, a result which proves coercivity. We point out for later use that the exact solution satisfies the global energy equality, $a(\mathbf{u},\mathbf{u}) - l(\mathbf{u}) = 0$ (obtained form setting $\mathbf{v} = \mathbf{u}$ in (2.31)). In expanded form, this is given by

$$\int_\Omega \mathbf{p}\cdot\mathbf{p}d\mathbf{x} + \int_{\partial\Omega} \alpha u^2 ds = \int_\Omega uf d\mathbf{x} + \int_{\partial\Omega_D} (\alpha u + \mathbf{p}\cdot\mathbf{n}) g_D ds + \int_{\partial\Omega_N} u g_N ds \tag{2.36}$$

which again involves linear and quadratic terms in $\mathbf{u}$, with the latter being strictly positive.

2.5.1 Discrete Problem

Statement (2.31) together with (2.32) and (2.33) is the point of departure for LDG discretization. We look for a discrete solution $\mathbf{u}_h \in \mathbf{X}_h$ such that

$$a(\mathbf{v},\mathbf{u}_h) = l(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{X}_h \tag{2.37}$$

or

$$\begin{aligned}
&\sum_{j=1}^{N_e} \left\{ \int_{\Omega_j} (\nabla v \cdot \mathbf{p}_h - f) d\mathbf{x} - \int_{\partial\Omega_j} v\hat{p} ds \right\} = 0 \\
&\sum_{j=1}^{N_e} \left\{ \int_{\Omega_j} (\mathbf{q}\cdot\mathbf{p}_h + \nabla\cdot\mathbf{q}u_h) d\mathbf{x} - \int_{\partial\Omega_j} \mathbf{q}\cdot\mathbf{n}\hat{u} ds \right\} = 0, \quad \forall \mathbf{v} \in \mathbf{X}_h.
\end{aligned} \tag{2.38}$$

This forms a system of equations whose unknowns are u_h and $\mathbf{p}_h$. As pointed out earlier, it is possible with our choice of numerical fluxes, to eliminate $\mathbf{p}_h$ locally, at elemental level. This results in a set of equations for u_h whose coefficient matrix is symmetric and positive definite.

2.5.2 *A-priori* Error Estimate

The LDG algorithm can be shown to converge at order $k+1$ for the solution in the L_2 norm and order k for the solution gradient when polynomials of degree k are used in the numerical approximation for general triangulations [22], provided of course, the solution is sufficiently smooth. On cartesian meshes, however, Cockburn proved in [25] that the solution gradient

super-converges at order $k + 1/2$ in the L_2 norm.

2.5.3 Example: Convergence in 2-D

We now examine the convergence behavior of LDG discretization for the 2-dimensional Poisson equation using piecewise linear elements. For this test case, we solve

$$\begin{aligned} -\nabla^2 u - f = 0, \quad f = 0 \quad \text{on} \quad \Omega = [0,1] \times [0,1] \\ u_x(0,y) = u_x(1,y) = u_y(x,0) = 0, u(x,1) = \cos(\pi x) \end{aligned} \tag{2.39}$$

for which the exact solution is $u = \frac{\cosh(\pi y)}{\cosh(\pi)} \cos(\pi x)$. The computational mesh and the numerical solution are shown in figures (2-3) and (2-4) while the grid convergence results are displayed in table (2.2).

h	$\|u - u_h\|_2$	Order	$\|\mathbf{p} - \mathbf{p}_h\|_2$	Order
1/8	3.370^{-3}	-	9.993^{-2}	-
1/16	8.717^{-4}	1.95	4.918^{-2}	1.02
1/32	2.214^{-4}	1.98	2.445^{-2}	1.01
1/64	5.577^{-5}	1.99	1.220^{-2}	1.00

Table 2.2: L_2 Errors and Orders of Convergence

The grid convergence results are in line with the *a-priori* estimates of order $k + 1$ and k convergence rate for the L_2 norm of the error in the solution and the solution gradient, respectively.

2.6 DG Implementation for the Convection-Diffusion Equation

In this section, we review the discontinuous Galerkin method for the convection-diffusion equation. The model problem is given by (2.4). This equation is different from that of the pure diffusion case only in the presence of the convection term, which is easily discretized by the discontinuous Galerkin algorithm. We treat the convection-diffusion equation by

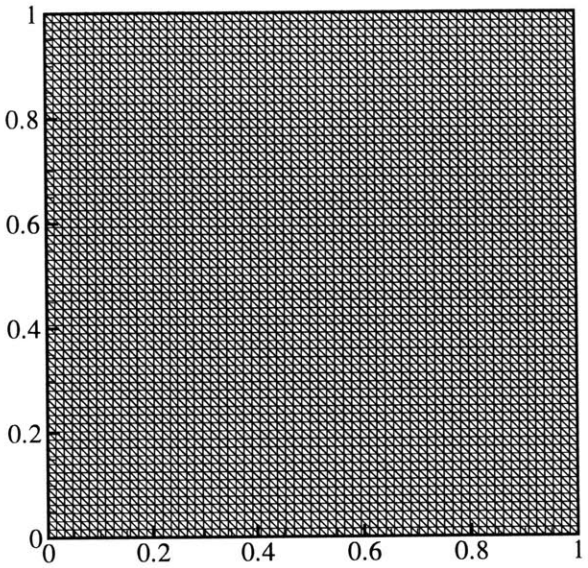


Figure 2-3: $h = 1/64$ Computational Mesh

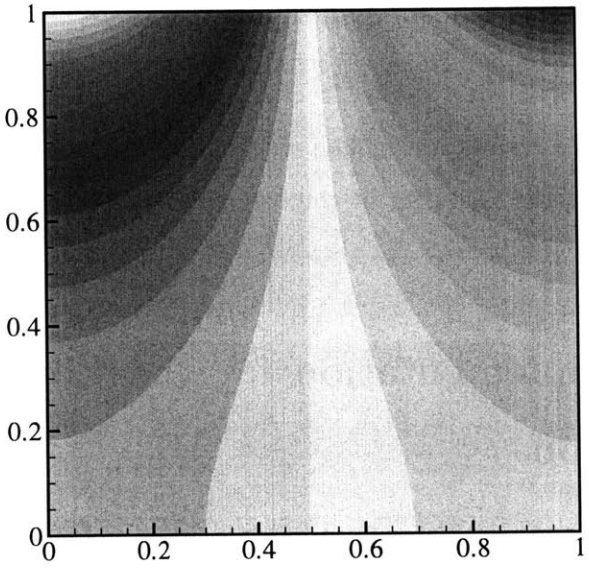


Figure 2-4: $h = 1/64$ LDG solution

applying the DG algorithm for first-order, hyperbolic problems to the convection term and the LDG discretization to the diffusion operator. After re-writing the governing equation as a system of first-order equations as shown in (2.5), we multiply (2.5) with arbitrary test functions v and $\mathbf{q}$ and integrate by parts over each subdomain Ω_j. Replacing the multi-valued inter-subdomain fluxes by appropriate numerical interface fluxes leads to

$$\sum_{j=1}^{N_e}\left\{\int_{\Omega_j}(\nabla v\cdot(-\boldsymbol{a}u+\mathbf{p})-vf)d\mathbf{x}+\int_{\partial\Omega_j}v(\hat{h}-\hat{p})ds\right\}=0$$
$$\sum_{j=1}^{N_e}\left\{\int_{\Omega_j}(\mathbf{q}\cdot\mathbf{p}+\nabla\cdot\mathbf{q}u)d\mathbf{x}-\int_{\partial\Omega_j}\mathbf{q}\cdot\mathbf{n}\hat{u}ds\right\}=0,\quad\forall\mathbf{v}\in\mathbf{X}. \tag{2.40}$$

The variational continuous problem is then formulated as: Find $\mathbf{u}\in\mathbf{X}$ such that

$$a(\mathbf{v},\mathbf{u})-l(\mathbf{v})=0,\quad\forall\mathbf{v}\in\mathbf{X}$$

where $a:\mathbf{X}\times\mathbf{X}\mapsto\Re$ and $l:\mathbf{X}\mapsto\Re$ are given by

$$\begin{aligned}
a(\mathbf{v},\mathbf{w}) &= \int_\Omega\left\{\boxed{\nabla v\cdot(-\boldsymbol{a}w+\mathbf{r})}+\boxed{\boxed{\mathbf{q}\cdot\mathbf{r}+\nabla\cdot\mathbf{q}w}}\right\}d\mathbf{x}\\
&+ \int_\Gamma\left\{\boxed{[v]\cdot(\boldsymbol{a}\{w\}+\frac{1}{2}|\boldsymbol{a}\cdot\mathbf{n}|[w]-\{\mathbf{r}\}-\boldsymbol{C}_{12}\cdot[[\mathbf{r}]])}-\boxed{\boxed{[\mathbf{q}\cdot\mathbf{n}](\{w\}-\boldsymbol{C}_{21}\cdot[w])}}\right\}ds\\
&- \int_{\partial\Omega_D}\boxed{v(\mathbf{r}\cdot\mathbf{n}-\alpha w-a^+w)}ds-\int_{\partial\Omega_N}\boxed{\boxed{\mathbf{q}\cdot\mathbf{n}w}}ds \qquad (2.41)\\
l(\mathbf{v}) &= \int_\Omega\boxed{vf}d\mathbf{x}+\int_{\partial\Omega_D}(\boxed{\alpha v-va^-}+\boxed{\boxed{\mathbf{q}\cdot\mathbf{n}}})g_Dds+\int_{\partial\Omega_N}\boxed{vg_N}ds. \qquad (2.42)
\end{aligned}$$

As before, boxed-quantities are associated with the conservation law while the double-boxed quantities are connected with the definition of the auxiliary variables. For the convection-diffusion equation, $a(\mathbf{v},\mathbf{v})$ is given by

$$a(\mathbf{v},\mathbf{v})=\int_\Omega\mathbf{q}\cdot\mathbf{q}d\mathbf{x}+\int_\Gamma\frac{1}{2}|\boldsymbol{a}\cdot\mathbf{n}|[v]^2ds+\int_{\partial\Omega_D}(a^+v^2-\frac{1}{2}\boldsymbol{a}\cdot\mathbf{n}v^2+\alpha v^2)ds. \tag{2.43}$$

We see again, that $(a(\mathbf{v},\mathbf{v}) \geq 0, \forall \mathbf{v} \in \mathbf{X})$; the coercivity condition we need. Globally, we have the following energy equality $a(\mathbf{u},\mathbf{u}) = l(\mathbf{u})$, which becomes

$$\int_\Omega \mathbf{p}\cdot\mathbf{p}d\mathbf{x} + \int_\Gamma \frac{1}{2}|\boldsymbol{a}\cdot\mathbf{n}|[u]^2 ds + \int_{\partial\Omega_D}(a^+u^2 - \frac{1}{2}\boldsymbol{a}\cdot\mathbf{n}u^2 + \alpha u^2)ds = \int_\Omega uf d\mathbf{x} + \int_{\partial\Omega_D}(\alpha u + \mathbf{p}\cdot\mathbf{n})g_D ds + \int_{\partial\Omega_N} ug_N ds. \tag{2.44}$$

2.6.1 Discrete Problem

Statement (2.41) together with (2.41) and (2.42) is the point of departure for LDG discretization. Consider the solution $\mathbf{u}_h \in \mathbf{X}_h$ such that

$$a(\mathbf{v},\mathbf{u}_h) = l(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{X}_h \tag{2.45}$$

or

$$\sum_{j=1}^{N_e}\left\{\int_{\Omega_j}(\nabla v\cdot(-\boldsymbol{a}u_h + \mathbf{p}_h) - vf)d\mathbf{x} + \int_{\partial\Omega_j} v(\hat{h} - \hat{p})ds\right\} = 0$$
$$\sum_{j=1}^{N_e}\left\{\int_{\Omega_j}(\mathbf{q}\cdot\mathbf{p}_h + \nabla\cdot\mathbf{q}u_h)d\mathbf{x} - \int_{\partial\Omega_j}\mathbf{q}\cdot\mathbf{n}\hat{u}ds\right\} = 0, \quad \forall \mathbf{v} \in \mathbf{X}_h. \tag{2.46}$$

As is the case with the Poisson equation, the auxiliary variables, $\mathbf{p}_h$, can be eliminated at elemental level. The presence of the convection term, however, means that the set of resulting equations has a non-symmetric coefficient matrix.

Chapter 3

Bounds for Linear Functional Outputs: Scalar Symmetric Case

In this chapter, we develop the basic framework for the algorithm that produces upper and lower bounds for linear functional outputs of coercive 2^{nd}-order elliptic partial differential equations with respect to the outputs obtained from the exact solution. Unlike [30, 43], however, we base our method on the Local Discontinuous Galerkin scheme, which has the advantage of not requiring the complicated "equilibration" procedure that is necessary in existing implicit *a-posteriori* error bounding algorithms. The numerical fluxes that are naturally produced by the Discontinuous Galerkin discretization to ensure inter-subdomain coupling also serve as "equilibrated" fluxes in the context of implicit error-bounding. The algorithm is in some ways an extension of those presented in [35, 34, 37] as they are both based on formulating the output as the solution to a constrained minimization problem with an augmented Lagrangian. The objective function involves a "quadratic" energy term that derives from the coercivity of the underlying governing partial differential equations plus the linear functional output of interest. The equilibrium governing equations enter as constraints to the minimization. The algorithm may also be seen as an error bounding algorithm for Discontinuous Galerkin methods as it is specifically built for and based on the properties of these schemes.

3.1 Problem Definition

We take as model problem the Poisson equation, given by (2.2). We are interested in obtaining upper and lower bounds for linear functional outputs of the form

$$S = \int_\Omega u f_0^V d\mathbf{x} + \int_{\partial\Omega_D} \nabla u \cdot \mathbf{n} g_0 ds + \int_{\partial\Omega_N} u f_0^S ds, \quad f_0^V \in L_2(\Omega) \tag{3.1}$$

which are functionals of the solution to (2.2). Here, $f_0^V \in L_2(\Omega)$ and $g_0, f_0^S \in L_2(\partial\Omega)$ are given functions.

We recall from (2.32) and (2.33) that with the definitions $\mathbf{v} = [v, \mathbf{q}]^T, \mathbf{q} = [q_1, q_2, q_3]^T, \mathbf{u} = [u, \mathbf{p}]^T, \mathbf{p} = [p_1, p_2, p_3]^T$, (2.3) may be written in variational form as: Find $\mathbf{u} \in \mathbf{X}$ such that

$$a(\mathbf{v}, \mathbf{u}) = l(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{X}. \tag{3.2}$$

The output may then be written as $S = l^0(\mathbf{u})$ with

$$l^0(\mathbf{v}) = \int_\Omega v f_0^V d\mathbf{x} + \int_{\partial\Omega_D} \mathbf{q} \cdot \mathbf{n} g_0 ds + \int_{\partial\Omega_N} v f_0^S ds. \tag{3.3}$$

3.2 Lower Bound Formulation: The Lagrangian

We now proceed to develop the algorithm for the computation of lower bounds for S. As we shall see later on, upper bounds may be obtained in an analogous manner with only a slight modification of the algorithm. Following the methodology of Patera *et al.* [35, 34], we introduce the Lagrangian, $\mathcal{L} : \mathbf{X} \times \mathbf{X} \mapsto \Re$, as

$$\mathcal{L}(\mathbf{v}, \Psi_\mathbf{v}) = \kappa(a(\mathbf{v}, \mathbf{v}) - l(\mathbf{v})) + l^0(\mathbf{v}) + a(\Psi_\mathbf{v}, \mathbf{v}) - l(\Psi_\mathbf{v}), \quad \forall \mathbf{v}, \Psi_\mathbf{v} \in \mathbf{X} \tag{3.4}$$

where $\Psi_\mathbf{v} = [\psi_v, \psi_\mathbf{q}]^T$ and $\kappa \geq 0$ is an optimization parameter. We recall from chapter 2 that for the Poisson equation, $a(\mathbf{v}, \mathbf{v}) - l(\mathbf{v})$ is given by

$$
\begin{aligned}
a(\mathbf{v},\mathbf{v}) - l(\mathbf{v}) &= \int_{\Omega} (\mathbf{q}\cdot\mathbf{q} - vf)d\mathbf{x} + \int_{\partial\Omega_D} (\alpha v(v-g_D) - \mathbf{q}\cdot\mathbf{n}g_D)ds \\
&- \int_{\partial\Omega_N} vg_N ds. \qquad (3.5)
\end{aligned}
$$

where α is a penalization parameter to enforce Dirichlet boundary conditions and that $a(\mathbf{u},\mathbf{u}) - l(\mathbf{u}) = 0$. To refresh our memory, the expression for $a(\mathbf{v},\mathbf{w}) - l(\mathbf{v})$ is given by

$$
\begin{aligned}
a(\mathbf{v},\mathbf{w}) - l(\mathbf{v}) &= \sum_{j=1}^{N_e} \Big\{ \int_{\Omega_j} \nabla v \cdot \mathbf{r} d\mathbf{x} - \int_{\partial\Omega_j} v\hat{r} ds \\
&+ \int_{\Omega_j} (\mathbf{q}\cdot\mathbf{r} + \nabla\cdot\mathbf{q}w)d\mathbf{x} - \int_{\partial\Omega_j} \mathbf{q}\cdot\mathbf{n}\hat{w} ds \Big\}.
\end{aligned}
$$

The output, $S = l^0(\mathbf{u})$, may then be expressed in terms of the following constrained minimization statement

$$
S = l^0(\mathbf{u}) = \inf_{\mathbf{v}\in\mathbf{X}} \sup_{\mathbf{\Psi_v}\in\mathbf{X}} \mathcal{L}(\mathbf{v},\mathbf{\Psi_v}). \qquad (3.6)
$$

We see this since

$$
\sup_{\mathbf{\Psi_v}\in\mathbf{X}} \mathcal{L}(\mathbf{v},\mathbf{\Psi_v}) = \begin{cases} l^0(\mathbf{u}) & \text{if} \quad a(\mathbf{\Psi_v},\mathbf{v}) - l(\mathbf{\Psi_v}) = 0, \quad \forall \mathbf{\Psi_v} \\ \infty & \text{if} \quad a(\mathbf{\Psi_v},\mathbf{v}) - l(\mathbf{\Psi_v}) \neq 0, \quad \forall \mathbf{\Psi_v}. \end{cases} \qquad (3.7)
$$

The maximization over $\mathbf{\Psi_v}$ forces the minimization over $\mathbf{X}$ to select the $\mathbf{v}$ that satisfies the governing equation; the minimizer $\mathbf{u}$. Since $a(\mathbf{u},\mathbf{u}) - l(\mathbf{u}) = 0$ and $a(\mathbf{\Psi_v},\mathbf{u}) - l(\mathbf{\Psi_v}) = 0$, we obtain $\mathcal{L} = l^0(\mathbf{u})$. Furthermore, from duality (see, for example [13]), we can claim that

$$
S = l^0(\mathbf{u}) = \inf_{\mathbf{v}\in\mathbf{X}} \sup_{\mathbf{\Psi_v}\in\mathbf{X}} \mathcal{L}(\mathbf{v},\mathbf{\Psi_v}) = \boxed{\sup_{\mathbf{\Psi_v}\in\mathbf{X}} \inf_{\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v},\mathbf{\Psi_v})}. \qquad (3.8)
$$

The last equality requires $\mathcal{L}(\mathbf{v}, \Psi_{\mathbf{v}})$ to be sufficiently regular, a condition which is satisfied in our case. From the above relations it follows that

$$S = \sup_{\Psi_{\mathbf{v}} \in \mathbf{X}} \inf_{\mathbf{v} \in \mathbf{X}} \mathcal{L}(\mathbf{v}, \Psi_{\mathbf{v}}) \geq \boxed{\inf_{\mathbf{v} \in \mathbf{X}} \mathcal{L}(\mathbf{v}, \Psi_{\mathbf{v}}),} \quad \forall \Psi_{\mathbf{v}} \in \mathbf{X}. \tag{3.9}$$

We point out that the boxed quantity in equation (3.9) is, in fact, and expression for a lower bound for the output, S. We note that this is true for *any* $\Psi_{\mathbf{v}}$.

3.2.1 Lower Bound Evaluation: A Simple Example

Before we proceed to derive an expression for the lower bound for S, we digress momentarily to look at the following example. We define a function, $Z = Z(y_1, y_2)$, given by

$$Z(y_1, y_2) = A_1 y_1^2 + A_2 y_1 + B_1 y_2 + B_2 \tag{3.10}$$

for $\mathbf{y} = [y_1, y_2]^T \in \Re$ and where $\mathbf{A} = [A_1, A_2]^T, \mathbf{B} = [B_1, B_2]^T \in \Re$ are arbitrary constants. We then perform the following minimization

$$Z^* = \min_{y_1, y_2 \in \Re} Z(y_1, y_2) \tag{3.11}$$

and obtain the result

$$\boxed{Z^* = \begin{cases} \text{case I} & -\frac{A_2^2}{4A_1} + B_2 \quad \text{if } A_1 > 0, B_1 = 0, \\ \text{case II} & -\infty \qquad\quad \text{otherwise.} \end{cases}} \tag{3.12}$$

From (3.12), we see that when a function is linear and quadratic in its arguments, the unconstrained minimum is either a constant or unbounded from below, depending on the coefficients of its polynomials. This simple problem is, in fact, an analog of (3.9) in that

both Z and $\mathcal{L}$ are polynomial functions of their arguments. Here $\mathbf{y}$ plays the same role as $\mathbf{v}$ in (3.9) while coefficients $\mathbf{A}, \mathbf{B}$ are the counterparts to $\mathbf{\Psi_v}$. The strategy for the remainder of this chapter and in the chapters is then given as follows

1. Define the Lagrangian in a form analogous to (3.12)
2. Choose $\mathbf{\Psi_v}$ to ensure that we obtain case I in (3.12)

3.2.2 Lower Bound Evaluation: S^-

To evaluate a lower bound for S, we attempt to minimize $\mathcal{L}$ over all $\mathbf{v} = [v, \mathbf{q}]^T$. First, setting the variation of $\mathcal{L}$ with respect to $\mathbf{q}$ equal to zero results in

$$\int_\Omega (2\kappa\mathbf{q} + \psi_\mathbf{q} + \nabla\psi_v) \cdot \delta\mathbf{q} d\mathbf{x} - \int_\Gamma [\psi_v]\delta\hat{q} ds - \int_{\partial\Omega_D} (\psi_v + \kappa g_D - g_0)\delta\mathbf{q} \cdot \mathbf{n} ds = 0 \quad (3.13)$$

which implies the following constraints

$$\boxed{[\psi_v]|_\Gamma = \mathbf{0}, \quad \psi_v|_{\partial\Omega_D} = -\kappa g_D + g_0.} \quad (3.14)$$

and produces the minimizer $\mathbf{q} = -\frac{1}{2\kappa}(\psi_\mathbf{q} + \nabla\psi_v)$. Second, setting the variation of $\mathcal{L}$ with respect to v equal to zero results in

$$\begin{aligned} &\int_\Omega (\nabla \cdot \psi_\mathbf{q} - \kappa f + f_0^V)\delta v d\mathbf{x} - \int_\Gamma [\psi_\mathbf{q} \cdot \mathbf{n}]\delta\hat{v} ds \\ &- \int_{\partial\Omega_N} (\psi_\mathbf{q} \cdot \mathbf{n} + \kappa g_N - f_0^S)\delta v ds + \int_{\partial\Omega_D} (\kappa(2\alpha v - g_D) + \alpha\psi_v)\delta v ds = 0. \end{aligned} \quad (3.15)$$

which requires

$$\boxed{\begin{aligned} &\nabla \cdot \psi_\mathbf{q} - \kappa f + f_0^V = 0, \quad [\psi_\mathbf{q} \cdot \mathbf{n}]|_\Gamma = 0 \\ &v|_{\partial\Omega_D} = g_D + \frac{1}{2\alpha\kappa} g_0, \quad (\psi_\mathbf{q} \cdot \mathbf{n})|_{\partial\Omega_N} = -\kappa g_N + f_0^S. \end{aligned}} \quad (3.16)$$

We point out that in (3.16), the essential boundary condition on $\partial\Omega_D$ is satisfied in the limit of the penalization parameter, $\alpha \to \infty$. Defining $\mathbf{X}^C$ as the following subset of $\mathbf{X}$ satisfying the following conditions

$$\begin{aligned}
&\mathbf{X}^C \equiv \{\boldsymbol{\Psi}_\mathbf{v} \in \mathbf{X} \quad \text{s.t.} \\
&[\psi_v]|_\Gamma = \mathbf{0}, \quad [\psi_\mathbf{q} \cdot \mathbf{n}]|_\Gamma = 0 \\
&\psi_v|_{\partial\Omega_D} = -\kappa g_D + g_0, \quad (\psi_\mathbf{q} \cdot \mathbf{n})|_{\partial\Omega_N} = -\kappa g_N + f_0^S \\
&\nabla \cdot \psi_\mathbf{q} - \kappa f + f_0^V = 0\}
\end{aligned} \tag{3.17}$$

we have

$$S^-(\boldsymbol{\Psi}_\mathbf{v}) = \inf_{\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v}, \boldsymbol{\Psi}_\mathbf{v}) = \begin{cases} \mathcal{L}^*(\boldsymbol{\Psi}_\mathbf{v}) & \text{if } \boldsymbol{\Psi}_\mathbf{v} \in \mathbf{X}^C, \\ -\infty & \text{otherwise.} \end{cases} \tag{3.18}$$

where

$$\begin{aligned}
\mathcal{L}^*(\boldsymbol{\Psi}_\mathbf{v}) \;=\; & -\int_\Omega \left(\frac{1}{4\kappa}(\psi_\mathbf{q} + \nabla\psi_v) \cdot (\psi_\mathbf{q} + \nabla\psi_v) + \psi_v f\right) d\mathbf{x} \\
& - \int_{\partial\Omega_D} \left(\psi_\mathbf{q} \cdot \mathbf{n} g_D + \frac{1}{4\alpha\kappa} g_0^2\right) ds - \int_{\partial\Omega_N} \psi_v g_N ds.
\end{aligned} \tag{3.19}$$

We note that in general, the minimization of $\mathcal{L}$ in (3.9) over v is unbounded. For $\boldsymbol{\Psi}_\mathbf{v} \in \mathbf{X}^C$, however, we do obtain a bounded minimum whose expression is given by (3.19). Here, we make the decision to set $\alpha \to \infty$, as the term contributes negatively to the lower bound, S. It is then a simple matter to evaluate (3.19) to obtain a lower bound for S. We note that *any* $\psi_v, \psi_\mathbf{q}$ satisfying the conditions stated in (3.17) would produce a lower bound for the output. Even so, the choice of ψ_v and $\psi_\mathbf{q}$ plays a critical role in the accuracy of the computed bounds.

3.3 Calculation of Lagrange Multipliers: Infinite-Dimensional Case

In this section, we develop an algorithm for the computation of the Lagrange multipliers ψ_v and $\psi_{\mathbf{q}}$ that will lead to accurate bounds. We will look for Lagrange multipliers ψ_v and $\psi_{\mathbf{q}}$ in a finite-dimensional subspace of $\mathbf{X}_h^C \subset \mathbf{X}^C \subset \mathbf{X}$ and we should expect that as $\mathbf{X}_h^C \to \mathbf{X}^C$ one obtains $S^-(\mathbf{\Psi}_{\mathbf{u}_h}) \equiv \mathcal{L}^*(\mathbf{\Psi}_{\mathbf{u}_h}) \to S$. The procedure for achieving this involves deriving optimal values for $\mathbf{\Psi_v}$, $\mathbf{\Psi_u} \equiv [\psi_u, \psi_{\mathbf{p}}]^T$ in the infinite-dimensional case; where the bound would be the exact output, and then approximating the resulting expressions discretely.

