The Dissertation Committee for Truman Everett Ellis
certifies that this is the approved version of the following dissertation:

Space-Time Discontinuous Petrov-Galerkin Finite Elements for Transient Fluid Mechanics

Committee:

Leszek F. Demkowicz, Supervisor

Robert D. Moser, Co-Supervisor

Thomas J.R. Hughes

Clint N. Dawson

Tan Bui

Space-Time Discontinuous Petrov-Galerkin Finite Elements for Transient Fluid Mechanics

by

Truman Everett Ellis, B.S.; M.S.; M.S.C.S.E.M.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

Dedicated to my grandpa, George Lowell Ellis.

Acknowledgments

The past six years have been some of the most significant and meaningful of my life. Foremost I need to thank my advisor Leszek Demkowicz. Your passion for research and dedication to following the math have been an inspiration and a revolution to how I view scientific computing. Thank you and Stasia for your incredible hospitality. I've always told people that I lucked out by finding an advisor who genuinely cares about the well-being of his students. To my co-advisor Robert Moser, thank you for always being ready to discuss the deeper details of fluid dynamics while inspiring me to consider the larger context of how computational science fits into society.

This work could not have been completed without the frequent help and expertise of Nathan Roberts, who is largely responsible for the development of the Camellia DPG library which was instrumental to obtaining the results in this thesis. Jesse Chan, your mathematical insights made possible the proofs contained here. Thank you for your patience when I wanted to run something by you. I thoroughly enjoyed our conversations on philosophy, theology, politics, mathematics, relationships, and much less serious topics.

I am grateful to my committee – Tom Hughes, Clint Dawson, and Tan Bui-Thanh – for suggesting interesting lines of research and offering perspectives on how my research fits into the larger world of computational science.

I owe a great debt of gratitude to Robert Rieben and Tzanio Kolev at Lawrence Livermore National Laboratory for seeing promise in a young graduate student and entrusting me with a project of real consequence and interest. My four summers at LLNL had a most profound influence on my career and perhaps more importantly, my appreciation for craft beers. Seriously, I have you to blame for my obsession with sour ales.

To my friends at ICES who made this such an enjoyable journey, thank you. Matthias Taus, I couldn't have passed Methods of Applied Math without you. Hanging out with you and Olivia was always fun. *Prosit!* Lindley Graham, you've been a good friend. Omar Al Hinai, despite the terrible business ideas, we had some great lunch discussions. Jesse and Jenny (and Liz), thank you for watching Charis so many times, she loves you guys. To Mike, Kathryn, Nick and Jade, John and Christa, Nora and Mat, Brendan, Federico, Sriram, Socratis, and so many others, thank you for my time in Austin so rewarding.

To new friends who have provided support and encouragement during a very challenging period of my life, you probably don't know how much you meant to me. Molly Mae Potter, thank you for the counseling and for opening your home to me when I needed it. You are an inspiring woman. Melissa, I enjoyed all the beers and adventures. Emily, you are such a good person. I grew a lot through my association with you.

To an old friend for many years – Lauren, thank you for the memories. I wish you peace and happiness in your new life.

This work is dedicated to my grandpa, George Lowell Ellis. Without his support and encouragement, I never would have started this work. To my parents – John and Vicki-Lynn – I enjoyed our weekly conversations, you two are awesome. To my brothers Kendrick and Morgan, thanks for all the California adventures. I love you all.

Space-Time Discontinuous Petrov-Galerkin Finite Elements for Transient Fluid Mechanics

Publication No. _____

Truman Everett Ellis, Ph.D.
The University of Texas at Austin, 2016

Supervisor: Leszek F. Demkowicz
Co-Supervisor: Robert D. Moser

Initial mesh design for computational fluid dynamics can be a time-consuming and expensive process. The stability properties and nonlinear convergence of most numerical methods rely on a minimum level of mesh resolution. This means that unless the initial computational mesh is fine enough, convergence can not be guaranteed. Any meshes below this minimum resolution level are termed to be in the "pre-asymptotic regime." This condition implies that meshes need to in some way anticipate the solution before it is known. On top of the minimum requirement that the surface meshes must adequately represent the geometry of the problem under consideration, resolution requirements on the volume mesh make the CFD practitioner's job significantly more time consuming.