To this end, we formulate the following constrained maximization problem. We maximize (3.19) with respect to $\mathbf{\Psi_v}$ subject to the constraint that $\mathbf{\Psi_v} \in \mathbf{X}^C$

$$\sup_{\mathbf{\Psi_v}\in\mathbf{X}} \inf_{\mathbf{\Lambda_v}\in\mathbf{X}} \Big\{ \mathcal{L}^*(\mathbf{\Psi_v}) + \int_\Omega \lambda_v(\nabla\cdot\psi_{\mathbf{q}} - \kappa f + f_0^V)d\mathbf{x} - \int_\Gamma (\lambda_{\mathbf{q}}\cdot\mathbf{n}[\psi_v] + \lambda_v[\psi_{\mathbf{q}}\cdot\mathbf{n}])ds - \int_{\partial\Omega_D} \lambda_{\mathbf{q}}\cdot\mathbf{n}(\psi_v + \kappa g_D - g_0)ds - \int_{\partial\Omega_N} \lambda_v(\psi_{\mathbf{q}}\cdot\mathbf{n} + \kappa g_N - f_0^S)ds \Big\} \tag{3.20}$$

where $\mathbf{\Lambda} = [\lambda_v, \lambda_{\mathbf{q}}]^T$. We point out that the same Lagrange multiplier, λ_v, is used to enforce the interface condition of $[\psi_{\mathbf{q}}\cdot\mathbf{n}] = 0$ and the elemental equilibrium condition. Note that if we choose different multipliers for the boundary and interior conditions, the maximization over $\psi_{\mathbf{q}}$ will force the two multipliers to be the same. Maximizing over $\psi_{\mathbf{q}}$ leads to

$$\int_\Omega (-\frac{1}{2\kappa}(\psi_{\mathbf{p}} + \nabla\psi_u)\cdot\delta\psi_{\mathbf{q}} + \lambda_u\delta\nabla\cdot\psi_{\mathbf{q}})d\mathbf{x} - \int_\Gamma \lambda_u[\delta\psi_{\mathbf{q}}\cdot\mathbf{n}]ds - \int_{\partial\Omega_D} \delta\psi_{\mathbf{q}}\cdot\mathbf{n}g_D ds - \int_{\partial\Omega_N} \delta\psi_{\mathbf{q}}\cdot\mathbf{n}\lambda_u ds = 0, \quad \forall\delta\psi_{\mathbf{q}} \in \mathbf{X} \tag{3.21}$$

and after some rearrangement we obtain,

$$-\int_{\Omega}\delta\psi_{\mathbf{q}}\cdot\{\frac{1}{2\kappa}(\psi_{\mathbf{p}}+\nabla\psi_u)+\nabla\lambda_u\}d\mathbf{x}+\int_{\partial\Omega_D}\delta\psi_{\mathbf{q}}\cdot\mathbf{n}(\lambda_u-g_D)ds=0,\quad\forall\delta\psi_{\mathbf{q}}\in\mathbf{X}. \tag{3.22}$$

Maximizing over ψ_v results in

$$\begin{aligned}&-\int_{\Omega}(\frac{1}{2\kappa}(\nabla\psi_u+\psi_{\mathbf{p}})\cdot\delta\nabla\psi_v+f\delta\psi_v)d\mathbf{x}-\int_{\Gamma}\lambda_{\mathbf{p}}\cdot\mathbf{n}[\delta\psi_v]ds-\int_{\partial\Omega_N}g_N\delta\psi_v ds\\&-\int_{\partial\Omega_D}\lambda_{\mathbf{p}}\cdot\mathbf{n}\delta\psi_v ds=0,\quad\forall\delta\psi_v\in\mathbf{X}\end{aligned} \tag{3.23}$$

which may be written as

$$\begin{aligned}&\int_{\Omega}-\delta\psi_v(\nabla^2\psi_u+\nabla\cdot\psi_{\mathbf{p}}-2\kappa f)d\mathbf{x}+\int_{\Gamma}[\delta\psi_v](\psi_{\mathbf{p}}+\nabla\psi_u+2\kappa\lambda_{\mathbf{p}})\cdot\mathbf{n}ds+\\&\int_{\partial\Omega_D}\delta\psi_v(\psi_{\mathbf{p}}+\nabla\psi_u+2\kappa\lambda_{\mathbf{p}})\cdot\mathbf{n}ds+\int_{\partial\Omega_N}\delta\psi_v\{(\psi_{\mathbf{p}}+\nabla\psi_u)\cdot\mathbf{n}+2\kappa g_N\}ds\\&=0,\quad\forall\delta\psi_v\in\mathbf{X}.\end{aligned} \tag{3.24}$$

After substitution of the constraint $\nabla\cdot\psi_{\mathbf{q}}-\kappa f+f_0^V=0\in\Omega$ into (3.24) we arrive at

$$\begin{aligned}&\int_{\Omega}-\delta\psi_v(\nabla^2\psi_u-\kappa f-f_0)d\mathbf{x}+\int_{\Gamma}[\delta\psi_v](\psi_{\mathbf{p}}+\nabla\psi_u+2\kappa\lambda_{\mathbf{p}})\cdot\mathbf{n}ds+\\&\int_{\partial\Omega_D}\delta\psi_v(\psi_{\mathbf{p}}+\nabla\psi_u+2\kappa\lambda_{\mathbf{p}})\cdot\mathbf{n}ds+\int_{\partial\Omega_N}\delta\psi_v\{(\psi_{\mathbf{p}}+\nabla\psi_u)\cdot\mathbf{n}+2\kappa g_N\}ds\\&=0,\quad\forall\delta\psi_v\in\mathbf{X}.\end{aligned} \tag{3.25}$$

From (3.22) we have

$$\nabla\lambda_u=-\frac{1}{2\kappa}(\psi_{\mathbf{p}}+\nabla\psi_u),\quad\lambda_u|_{\partial\Omega_D}=g_D$$

and we also observe that (3.25) requires

$$\nabla\lambda_u \cdot \mathbf{n} = -\frac{1}{2\kappa}(\psi_\mathbf{p} + \nabla\psi_u) \cdot \mathbf{n}|_{\partial\Omega_N} = g_N$$

which, when combined with (3.24) results in

$$\begin{aligned} &2\kappa \int_\Omega \delta\psi_v(\nabla^2\lambda_u + f)d\mathbf{x} + \int_\Gamma [\delta\psi_v](\psi_\mathbf{p} + \nabla\psi_u) \cdot \mathbf{n}ds + 2\kappa \int_\Gamma \lambda_\mathbf{p} \cdot \mathbf{n}[\delta\psi_v]ds \\ &-2\kappa \int_{\partial\Omega_N} \delta\psi_v(\nabla\lambda_u \cdot \mathbf{n} - g_N)ds = 0, \quad \forall\delta\psi_v \in \mathbf{X}^C. \end{aligned} \tag{3.26}$$

Note that (3.26) is identical to the primal problem and therefore $\lambda_u = u, \nabla\lambda_u = \nabla u$ and therefore $\lambda_\mathbf{p} = \mathbf{p}$. From equations (3.22) and (3.25) we arrive at

$$\begin{aligned} &\nabla^2 u + f = 0, \quad u|_{\partial\Omega_D} = g_D, \quad (\nabla u \cdot \mathbf{n})|_{\partial\Omega_N} = g_N \quad \text{(primal problem)} \\ &\nabla^2\psi_u - (\kappa f + f_0^V) = 0, \quad \psi_u|_{\partial\Omega_D} = -\kappa g_D + g_0, \\ &(\nabla\psi_u \cdot \mathbf{n})|_{\partial\Omega_N} = -\kappa g_N + f_0^S \quad \text{(“lifted” dual problem)} \\ &\text{Also,} \quad \psi_\mathbf{p} = -2\kappa\lambda_\mathbf{p} - \nabla\psi_u, \quad \lambda_\mathbf{p} = \nabla\lambda_u = \nabla u. \end{aligned} \tag{3.27}$$

The set of equations above produces ψ_u and $\psi_\mathbf{p}$ that, in the infinite-dimensional case, satisfy (3.18) and result in the exact bounds for S. In practice, however, we would of course be working with finite-dimensional discretizations and the approximations to $\mathbf{u}$ and $\mathbf{\Psi_u}$ obtained will satisfy the conditions in (3.18) in only a weak sense and do not satisfy all the constraints required by X^C. However, a simple postprocessing of the computed approximations is sufficient to obtain valid multipliers that will guarantee a lower bound in (3.19).

3.3.1 Alternative Derivation

Here, we present an alternative derivation for the optimal $\mathbf{\Psi_u}$ in the infinite-dimensional limit. To do so, we first define the primal and dual problems as

$$
\begin{aligned}
-\nabla^2 u - f = 0 \quad &\text{in} \quad \Omega \\
u = g_D \quad &\text{on} \quad \partial\Omega_D \\
\nabla u = g_N \quad &\text{on} \quad \partial\Omega_N \qquad \text{(Primal Problem)} \\
-\nabla^2 \zeta_u + f_0^V = 0 \quad &\text{in} \quad \Omega \\
\zeta_u = g_0 \quad &\text{on} \quad \partial\Omega_D \\
\nabla \zeta_u = f_0^S \quad &\text{on} \quad \partial\Omega_N \qquad \text{(Dual Problem)}
\end{aligned} \tag{3.28}
$$

where $\zeta_{\mathbf{u}} = [\zeta_u, \zeta_{\mathbf{p}}]^T$ is the dual solution. We then postulate that $\boldsymbol{\Psi}_{\mathbf{u}}$ takes the form

$$
\begin{aligned}
\psi_u &= \alpha_u u + \beta_u \zeta_u \\
\psi_{\mathbf{p}} &= \alpha_{\mathbf{p}} \mathbf{p} + \beta_{\mathbf{p}} \zeta_{\mathbf{p}}
\end{aligned} \tag{3.29}
$$

and show why this *must* be the case. To satisfy the condition

$$
\nabla \cdot \psi_{\mathbf{p}} - \kappa f + f_0^V = 0 \qquad \text{in} \quad \Omega
$$

given the definitions of the primal and dual problems implies that $\psi_{\mathbf{p}}$ is necessarily a linear combination of $\mathbf{p}, \zeta_{\mathbf{p}}$. We have

$$
\alpha_{\mathbf{p}} = -\kappa, \quad \beta_{\mathbf{p}} = -1
$$

since

$$
\begin{aligned}
-\nabla \mathbf{p} - f = 0 \quad &\text{in} \quad \Omega \quad \text{(Primal Problem)} \\
-\nabla \zeta_{\mathbf{p}} + f_0^V = 0 \quad &\text{in} \quad \Omega \quad \text{(Dual Problem).}
\end{aligned}
$$

We also have the minimizer

$$
\mathbf{p} = -\frac{1}{2\kappa}(\psi_{\mathbf{p}} + \nabla \psi_u) \tag{3.30}
$$

which we now re-write as

$$\mathbf{p} = -\frac{1}{2\kappa}\Big\{(-\kappa\mathbf{p} - \zeta_{\mathbf{p}}) + \nabla(\alpha_u u + \beta_u \zeta_u)\Big\}. \tag{3.31}$$

To obtain a formulation free of bound gap in the infinite-dimensional limit, (3.31) must be self-consistent. Since $\mathbf{p} = \nabla u$ and $\zeta_{\mathbf{p}} = \nabla\zeta$, this requires $\alpha_u = -\kappa, \beta_u = 1$; at which point (3.31) would be given by

$$\begin{aligned} \mathbf{p} &= -\frac{1}{2\kappa}\Big\{(-\kappa\mathbf{p} - \zeta_{\mathbf{p}}) + \nabla(-\kappa u + \zeta_u)\Big\} \\ &= -\frac{1}{2\kappa}\Big\{-\kappa\mathbf{p} - \kappa\nabla u - \zeta_{\mathbf{p}} + \nabla\zeta_u\Big\} \\ &= \mathbf{p}. \end{aligned} \tag{3.32}$$

This means the minimizer, (3.30), is self-consistent. We have thus shown that the optimal $\mathbf{\Psi_u}$ is a linear combination of the primal and dual solutions.

3.4 Calculation of $\mathbf{\Psi_u}$: Finite-Dimensional Case

We now proceed to compute $\mathbf{\Psi}_{\mathbf{u}_h}$ such that the conditions put forth in (3.18) are met. We first solve (3.28) by LDG as described in chapter 2 and obtain $u_h, \hat{\mathbf{p}}_h$ and $\zeta_{u_h}, \hat{\zeta}_{\mathbf{p}_h}$. From the solution we also obtain interface fluxes $\hat{p}, \hat{\zeta}_{\mathbf{p}}$. Next, we average, by taking the mean of all nodal values of u_h, ζ_{u_h} at all elemental vertices and interfaces to obtain $\tilde{u}_h, \tilde{\zeta}_{u_h}$. To satisfy the necessary condition of continuous ψ_u, we set

$$\boxed{\psi_{u_h} = -\kappa\tilde{u}_h + \tilde{\zeta}_{u_h}.} \tag{3.33}$$

Finally, we modify the boundary values of ψ_{u_h} thus obtained such that essential boundary conditions are satisfied point-wise. Note that ψ_{u_h} is an approximation to the "lifted" dual function in (3.27) and meets all the constraints in (3.17).

3.4.1 Elemental Reconstruction of $\psi_{\mathbf{p}_h}$

We recall that $\psi_{\mathbf{p}}$ must satisfy

$$\nabla \cdot \psi_{\mathbf{p}} - \kappa f + f_0^V = 0, \quad [\psi_{\mathbf{p}} \cdot \mathbf{n}]|_\Gamma = 0, \quad (\psi_{\mathbf{p}} \cdot \mathbf{n})|_{\partial\Omega_N} = -\kappa g_N + f_0^S.$$

Since $\mathbf{p}_h$ and $\zeta_{\mathbf{p}_h}$ only satisfy these conditions in a weak sense, their values must be appropriately post-processed prior to insertion into (3.19). To obtain $\psi_{\mathbf{p}}$, we start by making the following observations regarding LDG solutions

- $\hat{p}_h, \hat{\zeta}_{\mathbf{p}_h}$ are unique across elemental interfaces
- $\int_{\partial\Omega_j} \hat{p}_h ds = -\int_{\Omega_j} f d\mathbf{x}$ (seen by setting test function to one on Ω_j and zero elsewhere)
- $\int_{\partial\Omega_j} \hat{\zeta}_{\mathbf{p}_h} ds = \int_{\Omega_j} f_0^V d\mathbf{x}$ (seen by setting test function to one on Ω_j and zero elsewhere)

We can therefore satisfy all the necessary conditions by solving *locally*, in each element for

$$\nabla \cdot \tilde{\mathbf{p}}_h - f = 0, \quad \nabla \cdot \tilde{\zeta}_{\mathbf{p}_h} + f_0 = 0 \tag{3.34}$$

with the boundary condition

$$\tilde{\mathbf{p}}_h \cdot \mathbf{n} = \hat{p}, \quad \tilde{\zeta}_{\mathbf{p}_h} \cdot \mathbf{n} = \hat{\zeta}_{\mathbf{p}}. \tag{3.35}$$

Equations (3.34) and (3.35) can be shown in general to possess a solution [30], we can therefore solve the aforementioned equations and set

$$\boxed{\psi_{\mathbf{p}_h} = -\kappa \tilde{\mathbf{p}}_h - \tilde{\zeta}_{\mathbf{p}_h}.} \tag{3.36}$$

In particular, for P_1 elements in two dimensions, we obtain six equations for $\tilde{\mathbf{p}}_h$ and $\tilde{\zeta}_{\mathbf{p}_h}$ with a non-singular coefficient matrix-implying a unique solution. The resulting $\psi_{u_h}, \psi_{\mathbf{p}_h}$ satisfy (3.18) in a point-wise manner as long as f and f_0^V are constant on Ω_j. General polynomial forms of f and f_0^V may also be considered provided a higher degree polynomial is used in

the discretization. This property allows us to bound the output of the exact solution. In general, the algorithm as given is applicable to the Poisson equation defined over a polygonal domain with piecewise polynomial forcing.

Finally, we summarize the steps for the computation of $\boldsymbol{\Psi}_h$ as follows

1. Solve (3.28) by LDG for $u_h, \hat{p}_h$ and $\zeta_{u_h}, \hat{\zeta}_{\mathbf{p}_h}$
2. Average u_h and ζ_{u_h} at all elemental vertices and interfaces to obtain $\tilde{u}_h$ and $\tilde{\zeta}_{u_h}$
3. Solve (3.34) *locally*, in each element and impose (3.35) at all elemental boundaries and obtain $\tilde{\mathbf{p}}_h$ and $\tilde{\zeta}_{\mathbf{p}_h}$
4. Set $\psi_{u_h} = -\kappa\tilde{u}_h + \tilde{\zeta}_{u_h}$
5. Set $\psi_{\mathbf{p}_h} = -\kappa\tilde{\mathbf{p}}_h - \tilde{\zeta}_{\mathbf{p}_h}$

Algorithm 1: Computation of $\boldsymbol{\Psi}_{\mathbf{u}_h}$

Note that (1) involves only two global solutions (for the primal and dual problems) and that all other computations involve *local* operations.

3.5 Upper Bounds

We now proceed to the computation of upper bounds. The procedure outlined in the previous section can be easily extended to the calculation of upper bounds for our output of interest. From (3.6), it is seen that the lower bound of the output is given by the constrained minimization statement

$$S = \inf_{\mathbf{v}\in\mathbf{X}} \sup_{\boldsymbol{\Psi}_\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v}, \boldsymbol{\Psi}_\mathbf{v}) \geq \inf_{\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v}, \boldsymbol{\Psi}_\mathbf{v}) = S^{-}(\boldsymbol{\Psi}_\mathbf{v}). \tag{3.37}$$

and the the upper bound may be acquired in an analogous manner by calculating a lower bound for $-S$ [35]. That is, we introduce $\tilde{\mathcal{L}}$

$$\tilde{\mathcal{L}}(\mathbf{v}, \boldsymbol{\Psi}_\mathbf{v}) = \kappa(a(\mathbf{v},\mathbf{v}) - l(\mathbf{v})) - l^0(\mathbf{v}) + a(\boldsymbol{\Psi}_\mathbf{v}, \mathbf{v}) - l(\boldsymbol{\Psi}_\mathbf{v}). \tag{3.38}$$

A lower bound for $-l(\mathbf{u})$ is given by

$$-l(\mathbf{u}) = -S = \inf_{\mathbf{v}\in\mathbf{X}} \sup_{\mathbf{\Psi}_v\in\mathbf{X}} \tilde{\mathcal{L}}(\mathbf{v}, \mathbf{\Psi}_v) \geq \inf_{\mathbf{v}\in\mathbf{X}} \tilde{\mathcal{L}}(\mathbf{v}, \mathbf{\Psi}_\mathbf{v}) \tag{3.39}$$

and therefore

$$l(\mathbf{u}) = S \leq -\inf_{\mathbf{v}\in\mathbf{X}} \tilde{\mathcal{L}}(\mathbf{v}, \mathbf{\Psi}_\mathbf{v}). \tag{3.40}$$

Note that a meaningful bound can only be obtained for $\mathbf{\Psi} \in X^C$ We point out that the computation of upper bounds requires the solution of the primal problem-which we have already obtained for the computation of lower bounds, as well as the negative of the dual solution-which we also possess from lower bound computations.

3.6 Bound Optimization

The bound gap, $\Delta S = S^+ - S^-$, is given by

$$\boxed{S^+ - S^- = \int_\Omega \left\{ \frac{\kappa}{2}(\tilde{\mathbf{p}}_h - \nabla\tilde{u}_h)\cdot(\tilde{\mathbf{p}}_h - \nabla\tilde{u}_h) + \frac{1}{2\kappa}(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h})\cdot(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h}) \right\} d\mathbf{x}} \tag{3.41}$$

To see this, we go back to the expression for $S^\pm$. Denoting the solutions to the adjoint equation corresponding to the lower bound computation $\zeta^-_{u_h}, \zeta^-_{\mathbf{p}_h}$ and $\zeta^+_{u_h}, \zeta^+_{\mathbf{p}_h}$ for the upper bound, we have the following expression for the upper and lower bounds

$$\begin{aligned} S^\pm \quad &= \pm \int_\Omega \left\{ \frac{1}{4\kappa} \Big[\kappa^2(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h)\cdot(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h) + \kappa(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h)\cdot(\tilde{\zeta}^\pm_{\mathbf{p}_h} - \nabla\tilde{\zeta}^\pm_{u_h}) \right. \\ &\quad \left. + (\tilde{\zeta}^\pm_{\mathbf{p}_h} - \nabla\tilde{\zeta}^\pm_{u_h})\cdot(\tilde{\zeta}^\pm_{\mathbf{p}_h} - \nabla\tilde{\zeta}^\pm_{u_h}) \Big] + (-\kappa\tilde{u}_h + \tilde{\zeta}^\pm_{u_h}) f \right\} d\mathbf{x} \\ &\quad \mp \int_{\partial\Omega_D} (\kappa\tilde{\mathbf{p}}_h + \tilde{\zeta}^\pm_{\mathbf{p}_h})\cdot\mathbf{n} g_D ds \mp \int_{\partial\Omega_N} (\kappa\tilde{u}_h + \tilde{\zeta}^\pm_{u_h}) g_N ds. \end{aligned} \tag{3.42}$$

Noting that $\tilde{\zeta}^+_{u_h} = -\tilde{\zeta}^-_{u_h}$ and $\tilde{\zeta}^+_{\mathbf{p}_h} = -\tilde{\zeta}^-_{\mathbf{p}_h}$, we can write the bound gap as

$$\begin{aligned}\Delta S = \quad & \int_\Omega \Big\{ \frac{\kappa}{2}(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h)\cdot(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h) + \frac{1}{2\kappa}(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h})\cdot(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h}) \\ & -2\kappa\tilde{u}_h f \Big\} d\mathbf{x} - \int_{\partial\Omega_D} 2\kappa\tilde{\mathbf{p}}_h\cdot\mathbf{n} g_D ds - \int_{\partial\Omega_N} 2\kappa\tilde{u}_h g_N ds\end{aligned}$$

which can be written as

$$\begin{aligned}\Delta S = \quad & \int_\Omega \Big\{ \frac{\kappa}{2}(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h)\cdot(\tilde{\mathbf{p}}_h + \nabla\tilde{u}_h) + \frac{1}{2\kappa}(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h})\cdot(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h}) \\ & -2\kappa\tilde{\mathbf{p}}_h\cdot\nabla\tilde{u}_h \Big\} d\mathbf{x}\end{aligned}$$

since

$$\int_\Omega \nabla\tilde{u}_h\cdot\tilde{\mathbf{p}}_h - \tilde{u}_h f d\mathbf{x} - \int_{\partial\Omega_D}\tilde{\mathbf{p}}_h\cdot\mathbf{n} g_D ds - \int_{\partial\Omega_N}\tilde{u}_h g_N ds = 0,$$

which confirms (3.41). To obtain κ, we minimize (3.41) with respect to κ by setting

$$\frac{\partial}{\partial\kappa}\Delta S = 0$$

which leads to

$$\int_\Omega \Big\{ \frac{1}{2}(\tilde{\mathbf{p}}_h - \nabla\tilde{u}_h)\cdot(\tilde{\mathbf{p}}_h - \nabla\tilde{u}_h) - \frac{1}{2\kappa^2}(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h})\cdot(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h}) \Big\} d\mathbf{x} = 0$$

and

$$\boxed{\kappa = \sqrt{\frac{\sum_{j=1}^{N_e}\int_{\Omega_j}(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h})\cdot(\tilde{\zeta}_{\mathbf{p}_h} - \nabla\tilde{\zeta}_{u_h})d\mathbf{x}}{\sum_{j=1}^{N_e}\int_{\Omega_j}(\tilde{\mathbf{p}}_h - \nabla\tilde{u}_h)\cdot(\tilde{\mathbf{p}}_h - \nabla\tilde{u}_h)d\mathbf{x}}}.} \qquad (3.43)$$

3.7 Error Bound Algorithm Example: Volumetric Outputs

For this test case, we solve

$$\begin{aligned} -\nabla u^2 - f = 0, \quad f = 1 \quad \text{on} \quad \Omega = [0,1] \times [0,1] \\ u = 0 \quad \text{on} \quad \partial\Omega \end{aligned} \tag{3.44}$$

with LDG discretization and choose as output

$$l^0(\mathbf{u}) = \int_\Omega f_0^V u d\mathbf{x}$$

with f_0^V taken to be

$$\begin{aligned} \int_\Omega f_0^V u d\mathbf{x} \quad \text{with} \quad f_0^V = 1 \qquad \text{if} \qquad & x_1 \leq \frac{1}{2} \quad \text{and} \quad x_2 \geq \frac{1}{2}, \\ \text{or} \qquad & x_1 \geq \frac{1}{2} \quad \text{and} \quad x_2 \leq \frac{1}{2} \\ = 0 \quad \text{otherwise.} & \end{aligned}$$

The grid convergence results are shown in the table (3.1). The problem setup, computational mesh as well as primal and dual solution are shown in figures (3-1)-(3-4). We see that the optimal convergence rate of order h^{2k} is indeed obtained.

h	S	S^-	S^+	Order
1/8	1.757213×10^{-2}	1.666820×10^{-2}	1.782455×10^{-2}	-
1/16	1.757213×10^{-2}	1.734121×10^{-2}	1.763837×10^{-2}	1.96
1/32	1.757213×10^{-2}	1.751415×10^{-2}	1.758905×10^{-2}	1.99

Table 3.1: ΔS Grid Convergence: Poisson

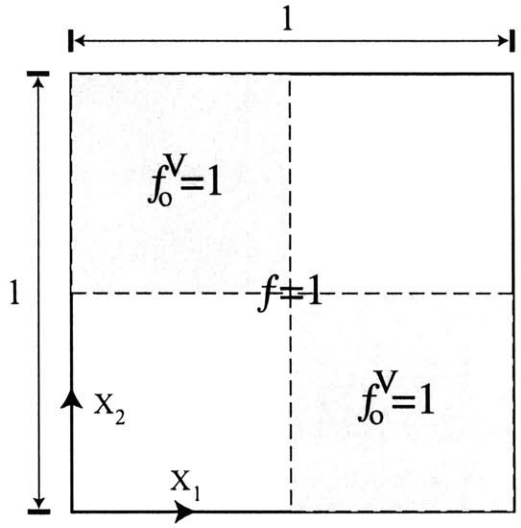


Figure 3-1: Problem Setup: Poisson

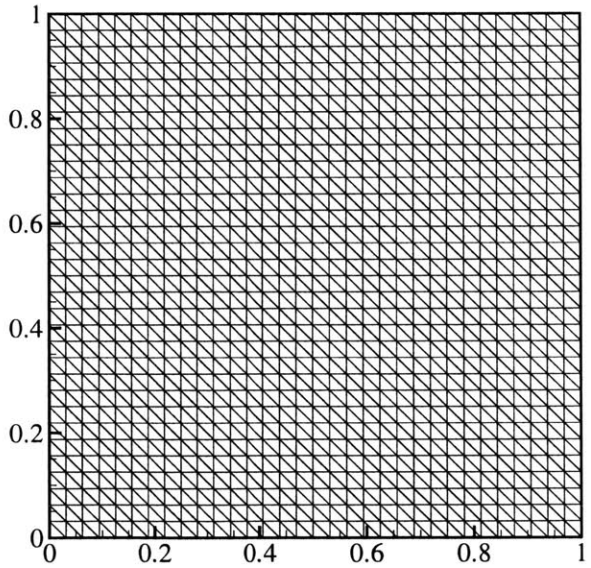


Figure 3-2: $h = 1/32$ Computational Mesh: Poisson

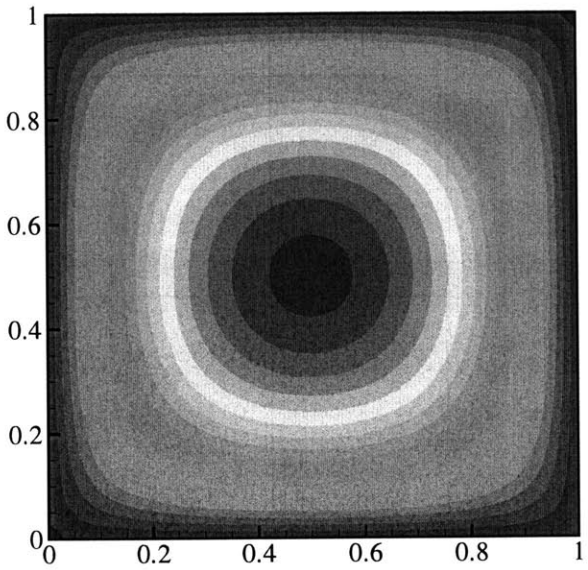


Figure 3-3: Primal solution: Poisson, u_h

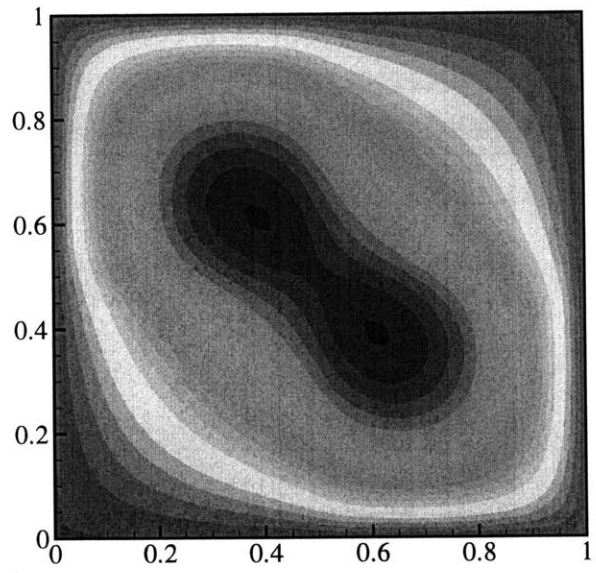


Figure 3-4: Dual solution: Poisson, u_h

Chapter 4

Bounds for Linear Functional Outputs: Nonsymmetric Case

In this chapter, we turn our attention to the development of the error bounding algorithm for linear functional outputs of second order scalar nonsymmetric coercive PDEs; in particular, the convection-diffusion equation. In most respects, the error bounding algorithm for the nonsymmetric case proceeds in much the same manner as in the symmetric case considered in the previous chapter. However, there are some subtleties associated with the nonsymmetric terms that needs to be addressed in order to produce an effective algorithm.

In the convective limit, however, we face additional challenges. The basic method developed thus far produces bounds which deteriorate significantly in the convective limit when the solution is not well-resolved. This is in agreement with the results reported in [43]. This issue is of particular importance here since a primary motivation for using discontinuous finite elements both in constructing an error bounding algorithm and in the numerical solution of PDEs is to exploit the algorithm's ability to handle convection-dominated problems. In this chapter, we also propose an approach to recover accurate bounds for convection-dominated problems.