In contrast to most other numerical methods, the discontinuous Petrov-Galerkin finite element method retains exceptional stability on extremely coarse

meshes. DPG is also inherently very adaptive. It is possible to compute the residual error without knowledge of the exact solution, which can be used to robustly drive adaptivity. This results in a very automated technology, as the user can initialize a computation on the coarsest mesh which adequately represents the geometry then step back and let the program solve and adapt iteratively until it resolves the solution features.

A common complaint of minimum residual methods by computational fluid dynamics practitioners is that they are not locally conservative. In this thesis, this concern is addressed by developing a locally conservative DPG formulation by augmenting the system with Lagrange multipliers. The resulting DPG formulation is then proved to be robust and shown to produce superior numerical results over standard DPG on a selection of test problems.

Adaptive convergence to steady incompressible and compressible Navier-Stokes solutions was explored in [18] and [65]. Space-time offers a natural extension to transient problems as it preserves the stability and adaptivity properties of DPG in the time dimension. Space-time also offers more extensive parallelization capability than problems treated with traditional time stepping as it allows multigrid concurrently in both space and time. A proof of concept space-time DPG formulation is developed for transient convection-diffusion. The robust test norms derived for steady convection-diffusion are extended to the space-time case and proofs of robustness are provided. Numerical results verify the robust behavior and near L^2 optimality of the resulting solutions.

The space-time formulation for convection-diffusion is then extended

to transient incompressible and compressible Navier-Stokes by analogy. Several numerical experiments are performed, but a mathematical analysis is not attempted for these nonlinear problems. Several side topics are explored such as a study of the compressible Navier-Stokes equations under various variable transformations and the development of consistent test norms through the concept of physical entropy.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Computational science has revolutionized the engineering design process – enabling design analysis and optimization to be done virtually before expensive physical prototypes need to be built. However, some fields of engineering analysis lend themselves to a computational approach much easier than others. Fluid dynamics has long been one of the most challenging engineering disciplines to simulate via numerical techniques. Aside from the inherent modeling challenges presented by fluid turbulence, many fluid flows can be characterized as singularly perturbed problems – problems in which the viscosity length scale is many orders of magnitude smaller than the large scale features of the flow. This has necessitated the need for meshes with large gradations in resolution to enable resolution of boundary layers while being computationally efficient in the free stream. Traditionally, these meshes would be custom designed by a domain expert who could predict which parts of the domain would need more resolution than others. On top of this, many numerical techniques would fail to converge unless the presented initial mesh was in the "asymptotic regime", i.e. the physics (viscous effects) could by somewhat sufficiently represented. These requirements made mesh generation

a laborious and far from automated procedure.

1.1.1 A Robust Adaptive Method for CFD

The failure of many numerical methods in the "pre-asymptotic regime" can be characterized mathematically as a loss of stability on coarse meshes. Discrete stability and convergence for linear problems is guaranteed by the famous discrete inf-sup condition of Babuška [5]. For mixed formulations, including the classical variational formulation for the Stokes problem, the condition reduces to the celebrated Ladyženskaya-Babuška-Brezzi (LBB) condition relating approximation spaces for velocity and pressure [26]. Leszek Demkowicz and Jay Gopalakrishnan first proposed the discontinuous Petrov-Galerkin method in 2009[28, 29] in order to address stability issues for a very broad class of problems. The DPG method automatically satisfies the discrete inf-sup condition by computing on-the-fly optimal test functions. This enables DPG simulations to remain stable and convergent even in the pre-asymptotic regime. By nature, the DPG method also comes with a built-in error representation function, effectively eliminating the need for other a posteriori error estimators. Practically, this means that a simulation could start with just the coarsest mesh necessary to represent the geometry of the solution and adaptively refine toward a resolved solution in a very automatic way. Carried to its logical conclusion, this capability could significantly cut down on the time intensive manual mesh generation (and tweaking) that dominates a good amount of simulation and analysis time. Where a current numerical method

might falter on a poorly designed mesh, necessitating an engineer to manually enter the problem and fix the offending mesh nodes, a DPG simulation would converge on the poor mesh, mark the offending cells, refine, and continue toward a solution.