4.1 Problem Definition

We take as model problem the convection-diffusion equation, given by (2.4). As before, we are interested in obtaining upper and lower bounds on linear functional outputs derived from the solution to (2.4) of the form

$$S = \int_\Omega u f_0^V d\mathbf{x} + \int_{\partial\Omega_D} \nabla u \cdot \mathbf{n} g_0 ds + \int_{\partial\Omega_N} u f_0^S ds. \tag{4.1}$$

Here, $f_0^V \in L_2(\Omega)$ and $g_0, f_0^S \in L_2(\partial\Omega)$ are given functions. We recall from (2.41) and (2.42) that with the definitions $\mathbf{v} = [v, \mathbf{q}]^T, \mathbf{q} = [q_1, q_2, q_3]^T, \mathbf{u} = [u, \mathbf{p}]^T, \mathbf{p} = [p_1, p_2, p_3]^T$, (2.4) may be expressed in weak form as

$$a(\mathbf{v}, \mathbf{u}) = l(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{X}. \tag{4.2}$$

The output may then expressed as $S = l^0(\mathbf{u})$ with

$$l^0(\mathbf{v}) = \int_\Omega v f_0^V d\mathbf{x} + \int_{\partial\Omega_D} \mathbf{q} \cdot \mathbf{n} g_0 ds + \int_{\partial\Omega_N} v f_0^S ds. \tag{4.3}$$

4.2 Lower Bound: the Lagrangian

The strategy for producing upper and lower bounds on linear functional outputs of the convection-diffusion equation is much the same as that for the Poisson equation; we want to express the output as the solution to a constrained minimization problem whose lower bound may be expressed in a form analogous to (3.12). We then choose the Lagrange multipliers such that we obtain the analog of case I in (3.12). The Lagrangian, $\mathcal{L} : \mathbf{X} \times \mathbf{X} \rightarrow \Re$, is now defined as

$$\mathcal{L}(\mathbf{v}, \Psi_\mathbf{v}) = \kappa(a(\mathbf{v}, \mathbf{v}) - l(\mathbf{v})) + l^0(\mathbf{v}) + a(\Psi_v, \mathbf{v}) - l(\Psi_\mathbf{v}) + \Theta(\Psi_\mathbf{v}, \mathbf{v}), \quad \forall \mathbf{v}, \Psi_\mathbf{v} \in \mathbf{X}.$$

where $\boldsymbol{\Psi}_{\mathbf{v}} = [\psi_v, \psi_{\mathbf{q}}]^T$ and $\kappa \geq 0$. $\Theta(\boldsymbol{\Psi}_{\mathbf{v}}, \mathbf{v})$ is given by

$$\begin{aligned}\Theta(\boldsymbol{\Psi}_{\mathbf{v}}, \mathbf{v}) &= \int_\Gamma \frac{1}{2}\alpha_\Gamma |\boldsymbol{a}\cdot\mathbf{n}|[v]^2 ds + \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}(\psi_v \boldsymbol{a}\cdot\mathbf{q} + \nabla\psi_v\cdot\boldsymbol{a}v)d\mathbf{x} \\ &- \int_{\partial\Omega_j}\psi_v\boldsymbol{a}\cdot\mathbf{n}v ds\Big\}. \end{aligned} \tag{4.4}$$

We also recall that $a(\mathbf{v},\mathbf{w}) - l(\mathbf{v})$ is given by

$$\begin{aligned}a(\mathbf{v},\mathbf{w}) - l(\mathbf{v}) &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}(\nabla v\cdot(-\boldsymbol{a}w + \mathbf{r}) - vf)d\mathbf{x} + \int_{\partial\Omega_j} v(\hat{h} - \hat{r})ds \\ &+ \int_{\Omega_j}(\mathbf{q}\cdot\mathbf{r} + \nabla\cdot\mathbf{q}w)d\mathbf{x} - \int_{\partial\Omega_j}\mathbf{q}\cdot\mathbf{n}\hat{w}ds\Big\} \end{aligned} \tag{4.5}$$

for $\mathbf{w} = [w, \mathbf{r}]^T$. Here α_Γ is a positive penalizing parameter. We now make a few comments concerning $\Theta(\boldsymbol{\Psi}_{\mathbf{v}}, \mathbf{v})$. First, for $\mathbf{v} = \mathbf{u}$, one obtains $\Theta(\boldsymbol{\Psi}_{\mathbf{v}}, \mathbf{u}) = 0$. The penalization parameter, α_Γ in $\Theta(\boldsymbol{\Psi}_{\mathbf{v}}, \mathbf{v})$ plays an analogous role to that of α in $a(\mathbf{v},\mathbf{w})$ (see (2.41)); namely, to restrict the space over which we minimize $\mathcal{L}^*$. In the limit of $\alpha, \alpha_\Gamma \to \infty$, essential boundary conditions and continuity of solution across elemental interfaces, $[v] = 0$, will be satisfied *exactly*. This obviously reduces the space over which we minimize $\mathcal{L}$ and produces better bounds. Second, the second and third terms in (4.4), when added to $a(\boldsymbol{\Psi}_{\mathbf{v}}, \mathbf{v})$ results in the *non*-integrated by parts form of the operator such that we have

$$\begin{aligned}\tilde{a}(\mathbf{v},\mathbf{w}) - \tilde{l}(\mathbf{v}) &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}(v\boldsymbol{a}\cdot\mathbf{r} + \nabla v\cdot\mathbf{r} - vf)d\mathbf{x} + \int_{\partial\Omega_j} v(\hat{h} - \boldsymbol{a}\cdot\mathbf{n}w - \hat{r})ds \\ &+ \int_{\Omega_j}(\mathbf{q}\cdot\mathbf{r} + \nabla\cdot\mathbf{q}w)d\mathbf{x} - \int_{\partial\Omega_j}\mathbf{q}\cdot\mathbf{n}\hat{w}ds\Big\}. \end{aligned} \tag{4.6}$$

We have chosen this form for the definition of the Lagrangian for the ease of post-processing later; as with the purely elliptic case, we will need to modify our finite dimensional solution to satisfy all of the necessary conditions for the computation of exact bounds. This form

allows us to compute bounds without having to perform *additional global problem solutions* beyond the solution of the primal and dual problems.

From (3.6), we see that the output may be expressed in terms of the following constrained minimization statement

$$S = l^0(\mathbf{u}) = \inf_{\mathbf{v}\in\mathbf{X}} \sup_{\mathbf{\Psi_v}\in\mathbf{X}} \mathcal{L}(\mathbf{v}, \mathbf{\Psi_v}), \quad \forall \mathbf{\Psi_v} \in \mathbf{X} \tag{4.7}$$

and from the same argument as the one used to obtain (3.8), we can write the following inequality

$$S \geq l^0(\mathbf{u}) = \inf_{\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v}, \mathbf{\Psi_v}). \tag{4.8}$$

To obtain a lower bound for S, we follow the algorithm developed in the previous chapter and attempt to minimize $\mathcal{L}(\mathbf{v}, \mathbf{\Psi_v})$ over all $\mathbf{v} = [v, \mathbf{q}]^T$. We start by setting $\alpha, \alpha_\Gamma \to \infty$, resulting in the following consequences

$$v|_{\partial\Omega_D} = g_D, \quad [v] = \mathbf{0}.$$

(Comment: leaving the penalization parameters in results in significantly longer algebra, which I thought distracts from the main point of the presentation, but can be done if necessary.) First, setting the variation of $\mathcal{L}$ with respect to $\mathbf{q}$ equal to zero results in

$$\int_\Omega (2\kappa\mathbf{q} + \psi_\mathbf{q} + \boldsymbol{a}\psi_v + \nabla\psi_v) \cdot \delta\mathbf{q} d\mathbf{x} - \int_\Gamma [\psi_v]\delta\hat{q} ds - \int_{\partial\Omega_D} (\psi_v + \kappa g_D - g_0)\delta\mathbf{q} \cdot \mathbf{n} ds = 0 \tag{4.9}$$

which necessitates

$$\boxed{[\psi_v] = \mathbf{0}, \quad \psi_v|_{\partial\Omega_D} = -\kappa g_D + g_0.} \tag{4.10}$$

and produces the minimizer $\mathbf{q} = -\frac{1}{2\kappa}(\psi_\mathbf{q} + \boldsymbol{a}\psi_v + \nabla\psi_v)$. Second, setting the variation of $\mathcal{L}$

with respect to v equal to zero results in

$$\int_{\Omega}(\nabla\cdot\psi_{\mathbf{q}}-\kappa f+f_0^V)\delta v d\mathbf{x}-\int_{\Gamma}[\psi_{\mathbf{q}}\cdot\mathbf{n}]\delta\hat{v}ds-\int_{\partial\Omega_N}(\psi_{\mathbf{q}}\cdot\mathbf{n}+\kappa g_N-f_0^S)\delta v ds=0 \quad (4.11)$$

which requires

$$\boxed{\begin{aligned}&\nabla\cdot\psi_{\mathbf{q}}-\kappa f+f_0^V=0\\&[\psi_{\mathbf{q}}\cdot\mathbf{n}]=0,\quad(\psi_{\mathbf{q}}\cdot\mathbf{n})|_{\partial\Omega_N}=-\kappa g_N+f_0^S.\end{aligned}\quad (4.12)}$$

Defining $\mathbf{X}^C$ as a subset of $\mathbf{X}$ satisfying (4.10) and (4.12), we have

$$\boxed{\inf_{\mathbf{v}\in\mathbf{X}}\mathcal{L}(\mathbf{v},\Psi_{\mathbf{v}})=\begin{cases}\mathcal{L}^* & \text{if } \Psi_{\mathbf{v}}\in\mathbf{X}^C,\\ -\infty & \text{otherwise.}\end{cases}\quad (4.13)}$$

where $\mathcal{L}^*=S^-$ is given by

$$\boxed{\begin{aligned}\mathcal{L}^* &= -\int_{\Omega}\frac{1}{4\kappa}(\psi_{\mathbf{q}}+\boldsymbol{a}\psi_v+\nabla\psi_v)\cdot(\psi_{\mathbf{q}}+\boldsymbol{a}\psi_v+\nabla\psi_v)+\psi_v f d\mathbf{x}\\ &\quad -\int_{\partial\Omega_D}(\psi_{\mathbf{q}}\cdot\mathbf{n}g_D-\frac{1}{2}\boldsymbol{a}\cdot\mathbf{n}g_D^2)ds-\int_{\partial\Omega_N}\psi_v g_N ds,\quad\forall\Psi_{\mathbf{v}}\in\mathbf{X}^C.\end{aligned}\quad (4.14)}$$

We need only to evaluate (4.14) to obtain a lower bound for S. Just as before, any $\Psi_{\mathbf{v}}$ satisfying (4.13) would, just as in the Poisson case, produce a lower bound for the output. The choice of $\Psi_{\mathbf{v}}$, however, is critical in obtaining bounds of acceptable quality.

4.3 Calculation of Lagrange Multipliers: Infinite-Dimensional Case

As pointed out in the previous section, the choice of $\Psi_{\mathbf{v}}$ is critical to the quality of the computed bounds. With the same approach we had employed earlier, we setup the following constrained maximization problem and maximize over $\Psi_{\mathbf{v}}$ subject to necessary constraints

$$\sup_{\boldsymbol{\Psi}_\mathbf{v}\in\mathbf{X}}\inf_{\boldsymbol{\Lambda}_\mathbf{v}\in X}\Big\{\mathcal{L}^*(\boldsymbol{\Psi}_\mathbf{v}) + \int_\Omega \lambda_v(\nabla\cdot\psi_\mathbf{q} - \kappa f + f_0^V)d\mathbf{x} - \int_\Gamma(\lambda_\mathbf{q}\cdot\mathbf{n}[\psi_v] + \lambda_v[\psi_\mathbf{q}\cdot\mathbf{n}])ds$$
$$- \int_{\partial\Omega_D}\lambda_\mathbf{q}\cdot\mathbf{n}(\psi_v + \kappa g_D - g_0)ds - \int_{\partial\Omega_N}\lambda_v(\psi_\mathbf{q}\cdot\mathbf{n} + \kappa g_N - f_0^S)ds\Big\} \tag{4.15}$$

where $\boldsymbol{\Lambda}_\mathbf{v} = [\lambda_v, \lambda_\mathbf{q}]^T$. Maximizing (4.15) over $\psi_\mathbf{q}$ leads to

$$-\int_\Omega \delta\psi_\mathbf{q}\cdot\left\{\frac{1}{2\kappa}(\psi_\mathbf{p} + \boldsymbol{a}\psi_u + \nabla\psi_u) + \nabla\lambda_u\right\}d\mathbf{x} + \int_{\partial\Omega_D}\delta\psi_\mathbf{q}\cdot\mathbf{n}(\lambda_u - g_D)ds = 0,$$
$$\forall\delta\psi_\mathbf{q}\in\mathbf{X}. \tag{4.16}$$

Maximizing over ψ_v results in

$$-\int_\Omega\left\{\frac{1}{2\kappa}(\psi_\mathbf{p} + \boldsymbol{a}\psi_u + \nabla\psi_u)\cdot(\boldsymbol{a}\delta\psi_v + \delta\nabla\psi_v) + \delta\psi_v f\right\}d\mathbf{x} - \int_\Gamma\lambda_\mathbf{p}\cdot\mathbf{n}[\delta\psi_v]ds$$
$$-\int_{\partial\Omega_N}\delta\psi_v g_N ds - \int_{\partial\Omega_D}\lambda_\mathbf{p}\cdot\mathbf{n}\delta\psi_v ds = 0, \quad \delta\psi_v\in\mathbf{X}$$

which may be written as

$$-\int_\Omega\{(\psi_\mathbf{p} + \nabla\psi_u + \boldsymbol{a}\psi_u)\cdot\boldsymbol{a} - (\nabla\cdot\psi_\mathbf{p} + \boldsymbol{a}\cdot\nabla\psi_u + \nabla^2\psi_u) + 2\kappa f\}\delta\psi_v d\mathbf{x} -$$
$$\int_\Gamma[\delta\psi_v](\psi_\mathbf{p} + \boldsymbol{a}\psi_u + \nabla\psi_u + 2\kappa\lambda_\mathbf{p})\cdot\mathbf{n}ds - \int_{\partial\Omega_N}\delta\psi_v\{(\psi_\mathbf{p} + \boldsymbol{a}\psi_u + \nabla\psi_u)\cdot\mathbf{n} + 2\kappa g_N\}ds$$
$$-\int_{\partial\Omega_D}\delta\psi_v(\psi_\mathbf{p} + \boldsymbol{a}\psi_u + \nabla\psi_u + 2\kappa\lambda_\mathbf{p})\cdot\mathbf{n}ds = 0, \quad \delta\psi_v\in\mathbf{X} \tag{4.17}$$

and after further simplification and substitution we arrive at

$$-\int_{\Omega}\delta\psi_v(\boldsymbol{a}\cdot\psi_{\mathbf{p}}+\boldsymbol{a}^2\psi_u-\nabla^2\psi_u+\kappa f+f_0^V)d\mathbf{x}-$$
$$\int_{\Gamma}[\delta\psi_v](\psi_{\mathbf{p}}\cdot+\boldsymbol{a}\psi_u+\nabla\psi_u+2\kappa\lambda_{\mathbf{p}})\cdot\mathbf{n}ds-\int_{\partial\Omega_N}\delta\psi_v\{(\psi_{\mathbf{p}}+\boldsymbol{a}\psi_u+\nabla\psi_u)\cdot\mathbf{n}+2\kappa g_N\}ds$$
$$-\int_{\partial\Omega_D}\delta\psi_v(\psi_{\mathbf{p}}+\boldsymbol{a}\psi_u+\nabla\psi_u+2\kappa\lambda_{\mathbf{p}})\cdot\mathbf{n}ds=0,\quad\delta\psi_v\in\mathbf{X}. \tag{4.18}$$

From (4.16) and (4.18) we have

$$\nabla\lambda_u=-\frac{1}{2\kappa}(\psi_{\mathbf{p}}+\boldsymbol{a}\psi_u+\nabla\psi_u),\quad\lambda_u|_{\partial\Omega_D}=g_D,\quad(\nabla\lambda_u\cdot\mathbf{n})|_{\partial\Omega_N}=g_N \tag{4.19}$$

which, when combined with (4.17) results in

$$\int_{\Omega}\delta\psi_u(\boldsymbol{a}\cdot\nabla\lambda_p-\nabla^2\lambda_u-f)d\mathbf{x}+\int_{\partial\Omega_N}\delta\psi_u(\lambda_{\mathbf{p}}\cdot\mathbf{n}-g_N)ds=0,\quad\delta\psi_u\in\mathbf{X}. \tag{4.20}$$

From (4.16), (4.18) and (4.20) we see that $\lambda_v=u$ which obviously implies $\nabla\lambda_u=\nabla u$ from which we infer $\lambda_{\mathbf{p}}=\mathbf{p}$. We arrive at the following set of equations for ψ_u and $\psi_{\mathbf{p}}$

$$\begin{aligned}
&\boldsymbol{a}\cdot\nabla u-\nabla^2u-f=0,\quad u|_{\partial\Omega_D}=g_D\quad(\nabla u\cdot\mathbf{n})|_{\partial\Omega_N}=g_N\quad\text{(primal problem)}\\
&\boldsymbol{a}\cdot\nabla\psi_u+\nabla^2\psi_u-(\kappa f+f_0)+2\kappa\boldsymbol{a}\cdot\nabla u=0,\quad\psi_u|_{\partial\Omega_D}=-\kappa g_D+g_0,\\
&(\nabla\psi_v\cdot\mathbf{n})|_{\partial\Omega_N}=-\kappa g_N+f_0^S\quad\text{(“lifted” dual problem)}\\
&\text{also,}\quad\psi_{\mathbf{p}}=-2\kappa\mathbf{p}-\boldsymbol{a}\psi_u-\nabla\psi_u,\quad\mathbf{p}=\nabla u.
\end{aligned} \tag{4.21}$$

As in the Poisson implementation we saw in the previous chapter, the set of equations given above would, in the limit of infinite-dimensional discretization, produce solutions $\mathbf{u}$ and $\boldsymbol{\Psi}_{\mathbf{u}}$ which fulfill all the necessary conditions put forth in (4.10) and (4.12) and result in the exact bound for S. When working with finite-dimensional approximations, however, one must modify numerical solutions $\mathbf{u}_h$ and $\boldsymbol{\Psi}_{\mathbf{u}_h}$ before they can be used to evaluate (4.14) as the aforementioned conditions are no longer automatically satisfied.

4.4 Computation of $\Psi_{\mathbf{u}}$: Finite-Dimensional Case

We now proceed to compute $\Psi_{\mathbf{u}_h}$ such that the conditions in (4.13) are met. It is convenient to first define $\zeta_{\mathbf{u}} = [\zeta_u, \zeta_{\mathbf{p}}]^T$, where

$$\zeta_u \equiv \kappa u + \psi_u, \quad \zeta_{\mathbf{p}} \equiv -\kappa \mathbf{p} - \psi_{\mathbf{p}} \quad \text{(dual solution)}. \tag{4.22}$$

We then solve

$$\begin{aligned}
\nabla \cdot (\boldsymbol{a}u) - \nabla^2 u - f = 0 \quad &\text{in} \quad \Omega \\
u = g_D \quad &\text{on} \quad \partial\Omega_D \\
\nabla u = g_N \quad &\text{on} \quad \partial\Omega_N \qquad \text{(Primal Problem)} \\
-\nabla \cdot (\boldsymbol{a}\zeta_u) - \nabla^2 \zeta_u + f_0^V = 0 \quad &\text{in} \quad \Omega \\
\zeta_u = g_0 \quad &\text{on} \quad \partial\Omega_D \\
\nabla \zeta_u = f_0^S \quad &\text{on} \quad \partial\Omega_N \qquad \text{(Dual Problem)}
\end{aligned} \tag{4.23}$$

with LDG and obtain $u_h, \hat{\mathbf{p}}_h, \hat{h}(u_h)$, and $\zeta_{u_h}, \hat{\zeta}_{\mathbf{p}_h}, \hat{h}(\zeta_{u_h})$. We average u_h, ζ_{u_h} at all elemental vertices and interfaces, as outlined in section 3.4, to obtain $\tilde{u}_h, \tilde{\zeta}_{u_h}$; and to satisfy the necessary condition of continuous ψ_{u_h}, we set

$$\boxed{\psi_{u_h} = -\kappa \tilde{u}_h + \tilde{\zeta}_{u_h}.} \tag{4.24}$$

4.4.1 Elemental Reconstruction

We recall that $\psi_{\mathbf{p}}$ must satisfy

$$\nabla \cdot \psi_{\mathbf{p}} - \kappa f + f_0^V = 0, \quad [\psi_{\mathbf{p}} \cdot \mathbf{n}]|_\Gamma = 0, \quad (\psi_{\mathbf{p}} \cdot \mathbf{n})|_{\partial\Omega_N} = -\kappa g_N + f_0^S.$$

To obtain $\psi_{\mathbf{p}}$, we start by making the following observations regarding LDG solutions

- $\hat{p}_h + \hat{h}(u_h), \hat{\zeta}_{\mathbf{p}_h} + \hat{h}(\zeta_h)$ are unique across elemental interfaces

- $\int_{\partial\Omega_j}(\hat{p}_h + \hat{h}(u_h))ds = -\int_{\Omega_j} f d\mathbf{x}$ (seen by setting test function to one on Ω_j and zero elsewhere)
- $\int_{\partial\Omega_j}(\hat{\zeta}_{\mathbf{p}_h} + \hat{h}(\zeta_h))ds = \int_{\Omega_j} f_0^V d\mathbf{x}$ (seen by setting test function to one on Ω_j and zero elsewhere)

We can therefore satisfy all the necessary conditions by solving *locally*, in each element

$$\nabla \cdot \tilde{\mathbf{p}}_h - f = 0, \quad \nabla \cdot \tilde{\zeta}_{\mathbf{p}_h} + f_0 = 0 \tag{4.25}$$

and by imposing

$$\tilde{\mathbf{p}}_h \cdot \mathbf{n} = \hat{p} + \hat{h}(u_h), \quad \tilde{\zeta}_{\mathbf{p}_h} \cdot \mathbf{n} = \hat{\zeta}_{\mathbf{p}} + \hat{h}(\zeta_h). \tag{4.26}$$

Finally, we set

$$\boxed{\psi_{\mathbf{p}_h} = -\kappa\tilde{\mathbf{p}}_h - \tilde{\zeta}_{\mathbf{p}_h}.} \tag{4.27}$$

The resulting $\psi_{u_h}, \psi_{\mathbf{p}_h}$, given by equations (4.24) and (4.27) satisfy the conditions in (4.10) and (4.12) in a point-wise manner as long as f and f_0^V are of polynomial form on Ω_j and a sufficiently high degree polynomial is used in the the discretization. This property allows us to bound the output of the exact solution. We then summarize the steps for the computation of $\mathbf{\Psi}_{\mathbf{u}_h}$ as follows

We point out here that the entire algorithm requires only two global solutions; to obtain approximations for the primal and dual problems. The postprocessing steps involve only *local* calculations. Had the definition of the Lagrangian not included Θ, we would have to perform *two additional* global solves to satisfy all the necessary conditions to produce bounds for S.

1. Solve (4.23) by LDG for $u_h, \hat{p}_h, \hat{h}(u_h)$ and $\zeta_{u_h}, \hat{\zeta}_{\mathbf{p}_h}, \hat{h}(\zeta_{u_h})$
2. Average u_h and ζ_{u_h} at all elemental vertices and interfaces to obtain $\tilde{u}_h$ and $\tilde{\zeta}_{u_h}$
3. Solve (4.25) *locally*, in each element and impose (4.26) at all elemental boundaries and obtain $\tilde{\mathbf{p}}_h$ and $\tilde{\zeta}_{\mathbf{p}_h}$
4. Set $\psi_{u_h} = -\kappa\tilde{u}_h + \tilde{\zeta}_{u_h}$
5. Set $\psi_{\mathbf{p}_h} = -\kappa\tilde{\mathbf{p}}_h - \tilde{\zeta}_{\mathbf{p}_h}$

Algorithm 2: Computation of $\boldsymbol{\Psi}_{\mathbf{u}_h}$

4.5 Computation of $S^+, \Delta S$ and κ

The upper bound, S^+, is acquired by minimizing $\tilde{\mathcal{L}}$, the Lagrangian defined with the negative output, just as shown in section 3.5 for the purely elliptic case. The ingredients required for the computation of upper bounds are the postprocessed primal solution and the negative of the postprocessed dual solution, both of which we have from lower bound calculations. The upper bounds are thus obtained with *no additional* cost. And like before, the bound gap can be decomposed into contributions from individual elements. For the convection-diffusion case, the bound gap is given by

$$\begin{aligned}\Delta S &= \sum_{j=1}^{N_e} \int_{\Omega_j} \frac{\kappa}{2}(\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - \tilde{\mathbf{p}}_h)\cdot(\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - \tilde{\mathbf{p}}_h) \\ &+ \frac{1}{2\kappa}(\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} - \tilde{\zeta}_{\mathbf{p}_h})\cdot(\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} - \tilde{\zeta}_{\mathbf{p}_h})d\mathbf{x}. \quad (4.28)\end{aligned}$$

Finally, κ is obtained by minimizing ΔS and is given by

$$\kappa = \sqrt{\frac{\sum_{j=1}^{N_e}\int_{\Omega_j}(\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} - \tilde{\zeta}_{\mathbf{p}_h})\cdot(\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} - \tilde{\zeta}_{\mathbf{p}_h})d\mathbf{x}}{\sum_{j=1}^{N_e}\int_{\Omega_j}(\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - \tilde{\mathbf{p}}_h)\cdot(\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - \tilde{\mathbf{p}}_h)d\mathbf{x}}} \quad (4.29)$$

4.6 Error Bound Algorithm Example: Convection-Diffusion Equation

In this example, we solve the following one-dimensional problem using the two-dimensional algorithm.

$$\begin{aligned} \nabla \cdot (\boldsymbol{a}u) - \nabla u^2 = 0, \quad \boldsymbol{a} = [10, 0]^T \quad \text{on} \quad \Omega = [0,1] \times [0,1] \\ u_{x_2}(x_1, x_2) = 0 \quad \text{on} \quad \partial\Omega_w \\ u(0, x_2) = 1, \quad u(1, x_2) = 0 \end{aligned} \tag{4.30}$$

with LDG discretization and choose as output $l^0(\mathbf{u}) = \int_\Omega u d\mathbf{x}$. The problem setup, primal solution and the dual solution are shown in figures (4-1)-(4-3) while the grid convergence results are shown in table (4.1). We see that the optimal convergence rate of h^2 is indeed obtained for linear elements.

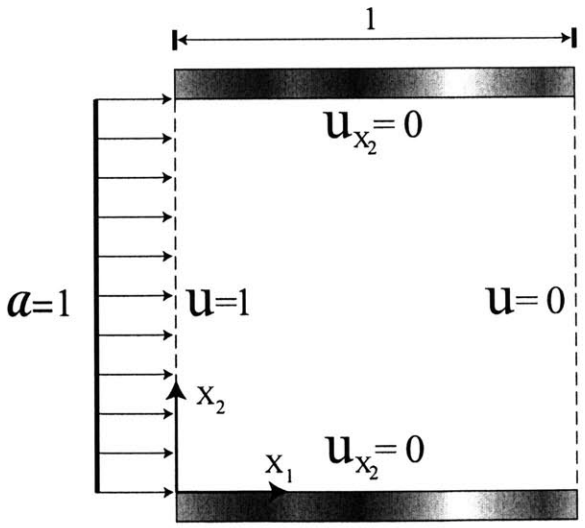


Figure 4-1: Problem Setup: CD1

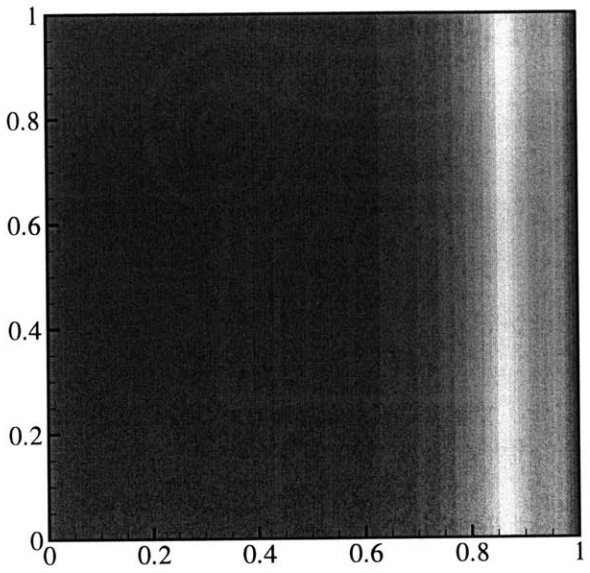


Figure 4-2: Primal solution: CD1, u_h

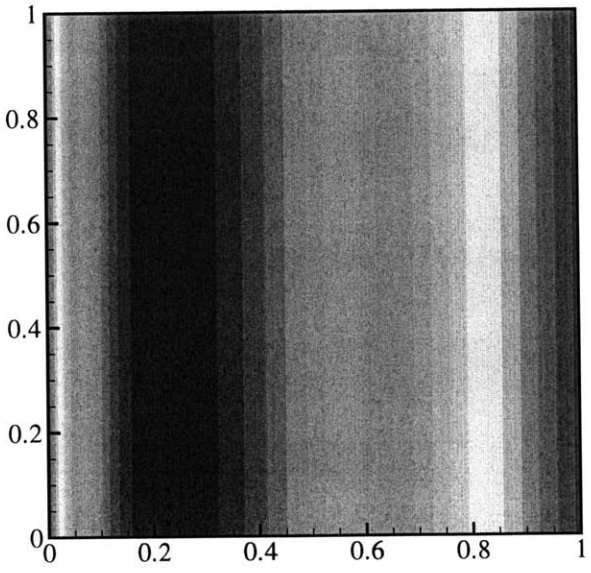


Figure 4-3: Dual solution: CD1, u_h

h	S	S^-	S^+	Order	κ
1/8	0.90	0.863085	0.937002	-	0.0930
1/16	0.90	0.890797	0.909282	2.00	0.0947
1/32	0.90	0.897770	0.902317	2.02	0.0967

Table 4.1: ΔS Grid Convergence: CD1

4.7 High Peclet Number Problems

In the previous sections, we have seen the error bounding algorithm applied to the Poisson and convection-diffusion equations. For outputs of the form $l^0(\mathbf{u}) = \int_\Omega f_0^V u d\mathbf{x}$, the algorithm preformed well and produced upper and lower bounds with a bound gap within a few percent of the exact value even when fairly coarse meshes were used in the numerical computation. We now examine the performance of the error bounding algorithm as applied to the more challenging problem of high Peclet number flows. Such problems are of considerable interest in engineering and computational physics and it would be highly desirable to have an algorithm capable of handling them. We return to the seemingly innocuous 1-D convection-diffusion equation problem

$$\begin{aligned} \nabla \cdot (\boldsymbol{a}u) - \nabla u^2 = 0, \quad \boldsymbol{a} = [\mathrm{Pe}, 0]^T \quad &\text{on} \quad \Omega = [0,1] \times [0,1] \\ u_{x_2}(x_1, x_2) = 0 \quad &\text{on} \quad \partial\Omega_w \\ u(0, x_2) = 1, \quad u(1, x_2) = 0 \end{aligned} \tag{4.31}$$

for various Pe using LDG with $h = 1/8$. The analytical solution is given by $1 - \frac{e^{Pex}-1}{e^{Pe}-1}$ and the results obtained using the previously described algorithm are shown in table (4.2). We see a sharp deterioration in bound quality with increasing Peclet number, to the point where the bounds are essentially meaningless.