Another benefit to the enhanced stability properties of DPG is the ability to consider high order and *hp*-adaptive methods. Many popular numerical methods for CFD (such as the discontinuous Galerkin method) are stable for low polynomial orders, but require additional stabilizing terms for higher orders. Additionally, one of the longstanding issues with *hp*-adaptive techniques was that they suffered stability problems when the polynomial order rose to high. Polynomial order presents no issue at all to DPG methods – allowing us to recover the high order convergence rates of high uniform p methods or even the exponential convergence rates of *hp* methods.

The biggest limitation to past explorations of the DPG method is that they were all limited to steady state problems. Obviously this seriously limits the variety of interesting problems we could consider. The easiest extension of steady DPG to transient problems would be to do an implicit time stepping technique in time and use DPG for only the spatial solve at each time step. We did indeed explore this approach, but it didn't seem to be a natural fit with the adaptive features of DPG. Clearly the CFL condition was not binding since we were interested in implicit time integration schemes, but the CFL condition can be a guiding principle for temporal accuracy in this case. So if we are interested in temporally accurate solutions, we are limited by the fact

that our smallest mesh elements (which may be orders of magnitude smaller than the largest elements) are constrained to proceed at a much smaller time step than the mesh as a whole. We can either restrict the whole mesh to the smallest time step, or we can attempt some sort of local time stepping. A space-time DPG formulation presents an attractive choice as we will be able to preserve our natural adaptivity from the steady problems while extending it in time. Thus we achieve an adaptive solution technique for transient problems in a unified framework. The obvious downside to such an approach is that for 2D spatial problems, we now have to compute on a three dimensional mesh while a spatially 3D problem becomes four dimensional.

1.1.2 Investigating a New Methodology

Much of science is driven by curiosity, and this especially holds for computational science. There is inherent value in exploring new methodologies because they may hold the keys to solving new problems or old problems in a better way. A new method may also help us to better understand existing methods. The variational multiscale approach to finite element analysis helped to elucidate on some of the success of the much older streamline upwind Petrov-Galerkin method while generalizing and improving it. The DPG method itself can be viewed as a generalization of least-squares finite elements from a multiscale point of view[9] or even of mixed methods[25].

Curiosity similarly motivates the desire to explore a space-time DPG formulation for computational fluid dynamics. Based on our past experience

with steady DPG, we anticipate space-time DPG to be a very interesting technique that could extend the automaticity of DPG in very novel ways.

1.1.3 DPG + X

DPG is admittedly, a very costly method at present. We have ideas about how to reduce the effective cost, but DPG may never be as fast as more traditional methods designed explicitly for CFD. Ultimately, there is no reason why we can't combine DPG with another method to gain the benefits of both. We could let DPG handle the initial coarse mesh and adaptively start refining toward a mesh that is sufficiently fine for another method to take over. The other method could then use traditional *a posteriori* error approximation to arrive at a fully resolved solution. This leverages the benefits of using DPG in an automated way on coarse meshes where the cost is less significant while benefiting from the computational efficiency of whatever method is coupled to it. If the other method is finite element based, this could possibly be done as simply as swapping out the test functions being used – perhaps the mesh is fine enough that we can do without the optimal test functions. We only mention this as a possible use of DPG; we are not going to look into such coupling in this research.

1.1.4 DPG for HPC

Many of the features inherent in the DPG method appear promising in the context of high performance computing. Our goal is to design a method

that eliminates human intervention as much as possible. The superior stability of the method promises to prevent a simulation from crashing which could eliminate expensive restarts on large systems. Preliminary studies on convection-diffusion suggest exceptional robustness of the method in terms of diminishing viscosity, promising successful application to a large class of flow problems. The adaptivity lent by the error representation function provides a reliable and automated way to start from a coarse mesh and only refine toward solution features in need. This uses compute resources much more efficiently than uniform refinements, allowing larger simulations with fewer resources. These features combine to produce a high degree of automaticity. Ultimately, it is desirable that an engineer could produce a rough mesh that just captures the geometry of the problem and start a DPG simulation that automatically picks up solution features without the user needing to jump back in and fix things.