4.7.1 High Peclet Number Algorithm: 1-D Analysis

To better understand this behavior, we discretize the same one-dimensional problem in one dimension and carry out the bounding algorithm. In other words, we look at

Pe	S	S^+	S^-
10	0.900	0.932837	0.867206
100	0.990	2.465348	-0.485347
1000	0.999	21.08058	-19.08258

Table 4.2: ΔS vs. Pe

$$\frac{\partial}{\partial x}(\boldsymbol{a}u) - \frac{\partial^2 u}{\partial x^2} = 0 \quad \text{on} \quad \Omega = [0,1], \quad u(0) = 1, \quad u(1) = 0$$

which we re-write as

$$\frac{\partial F_T}{\partial x} = 0, \quad F_T = \boldsymbol{a}u - \frac{\partial u}{\partial x}.$$

We can show that the solution is given by

$$u = 1 - \frac{e^{Pex} - 1}{e^{Pe} - 1}, \quad F_T = 1.$$

We note from (4.13) that $\psi_{\mathbf{p}}$ must satisfy

$$-[\psi_{\mathbf{p}} \cdot \mathbf{n}] = 0, \quad \nabla \cdot \psi_{\mathbf{p}} - \kappa f + f_0^V = 0$$

which in 1-D simply implies continuous $\psi_{\mathbf{p}}$ satisfying the above differential equation. We can again write $\psi_{\mathbf{p}}$ as a weighted sum of primal and adjoint solutions such that

$$\psi_{\mathbf{p}} = \kappa F_T + G_T$$

corresponding to the conservation equations

$$\frac{\partial F_T}{\partial x} - f = 0, \quad \frac{\partial G_T}{\partial x} + f_0^V = 0.$$

We also recall that ψ_u needs to be continuous and satisfy all essential boundary conditions, but is otherwise free to take on any value. We further note that for this test problem, where $f = 0$, the analytical solution for F_T is a constant. This constant is determined by the boundary conditions imposed. Following the steps outlined earlier, we solve

$$\frac{\partial}{\partial x}(\boldsymbol{a}u) - \frac{\partial^2 u}{\partial x^2} = 0, \quad -\frac{\partial}{\partial x}(\boldsymbol{a}\zeta_u) - \frac{\partial^2 \zeta_u}{\partial x^2} + f_0^V = 0 \tag{4.32}$$

and define

$$F_T = \boldsymbol{a}u - \mathbf{p}, \quad G_T = -\boldsymbol{a}\zeta_u - \zeta_{\mathbf{p}}. \tag{4.33}$$

Recalling that the bound gap, ΔS, is given by

$$\Delta S = \sum_{j=1}^{N_e} \int_{\Omega_j} \frac{\kappa}{2}(\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - F_{T_h})^2 + \frac{1}{2\kappa}(\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} + G_{T_h})^2 d\mathbf{x} \tag{4.34}$$

we can split the bound gap into contributions from the primal and adjoint solutions. Focusing on the primal contribution for the moment, we see that bound gap is minimized whenever we have

$$\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - F_{T_h} = 0. \tag{4.35}$$

Since F_T is constant and equal to the total flux, the bound gap would be minimized if $\boldsymbol{a}\tilde{u}_h + \nabla\tilde{u}_h$ is also constant and equal to the total flux. This is indeed the case in the limit of infinite-dimensional discretization and closely approximated when the solution details are

well resolved in finite-dimensional discretizations. When the mesh employed fails to resolve the solution details, however (from the analytical solution it is seen that for large values of Pe a sharp boundary layer develops at $x = 1$), $\boldsymbol{a}\tilde{u}_h + \nabla\tilde{u}_h$ can deviate significantly from the total flux and thus resulting in significant contributions to the bound gap.

We further observe that while $\nabla\tilde{u}_h$ can be rather inaccurate in under-resolved boundary layers, the local flux conservation property of LDG discretization ensures that F_{T_h} is not sensitive to the presence of boundary layers. This is especially true here since F_T is constant. We can therefore expect to obtain accurate F_{T_h} even when very coarse discretizations are employed; an expectation that is confirmed by numerical results. This suggests a simple remedy for the high Peclet number problem; we *locally* refine the solution space for $\tilde{u}_h$ and minimize the quantity

$$\int_{\Omega_j} (\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - F_{T_h})^2 d\mathbf{x} \tag{4.36}$$

for each element in Ω using the F_{T_h} from the coarse mesh computation. Like F_{T_h}, G_{T_h} is also insensitive to the presence of boundary layers and the same strategy may be applied to minimize

$$\int_{\Omega_j} (\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} + G_{T_h})^2 d\mathbf{x}. \tag{4.37}$$

We then have the following steps for the 1-D error bounding algorithm for high Peclet number problems

4.7.2 High Peclet Number Algorithm Example: 1-D Case

We now return to the same 1-D test problem analyzed earlier. We solve (4.32) with boundary conditions $u(0) = 1$ and $u(1) = 0$ for various Pe using LDG with $h = 1/8$ and P_1 elements using the proposed algorithm for high Peclet number problems. The only decision remaining is the manner in which the ψ_v solution space is locally enriched. For this test problem, we

1. Solve primal problem by LDG and obtain F_{T_h}
2. Solve dual problem by LDG and obtain G_{T_h}
3. For each element, Ω_j, locally refine the solution space for $\tilde{u}_h$ and minimize (4.36) subject to the constraint that the degrees of freedom corresponding to the unrefined solution space be frozen at their original values
4. For each element, Ω_j, locally refine the solution space for $\tilde{\zeta}_{u_h}$ and minimize (4.37) subject to the constraint that the degrees of freedom corresponding to the unrefined solution space be frozen at their original values
5. Average u_h and ζ_{u_h} at elemental interfaces and set $\psi_u = -\kappa\tilde{u}_h + \tilde{\zeta}_{u_h}$

Algorithm 3: 1-D High Peclet Number Problem

employ local p-refinement in the solution space of v_h and ζ_{v_h} with Gauss-Legendre-Lobatto nodal basis points. The algorithm is tested using various degrees of local refinement at different Peclet numbers, the results of which are displayed in tables (4.3) and (4.4).

modes	S	S^-	S^+
2	0.99	-0.405652	2.385652
4	0.99	0.880536	1.099464
8	0.99	0.989849	0.990151

Table 4.3: ΔS vs. P_k, Pe=100

modes	S	S^-	S^+
2	0.995	-7.632294	9.628293
4	0.995	-0.573043	2.569043
8	0.995	0.842074	1.153926

Table 4.4: ΔS vs. P_k, Pe=500

In the results above, two modes represents no local refinement while eight modes involve a local P_7 solution space for v_h and ζ_{v_h}. We see that for sufficiently high k, we do indeed recover the ability to obtain meaningful bounds with a coarse global working mesh.

4.7.3 High Peclet Number Algorithm Example: 2-D Case

In the two-dimensional implementation of the high Peclet number algorithm we consider triangular elements. As no simple tensor product basis exists for local p-refinement, we consider local h-refinement instead. Furthermore, unlike the 1-D test case, F_T is *not* constant for multi-dimensional problems; these factors making local h-refinement more attractive, which we therefore employ for triangular elements. As in the 1-D case, we minimize over each element, Ω_j, the quantity

$$\int_{\Omega_j} \frac{\kappa}{2}(\boldsymbol{a}\tilde{u}_h - \nabla\tilde{u}_h - F_{T_h})^2 + \frac{1}{2\kappa}(\boldsymbol{a}\tilde{\zeta}_{u_h} + \nabla\tilde{\zeta}_{u_h} + G_{T_h})^2 d\mathbf{x} \tag{4.38}$$

subject to all necessary constraints in (4.13). Dividing each element, Ω_j, into n_e elements, ω_k such that

$$\Omega_j = \sum_{k=1}^{n_e} \omega_k$$

we minimize (4.38) over Ω_j subject to the conditions that

$$-[\psi_{\mathbf{p}} \cdot \mathbf{n}]|_\gamma = 0, \quad (\nabla \cdot \psi_{\mathbf{p}} - \kappa f + f_0^V)|_{\omega_k} = 0 \tag{4.39}$$

where γ is the set of all internal elemental interfaces in Ω_j. In addition, we require that

$$(F_{T_h} \cdot \mathbf{n})|_{\partial\omega\cap\partial\Omega_j} = \tilde{\mathbf{p}}_h \cdot \mathbf{n}, \quad (G_{T_h} \cdot \mathbf{n})|_{\partial\omega\cap\partial\Omega_j} = \tilde{\zeta}_{\mathbf{p}_h} \cdot \mathbf{n}. \tag{4.40}$$

Unlike the 1-D implementation, here we allow F_{T_h}, G_{T_h} to vary in the interior of Ω_j instead of being frozen at the coarse mesh values.

1. Step through Algorithm 2 to obtain $\tilde{\mathbf{p}}_h$ and $\tilde{\zeta}_{\mathbf{p}_h}$
2. *Locally* refine the solution space for $\tilde{u}_h, F_{T_h}$ and $\tilde{\zeta}_{u_h}, G_{T_h}$ such that $\Omega_j = \sum_{k=1}^{n_e} \omega_k$
3. *Locally*, over each element Ω_j, minimize (4.38) subject to the constraints in (4.39) and (4.40). As in the 1-D implementation, degrees of freedom for u_h, ζ_{u_h} corresponding to the unrefined solution space are frozen at their original values
4. Average u_h and ζ_{u_h} over all elemental vertices and interfaces to obtain $\tilde{u}_h$ and $\tilde{\zeta}_{u_h}$
5. Set $\psi_u = -\kappa\tilde{u}_h + \tilde{\zeta}_{u_h}$ and $\psi_{\mathbf{p}_h} = -\kappa F_{T_h} + G_{T_h}$

Algorithm 4: 2-D High Peclet Number Problem

4.7.4 High Peclet Number Algorithm Example: 2-D Numerical Example

For the 2-D high Peclet number test case we solve

$$\begin{aligned} &\boldsymbol{a} \cdot \nabla u - \nabla u^2 = 0, \quad \boldsymbol{a} = [\mathrm{Pe}, 0]^T \quad \text{on} \quad \Omega = [0,1] \times [0,1] \\ &u(0,y) = 1, \quad u(1,y) = 1, \quad u(x,1) = 0 \end{aligned} \tag{4.41}$$

and for $x = 0$

$$\begin{aligned} u(0,y) &= 1 && \text{if} \quad \epsilon \leq y \leq 1-\epsilon \\ &= \frac{y}{\epsilon} && \text{if} \quad y < \epsilon \\ &= \frac{1-y}{\epsilon} && \text{if} \quad y > 1-\epsilon \end{aligned} \tag{4.42}$$

for $\epsilon = 6.25 \times 10^{-2}$. Using LDG discretization with a global mesh size of $h = 1/16$, the algorithm is tested for $Pe = 100$ and $Pe = 1000$. The output selected here is

$$l^0(\mathbf{u}) = \int_0^1 u(0,y)\frac{\partial u}{\partial x}(1,y)dy,$$

the results of which are shown in tables (4.5) and (4.6), as well as in figures (4-4) and (4-5). Here l_e refers to the local h-refinement of Ω_j, so that $l_e = 4$ implies a local mesh size $1/4$

that of the global working mesh. We see that with the benefit of local h-refinement, we can treat significantly higher Peclet number problems without resolving the global solution. We also point out that for this test case, S_h is seen to be very inaccurate. This is caused by the lack of resolution of the boundary layer at $x_1 = 1$ and the fact that the output of choice depends on the gradient of the boundary layer. Our local optimization strategy produces bounds which are actually significantly closer to the true value of S than S_h.

l_e	S	S_h	S^-	S^+
1	-76.2584	-41.6761	-108.524	-43.0567
4	-76.2584	-41.6761	-80.0124	-72.4304
8	-76.2584	-41.6761	-77.7969	-74.4304

Table 4.5: ΔS vs. l_e: CD2, Pe=100

l_e	S	S_h	S^-	S^+
1	-890.804	-79.9276	-9247.23	7553.74
16	-890.804	-79.9276	-1112.81	-662.954
32	-890.804	-79.9276	-957.879	-819.954

Table 4.6: ΔS vs. l_e: CD2, Pe=1000

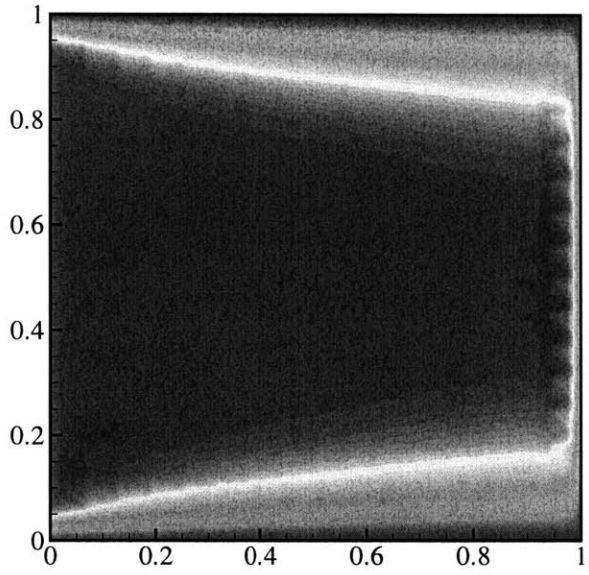


Figure 4-4: $h = 1/16$ primal solution: CD2, Pe=100

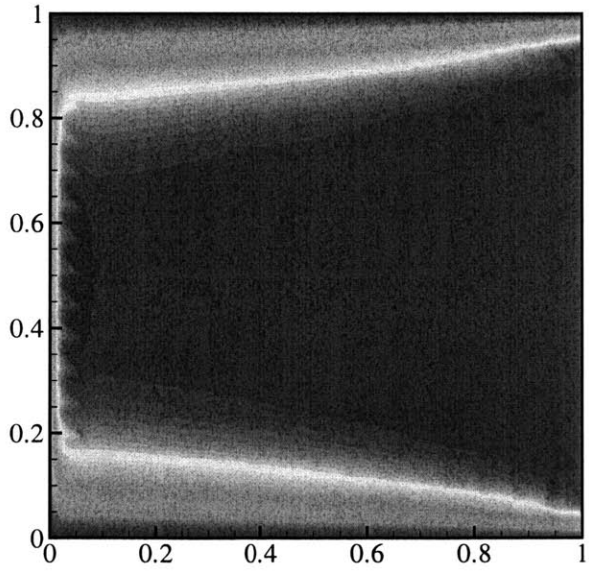


Figure 4-5: $h = 1/16$ dual solution: CD2, Pe=100

Chapter 5

Bounds for Linear Functional Outputs: Symmetric Positive-Definite Systems

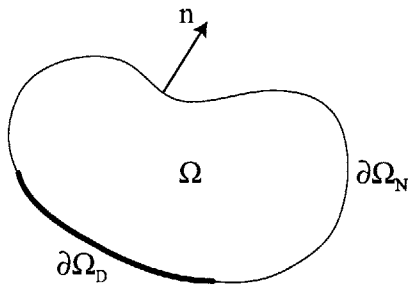
In this chapter, we focus on the implementation of our error bound algorithm for symmetric, positive-definite, systems. For simplicity, we select as our test problem the plane stress/strain model of linear elasticity, even though the methodology is not restricted to two-dimensional problems. The LDG discretization of the governing equations, as we shall see, extends directly from that of its implementation on scalar problems such as the Poisson equation. On the error bounding side of the algorithm, much of what was developed for the Poisson test case applies here, with only minor modifications required to accommodate the fact that we now have a system of governing equations. In the numerical solution of the equations of linear elasticity, the continuous Galerkin finite element method has long been established as the method of choice, as is the case with the numerical solution of all symmetric, positive-definite, problems. In the context of implicit a-*posteriori* error analysis, however, methods based on discontinuous Galerkin schemes offer certain advantages such as avoiding the complicated "equilibration" step, which can be tedious for scalar problems such as the Poisson equation [30] and even more complicated when systems of equations are involved [36], in the error bounding algorithm. It is this reason, what makes exploring DG discretizations for the equations of linear elasticity, worthwhile.

5.1 LDG Discretization: Plane-Stress Model

We start with the strong form of the governing equations

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau} - \boldsymbol{f} = 0 \quad &\text{in} \quad \Omega, \\ \mathbf{u} = \boldsymbol{g}_D \quad &\text{on} \quad \partial\Omega_D, \\ \boldsymbol{\tau} \cdot \mathbf{n} = \boldsymbol{g}_N \quad &\text{on} \quad \partial\Omega_N \end{aligned} \tag{5.1}$$

in domain Ω with boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, as shown below



with $\mathbf{u} = [u_1, u_2]^T$, $\boldsymbol{f} = [f_1, f_2]^T$, $\boldsymbol{g}_D$ the imposed Dirichlet data, $\boldsymbol{g}_N$ the imposed surface traction and

$$\boldsymbol{\tau} \equiv \begin{bmatrix} \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \end{bmatrix} = \boldsymbol{C} : \boldsymbol{\epsilon} \tag{5.2}$$

with $\boldsymbol{\epsilon} = (\nabla\mathbf{u} + (\nabla\mathbf{u})^T)/2$. Here we use the notation $(\nabla \cdot \boldsymbol{\tau})_i = \sum_{j=1}^{2} \partial_j \tau_{ij}$, $(\boldsymbol{C} : \boldsymbol{\epsilon})_{kl} = \sum_{i,j=1}^{2} \tau_{ij} C_{ijkl}$ and $\mathbf{v} \otimes \mathbf{n}$ the matrix whose ij^{th} components is $v_i n_j$. $\boldsymbol{C}$ is a fourth-order $2 \times 2 \times 2 \times 2$ material properties tensor given by

$$\boldsymbol{C} = \frac{E}{1-\nu^2}\begin{pmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \nu \end{bmatrix}_{k=1,l=1} & \begin{bmatrix} 0 & 1-\nu \\ 0 & 0 \end{bmatrix}_{k=1,l=2} \\ \begin{bmatrix} 0 & 0 \\ 1-\nu & 0 \end{bmatrix}_{k=2,l=1} & \begin{bmatrix} \nu & 0 \\ 0 & 1 \end{bmatrix}_{k=2,l=2} \end{pmatrix}. \tag{5.3}$$

Here, each quadrant represents a separate k,l block for $k,l = 1 \ldots 2$. E is the Young's modulus normalized to one and ν the Poisson ratio. Note that $\epsilon_{21} = \epsilon_{12}$ and $\tau_{21} = \tau_{12}$. The system of equations to be solved is then

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau} - \boldsymbol{f} &= 0 \\ \boldsymbol{\tau} - \boldsymbol{C} : \boldsymbol{\epsilon} &= 0. \end{aligned} \tag{5.4}$$

We consider partition T, of the domain Ω, into N_e non-overlapping subdomains such that $\Omega = \sum_{j=1}^{N_e} \Omega_j$ and introduce space $\mathbf{X} = \mathbf{V} \times \boldsymbol{\Sigma}$

$$\begin{aligned} \mathbf{V} &= \{\mathbf{v} \in L_2(\Omega)^2, \mathbf{V}|_{\Omega_j} \in H^1(\Omega_j)^2, \forall \Omega_j \in T\} \\ \boldsymbol{\Sigma} &= \{\boldsymbol{\sigma} \in L_2(\Omega)^4, \boldsymbol{\Sigma}|_{\Omega_j} \in H(div, \Omega_j)^2, \forall \Omega_j \in T\}. \end{aligned} \tag{5.5}$$

We then multiply (5.4) with arbitrary test functions $\mathfrak{v} = [\mathbf{v}, \boldsymbol{\sigma}]^T$ and integrate by parts over each subdomain Ω_j and replace all multi-valued subdomain interface fluxes with unique numerical interface fluxes to obtain

$$\begin{aligned} &\int_{\Omega_j} (\nabla \mathbf{v} : \boldsymbol{\tau} - \mathbf{v} \cdot \boldsymbol{f}) d\mathbf{x} - \int_{\partial\Omega_j} (\mathbf{v} \otimes \mathbf{n}) : \hat{\boldsymbol{\tau}} ds = 0 \\ &\int_{\Omega_j} (\boldsymbol{\sigma} : \boldsymbol{\tau} + (\nabla \boldsymbol{\sigma} \cdot \boldsymbol{C}) \cdot \mathbf{u}) d\mathbf{x} - \int_{\partial\Omega_j} ((\boldsymbol{\sigma} : \boldsymbol{C}) \cdot \hat{\mathbf{u}}) \cdot \mathbf{n} ds, \quad \forall \mathbf{V} \in \mathbf{X} \end{aligned} \tag{5.6}$$

where $(\nabla \boldsymbol{\sigma})_{ijk} = \sigma_{ij,k}$ and $(\nabla \boldsymbol{\sigma} \cdot \boldsymbol{C})_l = \sigma_{ij,k} C_{klij}$. Here $\hat{\boldsymbol{\tau}}$ and $\hat{\mathbf{u}}$ are the now familiar

subdomain interface fluxes. For the definition of these fluxes, we use the *direct* extension from the scalar, Poisson equation implementation and take

$$\begin{aligned} \hat{\boldsymbol{\tau}} &= \frac{1}{2}(\boldsymbol{\tau}^+ + \boldsymbol{\tau}^-) - C_{11}(\mathbf{u}^+ - \mathbf{u}^-) + C_{12}(\boldsymbol{\tau}^+ - \boldsymbol{\tau}^-) \\ \hat{\mathbf{u}} &= \frac{1}{2}(\mathbf{u}^+ + \mathbf{u}^-) - C_{12}(\mathbf{u}^+ - \mathbf{u}^-) \end{aligned} \tag{5.7}$$

where superscript $(-)$ refers to function values to the exterior of the subdomain interface and $(+)$ refers to function values to the interior of the subdomain interface for subdomain Ω_j. Here we set $C_{11} = 0$ and choose

$$C_{12} = \frac{1}{2}\text{sign}(\mathbf{b} \cdot \mathbf{n}).$$

For boundary interfaces, we employ

$$\begin{bmatrix} \hat{\boldsymbol{\tau}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}^+ - \alpha(\mathbf{u}^+ - \boldsymbol{g}_D) \otimes \mathbf{n}^+ \\ \boldsymbol{g}_D \end{bmatrix} \tag{5.8}$$

for Dirichlet boundaries where α is a positive penalization parameter and

$$\begin{bmatrix} \hat{\boldsymbol{\tau}} \cdot \mathbf{n} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_N \\ \mathbf{u}^+ \end{bmatrix} \tag{5.9}$$

for Neumann boundaries.

Defining $\mathbf{u} = [\mathbf{u}, \boldsymbol{\tau}]^T$, we write (5.1) in weak form as

$$a(\mathfrak{v}, \mathbf{u}) = l(\mathfrak{v}), \quad \forall \mathfrak{v} \in \mathbf{X} \tag{5.10}$$

where $a : \mathbf{X} \times \mathbf{X} \mapsto \Re$, $l : \mathbf{X} \mapsto \Re$ are given by

$$
\begin{aligned}
a(\mathfrak{v},\mathfrak{u}) &= a(\mathfrak{v},\mathfrak{u})_{eq.} + a(\mathfrak{v},\mathfrak{u})_{fl.} \\
l(\mathfrak{v}) &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j} \mathbf{v}\cdot\boldsymbol{f}d\mathbf{x} + \int_{\partial\Omega_j\cap\partial\Omega_D}((\mathbf{v}\otimes\mathbf{n}) : C_{11}\boldsymbol{g}_D\otimes\mathbf{n}^+)ds \\
&+ \int_{\partial\Omega_j\cap\partial\Omega_D}((\boldsymbol{\sigma}:\boldsymbol{C})\cdot\boldsymbol{g}_D)\cdot\mathbf{n}^+ds + \int_{\partial\Omega_j\cap\partial\Omega_N}\mathbf{v}\cdot\boldsymbol{g}_N ds\Big\} \qquad (5.11)
\end{aligned}
$$

where $a(\mathfrak{v},\mathfrak{u})_{eq.}$ and $a(\mathfrak{v},\mathfrak{u})_{fl.}$ are the equilibrium and flux components of $a(\mathfrak{v},\mathfrak{u})$, respectively. We have

$$
\begin{aligned}
a(\mathfrak{v},\mathfrak{u})_{eq.} &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}\nabla\mathbf{v} : \boldsymbol{\tau}d\mathbf{x} - \int_{\partial\Omega_j}(\mathbf{v}\otimes\mathbf{n}) : \hat{\boldsymbol{\tau}}ds \\
&- \int_{\partial\Omega_j\cap\partial\Omega_D}(\mathbf{v}\otimes\mathbf{n}) : (\boldsymbol{\tau} - C_{11}\mathbf{u}^+\otimes\mathbf{n}^+)ds\Big\} \\
a(\mathfrak{v},\mathfrak{u})_{fl.} &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}(\boldsymbol{\sigma}:\boldsymbol{\tau} + (\nabla\boldsymbol{\sigma}\cdot\boldsymbol{C})\cdot\mathbf{u})d\mathbf{x} - \int_{\partial\Omega_j\setminus\partial\Omega_j}((\boldsymbol{\sigma}:\boldsymbol{C})\cdot\hat{\mathbf{u}})\cdot\mathbf{n}ds \\
&- \int_{\partial\Omega_j\cap\partial\Omega_N}((\boldsymbol{\sigma}:\boldsymbol{C})\cdot\mathbf{u}^+)\cdot\mathbf{n}^+ds\Big\} \qquad (5.12)
\end{aligned}
$$

5.2 Energy Balance

Before we proceed to the derivation of the expression for energy balance, the algebra is significantly simplified if (5.4) is first written in "symmetric" form

$$
\begin{aligned}
-\tau_{11,x_1} - \tau_{12,x_2} - f_1 &= 0 \\
-\tau_{12,x_1} - \tau_{22,x_2} - f_2 &= 0 \\
(\tau_{11} - \nu\tau_{22}) - u_{1,x_1} &= 0 \\
(\nu\tau_{22} - \tau_{11}) - u_{2,x_2} &= 0 \\
2(1+\nu)\tau_{12} - (u_{2,x_1} + u_{1,x_2}) &= 0. \qquad (5.13)
\end{aligned}
$$

We then multiply the governing equations, by $\mathbf{u}$ and $\boldsymbol{\tau}$ and integrate by parts over each subdomain Ω_j to arrive at

$$\begin{aligned}
&\int_{\Omega_j}(u_{1,x_1}\tau_{11}+u_{1,x_2}\tau_{12}-u_1 f_1)d\mathbf{x}-\int_{\partial\Omega_j}u_1(\hat{\tau}_{11}n_1+\hat{\tau}_{12}n_2)ds=0\\
&\int_{\Omega_j}(u_{2,x_1}\tau_{21}+u_{2,x_2}\tau_{22}-u_2 f_2)d\mathbf{x}-\int_{\partial\Omega_j}u_2(\hat{\tau}_{21}n_1+\hat{\tau}_{22}n_2)ds=0\\
&\int_{\Omega_j}(\tau_{11}(\tau_{11}-\nu\tau_{22})+\tau_{11,x_1}u_1)d\mathbf{x}-\int_{\partial\Omega_j}\tau_{11}\hat{u}_1 n_1 ds=0\\
&\int_{\Omega_j}(\tau_{22}(\tau_{22}-\nu\tau_{11})+\tau_{22,x_2}u_2)d\mathbf{x}-\int_{\partial\Omega_j}\tau_{22}\hat{u}_2 n_2 ds=0\\
&\int_{\Omega_j}(2(1+\nu)\tau_{12}^2+\tau_{12,x_1}u_2+\tau_{12,x_2}u_1)d\mathbf{x}-\int_{\partial\Omega_j}\tau_{12}(\hat{u}_2 n_1+\hat{u}_1 n_2)ds=0 \qquad (5.14)
\end{aligned}$$

which may be written as

$$\begin{aligned}
&\int_{\Omega_j}((u_1\tau_{11})_{,x_1}+(u_1\tau_{12})_{,x_2}+(u_2\tau_{12})_{,x_1}+(u_2\tau_{22})_{,x_2})d\mathbf{x}-\\
&\int_{\partial\Omega_j}(u_1(\hat{\tau}_{11}n_1+\hat{\tau}_{12}n_2)+\hat{u}_1(\tau_{11}n_1+\tau_{12}n_2))ds+\\
&\int_{\partial\Omega_j}(u_2(\hat{\tau}_{12}n_1+\hat{\tau}_{22}n_2)+\hat{u}_2(\tau_{12}n_1+\tau_{22}n_2))ds+\\
&\int_{\Omega_j}(\tau_{11}^2+\tau_{22}^2+2(1+\nu)\tau_{12}^2-2\nu\tau_{11}\tau_{22}-u_1 f_1-u_2 f_2)d\mathbf{x}=0. \qquad (5.15)
\end{aligned}$$