DPG is very compute intensive compared to the associated communication and memory costs. Most of the work is spent in embarrassingly parallel local solves for the optimal test functions and local stiffness matrix assembly. Additionally, the stability properties of DPG make high order stability a triviality, and in general, high order methods tend to have a more attractive compute/communication profile than low order methods. In our codes, we use QR factorization for optimal test function solves, but this factorization is recyclable as we essentially have many right hand sides. The division of degrees of freedom into internal vs skeleton unknowns produces a global system which

can be statically condensed into a solve purely in terms of the skeleton degrees of freedom. In addition to significantly cutting down on the size of the global solve, this produces a embarrassingly parallel post-processing solve for the internal degrees of freedom. This property was one of the motivations behind the development of the hybridized discontinuous Galerkin [22] method. No matter what system of equations is being considered, DPG always produces a Hermitian (symmetric if real) positive definite stiffness matrix for the global solve. This property allows us to leverage the conjugate gradient algorithm as the foundation for iterative global solvers. As compute resources scale up, many more HPC simulations are increasingly becoming coupled in multiphysics simulations. Since the only requirement for a well-defined discrete DPG method is a well-defined continuous problem, it is certainly possible that each different part of the multiphysics simulation could be discretized with DPG – no need to develop many different methods for each part of the simulation. Already DPG has been successfully applied to a wide variety of problems in computational mechanics, as noted below.

1.2 Literature Review

We start this literature review by looking at various numerical methods that have been popular in the simulation of fluid dynamics problems. We then branch out to discuss the development of space-time finite elements in for various application domains. Finally we explore some of the recent developments in the discontinuous Petrov-Galerkin finite element method.

1.2.1 Methods for Computational Fluid Dynamics

Computational fluid dynamics has been one of the driving forces behind numerical analysis since computers first became available for scientific research and has followed the progression as simple methods give rise to more sophisticated ones with the maturation of computational science as a discipline. Finite difference methods were a popular choice in the early days of CFD, but these slowly gave way to finite volumes as the dominant choice. As the analysis techniques in computational science have matured, it has been increasingly desirable to be able to prove certain properties of numerical methods. The solid mathematical foundation of the finite element method renders it especially nice for analysis, and in recent years finite elements have been developing a growing following among CFD practitioners.

1.2.1.1 Finite Difference and Finite Volume Methods

Finite difference methods approximate derivatives in the strong form of the equation under consideration with finite difference approximations, but proofs of convergence rely on a distributional understanding of the equations (covering both differential equations and Rankine-Hugoniot conditions) and various forms of entropy conditions. These methods were first popularized for conservation laws by Lax who also introduced the idea of numerical flux and ideal of a monotone scheme. For fluid dynamics applications, popular finite difference schemes use numerical fluxes to reconstruct approximate derivatives at certain mesh points.

Finite volume methods can be derived from applying finite difference principles to the integral form of the conservation law under consideration. They are often derived by reference to a *control volume*. The primary benefit of finite volume methods over their finite difference counterparts is that they are much easier to develop for general unstructured meshes. Finite difference schemes typically require uniform or smoothly varying structured grids. Finite volume methods are typically low order (maximum of second order), but the emergence of discontinuous Galerkin finite element methods have provided a natural higher order extension to finite volume methods.

The presence of shocks in compressible Navier-Stokes simulations presents a difficult problem for any numerical method. The so called Gibbs phenomenon causes polynomial representations of unresolved discontinuous fields to develop undershoots and overshoots. The length scale of shocks in the solution of the Navier-Stokes equations is often on the order of several mean free paths of the fluid under consideration. So any simulation that does not resolve down to this level is going to have to deal with Gibbs effects. This can be a problem when the undershoots threaten to take density or energy negative which can quickly cause the entire solution to lose stability and return garbage. The three classical techniques used to counter this possibility in finite difference and finite volume schemes are artificial viscosity, total variation diminishing schemes, and slope limiters. Each of these techniques has its own flaws, whether loss of accuracy, limitations in multi dimensions, or numerous parameters that need to be tuned on a problem specific basis. The weighted essentially non-oscillatory

(WENO) scheme[53] remains a popular solution among many CFD practitioners and was itself an improvement on the earlier essentially non-oscillatory (ENO) scheme of Harten, Enquist, Osher, and Chakravarthy[43]. Despite the various implementation details, most of these methods for handling shocks can be interpreted as adding some sort of artificial diffusion into the numerical scheme. These means that the scheme is now solving a modified version of the original equations under consideration – one with artificially introduced diffusion terms.

1.2.1.2 Stabilized Finite Element Methods

Finite difference methods are very easy to implement, but remain limited to structured grids. Finite volume methods fix many of the limitations of finite differences, but are much harder to generalize to higher order and remain much more difficult to analyze mathematically. The rigorous mathematical foundation of finite element methods has lead to growing interest from computational scientists. Additionally, the finite element framework allows for weaker regularity constraints on the solution than implied by the strong form of the equations and a natural way to solve on general physical domains with arbitrarily high approximation order. Finite element methods found early success in the field of computational solid mechanics where the symmetric positive-definite nature of such problems allowed classical Bubnov-Galerkin methods to produce optimal or near-optimal results. Unfortunately, classical finite element methods perform poorly on singularly perturbed prob-

lems, and more general formulations had to be explored. Some of the early pioneers of finite elements for CFD include Oden, Zienkiewicz, Karniadakis, and Hughes[21].

Residual based stabilization has been a popular means of fixing the loss of robustness on singularly perturbed problems. A given bilinear form is modified by adding the strong form of the residual multiplied by a test function and scaled by some stabilization parameter τ (possibly a function). The classical example of this technique is streamline upwind Petrov-Galerkin (SUPG) method for convection-diffusion using piecewise linear continuous finite elements[14]. In addition to removing the spurious oscillations of Bubnov-Galerkin methods, SUPG recovers the optimal approximation in the H^1 norm in 1D.

Streamline Upwind Petrov-Galerkin Method. In general, the trial (approximating) and test (weighting) spaces in the finite element method need not the the same as they are in the Bubnov-Galerkin method. The term Petrov-Galerkin refers to methods in which the two space differ. The original motivation behind the method was that in 1D convection-diffusion, it is possible to recover the exact solution at nodal points using a finite difference method with "exact" artificial diffusion based on the mesh size h, the convection magnitude $\boldsymbol{\beta}$, and the viscosity ϵ. Tom Hughes, who developed the method, adapted these ideas to a finite element framework be modifying the test functions rather than by direct modification of the equations.

In the abstract, the convection-diffusion equation can be written as

$$Lu = (L_{adv} + L_{diff})u = f\,,$$

where $L_{adv}u := \nabla \cdot (\boldsymbol{\beta} u)$ is the advection operator and $L_{diff}u := -\epsilon \Delta u$ is the diffusion operator. If u is a linear combination of piece-wise linear basis functions ϕ_i, $i = 0, \cdots, N$, then within each element, the second order diffusion operator is zero. Given $b(u, v)$ and $l(v)$ from as the bilinear form and load from the standard Galerkin formulation, SUPG defines a new system $b_{SUPG}(u, v) = l_{SUPG}(v)$ where

$$b_{SUPG}(u, v) = b(u, v) + \sum_K \int_K \tau (L_{adv} v)(Lu - f)$$
$$l_{SUPG}(v) = l(v) + \sum_K \int_K \tau (L_{adv} v) f\,,$$

where τ is the SUPG parameter selected to match "exact diffusion" on uniform meshes, in which case SUPG gives the same results as the exact diffusion finite difference method. However, unlike exact diffusion finite differences, SUPG gives optimal H_0^1 approximation and nodal interpolation of the exact solution on nonuniform meshes and when $f \neq 0$. Unfortunately, SUPG loses this nodal interpolation property in higher dimensions, but still remains close to the H_0^1 best approximation. Though developed with first order elements in mind, the method can be generalized to higher order elements with a modification of τ. SUPG preserves *consistency* of the variational problem – since the stabilization is based on the residual, the exact solution satisfies the stabilized variational problem. This property does not hold for the exact diffusion finite difference or finite volume methods.