After summing over all subdomains, we end up with

$$\boxed{\begin{aligned}
\Xi(\mathbf{u},\mathbf{u})-l(\mathbf{u}) &= \int_{\Omega}(\tau_{11}^2+\tau_{22}^2+2(1+\nu)\tau_{12}^2-2\nu\tau_{11}\tau_{22}-\mathbf{u}\cdot\boldsymbol{f})d\mathbf{x}\\
&- \int_{\partial\Omega_D}((\boldsymbol{\tau}:\boldsymbol{C})\cdot\boldsymbol{g}_D)\cdot\mathbf{n}ds-\int_{\partial\Omega_N}(\mathbf{u}\otimes\mathbf{n}):\boldsymbol{\tau}_s ds. \qquad (5.16)
\end{aligned}}$$

5.3 Lower Bound: the Lagrangian

The strategy for the development of our algorithm for the governing equations of linear elasticity is much the same as that laid out for the Poisson equation; we first define a Lagrangian such that the output functional of interest may be expressed as the solution to a minimization problem analogous to (3.12), we then proceed to choose Lagrange multipliers such that a bounded minimum results. As we develop our error bounding algorithm for plane-stress problem, we start with a few definitions that would simplify algebra. Given a subdomain interface $\partial\Omega_K$ between subdomains $\Omega_{K^\pm}$, we define

$$[u_i] = (u_i^+ \mathbf{n}^+ + u_i^- \mathbf{n}^-), \quad [\boldsymbol{\tau} \cdot \mathbf{n}] = (\boldsymbol{\tau}^+ \cdot \mathbf{n}^+ + \boldsymbol{\tau}^- \cdot \mathbf{n}^-)$$

where $\mathbf{n}^\pm$ are the outward unit normals on $\partial\Omega_K$ belonging to $\Omega_{K^\pm}$. We consider outputs of the form

$$l^0(\mathbf{u}) = \int_\Omega \boldsymbol{f}_0^V \cdot \mathbf{u} d\mathbf{x} + \int_{\partial\Omega_D} \boldsymbol{g}_0 \cdot (\boldsymbol{\tau} \cdot \mathbf{n}) ds + \int_{\partial\Omega_N} \boldsymbol{f}_0^S \cdot \mathbf{u} ds \\ \forall \boldsymbol{f}_0^V \in L_2(\Omega), \quad \forall \boldsymbol{g}_0, \boldsymbol{f}_0^S \in L_2(\partial\Omega). \tag{5.17}$$

Following the methodology developed for the Poisson equation, we define the Lagrangian, $\mathcal{L} : \mathbf{X} \times \mathbf{X} \rightarrow \Re$ as

$$\mathcal{L}(\boldsymbol{\Psi}, \mathfrak{v}) = \kappa(\Xi(\mathfrak{v}, \mathfrak{v}) - l(\mathfrak{v})) + l^0(\mathfrak{v}) + a(\boldsymbol{\Psi}, \mathfrak{v}) - l(\boldsymbol{\Psi}) \tag{5.18}$$

with $\boldsymbol{\Psi} = [\boldsymbol{\psi}, \boldsymbol{\gamma}]^T$ and $\kappa \geq 0$. The output is then expressed as

$$S = \inf_{\mathbf{v}, \boldsymbol{\sigma} \in \mathbf{X}} \sup_{\boldsymbol{\psi}, \boldsymbol{\gamma} \in \mathbf{X}} \mathcal{L}(\boldsymbol{\Psi}, \mathfrak{v}) \tag{5.19}$$

and the inequality

$$S \geq \inf_{\mathbf{v}, \boldsymbol{\sigma} \in \mathbf{X}} \mathcal{L}(\boldsymbol{\Psi}, \mathfrak{v}) \tag{5.20}$$

can be shown to hold. The adjoint contribution to the Lagrangian is given by

$$\begin{aligned} a(\boldsymbol{\Psi},\mathfrak{v})_j - l(\mathfrak{v})_j &= \int_{\Omega_j} (\psi_{1,x_1}\sigma_{11} + \psi_{1,x_2}\sigma_{12} - \psi_1 f_1)d\mathbf{x} - \int_{\partial\Omega_j} \psi_1(\hat{\sigma}_{11}n_1 + \hat{\sigma}_{12}n_2)ds \\ &+ \int_{\Omega_j} (\psi_{2,x_1}\sigma_{21} + \psi_{2,x_2}\sigma_{22} - \psi_2 f_2)d\mathbf{x} - \int_{\partial\Omega_j} \psi_2(\hat{\sigma}_{21}n_1 + \hat{\sigma}_{22}n_2)ds \\ &+ \int_{\Omega_j} (\gamma_{11}(\sigma_{11} - \nu\sigma_{22}) + \gamma_{11,x_1}v_1)d\mathbf{x} - \int_{\partial\Omega_j} \gamma_{11}\hat{v}_1 n_1 ds \\ &+ \int_{\Omega_j} (\gamma_{22}(\sigma_{22} - \nu\sigma_{11}) + \gamma_{22,x_2}v_2)d\mathbf{x} - \int_{\partial\Omega_j} \gamma_{22}\hat{v}_2 n_2 ds \\ &+ \int_{\Omega_j} (2(1+\nu)\gamma_{12}\sigma_{12} + \gamma_{12,x_1}v_2 + \gamma_{12,x_2}v_1)d\mathbf{x} - \int_{\partial\Omega_j} \gamma_{12}(\hat{v}_2 n_1 + \hat{v}_1 n_2)ds. \end{aligned} \tag{5.21}$$

Before we proceed with the minimization, we set $\alpha \to \infty$, which has the effect of imposing the condition

$$\mathbf{v}|_{\partial\Omega_D} = \boldsymbol{g}_D$$

and producing better bounds for S as we are now minimizing over a smaller space. Setting the variation of $\mathcal{L}$ with respect to $\boldsymbol{\sigma}$ equal to zero leads to the minimizer

$$\begin{aligned} &2\kappa(\sigma_{11} - \nu\sigma_{22}) + (\gamma_{11} - \nu\gamma_{22}) + \psi_{1,x_1} = 0 \\ &2\kappa(\sigma_{22} - \nu\sigma_{11}) + (\gamma_{22} - \nu\gamma_{11}) + \psi_{2,x_2} = 0 \\ &4(1+\nu)\kappa\sigma_{12} + 2(1+\nu)\gamma_{12} + \psi_{2,x_1} + \psi_{1,x_2} = 0 \end{aligned} \tag{5.22}$$

which results in

$$\begin{aligned}
\sigma_{11} &= -\frac{1}{2\kappa}\left\{\frac{\psi_{1,x_1}+\nu\psi_{2,x_2}}{1-\nu^2}+\gamma_{11}\right\}\\
\sigma_{22} &= -\frac{1}{2\kappa}\left\{\frac{\nu\psi_{1,x_1}+\psi_{2,x_2}}{1-\nu^2}+\gamma_{22}\right\}\\
\sigma_{22} &= -\frac{1}{2\kappa}\left\{\frac{\psi_{2,x_1}+\psi_{1,x_2}}{2(1+\nu)}+\gamma_{12}\right\}. \qquad (5.23)
\end{aligned}$$

The minimization also requires the constrains

$$[\psi_i]|_\Gamma = \mathbf{0}, \quad \boldsymbol{\psi}|_{\partial\Omega_D} = -\kappa \boldsymbol{g}_D + \boldsymbol{g}_0 \qquad (5.24)$$

to be satisfied. Setting the variation of $\mathcal{L}$ with respect to $\mathbf{v}$ equal to zero results in the constraints

$$\begin{aligned}
&\gamma_{11,x_1} + \gamma_{12,x_2} - \kappa f_1 + f_{0_1}^V = 0\\
&\gamma_{12,x_1} + \gamma_{22,x_2} - \kappa f_2 + f_{0_2}^V = 0\\
&[\boldsymbol{\gamma}\cdot\mathbf{n}]|_\Gamma = \mathbf{0}, \quad [\boldsymbol{\gamma}\cdot\mathbf{n}]|_{\partial\Omega_N} = -\kappa \boldsymbol{g}_N + \boldsymbol{f}_0^S. \qquad (5.25)
\end{aligned}$$

We then have

$$\inf_{\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v},\boldsymbol{\Psi}) = \begin{cases}
\mathcal{L}^* & \text{if} \quad \gamma_{11,x_1} + \gamma_{12,x_2} - \kappa f_1 + f_{0_1}^V = 0,\\
& \qquad \gamma_{12,x_1} + \gamma_{22,x_2} - \kappa f_2 + f_{0_2}^V = 0,\\
& \qquad [\boldsymbol{\gamma}\cdot\mathbf{n}] = \mathbf{0},\\
& \qquad [\boldsymbol{\psi}] = \mathbf{0},\\
& \qquad \boldsymbol{\psi}|_{\partial\Omega_D} = -\kappa \boldsymbol{g}_D + \boldsymbol{g}_0\\
& \qquad (\boldsymbol{\gamma}\cdot\mathbf{n})|_{\partial\Omega_N} = -\kappa \boldsymbol{g}_N + \boldsymbol{f}_0^S\\
-\infty & \qquad \text{otherwise.}
\end{cases} \qquad (5.26)$$

where $\mathcal{L}^*$ is given by

$$\boxed{\begin{aligned}\mathcal{L}^* \quad &= -\int_\Omega \left\{ \frac{1}{4\kappa}(A(\psi_{1,x_1} + \gamma_{11} - \nu\gamma_{22}) + B(\psi_{2,x_2} + \gamma_{22} - \nu\gamma_{11}) + C^2) + \boldsymbol{\psi}\cdot\boldsymbol{f} \right\} d\mathbf{x} \\ &\quad - \int_{\partial\Omega_D} (\boldsymbol{\gamma}\cdot\boldsymbol{g}_D)\cdot\mathbf{n} ds - \int_{\partial\Omega_N} \boldsymbol{\psi}\cdot\boldsymbol{g}_N ds, \quad \forall \boldsymbol{\Psi} \in \mathbf{X}^C \end{aligned}} \tag{5.27}$$

with $\mathbf{X}^C$ given by the subset of $\mathbf{X}$ satisfying (5.26) and

$$\begin{aligned} A &= \left\{ \frac{\psi_{1,x_1} + \nu\psi_{2,x_2}}{1-\nu^2} + \gamma_{11} \right\} \\ B &= \left\{ \frac{\nu\psi_{1,x_1} + \psi_{2,x_2}}{1-\nu^2} + \gamma_{22} \right\} \\ C &= \sqrt{2(1+\nu)} \left\{ \frac{\psi_{2,x_1} + \psi_{1,x_2}}{2(1+\nu)} + \gamma_{12} \right\}. \end{aligned} \tag{5.28}$$

5.4 Computation of Lagrange Multipliers: Infinite-Dimensional Case

Just as in the scalar Poisson test case, we maximize (5.27) with respect to $\boldsymbol{\psi}$ and $\boldsymbol{\gamma}$ to obtain guidance for the appropriate choices of $\boldsymbol{\psi}$ and $\boldsymbol{\gamma}$. We perform the following constrained maximization

$$\begin{aligned} \sup_{\boldsymbol{\Psi}\in\mathbf{X}} \inf_{\boldsymbol{\Lambda}\in\mathbf{X}} \Big\{ \mathcal{L}^*(\boldsymbol{\psi},\boldsymbol{\gamma}) \quad &+ \quad \int_\Omega \boldsymbol{\lambda}\cdot\mathbf{r} d\mathbf{x} - \int_\Gamma ((\boldsymbol{\phi}\cdot\mathbf{n})\cdot[\boldsymbol{\psi}] + \boldsymbol{\lambda}\cdot[\boldsymbol{\gamma}\cdot\mathbf{n}]) ds \\ &- \quad \int_{\partial\Omega_D} (\boldsymbol{\phi}\cdot\mathbf{n})\cdot(\boldsymbol{\psi} + \kappa\boldsymbol{g}_D - \boldsymbol{g}_0) ds \\ &- \quad \int_{\partial\Omega_N} \boldsymbol{\lambda}\cdot(\boldsymbol{\gamma}\cdot\mathbf{n} + \kappa\boldsymbol{g}_N - \boldsymbol{f}_0^S) \Big\} ds \end{aligned} \tag{5.29}$$

where $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}, \boldsymbol{\phi}]^T$. $\mathbf{r} = [r_1, r_2]^T$ is given by

$$\begin{aligned} r_1 &= \gamma_{11,x_1} + \gamma_{12,x_2} - \kappa f_1 + f_{0_1}^V \\ r_2 &= \gamma_{12,x_1} + \gamma_{22,x_2} - \kappa f_2 + f_{0_2}^V. \end{aligned} \tag{5.30}$$

Maximizing $\mathcal{L}^*$ over $\boldsymbol{\gamma}$ leads to

$$
\begin{aligned}
\delta\gamma_{11}: \quad & \int_\Omega (\tau_{11} - \nu\tau_{22})\delta\gamma_{11} d\mathbf{x} - \int_\Omega \bar{\lambda}_{1,x_1}\delta\gamma_{11} d\mathbf{x} \\
& + \int_{\partial\Omega_D} (\bar{\lambda}_1 - g_1) n_1 \delta\gamma_{11} ds = 0, \\
\delta\gamma_{22}: \quad & \int_\Omega (\tau_{22} - \nu\tau_{11})\delta\psi_{q_2} d\mathbf{x} - \int_\Omega \bar{\lambda}_{2,x_2}\delta\gamma_{22} d\mathbf{x} \\
& + \int_{\partial\Omega_D} (\bar{\lambda}_2 - g_1) n_2 \delta\gamma_{22} ds = 0, \\
\delta\gamma_{12}: \quad & \int_\Omega (1+\nu)\tau_{12}\delta\gamma_{12} d\mathbf{x} - \int_\Omega (\bar{\lambda}_{1,x_2} + \bar{\lambda}_{2,x_1})\delta\gamma_{12} d\mathbf{x} \\
& + \int_{\partial\Omega_D} ((\bar{\lambda}_1 - g_1) n_1 + (\bar{\lambda}_2 - g_2) n_1)\delta\gamma_{12} ds = 0, \qquad \forall \delta\boldsymbol{\gamma} \in \mathbf{X}
\end{aligned} \tag{5.31}
$$

from which it follows that

$$
\begin{aligned}
\bar{\lambda}_{1,x_1} &= (\tau_{11} - \nu\tau_{22}), \quad \bar{\lambda}_1|_{\partial\Omega_D} = g_1 \\
\bar{\lambda}_{2,x_2} &= (\tau_{22} - \nu\tau_{11}), \quad \bar{\lambda}_2|_{\partial\Omega_D} = g_2
\end{aligned}
$$

and

$$
(1+\nu)\tau_{12} + (\bar{\lambda}_{2,x_1} + \bar{\lambda}_{1,x_2}) = 0.
$$

Maximizing $\mathcal{L}^*$ over $\boldsymbol{\psi}$ leads to

$$
\begin{aligned}
\delta\psi_1: \quad & \int_\Omega \frac{\tau_{11}-\nu\tau_{22}}{1-\nu^2}\delta\psi_{1,x_1}d\mathbf{x} + \int_\Omega \tau_{12}\delta\psi_{1,x_2}d\mathbf{x} - \int_\Omega f_1\delta\psi_1 d\mathbf{x} + \\
& \int_\Omega \frac{\tau_{22}-\nu\tau_{11}}{1-\nu^2}\nu\delta\psi_{1,x_1}d\mathbf{x} - \int_\Gamma (\bar{\phi}_{11}n_1 + \bar{\phi}_{12}n_2)[\delta\psi_1]ds - \\
& \int_{\partial\Omega_N} g_{N_1}n_1\delta\psi_1 ds - \int_{\partial\Omega_D}(\bar{\phi}_{11}n_1 + \bar{\phi}_{12}n_2)\delta\psi_1 ds = 0, \\
\delta\psi_2: \quad & \int_\Omega \frac{\tau_{22}-\nu\tau_{11}}{1-\nu^2}\delta\psi_{2,x_2}d\mathbf{x} + \int_\Omega \tau_{12}\delta\psi_{2,x_1}d\mathbf{x} - \int_\Omega f_2\delta\psi_2 d\mathbf{x} + \\
& \int_\Omega \frac{\tau_{11}-\nu\tau_{22}}{1-\nu^2}\nu\delta\psi_{2,x_2}d\mathbf{x} - \int_\Gamma (\bar{\phi}_{12}n_1 + \bar{\phi}_{22}n_2)[\delta\psi_2]ds - \\
& \int_{\partial\Omega_N} g_{N_2}n_2\delta\psi_2 ds - \int_{\partial\Omega_D}(\bar{\phi}_{21}n_1 + \bar{\phi}_{22}n_2)\delta\psi_2 ds = 0, \qquad \forall\boldsymbol{\psi}\in\mathbf{X}
\end{aligned}
$$

which may be written as

$$
\begin{aligned}
\delta\psi_1: \quad & -\int_\Omega (\tau_{11,x_1} + \tau_{12,x_2} + f_1)\delta\psi_1 d\mathbf{x} + \int_{\partial\Omega}(\tau_{11}n_1 + \tau_{12}n_2)\delta\psi_1 ds \\
& + \int_\Gamma (\tau_{1j}n_j - \phi_{1j}n_j)[\delta\psi_1]ds - \int_{\partial\Omega_N}(\tau_{1j}n_j - g_{N_1}n_1)\delta\psi_1 ds \\
& + \int_{\partial\Omega_D}(\tau_{1j}n_j - \bar{\phi}_{1j}n_j)\delta\psi_1 ds = 0, \\
\delta\psi_2: \quad & -\int_\Omega (\tau_{12,x_1} + \tau_{22,x_2} - f_2)\delta\psi_2 d\mathbf{x} + \int_{\partial\Omega}(\tau_{12}n_1 + \tau_{22}n_2)\delta\psi_2 ds \\
& + \int_\Gamma (\tau_{2j}n_j - \phi_{2j}n_j) - [\delta\psi_2]ds - \int_{\partial\Omega_N}(\tau_{2j}n_j - g_{N_2}n_2)\delta\psi_2 ds \\
& + \int_{\partial\Omega_D}(\tau_{2j}n_j - \bar{\phi}_{2j}n_j)\delta\psi_2 ds = 0, \qquad \forall\boldsymbol{\psi}\in\mathbf{X}
\end{aligned} \tag{5.32}
$$

from equations (5.31)-(5.32) we arrive at

$$
\boxed{
\begin{aligned}
& -\frac{1}{1-\nu^2}(\bar{\lambda}_{1,x_1} + \nu\bar{\lambda}_{2,x_2})_{,x_1} - \frac{1}{2(1+\nu)}(\bar{\lambda}_{1,x_2} + \bar{\lambda}_{2,x_1})_{,x_2} - f_1 = 0, \\
& \bar{\lambda}_1|_{\partial\Omega_D} = g_1, \quad \bar{\phi}_{1j}n_j|_{\partial\Omega_N} = g_{N_1}, \quad \phi_{ij} = \tau_{ij} \\
& -\frac{1}{1-\nu^2}(\bar{\lambda}_{2,x_2} + \nu\bar{\lambda}_{1,x_1})_{,x_2} - \frac{1}{2(1+\nu)}(\bar{\lambda}_{1,x_2} + \bar{\lambda}_{2,x_1})_{,x_2} - f_2 = 0, \\
& \bar{\lambda}_2|_{\partial\Omega_D} = g_2, \quad \bar{\phi}_{2j}n_j|_{\partial\Omega_N} = g_{N_2}. \quad \text{(primal problem)}
\end{aligned}
} \tag{5.33}
$$

We recognize from (5.33) that $\mathbf{\Lambda} = \mathbf{U}$. Furthermore, by combining the equilibrium requirements imposed by equations (5.26) with (5.23) we can write

$$\begin{aligned}
&-\left\{\frac{1}{1-\nu^2}(\bar{\psi}_{1,x_1}+\nu\bar{\psi}_{2,x_2})+\bar{\gamma}_{11}\right\}_{,x_1}-\left\{\frac{1}{2(1+\nu)}(\bar{\psi}_{1,x_2}+\bar{\psi}_{2,x_1})+\bar{\gamma}_{12}\right\}_{,x_2}\\
&+2\kappa f_1=0\\
&-\left\{\frac{1}{1-\nu^2}(\bar{\psi}_{2,x_2}+\nu\bar{\psi}_{1,x_1})+\bar{\gamma}_{22}\right\}_{,x_2}-\left\{\frac{1}{2(1+\nu)}(\bar{\psi}_{1,x_2}+\bar{\psi}_{2,x_1})+\bar{\gamma}_{12}\right\}_{,x_1}\\
&+2\kappa f_2=0
\end{aligned} \tag{5.34}$$

which after some simplifications and the application of boundary conditions results in the following familiar set of equations

$$\boxed{\begin{aligned}
&-\frac{1}{1-\nu^2}(\bar{\psi}_{1,x_1}+\nu\bar{\psi}_{2,x_2})_{,x_1}-\frac{1}{2(1+\nu)}(\bar{\psi}_{1,x_2}+\bar{\psi}_{2,x_1})_{,x_2}+\kappa f_1+f_{0_1}^V=0,\\
&\bar{\psi}_1|_{\partial\Omega_D}=-\kappa g_1,\quad \bar{\gamma}_{1j}n_j|_{\partial\Omega_N}=-\kappa g_{N_1}+f_{0_1}^S\\
&-\frac{1}{1-\nu^2}(\bar{\psi}_{2,x_2}+\nu\bar{\psi}_{1,x_1})_{,x_2}-\frac{1}{2(1+\nu)}(\bar{\psi}_{1,x_2}+\bar{\psi}_{2,x_1})_{,x_1}+\kappa f_2+f_{0_2}^V=0,\\
&\bar{\psi}_2|_{\partial\Omega_D}=-\kappa g_2,\quad \bar{\gamma}_{2j}n_j|_{\partial\Omega_N}=-\kappa g_{N_2}+f_{0_2}^S.\quad (\text{“lifted” dual problem})
\end{aligned}} \tag{5.35}$$

It is convenient to define

$$\boldsymbol{\zeta}\equiv\kappa\mathbf{u}+\bar{\boldsymbol{\psi}},\quad \boldsymbol{\chi}\equiv\kappa\boldsymbol{\tau}+\bar{\boldsymbol{\gamma}}\quad (\text{dual solution}).$$

In the limit of infinite-dimensional discretization Equations (5.33), (5.35) and (5.23) produce $\bar{\boldsymbol{\psi}},\bar{\boldsymbol{\gamma}}$ that satisfy the sufficient and necessary conditions of (5.26) and produce the exact bound for S. When working with finite-dimensional spaces, however, $\mathbf{u}_h,\boldsymbol{\tau}_h$ and $\boldsymbol{\zeta}_h,\boldsymbol{\chi}_h$ satisfy (5.26) in only a weak sense and post-processing of the numerical solution is required to produce bounds for the exact solution.

5.5 Computation of $\bar{\Psi}$: Finite-Dimensional Case

We now proceed to the computation of $\bar{\boldsymbol{\Psi}}_h$, such that all the conditions in (5.26) are met. Much of the methodology developed for the Poisson test case is also applicable here. The

fact that we are now dealing with a system rather than a scalar equation, however, requires us to modify the earlier algorithm to accommodate the additional challenges posed. We start by solving the following pair of *global* problems

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau}_h - \boldsymbol{f} &= 0 \quad \text{(primal)} \\ -\nabla \cdot \boldsymbol{\chi}_h + \boldsymbol{f}_0 &= 0 \quad \text{(dual)} \end{aligned} \tag{5.36}$$

to obtain $\mathbf{u}_h, \hat{\tau}$ and $\boldsymbol{\zeta}_h, \hat{\chi}$. We then average $\mathbf{u}_h$ and $\boldsymbol{\zeta}_h$ at all elemental vertices and interfaces to obtain $\tilde{\mathbf{u}}_h$ and $\tilde{\boldsymbol{\zeta}}_h$ in order to satisfy the condition of continuous $\bar{\boldsymbol{\psi}}$. We set

$$\boxed{\bar{\boldsymbol{\psi}}_h = -\kappa \tilde{\mathbf{u}}_h + \tilde{\boldsymbol{\zeta}}_h.} \tag{5.37}$$

5.5.1 Elemental Reconstruction of $\bar{\gamma}$: P_1 Case

We recall that in the error bound algorithm for the Poisson equation, we post-process $\psi_{\mathbf{p}_h}$ by setting the solution value equal to that of the numerical flux such that

$$\int_{\partial\Omega_j} \psi_{\mathbf{p}_h} \cdot \mathbf{n} ds = \int_{\partial\Omega_j} \hat{h} ds, \quad \forall \Omega_j \in T_h$$

to satisfy the equilibrium and continuous normal stress conditions. For P_1 elements, this results in six equations and six unknowns per element that we can then solve for $\psi_{\mathbf{p}_h}$. For plane stress problems, a straight-forward extension of this strategy results in 12 equations and nine unknowns for the same P_1 element. The set of equations produces a singular matrix that has, in general, no solution. Additional steps, must therefore, be employed to generate $\bar{\gamma}_h$ such that the requisite conditions are met. In general, we need to satisfy

$$\begin{aligned} \bar{\gamma}_{11,x_1} + \bar{\gamma}_{12,x_2} - \kappa f_1 + f_{0_1}^V &= 0 \\ \bar{\gamma}_{12,x_1} + \bar{\gamma}_{22,x_2} - \kappa f_2 + f_{0_2}^V &= 0 \quad \text{in} \quad \Omega_j \end{aligned} \tag{5.38}$$

and

$$\begin{aligned} \bar{\gamma}_{11} n_1 + \bar{\gamma}_{12} n_2 &= \hat{\gamma}_{1j} n_j \\ \bar{\gamma}_{12} n_1 + \bar{\gamma}_{22} n_2 &= \hat{\gamma}_{2j} n_j \quad \text{on} \quad \partial\Omega_j \end{aligned} \tag{5.39}$$

over each individual element, Ω_j. When working with P_1 elements, however, satisfying (5.39) insures that (5.38) is also satisfied when $\boldsymbol{f}$ and $\boldsymbol{f}_0^V$ are constant. Unfortunately, as pointed out earlier, (5.39) results in 12 conditions when there are only nine degrees of freedom available in each element to satisfy those conditions. To overcome this difficulty, we subdivide each element into three subelements, as shown in figure (5-1). The requirements imposed by (5.26) must now be satisfied in each subelement and across each subelemental interface. Together with (5.39), we now have 30 equations and 27 unknowns. A quick examination of the set of equations reveals that we have some redundancy in the equation set, and that only 27 of those equations are linearly independent; giving us a system of equations that can readily be solved.