We can interpret the residual based stabilization terms as modifying the test functions from the original bilinear form:

$$b(u, \tilde{v})$$

where the SUPG test function $\tilde{v}$ is defined element-wise as

$$\tilde{v} = \phi + \tau L_{adv}\phi \,.$$

That is, we perturb our original test functions by a scaled advection operator applied to the original test function. For low order C^0 test functions, this naturally gives each test functions an upwind bias. This introduces an important idea – that stability and optimal convergence can be achieved through suitable choice of test functions.

Variational Multiscale Methods. The variational multiscale method generalized and systematized the ideas behind SUPG for a larger class of problems. The motivation was that blind application of Bubnov-Galerkin does not produce robust results in the presence of multiscale physics[45]. The approach is to decompose the solution into a coarse and fine scale: $u = \bar{u} + u'$. The coarse scale, $\bar{u}$, is solved numerically, while attempting to solve for the fine scales, u', analytically. One issue that arises in this process is approximating the fine-scale Green's function for the operator under consideration which is usually nonlocal. Similarly, the effect of the fine scales on the coarse scales in nonlocal. The variational multiscale method gives a framework from which

stabilized methods can be derived for large classes of problems, but deep analysis of the problem at hand is required to derive the effect of the fine scales on the coarse scales. Computationally, VMS methods allow for computation with standard C^0 finite elements which avoids the annoying propagation of unknowns in discontinuous Galerkin methods.

Discontinuous Galerkin Methods. Discontinuous Galerkin finite elements were first introduced by Reed and Hill in 1973 for neutron transport problems[61]. Early contributors included Babuška, Lions, Nitsche, and Zlamal, but Arnold, Brezzi, Cockburn, and Marini put together a unified analysis of DG methods for elliptic problems in [4]. Of particular interest to our work in CFD is the work by Cockburn and Shu on DG for conservation laws starting with [23]. The method combines attractive features of the finite element and finite volume methods and has become hugely successful for fluid dynamics simulations. DG is a finite element method with the same rigorous mathematical foundation and other benefits of FEM, but uses a nonconforming basis such that basis functions are discontinuous across elements. In fact, the lowest order DG method is identical to the first order finite volume method. There is no explicit continuity between elements (though approximate conformity is enforced in a weak sense). In the vein of finite volume methods, a numerical flux is used to facilitate communication between neighboring elements. The numerical flux also introduces stabilization to the method, allowing it to simulate convection dominated flows. The piecewise discontinuous nature of DG

allows for very simple h and p adaptivity and straightforward parallelization.

Like in finite volume methods, the numerical flux is some function of the edge values from two neighboring elements, The numerical flux can also be interpreted as a form of stabilization[13]. Consider the steady 1D advection equation:

$$\frac{\partial(\beta(x)u)}{\partial x} = f\,, \quad u(0) = u_0\,.$$

Multiply by test function v and integrate by parts over each element $K = [x_K, x_{K+1}]$:

$$-\int_K \beta(x)u\frac{\partial v}{\partial x} + \beta uv|_{x_K}^{x_{K+1}} = \int_K fv\,.$$

The global formulation is formed by summing up each of the local contributions. Since our discretization is piece-wise discontinuous, boundary terms are double-valued, and we need to make a choice about which ones to use. Let $u(x_K^-)$ denote the upwind value at point x^K (left side for β positive), and $u(x_K^+)$ the downwind side. Then for element K, $u(x_K^+)$ and $u(x_{K+1}^-)$ refer to the values local to that element while $u(x_K^-)$ and $u(x_{K+1}^+)$ refer to the values from its two neighboring elements. The stable choice is always choose the upwind value for u while choosing the element local value for v. Choosing downwind values of u will give an unconditionally unstable method, while choosing average values will result in something similar to an H^1 conforming continuous Galerkin discretization[13]. The upwind bilinear form for positive β on element K will then be

$$-\int_K \beta(x)u\frac{\partial v}{\partial x} + \beta(x_{K+1})u(x_{K+1}^-)v(x_{K+1}^-) - \beta(x_K)u(x_K^-)v(x_K^+) = \int_K fv\,.$$