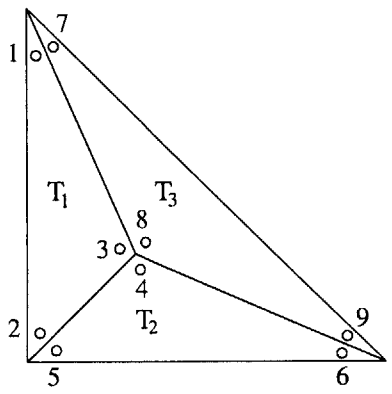


Figure 5-1: P_1 Subelement Layout

5.5.2 Elemental Reconstruction of $\bar{\gamma}$: P_2 Case

When working with P_2 elements, we are confronted with the same difficulty as the P_1 case, with insufficient degrees of freedom to satisfy the necessary constraints. We therefore resort

to the same strategy of dividing each element into three subelements to gain the extra degrees of freedom required to satisfy all the constrains, as shown in figure (5-2). The flux constraints in (5.39) impose 18 conditions while satisfying (5.26) in each of the three subelements and across subelemental interfaces results in an additional 36 conditions for a total of 54 conditions to be satisfied. The three subelements contain 18 nodes and 54 degrees of freedom, exactly equal to the number of constraints. As in the linear case, however, three of the conditions are redundant and we end up with more unknowns than equations. The system of equations have no unique solution and we might try to solve the system by singular value decomposition. Since ψ_h is given by (5.37), the bound gap can be optimized locally with respect to $\bar{\gamma}_h$ through *local* constrained minimization, and this is the approach we recommend and implement in all numerical test cases. Over each element Ω_j, $\bar{\gamma}_h$ is given by

$$\begin{aligned} \min \Delta S(\tilde{\boldsymbol{\tau}}_h, \tilde{\boldsymbol{\chi}}_h) \quad \text{subject to} \quad & -\nabla \cdot \tilde{\boldsymbol{\tau}}_h - \boldsymbol{f} = 0 \\ \text{and} \quad & -\nabla \cdot \tilde{\boldsymbol{\chi}}_h + \boldsymbol{f}_0^V = 0 \\ \text{and} \quad & \tilde{\boldsymbol{\tau}}_h \cdot \mathbf{n}|_{\partial\Omega_j} = \hat{\boldsymbol{\tau}} \cdot \mathbf{n} \\ \text{and} \quad & \tilde{\boldsymbol{\chi}}_h \cdot \mathbf{n}|_{\partial\Omega_j} = \hat{\boldsymbol{\chi}} \cdot \mathbf{n}. \end{aligned} \tag{5.40}$$

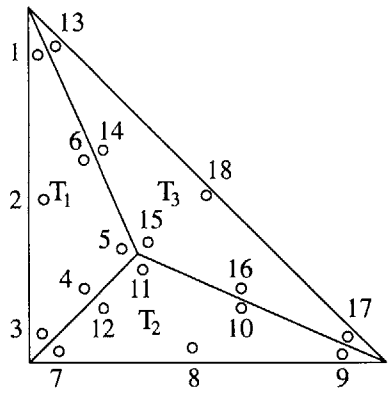


Figure 5-2: P_2 Subelement Layout

We then have the following steps for the computation of $\mathbf{\Psi}_h$

1. Solve (5.36) by LDG for $\mathbf{u}_h, \boldsymbol{\zeta}_h$ and $\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\chi}}$
2. Average $\mathbf{u}_h, \boldsymbol{\zeta}_h$ at all elemental vertices and interfaces to obtain $\tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\zeta}}_h$
3. Subdivide each element, $\Omega_j, \forall \Omega_j \in \Omega$ into three subelements
4. Perform local constrained optimization as given in (5.40)
5. Set $\bar{\boldsymbol{\psi}}_h = -\kappa \tilde{\mathbf{u}}_h + \tilde{\boldsymbol{\zeta}}_h$ and $\bar{\boldsymbol{\gamma}}_h = -\kappa \tilde{\boldsymbol{\tau}}_h - \tilde{\boldsymbol{\chi}}_h$

Algorithm 5: Computation of $\bar{\mathbf{\Psi}}_h$: Plane Stress

5.6 Bound Optimization

From the choice of $\mathbf{\Psi}_h$ outlined in the previous sections, the bound gap is given by

$$\begin{aligned} \Delta S \equiv S^+ - S^- &= \sum_{j=1}^{N_e} \Big\{ \frac{1}{2} \int_{\Omega_j} (\kappa (A_p^2 + B_p^2 - 2\nu A_p B_p + C_p^2) \\ &+ \frac{1}{\kappa} (A_d^2 + B_d^2 - 2\nu A_d B_d + C_d^2)) d\mathbf{x} \Big\} \end{aligned} \tag{5.41}$$

where

$$\begin{aligned} A_p &= \left\{ \frac{\tilde{u}_{1,x_1} + \nu \tilde{u}_{2,x_2}}{1-\nu^2} - \tau_{11} \right\} \\ B_p &= \left\{ \frac{\nu \tilde{u}_{1,x_1} + \tilde{u}_{2,x_2}}{1-\nu^2} - \tau_{22} \right\} \\ C_p &= 2(1+\nu) \left\{ \frac{\tilde{u}_{2,x_1} + \tilde{u}_{1,x_2}}{2(1+\nu)} - \tau_{12} \right\} \\ A_d &= \left\{ \frac{\tilde{\zeta}_{1,x_1} + \nu \tilde{\zeta}_{2,x_2}}{1-\nu^2} - \chi_{11} \right\} \\ B_d &= \left\{ \frac{\nu \tilde{\zeta}_{1,x_1} + \tilde{\zeta}_{2,x_2}}{1-\nu^2} - \chi_{22} \right\} \\ C_d &= 2(1+\nu) \left\{ \frac{\tilde{\zeta}_{2,x_1} + \tilde{\zeta}_{1,x_2}}{2(1+\nu)} - \chi_{12} \right\}. \end{aligned}$$

With the expression for bound gap in hand, it is a simple matter of setting $\frac{\partial \Delta S}{\partial \kappa} = 0$ to optimize the bound with respect to κ, and we arrive at

$$\boxed{\kappa = \sqrt{\frac{\sum_{j=1}^{N_e} \int_{\Omega_j} A_p^2 + B_p^2 - 2\nu A_p B_p + C_p^2 d\mathbf{x}}{\sum_{j=1}^{N_e} \int_{\Omega_j} A_d^2 + B_d^2 - 2\nu A_d B_d + C_d^2 d\mathbf{x}}}.} \tag{5.42}$$

5.7 Error Bound Algorithm Example: Plane-Stress Convergence

We now turn our attention to the performance of the error bound algorithm we have developed for the plane-stress model. For this test case we employ P_2 elements and examine the following setup

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau} - \boldsymbol{f} = 0 \quad \text{on} \quad & \Omega = [0,1] \times [0,1] \\ \text{with} \quad & f_1 = 0, \quad f_2 = x_2 - 1 \\ \text{and} \quad & \tau_{11} = \tau_{12} = 0|_{x_1=0,1} \\ \text{and} \quad & u_2 = \tau_{12} = 0|_{x_2=1} \\ \text{and} \quad & \tau_{12} = 0|_{x_2=0}, \quad \tau_{22} = -\frac{1}{2}\nu x_1^2|_{x_2=0}. \end{aligned}$$

for which the analytical solution is given by

$$\begin{aligned} u_1 &= \nu x_1 (\frac{1}{2}x_2^2 - x_2) + \frac{1}{6}\nu^2 x_1^3 + C \\ u_2 &= -\frac{1}{6}(x_2^3 - 3x_2^2 + 2) - \frac{1}{2}x_1^2(x_2 - 1) \end{aligned} \tag{5.43}$$

where C is an arbitrary constant taken to be zero here. We also note that $\tau_{11} = \tau_{12} = 0$ and $\tau_{22} = -(\frac{1}{2}x_2^2 - x_2) - \frac{1}{2}\nu x_1^2$. The output is taken to be the average x_2-deflection, $\int_\Omega u_2 d\mathbf{x}$. The primal and dual solutions are shown in figures (5-3)-(5-6) and the bound gap grid convergence results shown in table (5.1), from which we see that ΔS exhibits the optimal convergence rate of h^{2k} is obtained.

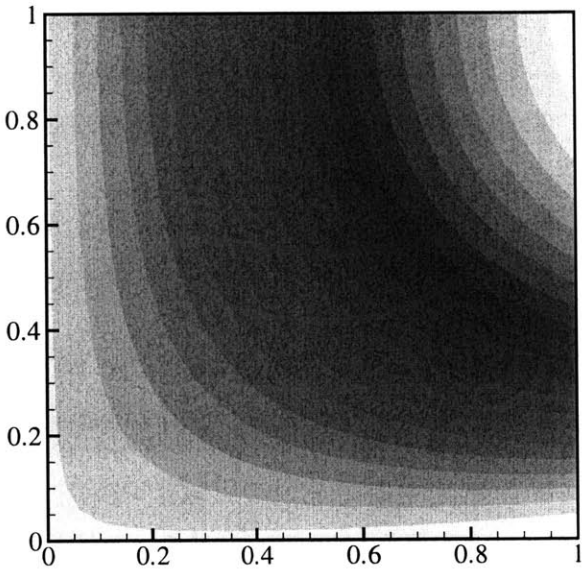


Figure 5-3: $h = 1/16$ u_1 Contour: PS1

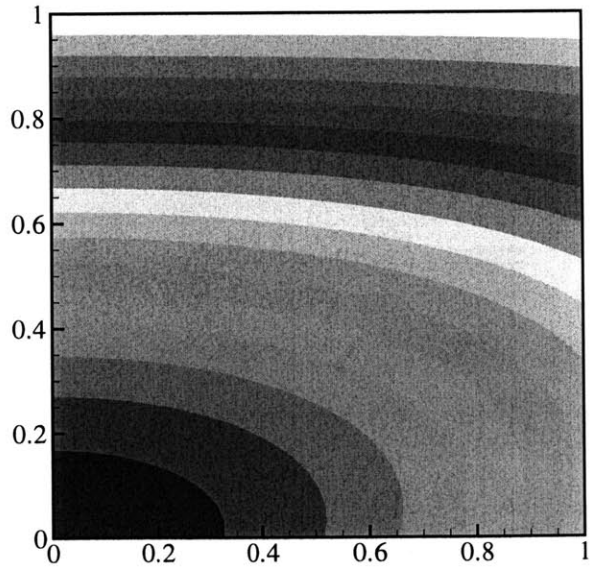


Figure 5-4: $h = 1/16$ u_2 Contour: PS1

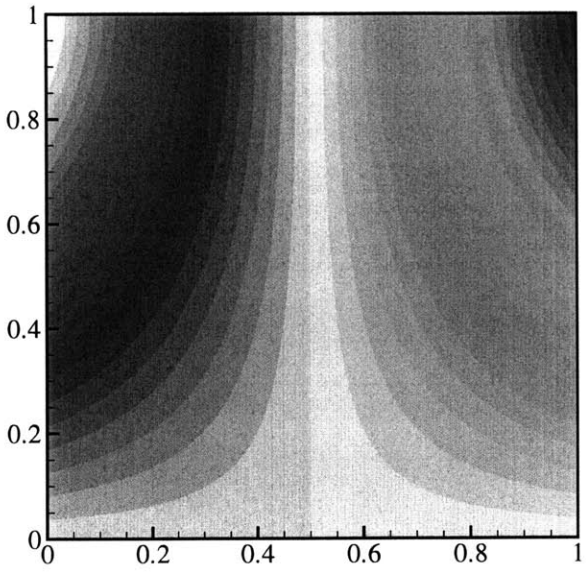


Figure 5-5: $h = 1/16$ ζ_1 Contour: PS1

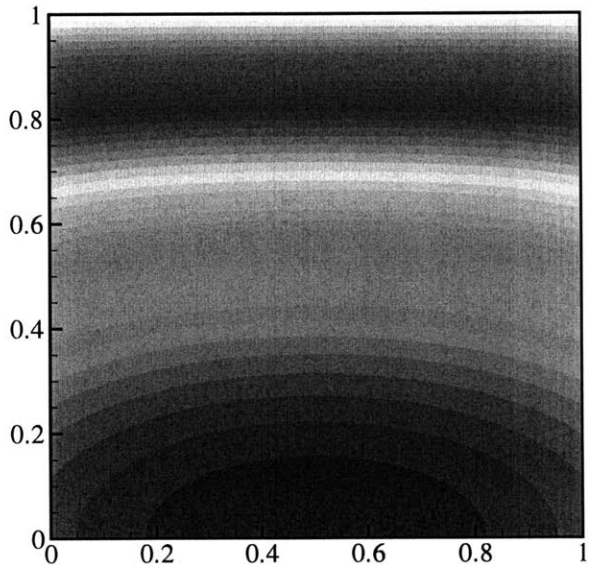


Figure 5-6: $h = 1/16$ ζ_2 Contour: PS1

h	S	S^-	S^+	Order	κ
1/4	$-3/16$	-0.187509843	-0.187491020	-	1.561
1/8	$-3/16$	-0.187500656	-0.187499312	3.81	1.755
1/16	$-3/16$	-0.187500043	-0.187499951	3.87	1.927

Table 5.1: ΔS Grid Convergence: PS1

5.8 Error Bound Algorithm Example: Uniformly Loaded Plate

We now turn to the more challenging problem of uniformly loaded plate with symmetric cut-outs, as shown in figure (5-7). Given the symmetry of the problem, we need to analyze only one-quarter of the domain. Here we use again P_2 element discretization and choose to discretize the positive $x_1 - x_2$ plane, the setup of which is illustrated in figure (5-8). Conditions of zero normal displacement, $\mathbf{u} \cdot \mathbf{n} = 0$ and zero shear stress, $\tau_{12} = 0$ are imposed along the symmetry boundaries along the x_1 and x_2 axis. On all other boundaries we impose $\boldsymbol{\tau} \cdot \mathbf{n} = 0$ with the exception of the loading at $x_1 = 1$. We take the loading $P = 1$ and choose as output the average x_2-deflection on the upper surface of the plate

$$l^0(\mathbf{u}) = \int_0^1 u_2(x_1, 1) dx_1.$$

The presence of corner-singularities prevents the optimal rate of convergence in ΔS from being obtained, even though effective bounds were still acquired; the results of which are shown in table (5.2) and figures (5-9)-(5-12). Here S^* is the output computed from the solution obtained on the finest mesh, containing 15360 elements.

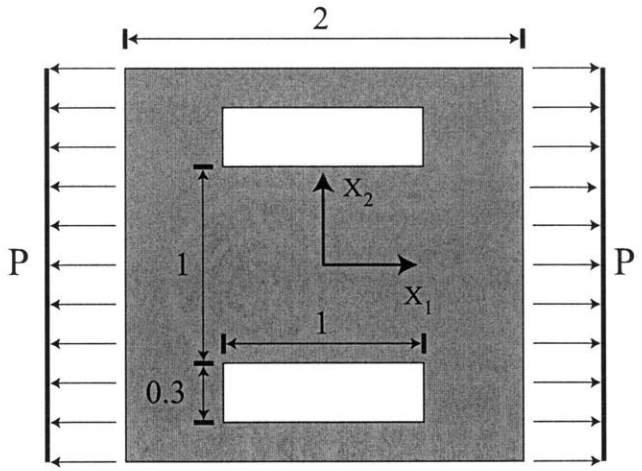


Figure 5-7: Problem Setup: PS2

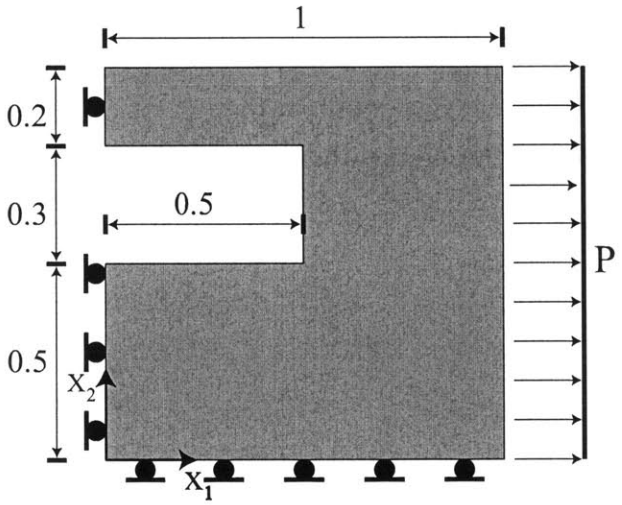


Figure 5-8: Computational Domain: PS2

Elements	S^*	S^-	S^+	Order
240	−0.443167	-0.456544	-0.431764	-
960	−0.443167	-0.445922	-0.440745	2.26
3840	−0.443167	-0.443760	-0.442652	2.22

Table 5.2: ΔS Grid Convergence: PS2

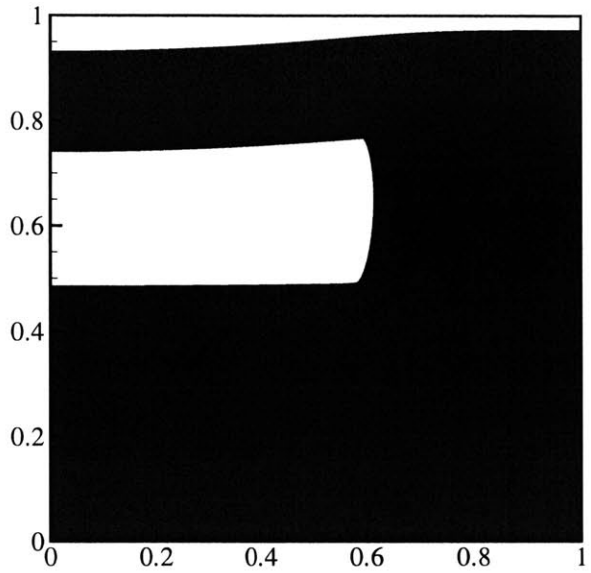


Figure 5-9: Deformed Geometry: PS2

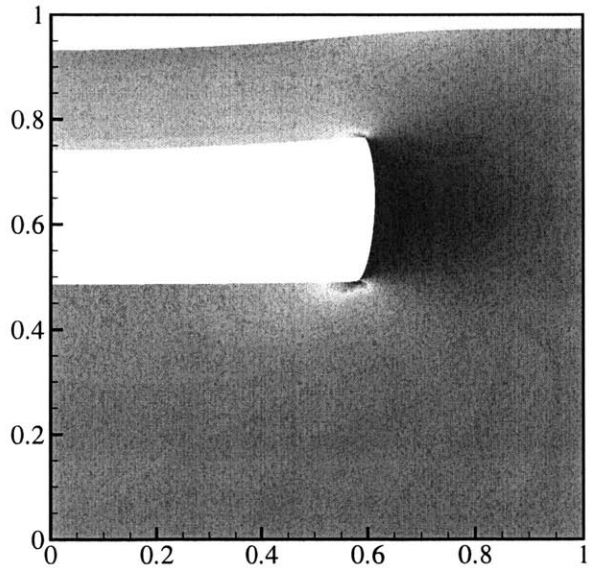


Figure 5-10: τ_{11} Contour: PS2

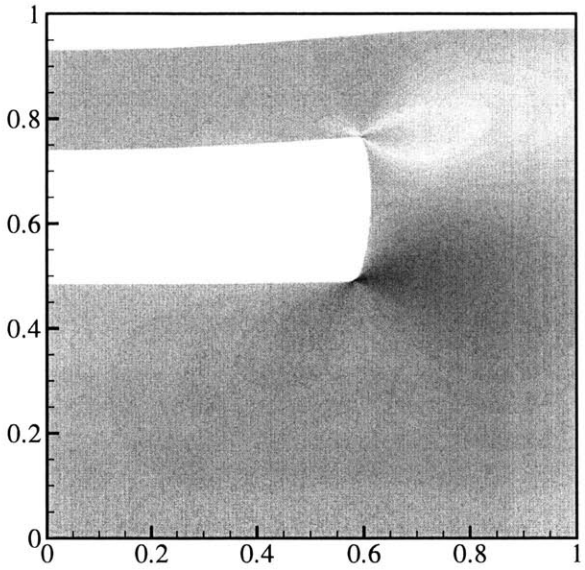


Figure 5-11: τ_{12} Contour: PS2

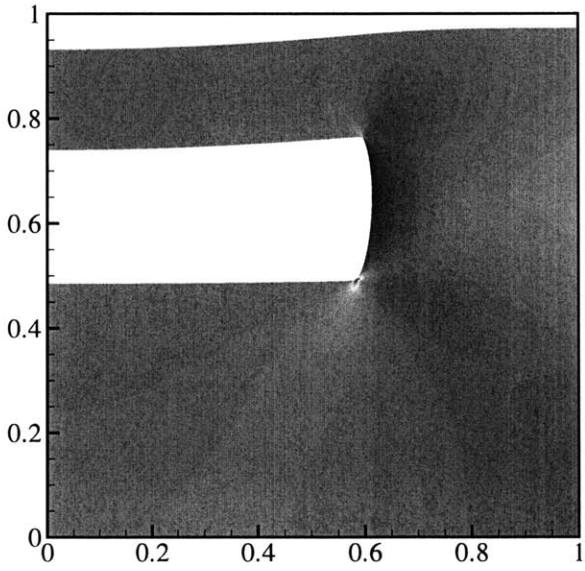


Figure 5-12: τ_{22} Contour: PS2

Chapter 6

Bounds for Linear Functional Outputs: Symmetric Indefinite Systems

In addition to receiving much attention from the compressible computational fluid dynamics community in the past decade and a half, the discontinuous Galerkin method has, in the last several years, commanded increasing interest from the community of *incompressible* computational fluid dynamics. In the numerical solution of the equations of incompressible fluid flow, such as the Stokes or Navier-Stokes equations, the treatment of the incompressibility constraint is an issue that has occupied significant research effort. Within the classical C_0, continuous, Galerkin finite element community, the incompressibility constraint and the resulting stability issues are resolved either through the use of mixed finite element interpolations or the explicit addition of numerical viscosity. The first option is considered to be the more elegant of the two in that mixed interpolation preserves the discretization as a *projection* method. In either instance, modifications are made to the basic continuous equal-order Galerkin discretization to achieve a numerical scheme satisfying the *inf-sup* condition. Numerous elements based on mixed interpolation have been developed and applied successfully in incompressible linear elasticity, Stokes flow and the Navier-Stokes equations. With the local discontinuous Gakerkin alorithm, however, it is possible to employ simple, equal-order interpolations and stabilize incompressibility through proper definition of numerical interface

fluxes while preserving the property of projection method. This was done in [19, 18], where LDG discretization was extended to the Stokes and Oseen equations.

The incompressibility constraint has also been a source of significant difficulty for those seeking to provide error bounds on outputs derived from the solutions to saddle problems such as Stokes flow and the Oseen equations with respect to the exact outputs [30]. Just as we have exploited the capabilities of LDG to produce meaningful error bounds with reference to the exact solution for high Peclet number problems in the previous chapter, so too, do we appeal to the properties of LDG discretization once again to come up with an algorithm to bound the linear functional outputs of saddle problems with respect to those produced by the exact solutions here.

6.1 LDG Discretization

We start with the strong form of the Stokes equation

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau}' + \nabla p - \boldsymbol{f} = 0 \quad &\text{in} \quad \Omega, \\ \nabla \cdot \mathbf{u} = 0 \quad &\text{in} \quad \Omega, \\ \mathbf{u} = \boldsymbol{g}_D \quad &\text{on} \quad \partial\Omega_D, \\ \boldsymbol{\tau} \cdot \mathbf{n} = \boldsymbol{g}_N \quad &\text{on} \quad \partial\Omega_N \end{aligned} \tag{6.1}$$

for Ω in $\Re^2$ with $\mathbf{u} = [u_1, u_2]^T, \mathbf{f} = [f_1, f_2]^T$, $\boldsymbol{g}_D$ the imposed Dirichlet data and $\boldsymbol{g}_N$ the imposed surface traction and

$$\boldsymbol{\tau}' \equiv \begin{bmatrix} \tau'_{11} & \tau'_{12} \\ \tau'_{21} & \tau'_{22} \end{bmatrix} = \mu \begin{bmatrix} 2\frac{\partial u_1}{\partial x_1} & \frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} & 2\frac{\partial u_2}{\partial x_2} \end{bmatrix}. \tag{6.2}$$

Here we use the standard notation $\nabla \cdot \boldsymbol{\tau}' = \sum_{j=1}^{d} \partial_j \tau'_{ij}$ and $\mathbf{v} \otimes \mathbf{n}$ the matrix whose ij^{th} components is $v_i n_j$. Note that $\tau'_{21} = \tau'_{12}$. We introduce space $\mathbf{X} = \mathbf{V} \times Q \times \boldsymbol{\Sigma}$, where

$$
\begin{aligned}
\mathbf{V} &= \{\mathbf{v} \in L_2(\Omega)^2, \mathbf{V}|_{\Omega_j} \in H^1(\Omega_j)^d, \forall \Omega_j \in T\} \\
Q &= \{q \in L_2(\Omega), Q|_{\Omega_j} \in H^1(\Omega_j), \forall \Omega_j \in T\} \\
\mathbf{\Sigma} &= \{\boldsymbol{\sigma}' \in L_2(\Omega)^4, \mathbf{\Sigma}|_{\Omega_j} \in H(div, \Omega_j)^2, \forall \Omega_j \in T\}
\end{aligned} \tag{6.3}
$$

and multiply (6.1) with arbitrary test functions $\mathfrak{v} = [\mathbf{v}, q, \boldsymbol{\sigma}']^T$ and integrate by parts over each subdomain Ω_j. After replacing all multi-valued inter-subdomain fluxes with unique numerical interface fluxes to obtain

$$
\begin{aligned}
&\int_{\Omega_j} (\nabla \mathbf{v} : \boldsymbol{\tau}' - \nabla \cdot \mathbf{v} p - \mathbf{v} \cdot \boldsymbol{f}) d\mathbf{x} - \int_{\partial\Omega_j} ((\mathbf{v} \otimes \mathbf{n}) : \hat{\boldsymbol{\tau}}' - \mathbf{v} \cdot \mathbf{n}\hat{p}) ds = 0 \\
&- \int_{\Omega_j} \nabla q \cdot \mathbf{u} d\mathbf{x} + \int_{\partial\Omega_j} q \hat{\mathbf{u}}_p \cdot \mathbf{n} ds = 0 \\
&\int_{\Omega_j} (\boldsymbol{\sigma}' : \boldsymbol{\tau}' + 2\mu \mathbf{u} \cdot (\nabla \cdot \boldsymbol{\sigma}')) d\mathbf{x} - \int_{\partial\Omega_j} 2\mu (\boldsymbol{\sigma}' \cdot \hat{\mathbf{u}}_{\boldsymbol{\tau}}) \cdot \mathbf{n} ds = 0, \quad \forall \mathfrak{v} \in \mathbf{X}.
\end{aligned} \tag{6.4}
$$

Here, $\hat{\boldsymbol{\tau}}', \hat{p}, \hat{\mathbf{u}}_p$ and $\hat{\mathbf{u}}_{\boldsymbol{\tau}}$ are the numerical subdomain interface fluxes. We also note that even though $\hat{\mathbf{u}}_{\boldsymbol{\tau}}$ and $\hat{\mathbf{u}}_p$ are both associated with the velocity $\mathbf{u}$, they are defined differently as they belong to different conservation laws.

We now proceed with the introduction of the notations required to define these interface fluxes. Given two adjacent subdomains Ω_K^+ and Ω_K^- sharing an interface $\partial\Omega_K^{\pm}$, let $\mathbf{n}^+$ and $\mathbf{n}^-$ be the corresponding outward unit normals at an arbitrary point $\mathbf{x}$ on $\partial K^{\pm}$ and let $\mathbf{u}^{\pm}$ be the trace of $\mathbf{u} = [\mathbf{u}, p, \boldsymbol{\tau}']^T$ on $\partial\Omega_K^{\pm}$ from the interior of $\Omega_K^{\pm}$, we define interface average $\{\cdot\}$ and interface jump $[\cdot]$ values at $\mathbf{x} \in \partial\Omega_K^{\pm}$ as

$$
\begin{aligned}
\{\mathbf{u}\} = (\mathbf{u}^+ + \mathbf{u}^-)/2, \quad \{p\} = (p^+ + p^-)/2, \quad \{\boldsymbol{\tau}'\} = (\boldsymbol{\tau}'^+ + \boldsymbol{\tau}'^-)/2, \\
[\mathbf{u}] = \mathbf{u}^+ \cdot \mathbf{n}^+ + \mathbf{u}^- \cdot \mathbf{n}^-, \quad [p] = p^+ \mathbf{n}^+ + p^- \mathbf{n}^-, \quad [\boldsymbol{\tau}'] = \boldsymbol{\tau}'^+ \cdot \mathbf{n}^+ + \boldsymbol{\tau}'^- \cdot \mathbf{n}^-.
\end{aligned}
$$

We also define a matrix interface jump, $[[\mathbf{u}]]$ as

$$[[\mathbf{u}]] = \mathbf{u}^+ \otimes \mathbf{n}^+ + \mathbf{u}^- \otimes \mathbf{n}^-.$$

We are now ready to define subdoamin interface fluxes. For interior interfaces, we take

$$\begin{aligned}
\begin{bmatrix} \hat{\boldsymbol{\tau}}' \\ \hat{\mathbf{u}}_\tau \end{bmatrix} &= \begin{bmatrix} \{\boldsymbol{\tau}'\} \\ \{\mathbf{u}\} \end{bmatrix} - \begin{bmatrix} C_{11}[[\mathbf{u}]] \\ \mathbf{0} \end{bmatrix} \\
\begin{bmatrix} \hat{p} \\ \hat{\mathbf{u}}_p \end{bmatrix} &= \begin{bmatrix} \{p\} \\ \{\mathbf{u}\} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ D_{11}[p] \end{bmatrix}.
\end{aligned} \tag{6.5}$$

The role of C_{11} and D_{11} is to ensure stability of the method, and thus are known as stability coefficients. In addition to their impact on stability, they can also affect the accuracy of the method; for the implementation here we follow the choice in [19] and set $C_{11} = 1/h$ and $D_{11} = h$. For Dirichlet boundary interfaces, we use the numerical flux

$$\begin{aligned}
\begin{bmatrix} \hat{\boldsymbol{\tau}}' \\ \hat{\mathbf{u}}_\tau \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\tau}'^+ - \alpha(\mathbf{u}^+ - \boldsymbol{g}_D) \otimes \mathbf{n}^+ \\ \boldsymbol{g}_D \end{bmatrix} \\
\begin{bmatrix} \hat{p} \\ \hat{\mathbf{u}}_p \end{bmatrix} &= \begin{bmatrix} p^+ \\ \boldsymbol{g}_D \end{bmatrix}
\end{aligned} \tag{6.6}$$

where superscript $(+)$ refers to function values to the interior of the boundary interface and α is a positive penalization parameter for the purpose of enforcing essential boundary conditions. For Neumann boundary interfaces, we employ

$$\begin{bmatrix} (\hat{\boldsymbol{\tau}}' + \hat{p}\boldsymbol{I}) \cdot \mathbf{n}^+ \\ \hat{\mathbf{u}}_\tau, \hat{\mathbf{u}}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_N \\ \mathbf{u}^+ \end{bmatrix}. \tag{6.7}$$

We note that for Neumann boundaries, the *total* stress is prescribed. We can now write (6.1)

in weak form as

$$a(\mathfrak{v},\mathfrak{u}) = l(\mathfrak{v}), \quad \forall \mathfrak{v} \in \mathbf{X} \tag{6.8}$$

where $a : \mathbf{X} \times \mathbf{X} \mapsto \Re$, $l : \mathbf{X} \mapsto \Re$ are given by

$$\begin{aligned}
a(\mathfrak{v},\mathfrak{u}) &= a(\mathfrak{v},\mathfrak{u})_{eq.} + a(\mathfrak{v},\mathfrak{u})_{fl.} \\
l(\mathfrak{v}) &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j} \mathbf{v}\cdot\boldsymbol{f}d\mathbf{x} + \int_{\partial\Omega_j\cap\partial\Omega_D}((\mathbf{v}\otimes\mathbf{n}) : (\alpha\boldsymbol{g}_D\otimes\mathbf{n}^+) - q\boldsymbol{g}_D\cdot\mathbf{n}^+)ds \\
&+ \int_{\partial\Omega_j\cap\partial\Omega_D} 2\mu(\boldsymbol{\sigma}'\cdot\boldsymbol{g}_D)\cdot\mathbf{n}^+ds + \int_{\partial\Omega_j\cap\partial\Omega_N}\mathbf{v}\cdot\boldsymbol{g}_N ds\Big\}
\end{aligned} \tag{6.9}$$

where $a(\mathfrak{v},\mathfrak{u})_{eq.}$ and $a(\mathfrak{v},\mathfrak{u})_{fl.}$ are the equilibrium and flux components of $a(\mathfrak{v},\mathfrak{u})$, respectively. We have

$$\begin{aligned}
a(\mathfrak{v},\mathfrak{u})_{eq.} &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}(\nabla\mathbf{v} : \boldsymbol{\tau}' - \nabla\cdot\mathbf{v}p)d\mathbf{x} - \int_{\partial\Omega_j}((\mathbf{v}\otimes\mathbf{n}) : \hat{\boldsymbol{\tau}}' - \mathbf{v}\cdot\mathbf{n}\hat{p})ds \\
&- \int_{\Omega_j}\nabla q\cdot\mathbf{u}d\mathbf{x} + \int_{\partial\Omega_j\setminus\partial\Omega} q\hat{\mathbf{u}}_p\cdot\mathbf{n}ds + \int_{\partial\Omega_j\cap\partial\Omega_N} q\mathbf{u}^+\cdot\mathbf{n}^+ds \\
&- \int_{\partial\Omega_j\cap\partial\Omega_D}((\mathbf{v}\otimes\mathbf{n}) : (\boldsymbol{\tau}'^+ - \alpha\mathbf{u}^+\otimes\mathbf{n}^+) - \mathbf{v}\cdot\mathbf{n}p^+)ds\Big\} \\
a(\mathfrak{v},\mathfrak{u})_{fl.} &= \sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}(\boldsymbol{\sigma}' : \boldsymbol{\tau}' + 2\mu\mathbf{u}\cdot(\nabla\cdot\boldsymbol{\sigma}'))d\mathbf{x} - \int_{\partial\Omega_j\setminus\partial\Omega} 2\mu(\boldsymbol{\sigma}'\cdot\hat{\mathbf{u}}_{\boldsymbol{\tau}})\cdot\mathbf{n}ds \\
&- \int_{\partial\Omega_j\cap\partial\Omega_N} 2\mu(\boldsymbol{\sigma}'\cdot\mathbf{u}^+)\cdot\mathbf{n}^+ds\Big\}.
\end{aligned} \tag{6.10}$$

The LDG algorithm for the Stokes system, with the aforementioned choices of interface fluxes, has been shown in [19] to converge at order h^{k+1} for $\mathbf{u}$ and at order h^k for $\boldsymbol{\tau}'$ and p in the L_2 norm.