DG methods have proven to be extremely successful in the field of computational fluid dynamics (and many other fields) due to several properties that are very important to fluid dynamicists. They are automatically locally conservative since the test function span the space of constants. The lowest order case is identical to first order finite volume methods. However, the most audible criticism of the DG method is the proliferation of unknowns relative to continuous finite elements. For linear elements in 1D, there will be twice as many unknowns, for 2D quadrilateral elements, four times as many, and with 3D hexahedral elements, eight times as many. This problem is assuaged when higher order elements are used, in which case the ratio approaches one as the element order goes to infinity. Another issue with DG is that there is a pre-asymptotic regime where the solution may go unstable if the mesh is not fine enough. This is a relevant issue when comparing to DPG, but most other methods encounter this as well, so it is not vocalized as a DG specific problem. DG methods are also critiqued for having bad conditioning and optimal convergence in "weak norms."

Hybridized Discontinuous Galerkin Methods. The hybridized discontinuous Galerkin method was first introduced by Cockburn, Gopalakrishnan, and Lazarov[22] as a way to address some of the issues with the standard DG method – notably the proliferation of unknowns. HDG introduces numerical traces (result of integrating a gradient by parts) and numerical fluxes (result of integrating a divergence by parts) which are handled differently. New cou-

pling unknowns are introduced for the numerical trace that only live on the mesh skeleton. The global problem can then be reframed exclusively in terms of these numerical traces and interior degrees of freedom can be solved in a fully parallel post-processing step. Numerical fluxes are treated in the same fashion as standard DG and hence contribute the same stabilization needed for convection dominated problems.

1.2.2 Space-Time Finite Elements

Most finite element simulations of transient phenomena use a semi-discrete formulation. This means that the PDE is first discretized in space using finite elements and then the leftover system of ordinary differential equations in time is usually solved by a finite difference method. The benefit of this procedure is that it is simple to implement and well understood numerically. Hughes[47] notes that "It is frequently argued that finite elements represent a superior methodology to finite differences" and that it is not surprising that many efforts have been made to apply finite element technologies to the space-time domain. Some of the earliest proponents of this approach were Kaczkowski[49], Argyris and Scharpf[3], Fried[40], and Oden[59]. These techniques were built on the underlying concept of Hamilton's principle. Bajer and Bonthoux present a nice review in [6].

Space-time finite elements present an attractive way to handle meshes with moving boundaries. Lesoinne and Farhat[52] studied several techniques for solving on moving meshes including Arbitrary Lagrangian-Eulerian finite

volume and finte element schemes as well as space-time finite volume schemes. The authors derived Geometric Conservation Laws (GCL) as important constraints that a scheme must satisfy for a time-accurate solution. They found that except for the case of space-time finite elements, the GCLs imposed important constraints on the schemes under consideration.

Van der Vegt and van der Ven[74] motivate their space-time discontinuous Galerkin method for 3D inviscid compressible moving boundary problems:

> The separation between space and time becomes cumbersome for time-dependent domain boundaries, which require the mesh to follow the boundary movement. We will therefore not separate space and time but consider the Euler equations directly in four dimensional space and use basis functions in the finite element discretization which are discontinuous across element faces, both in space and time. We refer to this technique as the space-time discontinuous Galerkin finite element method. The space-time DG method provides optimal efficiency for adapting and deforming the mesh while maintaining a conservative scheme which does not require interpolation of data after mesh refinement or deformation.

Klaij *et al.* [50] then extended the method to compressible Navier-Stokes while Rhebergen *et al.* [63] developed the method for incompressible Navier-Stokes. Rhebergen and Cockburn[62] also developed a space-time HDG method for incompressible Navier-Stokes.