6.2 Lower Bound: the Lagrangian

The objective is to produce upper and lower bounds on linear functional outputs of the form

$$l^0(\mathbf{u}) = \int_\Omega \boldsymbol{f}_0^V \cdot \mathbf{u} d\mathbf{x} + \int_{\partial\Omega_D} \boldsymbol{g}_0 \cdot (\boldsymbol{\tau} \cdot \mathbf{n}) ds + \int_{\partial\Omega_N} \boldsymbol{f}_0^S \cdot \mathbf{u} ds$$
$$\forall \boldsymbol{f}_0^V \in L_2(\Omega), \quad \forall \boldsymbol{g}_0, \boldsymbol{f}_0^S \in L_2(\partial\Omega) \tag{6.11}$$

where $\boldsymbol{f}_0^V$, $\boldsymbol{f}_0^S$ and $\boldsymbol{g}_0$ are given functions. The overall strategy is the same as that employed in the Poisson equation and given by the simple example in section (3.2.1). We want to express the output of interest as the solution to a minimization problem analogous to (3.12), relying on the appropriate choices of Lagrange multipliers to ensure that we obtain the analog of case I. Unlike the problems we have thus far encountered, however, the "obvious" formulation of the Lagrangian fails to produce the desired results. We will therefore consider two different formulations of the Lagrangian; starting with the more intuitive approach. The basic components of the Lagrangian remains the same and are given by

$$\mathcal{L}(\boldsymbol{\Psi}, \mathfrak{v}) = \kappa(a(\mathfrak{v}, \mathfrak{v}) - l(\mathfrak{v})) + l^0(\mathfrak{v}) + a(\boldsymbol{\Psi}, \mathfrak{v}) - l(\boldsymbol{\Psi}) \tag{6.12}$$

with $\boldsymbol{\Psi} = [\boldsymbol{\psi}, \xi, \boldsymbol{\gamma}]^T$ and $\kappa \geq 0$. The output is then given by the following constrained minimization statement

$$S = \inf_{\mathbf{v},q,\boldsymbol{\sigma}} \sup_{\boldsymbol{\psi},\xi,\boldsymbol{\gamma}} \mathcal{L}(\boldsymbol{\Psi}, \mathfrak{v}) \tag{6.13}$$

and the inequality

$$S \geq \inf_{\mathbf{v},q,\boldsymbol{\sigma}} \mathcal{L}(\boldsymbol{\Psi}, \mathfrak{v}) \tag{6.14}$$

can be shown to hold. To obtain a lower bound for S, then, one needs to minimize $\mathcal{L}$ over all $\mathfrak{v}$. As with test cases covered in previous chapters, however, only for very specific choices

of Ψ can we actually evaluate $\mathcal{L}$ and produce meaningful lower bounds for S.

6.3 Initial Approach

In our first attempt at producing an error bounding algorithm for the Stokes system, we take a straight-forward extension from the method as developed for the Poisson equation as well as the governing equations of linear elasticity. This approach produces results with which we are, for the most part, familiar with. The fact that we are now dealing with a saddle problem, however, will have significant implications for our algorithm.

6.3.1 Energy Balance

To see what would happen when we apply our error bounding algorithm to the Stokes problem, we start by deriving the expression for energy equality. Setting $\mathfrak{v} = \mathbf{u}$, we have

$$
\begin{aligned}
&\sum_{j=1}^{N_e} \int_{\Omega_j} (u_{1,x_1}(\tau_{11} - p) + u_{1,x_2}\tau_{12} - u_1 f_1)d\mathbf{x} - \int_{\partial\Omega_j} u_1((\hat{\tau}_{11} - \hat{p})n_1 + \hat{\tau}_{12}n_2)ds = 0 \\
&\sum_{j=1}^{N_e} \int_{\Omega_j} (u_{2,x_1}\tau_{21} + u_{2,x_2}(\tau_{22} - p) - u_2 f_2)d\mathbf{x} - \int_{\partial\Omega_j} u_2(\hat{\tau}_{21}n_1 + (\hat{\tau}_{22} - p)n_2)ds = 0 \\
&\sum_{j=1}^{N_e} - \int_{\Omega_j} (p_{,x_1}u_1 + p_{,x_2}u_2)d\mathbf{x} + \int_{\partial\Omega_j} p(\hat{u}_1 n_1 + \hat{u}_2 n_2)ds = 0 \\
&\sum_{j=1}^{N_e} \frac{1}{2\mu}\left\{ \int_{\Omega_j} \tau_{11}^2 + 2\tau_{11,x_1}\mu u_1)d\mathbf{x} - \int_{\partial\Omega_j} 2\tau_{11}\mu\hat{u}_1 n_1 ds \right\} = 0 \\
&\sum_{j=1}^{N_e} \frac{1}{2\mu}\left\{ \int_{\Omega_j} (\tau_{22}^2 + 2\tau_{22,x_2}\mu u_2)d\mathbf{x} - \int_{\partial\Omega_j} 2\tau_{22}\mu\hat{u}_2 n_2 ds \right\} = 0 \\
&\sum_{j=1}^{N_e} \frac{1}{\mu}\left\{ \int_{\Omega_j} (\tau_{12}^2 + \mu(\tau_{12,x_1}u_2 + \tau_{12,x_2}u_1))d\mathbf{x} - \int_{\partial\Omega_j} \mu(\tau_{12}\hat{u}_2 n_1 + \hat{u}_1 n_2)ds \right\} = 0 \qquad (6.15)
\end{aligned}
$$

which after some simplification result in

$$
\begin{aligned}
a(\mathbf{u},\mathbf{u}) - l(\mathbf{u}) &= \int_\Omega \left\{ \frac{1}{2\mu}(\tau_{11}^2 + \tau_{22}^2 + 2\tau_{12}^2) - \mathbf{u}\cdot\boldsymbol{f} \right\} d\mathbf{x} \\
&+ \int_\Gamma (C_{11}[[\mathbf{u}]] : [[\mathbf{u}]] + D_{11}[p]^2) ds - \int_{\partial\Omega_N} \mathbf{u}\cdot\boldsymbol{g}_N ds \\
&+ \int_{\partial\Omega_D} (\alpha(\mathbf{u}^+ \otimes \mathbf{n}^+) : (\mathbf{u}^+ - \boldsymbol{g}_D) \otimes \mathbf{n}^+ - (\boldsymbol{\tau}'\cdot\boldsymbol{g}_D)\cdot\mathbf{n}^+) ds. \qquad (6.16)
\end{aligned}
$$

The adjoint contribution to the Lagrangian is given by

$$
\begin{aligned}
a(\boldsymbol{\Psi},\mathfrak{v}) - l(\boldsymbol{\Psi}) &= \int_\Omega (\psi_{1,x_1}(\sigma_{11} - q) + \psi_{1,x_2}\sigma_{12} - \psi_1 f_1) d\mathbf{x} - \int_{\partial\Omega} \psi_1((\hat{\sigma}_{11} - \hat{q})n_1 + \hat{\sigma}_{12}n_2) ds \\
&+ \int_\Omega (\psi_{2,x_1}\sigma_{21} + \psi_{2,x_2}(\sigma_{22} - q) - \psi_2 f_2) d\mathbf{x} - \int_{\partial\Omega} \psi_2(\hat{\sigma}_{21}n_1 + (\hat{\sigma}_{22} - \hat{q}n_2)) ds \\
&- \int_\Omega (\xi_{,x_1} v_1 + \xi_{,x_2} v_2) d\mathbf{x} + \int_{\partial\Omega} \xi(\hat{v}_1 n_1 + \hat{v}_2 n_2) ds \\
&+ \frac{1}{2\mu}\left\{ \int_\Omega \gamma_{11}\sigma_{11} + 2\gamma_{11,x_1}\mu v_1) d\mathbf{x} - \int_{\partial\Omega} 2\gamma_{11}\mu\hat{v}_1 n_1 ds \right\} \\
&+ \frac{1}{2\mu}\left\{ \int_\Omega \gamma_{22}\sigma_{22} + 2\gamma_{22,x_2}\mu v_2) d\mathbf{x} - \int_{\partial\Omega} 2\gamma_{22}\mu\hat{v}_2 n_2 ds \right\} \\
&+ \frac{1}{\mu}\left\{ \int_\Omega (\gamma_{12}\sigma_{12} + \mu(\gamma_{12,x_1} v_2 + \sigma_{12,x_2} v_1)) d\mathbf{x} - \int_{\partial\Omega} \gamma_{12}\mu(\hat{v}_2 n_1 + \hat{v}_1 n_2) ds \right\}. \qquad (6.17)
\end{aligned}
$$

We are now ready to minimize the Lagrangian. Setting the variation of $\mathcal{L}$ with respect to q equal to zero requires the constraint

$$
-\psi_{1,x_1} - \psi_{2,x_2} = 0 \qquad (6.18)
$$

to be satisfied *pointwise* in the domain, Ω in order to produce a bounded minimum for $\mathcal{L}$. Setting the variation of $\mathcal{L}$ with respect to $\mathbf{v}$ and $\boldsymbol{\sigma}$ equal to zero produces other constraints and minimizers which are completely analogous to that of which we have already encountered and can handle. The incompressibility condition presented by (6.18), however, poses a serious

difficulty. Short of actually producing the exact solution, it is *impossible* to satisfy (6.18) pointwise for any non-trivial problem.

6.4 Proposed Approach

From what we have seen earlier, one needs to define the Lagrangian in such a manner so as to not trigger the incompressibility constraint as given in (6.18) when attempting to evaluate (6.14). This does not imply that the incompressibility constraint cannot appear in any form; but rather that it is necessary to avoid having the condition appear in its strong form such as shown in (6.18). Here we present a method to achieve that.

6.4.1 Energy Balance: Alternative Approach

To develop our error bounding algorithm for the Stokes problem, we need to start by deriving an alternative expression for the energy equality, which lies at the heart of the proposed algorithm. Setting $\mathbf{v} = \mathbf{u}, q = p$ and $\boldsymbol{\sigma} = \boldsymbol{\tau}$ and using the following definition for $\boldsymbol{\tau}$

$$\boldsymbol{\tau} = \begin{bmatrix} 2\mu\frac{\partial u_1}{\partial x_1} - p & \mu(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2}) \\ \mu(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2}) & 2\mu\frac{\partial u_2}{\partial x_2} - p \end{bmatrix}, \tag{6.19}$$

we have, for each element Ω_j

$$\begin{aligned}
&\int_{\Omega_j} (u_{1,x_1}\tau_{11} + u_{1,x_2}\tau_{12} - u_1 f_1)d\mathbf{x} - \int_{\partial\Omega_j} u_1(\hat{\tau}_{11}n_1 + \hat{\tau}_{12}n_2)ds = 0 \\
&\int_{\Omega_j} (u_{2,x_1}\tau_{21} + u_{2,x_2}\tau_{22} - u_2 f_2)d\mathbf{x} - \int_{\partial\Omega_j} u_2(\hat{\tau}_{21}n_1 + \hat{\tau}_{22}n_2)ds = 0 \\
&\frac{1}{2\mu}\left\{\int_{\Omega_j} (\tau_{11}(\tau_{11} + p) + 2\tau_{11,x_1}\mu u_1)d\mathbf{x} - \int_{\partial\Omega_j} 2\tau_{11}\mu\hat{u}_1 n_1 ds\right\} = 0 \\
&\frac{1}{2\mu}\left\{\int_{\Omega_j} (\tau_{22}(\tau_{22} + p) + 2\tau_{22,x_2}\mu u_2)d\mathbf{x} - \int_{\partial\Omega_j} 2\tau_{22}\mu\hat{u}_2 n_2 ds\right\} = 0 \\
&\frac{1}{\mu}\left\{\int_{\Omega_j} (\tau_{12}\tau_{12} + \mu(\tau_{12,x_1}u_2 + \tau_{12,x_2}u_1))d\mathbf{x} - \int_{\partial\Omega_j} \mu(\tau_{12}\hat{u}_2 n_1 + \hat{u}_1 n_2)ds\right\} = 0
\end{aligned} \tag{6.20}$$

which may be written as

$$\sum_{j=1}^{N_e}\Big\{\int_{\Omega_j}((u_1\tau_{11})_{,x_1}+(u_1\tau_{12})_{,x_2}+(u_2\tau_{12})_{,x_1}+(u_2\tau_{22})_{,x_2})d\mathbf{x}-$$
$$\int_{\partial\Omega_j}(u_1(\hat{\tau}_{11}n_1+\hat{\tau}_{12}n_2)+\hat{u}_1(\tau_{11}n_1+\tau_{12}n_2))ds+$$
$$\int_{\partial\Omega_j}(u_2(\hat{\tau}_{12}n_1+\hat{\tau}_{22}n_2)+\hat{u}_2(\tau_{12}n_1+\tau_{22}n_2))ds+$$
$$\int_{\Omega_j}\Big\{\frac{1}{2\mu}(\tau_{11}^2+\tau_{22}^2+2\tau_{12}^2+p(\tau_{11}+\tau_{12}+2p))-u_1f_1-u_2f_2\Big\}d\mathbf{x}\Big\}=0. \quad (6.21)$$

After cancelling contributions from interior integrals with those of boundary integrals, we end up with

$$\begin{aligned}a(\mathbf{u},\mathbf{u})-l(\mathbf{u}) &= \int_\Omega\Big\{\frac{1}{2\mu}(\tau_{11}^2+\tau_{22}^2+2\tau_{12}^2+p(\tau_{11}+\tau_{12}))-\mathbf{u}\cdot\boldsymbol{f}\Big\}d\mathbf{x}\\ &+ \int_{\partial\Omega_D}(\alpha(\mathbf{u}^+\otimes\mathbf{n}^+):(\mathbf{u}^+-\boldsymbol{g}_D)\otimes\mathbf{n}^+-(\boldsymbol{\tau}\cdot\boldsymbol{g}_D)\cdot\mathbf{n})ds\\ &- \int_{\partial\Omega_N}\mathbf{u}\cdot\boldsymbol{g}_N ds+\int_\Gamma C_{11}[[\mathbf{u}]]:[[\mathbf{u}]]ds. \quad (6.22)\end{aligned}$$

The adjoint contribution to the Lagrangian is given by

$$\begin{aligned}a(\boldsymbol{\Psi},\mathfrak{v})-l(\boldsymbol{\Psi}) &= \int_{\Omega_j}(\psi_{1,x_1}\sigma_{11}+\psi_{1,x_2}\sigma_{12}-\psi_1f_1)d\mathbf{x}-\int_{\partial\Omega_j}\psi_1(\hat{\sigma}_{11}n_1+\hat{\sigma}_{12}n_2)ds\\ &+ \int_{\Omega_j}(\psi_{2,x_1}\sigma_{21}+\psi_{2,x_2}\sigma_{22}-\psi_2f_2)d\mathbf{x}-\int_{\partial\Omega_j}\psi_2(\hat{\sigma}_{21}n_1+\hat{\sigma}_{22}n_2)ds\\ &- \int_{\Omega_j}(\xi_{,x_1}v_1+\xi_{,x_2}v_2)d\mathbf{x}+\int_{\partial\Omega_j}\xi(v_1n_1+v_2n_2)ds\\ &+ \frac{1}{2\mu}\Big\{\int_{\Omega_j}(\gamma_{11}(\sigma_{11}+q)+2\gamma_{11,x_1}\mu v_1)d\mathbf{x}-\int_{\partial\Omega_j}2\gamma_{11}\mu\hat{v}_1n_1ds\Big\}\\ &+ \frac{1}{2\mu}\Big\{\int_{\Omega_j}(\gamma_{22}(\sigma_{22}+q)+2\gamma_{22,x_2}\mu v_2)d\mathbf{x}-\int_{\partial\Omega_j}2\gamma_{22}\mu\hat{v}_2n_2ds\Big\}\\ &+ \frac{1}{\mu}\Big\{\int_{\Omega_j}(\gamma_{12}\sigma_{12}+\mu(\gamma_{12,x_1}v_2+\sigma_{12,x_2}v_1))d\mathbf{x}-\int_{\partial\Omega_j}\gamma_{12}\mu(\hat{v}_2n_1+\hat{v}_1n_2)ds\Big\}. \quad (6.23)\end{aligned}$$

Before we proceed with the minimization, we add the term

$$\Theta \equiv \alpha_\Gamma \int_\Gamma C_{11}[[\mathbf{v}]] : [[\mathbf{v}]] ds$$

to the Lagrangian. We then set $\alpha, \alpha_\Gamma \to \infty$, which has the effect of imposing the conditions

$$\mathbf{v}|_{\partial\Omega_D} = \boldsymbol{g}_D, \quad [v_i]|_\Gamma = \mathbf{0}$$

and results in more accurate bounds as we now minimize over a smaller space. Setting the variation of $\mathcal{L}$ with respect to $\mathbf{v}$ equal to zero results in the constraints

$$\boxed{\begin{aligned} &-\xi_{,x_1} + \gamma_{11,x_1} + \gamma_{12,x_2} - \kappa f_1 + f_{0_1}^V = 0 \\ &-\xi_{,x_2} + \gamma_{12,x_1} + \gamma_{22,x_2} - \kappa f_2 + f_{0_2}^V = 0 \\ &[\boldsymbol{\gamma} \cdot \mathbf{n}]|_\Gamma = \mathbf{0}, \quad [\boldsymbol{\gamma} \cdot \mathbf{n}]|_{\partial\Omega_N} = -\kappa \boldsymbol{g}_N + \boldsymbol{f}_0^S. \end{aligned}} \quad (6.24)$$

While setting the variation of $\mathcal{L}$ with respect to q equal to zero results in the requirement

$$\boxed{\kappa(\sigma_{11} + \sigma_{22}) + \gamma_{11} + \gamma_{22} = 0.} \quad (6.25)$$

We point out that the above condition can be readily satisfied; this is in direct contrast to condition (6.18) in the first approach. Setting the variation of $\mathcal{L}$ with respect to $\boldsymbol{\sigma}$ equal to zero necessitates the conditions

$$\begin{aligned} &\kappa(2\sigma_{11} + q) + \gamma_{11} + 2\mu\psi_{1,x_1} = 0 \\ &\kappa(2\sigma_{22} + q) + \gamma_{22} + 2\mu\psi_{2,x_2} = 0 \\ &2\kappa\sigma_{12} + \gamma_{12} + \mu(\psi_{2,x_1} + \psi_{1,x_2}) = 0 \end{aligned} \quad (6.26)$$

as well as

$$\boxed{[\psi_i]_\Gamma = \mathbf{0}, \quad \boldsymbol{\psi}|_{\partial\Omega_D} = -\kappa \boldsymbol{g}_D + \boldsymbol{g}_0.} \quad (6.27)$$

This gives us

$$\boxed{\begin{aligned}\sigma_{11} &= -\frac{1}{4\kappa}\Big\{3\gamma_{11} + \gamma_{22} + 2\mu(\psi_{1,x_1} - \psi_{2,x_2})\Big\}\\ \sigma_{22} &= -\frac{1}{4\kappa}\Big\{\gamma_{11} + 3\gamma_{22} - 2\mu(\psi_{1,x_1} - \psi_{2,x_2})\Big\}\\ \sigma_{12} &= -\frac{1}{2\kappa}\Big\{\gamma_{12} + \mu(\psi_{2,x_1} + \psi_{1,x_2})\Big\}.\end{aligned}} \tag{6.28}$$

We note that the incompressibility constraint has been lifted. In the proposed formulation, the stringent constraint presented in (6.18) required to achieve a bounded minimum is replaced with condition (6.25), allowing us to proceed with the error bounding procedure. To simplify algebra, we introduce ζ and require that

$$\gamma_{11} + \gamma_{22} - 2\mu(\psi_{1,x_1} + \psi_{2,x_2}) - 2\zeta = 0 \tag{6.29}$$

which results in

$$\boxed{\begin{aligned}\sigma_{11} &= -\frac{1}{2\kappa}(\gamma_{11} + 2\mu\psi_{1,x_1} + \zeta)\\ \sigma_{22} &= -\frac{1}{2\kappa}(\gamma_{22} + 2\mu\psi_{2,x_2} + \zeta)\\ \sigma_{12} &= -\frac{1}{2\kappa}(\gamma_{12} + \mu(\psi_{2,x_1} + \psi_{1,x_2}))\end{aligned}} \tag{6.30}$$

The minimization of (6.14) then leads to the following results

$$\boxed{\inf_{\mathbf{v}\in\mathbf{X}} \mathcal{L}(\mathbf{v},\boldsymbol{\Psi}) = \begin{cases}\mathcal{L}^* & \text{if} \quad -\xi_{,x_1} + \gamma_{11,x_1} + \gamma_{12,x_2} - \kappa f_1 + f_{0_1}^V = 0,\\ & \quad -\xi_{,x_2} + \gamma_{12,x_1} + \gamma_{22,x_2} - \kappa f_2 + f_{0_2}^V = 0,\\ & \quad [\boldsymbol{\gamma}\cdot\mathbf{n}] = \mathbf{0},\\ & \quad [\psi_i] = \mathbf{0},\\ & \quad \boldsymbol{\psi}|_{\partial\Omega_D} = -\kappa\boldsymbol{g}_D + \boldsymbol{g}_0\\ & \quad (\boldsymbol{\gamma}\cdot\mathbf{n})|_{\partial\Omega_N} = -\kappa\boldsymbol{g}_N + \boldsymbol{f}_0^S\\ -\infty & \text{otherwise.}\end{cases}} \tag{6.31}$$

using the expression for σ_{ij} given in (6.30), we arrive at the following $\mathcal{L}^*$

$$\begin{aligned}\mathcal{L}^* &= -\int_\Omega \left\{ \frac{1}{8\mu\kappa}\Big[(\gamma_{11}+2\mu\psi_{1,x_1})^2+(\gamma_{22}+2\mu\psi_{2,x_2})^2-2\zeta^2 \right. \\ &\qquad \left. +2(\gamma_{12}+\mu(\psi_{2,x_1}+\psi_{1,x_2}))^2\Big] + \boldsymbol{\psi}\cdot\boldsymbol{f}\right\} d\mathbf{x} \\ &\quad -\int_{\partial\Omega_D}(\boldsymbol{\gamma}\cdot\boldsymbol{g}_D)\cdot\mathbf{n}ds - \int_{\partial\Omega_N}\boldsymbol{\psi}\cdot\boldsymbol{g}_N ds. \end{aligned} \tag{6.32}$$

Any $\boldsymbol{\Psi}$ satisfying the sufficient and necessary conditions for a bounded minimum given in (6.31) will produce a lower bound for S when inserted into the above expression. Even so, only very specific choices would result in bounds of acceptable quality. In the next section we shall outline the procedure with which to determine the optimal choice of $\boldsymbol{\Psi}$.

6.5 Computation of $\boldsymbol{\Psi}$: Infinite-Dimensional Case

We now proceed to the selection of $\boldsymbol{\Psi}$. We start by defining the dual variables as $\boldsymbol{\Phi} = [\boldsymbol{\phi}, \iota, \boldsymbol{\chi}]^T$ and the dual problem as

$$\begin{aligned} -\nabla\cdot\boldsymbol{\chi}+\nabla\iota+\boldsymbol{f}_0^V &= 0 \quad \text{in} \quad \Omega, \\ \nabla\cdot\boldsymbol{\phi} &= 0 \quad \text{in} \quad \Omega, \\ \boldsymbol{\phi} &= \boldsymbol{g}_0 \quad \text{on} \quad \partial\Omega_D, \\ \boldsymbol{\chi}\cdot\mathbf{n} &= \boldsymbol{f}_0^S \quad \text{on} \quad \partial\Omega_N \end{aligned} \tag{6.33}$$

where

$$\boldsymbol{\chi} \equiv \mu \begin{bmatrix} 2\frac{\partial\phi_1}{\partial x_1} & \frac{\partial\phi_2}{\partial x_1}+\frac{\partial\phi_1}{\partial x_2} \\ \frac{\partial\phi_2}{\partial x_1}+\frac{\partial\phi_1}{\partial x_2} & 2\frac{\partial\phi_2}{\partial x_2} \end{bmatrix}.$$

We postulate that the optimal value of $\boldsymbol{\Psi}$ must be a linear combination of the primal and dual solutions and then proceed to show that this is in fact the case. We thus look for $\boldsymbol{\Psi}$ of

the form

$$\boldsymbol{\Psi} = \boldsymbol{\alpha}\mathbf{u} + \boldsymbol{\beta}\boldsymbol{\Phi}. \tag{6.34}$$

From (6.24), we see that it is necessary to satisfy

$$\begin{aligned} -\xi_{,x_1} + \gamma_{11,x_1} + \gamma_{12,x_2} - \kappa f_1 + f_{0_1}^V = 0 \\ -\xi_{,x_2} + \gamma_{12,x_1} + \gamma_{22,x_2} - \kappa f_2 + f_{0_2}^V = 0 \end{aligned} \tag{6.35}$$

which, given the definition of the primal and dual problems imply that one *must* be able to express $\boldsymbol{\gamma}$ as a linear combination of the primal and dual solutions. We have, for the primal component of equations (6.35)

$$\begin{aligned} -\alpha_\xi p_{,x_1} + \alpha_\gamma(\tau_{11,x_1} + \tau_{12,x_2}) - \kappa f_1 = 0 \\ -\alpha_\xi p_{,x_2} + \alpha_\gamma(\tau_{12,x_1} + \tau_{22,x_2}) - \kappa f_2 = 0 \end{aligned} \tag{6.36}$$

which, given the definition of $\boldsymbol{\tau}$ requires

$$\alpha_\xi = 0, \qquad \alpha_\gamma = -\kappa.$$

The dual component of of equations (6.35) is given by

$$\begin{aligned} -\beta_\xi \iota_{,x_1} + \beta_\gamma(\chi_{11,x_1} + \chi_{12,x_2}) + f_{0_1}^V = 0 \\ -\beta_\xi \iota_{,x_2} + \beta_\gamma(\chi_{12,x_1} + \chi_{22,x_2}) + f_{0_2}^V = 0 \end{aligned} \tag{6.37}$$

which, given the definition of $\boldsymbol{\chi}$ requires

$$\beta_\xi = 1, \qquad \alpha_\gamma = -1.$$

Furthermore, in order to avoid bound gap, $\boldsymbol{\Psi}$, when inserted into the righthand side of (6.28), must result in a self-consistent set of equations. We can thus conclude

$$\alpha_{\boldsymbol{\psi}} = -\kappa, \qquad \beta_{\boldsymbol{\psi}} = 1.$$

In summary, we have

$$\begin{aligned} &\alpha_{\boldsymbol{\psi}} = -\kappa, \qquad \alpha_{\boldsymbol{\xi}} = 0, \qquad \alpha_{\boldsymbol{\gamma}} = -\kappa \\ &\beta_{\boldsymbol{\psi}} = 1, \qquad \beta_{\boldsymbol{\xi}} = -1, \qquad \beta_{\boldsymbol{\gamma}} = 1 \end{aligned} \tag{6.38}$$

resulting in the following expressions for $\boldsymbol{\Psi}$

$$\boxed{\begin{aligned} &\boldsymbol{\psi} = -\kappa \mathbf{u} + \boldsymbol{\phi} \\ &\gamma^p_{ij} = -\kappa\mu\left\{\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right\} + \kappa\delta_{ij}p = -\kappa\tau_{ij} \\ &\gamma^d_{ij} = -\mu\left\{\frac{\partial \phi_i}{\partial x_j} + \frac{\partial \phi_j}{\partial x_i}\right\} = \chi_{ij}. \end{aligned}} \tag{6.39}$$

$\boldsymbol{\gamma} = \boldsymbol{\gamma}^p + \boldsymbol{\gamma}^d$ is given by

$$\gamma_{ij} = -\kappa\mu\left\{\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right\} + \kappa\delta_{ij}p - \mu\left\{\frac{\partial \phi^d_i}{\partial x_j} + \frac{\partial \phi^d_j}{\partial x_i}\right\}. \tag{6.40}$$

We also note that the following relations hold if we set $\zeta = \kappa p$

$$\begin{aligned} -\tau_{11} - \tau_{22} + 2\kappa\mu(u_{1,x_1} + u_{2,x_2}) - 2\zeta &= 0 \\ \chi_{11} + \chi_{22} - 2\mu(\phi_{1,x_1} + \phi_{2,x_2}) &= 0 \end{aligned} \tag{6.41}$$

allowing (6.29) to be satisfied. $\boldsymbol{\Psi}$ is then given by

$$\boxed{\begin{aligned} \boldsymbol{\psi} &= -\kappa \mathbf{u} + \boldsymbol{\phi} \\ \boldsymbol{\gamma} &= \boldsymbol{\gamma}^p + \boldsymbol{\gamma}^d \\ \xi &= \iota \end{aligned} \qquad (6.42)}$$

6.6 Computation of $\boldsymbol{\Psi}$: Finite-Dimensional Case

We now proceed to the computation of $\boldsymbol{\Psi}_h$, such that all the conditions in (6.31) are met. We start by solving a pair of *global* problems; the primal problem given by (6.1) and the dual problem given by (6.33). From their solution we obtain $\mathbf{u}_h, \hat{\boldsymbol{\tau}}', \hat{p}$ as well as $\boldsymbol{\phi}_h, \hat{\boldsymbol{\chi}}, \hat{\iota}$. We then average $\mathbf{u}_h$ and $\boldsymbol{\phi}_h$ over all elemental vertices and interfaces to obtain $\tilde{\mathbf{u}}_h$ and $\tilde{\boldsymbol{\phi}}_h$ in order to satisfy the constraint of continuous $\boldsymbol{\psi}$. We set

$$\boxed{\boldsymbol{\psi}_h = -\kappa \tilde{\mathbf{u}}_h + \tilde{\boldsymbol{\phi}}_h. \qquad (6.43)}$$

6.6.1 Elemental Reconstruction of $\boldsymbol{\gamma}$

We start the reconstruction of $\boldsymbol{\gamma}_h$ by solving the *local* primal problem

$$\begin{aligned} -\nabla \cdot \tilde{\boldsymbol{\tau}}_h - \boldsymbol{f} &= 0 \quad \text{in} \quad \Omega_j \\ -\tilde{\tau}^h_{11} - \tilde{\tau}^h_{22} + 2\mu(\tilde{u}^h_{1,x_1} + \tilde{u}^h_{2,x_2}) - 2\zeta_h &= 0 \quad \text{in} \quad \Omega_j \\ \tilde{\boldsymbol{\tau}}_h \cdot \mathbf{n} &= \hat{\boldsymbol{\tau}}' \cdot \mathbf{n} - \hat{p} \quad \text{on} \quad \partial\Omega_j \end{aligned} \qquad (6.44)$$

and the *local* dual problem

$$\begin{aligned} -\nabla \cdot \tilde{\boldsymbol{\chi}}_h + \nabla \tilde{\iota}_h + \boldsymbol{f}^V_0 &= 0 \quad \text{in} \quad \Omega_j \\ \tilde{\chi}^h_{11} + \tilde{\chi}^h_{22} - 2\mu(\tilde{\phi}^h_{1,x_1} + \tilde{\phi}^h_{2,x_2}) &= 0 \quad \text{in} \quad \Omega_j \\ \tilde{\boldsymbol{\chi}}_h \cdot \mathbf{n} - \tilde{\iota}_h &= \hat{\boldsymbol{\chi}} \cdot \mathbf{n} - \hat{\iota} \quad \text{on} \quad \partial\Omega_j. \end{aligned} \qquad (6.45)$$

The solution of (6.44) and (6.45) involves subdividing each element, Ω_j into three subelements, as we have done in the linear elasticity implementation. Aside from the fact that we are dealing with different equations here, the methodology is identical to the linear elasticity case as regards local post-processing. After solving local problems (6.44) and (6.45), $\boldsymbol{\gamma}_h$ is given by

$$\boxed{\boldsymbol{\gamma}_h = -\kappa\tilde{\boldsymbol{\tau}}_h - \tilde{\boldsymbol{\chi}}_h.} \qquad (6.46)$$

We can then summarize the steps involved in the computation of $\boldsymbol{\Psi}_h$ as follows

1. Solve (6.1) and (6.33) by LDG for obtain $\mathbf{u}_h, \hat{\boldsymbol{\tau}}', \hat{p}$ and $\boldsymbol{\phi}_h, \hat{\boldsymbol{\chi}}, \hat{\iota}$.
2. Average $\mathbf{u}_h, \boldsymbol{\phi}_h$ at all elemental vertices and interfaces to obtain $\tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\phi}}_h$
3. Subdivide each element, $\Omega_j, \forall \Omega_j \in \Omega$ into three subelements
4. Perform local reconstruction of primal and dual solutions as given in (6.44) and (6.45)
5. Set $\boldsymbol{\psi}_h = -\kappa\tilde{\mathbf{u}}_h + \tilde{\boldsymbol{\phi}}_h$ and $\boldsymbol{\gamma}_h = -\kappa\tilde{\boldsymbol{\tau}}_h - \tilde{\boldsymbol{\chi}}_h$

Algorithm 6: Computation of $\boldsymbol{\Psi}_h$: Stokes

6.6.2 Bound Optimization

With the aforementioned choice of $\boldsymbol{\Psi}_h$, the bound gap is given by

$$\begin{aligned}\Delta S \;=\; \sum_{j=1}^{N_e}\int_{\Omega_j}\frac{1}{4\mu}\Bigg\{\kappa\Big[&(\tilde{\tau}^h_{11}-2\mu\tilde{u}^h_{1,x_1})^2+(\tilde{\tau}^h_{22}-2\mu\tilde{u}^h_{2,x_2})^2-2\zeta^2\\ &+2(\tilde{\tau}^h_{12}-\mu(\tilde{u}^h_{2,x_1}+\tilde{u}^h_{1,x_2}))^2\Big]+\frac{1}{\kappa}\Big[(\tilde{\chi}^h_{11}+2\mu\phi^h_{1,x_1})^2\\ &+(\tilde{\chi}^h_{22}+2\mu\tilde{\phi}^h_{2,x_2})^2+2(\tilde{\chi}^h_{12}+\mu(\tilde{\phi}^h_{2,x_1}+\tilde{\phi}^h_{1,x_2}))^2\Big]\Bigg\}d\mathbf{x}. \qquad (6.47)\end{aligned}$$

Note that (6.41) ensures

$$(\tilde{\tau}^h_{11}-2\mu\tilde{u}^h_{1,x_1})^2+(\tilde{\tau}^h_{22}-2\mu\tilde{u}^h_{2,x_2})^2-2\zeta^2\geq 0$$

which implies $\Delta S \geq 0$. To obtain the optimal value for κ, we set

$$\frac{\partial}{\partial \kappa} \Delta S = 0$$

which results in

$$\boxed{\kappa = \sqrt{\frac{\sum_{j=1}^{N_e} \int_{\Omega_j} \{(\tilde{\tau}_{11}^h - 2\mu \tilde{u}_{1,x_1}^h)^2 + (\tilde{\tau}_{22}^h - 2\mu \tilde{u}_{2,x_2}^h)^2 - 2\zeta^2 + 2(\tilde{\tau}_{12}^h - \mu(\tilde{u}_{2,x_1}^h + \tilde{u}_{1,x_2}^h))^2\} d\mathbf{x}}{\sum_{j=1}^{N_e} \int_{\Omega_j} \{(\tilde{\chi}_{11}^h + 2\mu \tilde{\phi}_{1,x_1}^h)^2 + (\tilde{\chi}_{22}^h + 2\mu \tilde{\phi}_{2,x_2}^h)^2 + 2(\tilde{\chi}_{12}^h + \mu(\tilde{\phi}_{2,x_1}^h + \tilde{\phi}_{1,x_2}^h))^2\} d\mathbf{x}}}.$$

6.7 Stokes Error Bound Example: Channel Flow

For this test case we solve

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau}' + \nabla p - \boldsymbol{f} = 0 \quad &\text{in} \quad \Omega = [0,1] \times [0,1], \\ \nabla \cdot \mathbf{u} = 0 \quad &\text{in} \quad \Omega, \\ \mathbf{u} = \boldsymbol{g}_D \quad &\text{on} \quad \partial\Omega \end{aligned} \tag{6.48}$$

which, given appropriate $\boldsymbol{g}_D$, produces sufficiently regular analytical solutions such that one can verify as to whether or not the optimal asymptotic convergence rate is achieved by the algorithm. For $\boldsymbol{f} = 0$ and boundary conditions

$$u_1 = \begin{cases} 4(x_2 - x_2^2) & \text{if} \quad x_1 = 0 \\ 8(x_2 - x_2^2) & \text{if} \quad x_1 = 1 \\ 0 & \text{if} \quad x_2 = 0, 1 \end{cases} \qquad u_2 = \begin{cases} \frac{1}{3} & \text{if} \quad x_2 = 0 \\ -\frac{1}{3} & \text{if} \quad x_2 = 1. \end{cases} \tag{6.49}$$

The exact solution to the problem is given by $u_1 = 4(1+x_1)(x_2 - x_2^2), u_2 = 4/3x_2^3 - 2x_2^2 + 1/3$ and $p = -4(x_1^2 + 2x_1) + 4(x_2^2 - x_2)$. We consider the output

$$l^0(\mathbf{u}) = \int_\Omega f_{0_2}^V u_2 d\mathbf{x}$$

with

$$f_{0_2}^V = \begin{cases} 1 & \text{if } \frac{1}{4} \leq x_i \leq \frac{3}{4} \\ 0 & \text{otherwise.} \end{cases} \tag{6.50}$$

The velocity vector plot and the pressure contours of the $h = 1/16$ solution using P_2 discretization are shown in figures (6-1) and (6-2). The grid convergence results for $\mathbf{u}_h$ and p_h are shown in table (6.1), where it is seen that the expected order h^{k+1} convergence rate for velocity and order h^k for pressure are indeed obtained. The ΔS grid convergence results are shown in table (6.2), where it is seen that the optimal asymptotic convergence rate of h^{2k} is also obtained.

h	$\|e_{\mathbf{u}}\|_{L_2(\Omega)}$	order	$\|e_p\|_{L_2(\Omega)}$	order
1/4	5.227515×10^{-4}	-	1.142526×10^{-3}	-
1/8	6.569722×10^{-5}	2.99	1.142526×10^{-3}	-
1/16	8.212365×10^{-6}	3.00	2.856415×10^{-4}	2.00

Table 6.1: $\mathbf{u}_h, p_h$ Grid Convergence: Stokes1

h	S	S^-	S^+	order
1/4	0.34375	0.3435539	0.3440859	-
1/8	0.34375	0.3437344	0.3437764	3.66
1/16	0.34375	0.3437489	0.3437519	3.81

Table 6.2: ΔS Grid Convergence: Stokes1

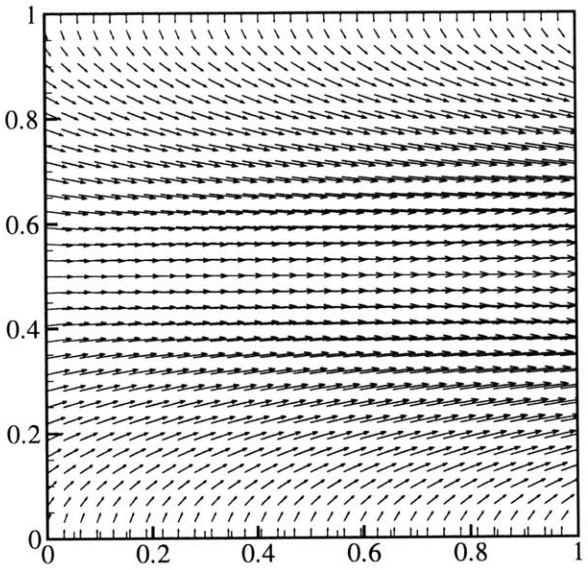


Figure 6-1: Velocity Vector: Stokes1

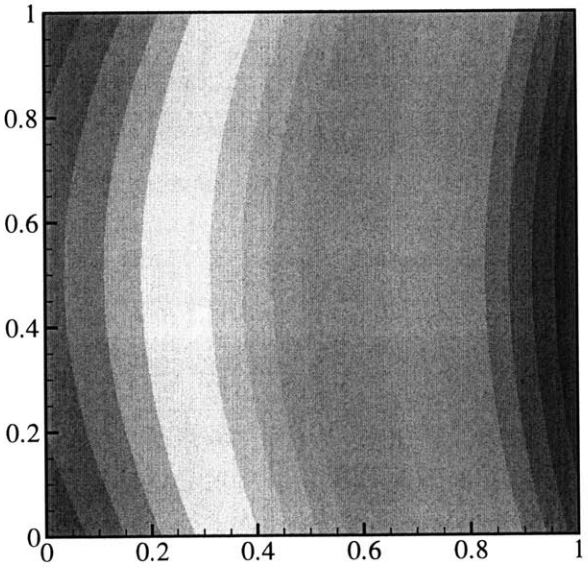


Figure 6-2: Pressure Contour: Stokes1

6.8 Stokes Error Bound Example: Drag on Square Cylinder

For this test case we solve the more interesting problem of

$$\begin{aligned} -\nabla \cdot \boldsymbol{\tau}' + \nabla(p + \bar{p}) - \boldsymbol{f} = 0 \quad &\text{in} \quad \Omega, \\ \nabla \cdot \mathbf{u} = 0 \quad &\text{in} \quad \Omega \end{aligned} \tag{6.51}$$

for $\boldsymbol{f} = 0$ for the geometry shown in (6-3) with the conditions of

$$\begin{aligned} \frac{\partial}{\partial x_1}\bar{p} = -1 \quad &\text{in} \quad \Omega, \\ \mathbf{u} = 0 \quad &\text{on} \quad \partial\Omega_s = \Omega_w \cup \Omega_{cylinder}, \\ \mathbf{u}, p \rightarrow \text{periodic} \quad &\text{on} \quad \partial\Omega \setminus \partial\Omega_s \end{aligned} \tag{6.52}$$

where $\partial\Omega_s$ refers to all solid boundary. We are interested in obtaining bounds for the output

$$l^0(\mathbf{u}) = \int_{\partial\Omega_{cylinder}} (\tau_{11} n_1 + \tau_{12} n_2) ds$$

which is also the total drag on the square cylinder. The velocity vector plot and the pressure contours of the solution obtained on the $h = 1/64$ mesh, P_2 discretization are shown in figures (6-4) and (6-5). We point out that while the singularities present in the problem prevent the optimal fourth-order convergence rate from being realized, effective bounds were nevertheless obtained; the evidence of which is shown in table (6.3). Here S^* is the output computed from the solution obtained on the $h = 1/64$ mesh.

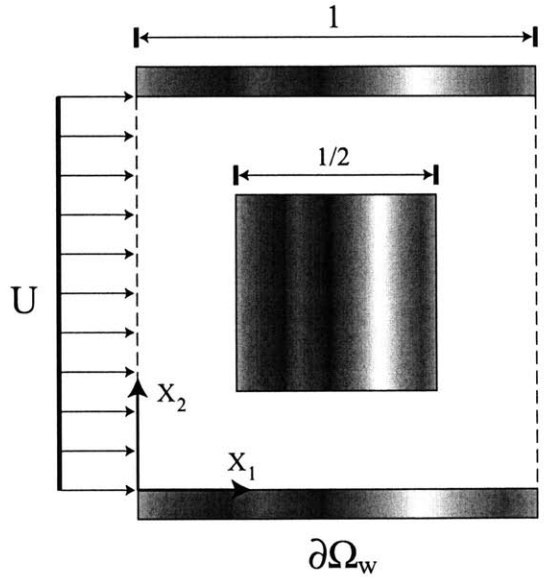


Figure 6-3: Problem Setup: Stokes2

Elements	S^*	S^-	S^+	Order
192	-0.484598588	-0.487292	-0.483790	-
768	-0.484598588	-0.485439	-0.484383	1.73
3072	-0.484598588	-0.484883	-0.484553	1.68

Table 6.3: ΔS Grid Convergence: Stokes2

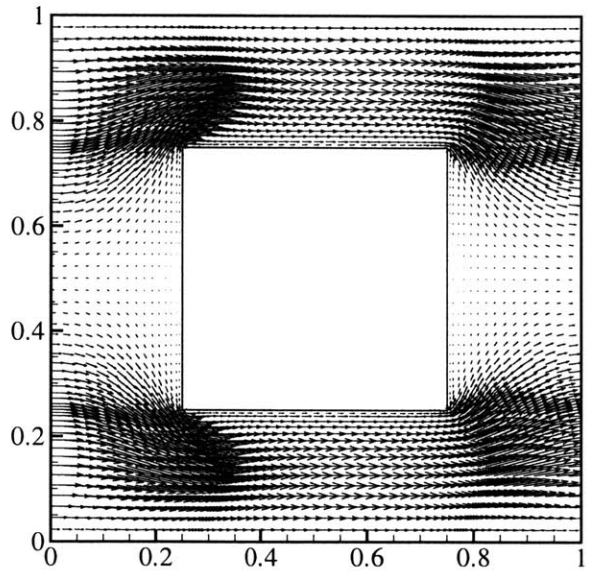


Figure 6-4: Velocity Vector: Stokes2

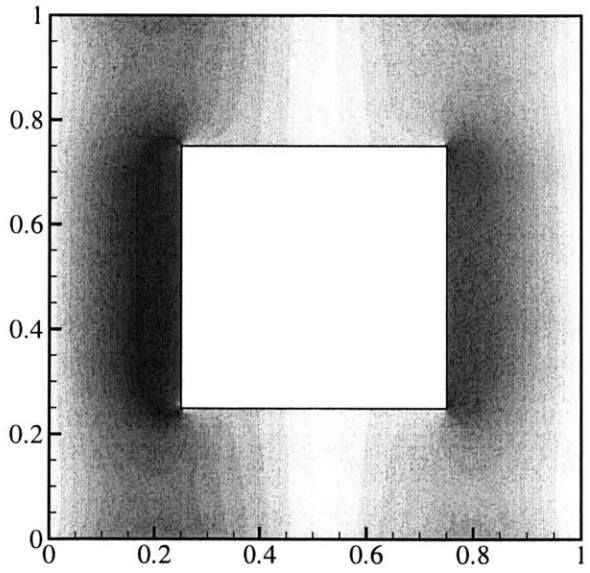


Figure 6-5: Pressure Contour: Stokes2

Chapter 7

Conclusion

Numerical solution of partial differential equations has by now become a standard tool for engineering design. Given an appropriate mathematical model describing a physical problem of interest, we have ever greater computational resources at our disposal to implement new and more powerful numerical algorithms deigned to accurately solve these problems. To fully exploit this tool, however, we need *guarantees* of accuracy and reliability. For these numerical solutions to be truly useful in the engineering decision making process, one must have confidence that the requisite level of precision has indeed been met. Providing this guarantee has been the subject of mush research of late.

7.1 Contribution

This thesis proposes a new approach to achieving the stated objective of evaluating the reliability of numerical solutions to partial differential equations. The proposed method is capable of providing strict bounds for linear functional outputs derived from the exact solution of linear coercive partial differential equations; and do so in a cost-effective manner. The method extends the capabilities of current error bounding algorithms in a variety of ways.

First, the algorithm is based on and therefore produces error bounds for the Local Discontinuous Galerkin (LDG) method, which has emerged as an important numerical method for both compressible and incompressible computational fluid dynamics. Up to now, only

asymptotic *a-posteriori* error estimates in the energy or L_2 norm are available for the LDG discretization. Asymptotic error estimates, of course, cannot provide guarantees that the desired level of precision has been met and thus compromises the usefulness of the numerical solution in the decision making process. In practical applications, we are also much more likely to be interested in the error in certain output functionals such as total surface stress, average boundary flux or deflection; quantities which are functions of the field variables but whose error cannot be bounded based on the error in the energy or L_2 norm. The proposed algorithm addresses these issues by producing *uniform* error bounds for linear functional outputs of linear coercive partial differential equations discretized with the LDG method. The upper and lower bounds on the linear functionals in question are uniform in that they *guarantee* the same functionals evaluated with the *exact* solution are bracketed within.

Secondly, basing the proposed method on the LDG discretization offers numerous advantages over existing *a-posteriori* error bounding algorithms. By exploiting the properties of discontinuous Galerkin discretization, we are able to treat problems that have thus far eluded our grasp. We can now count the high Peclet number convection-diffusion equation among those whose outputs we can effectively bound, even when the underlying numerical solution upon which the method is based fails to resolve the boundary layers that may be present in the exact solution. Another important contribution of the proposed method is its ability to handle saddle problems such as Stokes flow. The incompressibility constraint inherent in this class of partial differential equations has frustrated previous attempts to produce the type of uniform error bounds in discussion here for the Stokes problem. But here, with an approach that employs the LDG discretization as building block, we have succeeded in producing strict upper and lower bounds on the linear functionals of the Stokes problem. Like most existing error estimation routines, the method presented here requires only *local* computations beyond the solution of a global primal and dual problem. But unlike most implicit methods, a class of *a-posteriori* error estimation algorithms to which the proposed method also belongs, the method presented herein does not require the complicated equilibration procedure that implicit methods typically need and is thus computationally more efficient.

7.2 Recommendations

Within the framework of the method proposed by this thesis are several possible directions of future research. As the method is currently formulated, it is only applicable to polygonal domains. Since many problems involve curved boundaries, one should certainly investigate extending the algorithm to such cases. Closely related to this issue is the limitation of the current method that both boundary and interior forcing data be piecewise polynomial. Non-polynomial forcing is, of course, also frequently encountered in practice. In either case, the extension of the proposed algorithm would most likely involve non-polynomial basis functions, a task which the relative flexibility of discontinuous Galerkin discretization makes simpler.

More ambitious and far more difficult is the extension of the proposed algorithm to nonlinear problems. One might start with simple quadratic nonlinearity like those present in the Burger's equation. This would require a different formulation for the Lagrangian as the current formulation will not suffice. If this is possible, then, with the incompressibility constraint already resolved, one can reasonably contemplate a path to the incompressible Navier-Stokes equations; surely to be of interest to many. The feasibility of extending the type of strict *a-posteriori* error bounds presented in this work to nonlinear problems, however, remains an open question.

Bibliography

[1] M. Ainsworth and J.T. Oden. A unified approach to a posteriori error estimation based on element residual methods. *Numer. Math.*, 65:23–50, 1993.

[2] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Comp. Meth. Appl. Mech. Engrg.*, 142:1–88, 1997.

[3] D. N. Arnold. *An Interior Penalty Finite Element Methold with Discontinuous Elements.* PhD thesis, University of Chicago, Chicago, IL, 1978.

[4] D. N. Arnold. An interior penalty finite element methold with discontinuous elements. *SIAM J. Numer. Anal.*, 19:742–760, 1982.

[5] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2002.

[6] I. Babuska and W.C. Rheinboldt. A posteriori error estimates for adaptive finite element computations. *Int. J. Numer. Meth. Engrg.*, 123:1579–1615, 1978.

[7] I. Babuska and W.C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15:736–754, 1978.

[8] I. Babuska and W.C. Rheinboldt. A-posteriori error analysis of finite-element for one-dimensional problems. *SIAM J. Numer. Anal.*, 18:565–589, 1981.

[9] R. Bank and A. Weiser. Some a-posteriori error estimates for elliptic partial differential equations. *Math. Comp.*, 44:283–301, 1985.

[10] T. Barth and P. Charrier. Energy stable flux formulas for the discontinuous Galerkin discretization of first-order nonlinear conservation laws. Technical Report NAS-01-001, NASA, 2001.

[11] R. Becker, H. Kapp, and R. Rannacher. A-posteriori error estimation and mesh adaptation for finite element methods in elasto-plasticity. *SIAM J. Contr. Opt.*, 39:113–132, 2000.

[12] R. Becker and R. Rannacher. Weighted a-posteriori error control in finite element methods. Technical Report 96-1 (SFB359), IWR, Heidelberg, 1996. preprint.

[13] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Cambridge, Massachusetts, 1999. (2nd Ed.).

[14] R. Bustinza, G. Gatica, and B. Cockburn. An a-posteriori error estimate for the local discontinuous Galerkin finite element method applied to linear and nonlinear problems. *J. Sci. Comp.*, 22:147–185, 2005.

[15] P. Castillo. Performance of discontinuous Galerkin methods for elliptic PDEs. *SIAM J. Sci. Comput.*, 24:524–547, 2002.

[16] P. Castillo. An a-posteriori error estimate for the local discontinuous Galerkin finite element method. *J. Sci. Comp.*, 22:187–204, 2005.

[17] B. Cockburn, S. Hou, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:541–581, 1990.

[18] B. Cockburn, G. Kanschat, and D. Schotzau. The local discontinuous Galerkin method for Oseen equations. *Math. Comp.*, 73:569–593, 2003.

[19] B. Cockburn, G. Kanschat, D. Schotzau, and C. Schwab. Local discontinuous Galerkin methods for the Stokes system. *SIAM J. Numer. Anal.*, 40:319–343, 2002.

[20] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comput. Phys.*, 84:90–113, 1989.

[21] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework. *Math. Comp.*, 52:411–435, 1989.

[22] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion equations systems. *SIAM J. Numer. Anal.*, 28:1282–1309, 1998.

[23] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.

[24] B. Cockburn and C.-W. Shu. High-order methods for computational physics. In T.J. Barth and H. Deconinck, editors, *Lecture Notes in Computational Science and Engineering*, volume 9. Springer, 1999.

[25] B. Cockburn and C.-W. Shu. Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM J. Numer. Anal.*, 39:264–285, 2001.

[26] B.M. Fraeijis de Veubeke. *Stress Analysis.* John Wiley & Sons, New York, New York, 1965.

[27] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 131:3–47, 1997.

[28] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 24:974–1004, 2003.

[29] P. Houston, B. Senior, and E. Suli. hp-discontinuous Galerkin finite element methods for hyperbolic problems: Error analysis and adaptivity. *Internat. J. Numer. Methods Fluids*, 40:153–169, 2002.

[30] A.M. Sauer-Budgeand J.Bonet, A.Huerta, and J. Peraire. Computing bounds for linear functionals of exact weak solutions to Poisson's equation. *SIAM J. Numer. Anal.*, 2003. Submitted.

[31] C. Johnson and J. Pitkaranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46:1–26, 1986.

[32] L. Krivodonova and J.E. Flaherty. Error estimation for discontinuous Galerkin solutions of two-dimensional hyperbolic problems. *Adv. Comp. Math.*, 19:57–71, 2003.

[33] P. Ladevèze and D. Leguillon. Error estimation procedutre in the finite element method and applications. *SIAM J. Numer. Anal.*, 20:485–509, 1983.

[34] M. Paraschivoiu and A.T. Patera. A hierarchy duality approach to bounds for the outputs of partial differential equations. In P. Ladeveze and J.T. Oden, editors, *Comp. Meth. Appl. Mech. Engrg.* Elsevier, 1998.

[35] M. Paraschivoiu, J. Peraire, and A.T. Patera. A-posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations. *Comp. Meth. Appl. Mech. Engrg.*, 150:289–312, 1997.

[36] N. Páres, J. Bonet, A. Huerta, and J. Peraire. The computation of bounds for linear-functional outputs of weak solutions to the two-dimensional elasticity equations. *Elsevier Science*, 2004. Submitted.

[37] J. Peraire and A.T. Patera. Bounds for linear-functional outputs of coercive partial differential equations: local indicators and adaptive refinement. In P. Ladeveze and J.T. Oden, editors, *Advances in Adaptive Computational Methods in Mechanics.* Elsevier, 1998.

[38] T. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Numer. Anal.*, 28:133–140, 1991.

[39] N.A. Pierce and M.B. Giles. Adjoint and dfect bounding and correction for functional estimates. *J. Comput. Phys.*, 200:769–794, 2004.

[40] R. Rannacher and F.-T. Surrmeier. A feed-back approach to error control in finite element methods: applications to linear elasticity. *Comp. Mech.*, 19:434–446, 1997.

[41] R. Rannacher and F.-T. Surrmeier. A-posteriori error estimation and mesh adaptation for finite element methods in elasto-plasticity. *Comp. Meth. Appl. Mech. Engrg.*, 176:333–361, 1999.

[42] 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, Los Alamos, NM, 1973.

[43] A.M. Sauer-Budge and J. Peraire. Computing bounds for linear functionals of exact weak solutions to the advection-diffusion-reaction equation. *SIAM J. Sci. Comput.*, 2003. Submitted.

[44] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[45] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *J. Comput. Phys.*, 83:32–78, 1989.

[46] E. Suli, C. Schwab, and P. Houston. Discontinuous Galerkin Methods: Theory, Computation and Applications. In Hp-DGFEM for partial differential equations with non-negative characteristic form, editor, *Lecture Notes in Computational Science and Engineering*, volume 11. Springer, 2000.

[47] B. van Leer. Towards the ultimate conservation difference scheme, V. *J. Comput. Phys.*, 32:1–136, 1979.

[48] D.A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *J. Comput. Phys.*, 176:40–69, 2002.

[49] D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *J. Comput. Phys.*, 187:22–46, 2003.

[50] M. Zhang and C.-W. Shu. An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. *Math. Mod. Meth. Appl. Sci.*, 13:395–413, 2003.