Tezduyar and Behr[72] develop a deforming-spatial-domain/space-time

procedure coupled with Galerkin/least-squares to handle incompressible Navier-Stokes flows with moving boundaries and later Aliabadi and Tezduyar[2] apply the procedure to compressible flows. Hughes and Stewart[48] develop a general space-time multiscale framework for deriving stabilized methods for transient phenomena.

The tent-pitcher algorithm of Üngör[73] has become a popular way of mitigating the cost of space-time computations. The basic idea is that a space-time DG method can be solved element-by-element if the space-time mesh satisfies a cone constraint, i.e. the mesh faces can not be steeper in the temporal direction than a specified angle generated by the characteristics of the solution. In which case, each element is uncoupled from its neighbors, significantly increasing the efficiency of a solver. Since the cone condition evolves with the solution, the mesh must be generated on the fly based on the most recent solution information. Abedi, Petracovici, and Haber[1] applied this causal mesh generation to linear elastodynamics.

Space-time multigrid has been gaining attention lately as a means of extending the parallelism of simulations which are facing sequential bottlenecks. According to the website for the XBraid Project at Lawrence Livermore[39, 51]:

> Traditional sequential time-marching algorithms are a critical part of most computer simulations of a time-dependent problem, but these algorithms are currently facing a sequential bottleneck. This bottleneck is driven by the broad trend that future performance gains will come from greater concurrency, not faster clock

speeds. Previously, ever-increasing clock speeds decreased the compute time for each time step, thus allowing more time steps to be calculated without increasing the overall compute time. Now that clock speeds are stagnant, further refinements in time (i.e., increases in the number of time steps) will simply increase the simulations overall compute time. Many of these refinements in time will be required to maintain balance between spatial and temporal accuracies. Additionally, some simulations are already fully resolved in space, and it is unclear how such simulations will take advantage of the coming increases in concurrency.

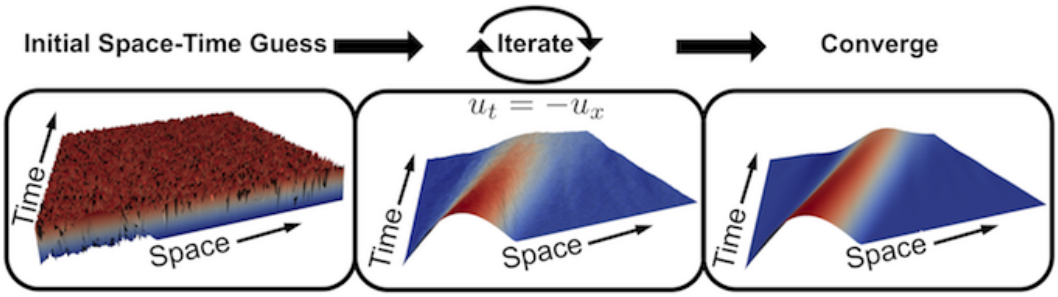


Figure 1.1: Multigrid in time with XBraid by LLNL[51]

Multigrid in time allows all times to be solved for simultaneously, dramatically improving parallelization opportunities within a simulation code, see Figure 1.1. The practical downside is that this is a much more expensive procedure on small to medium sized computer architectures. It's only on very large systems that have maxed out the strong scaling of a single timestep where this approach starts to pay off. Thus, for the proof of concept problems solved on moderate sized system in this dissertation, we only expect to reap the extra

cost without seeing the reward. However this concept has potential as we look towards exascale simulations in the future.

1.2.3 Discontinuous Petrov-Galerkin Method

The discontinuous Petrov-Galerkin finite element method with optimal test functions was first proposed by Demkowicz and Gopalakrishnan in 2009[29]. The basic ideas are fairly straight-forward; DPG minimizes the residual in a user defined energy norm. Consider a variational problem: find $u \in U$ such that

$$b(u, v) = l(v) \quad \forall v \in V$$

with operator $B : U \to V' \quad (V'$ is the dual space to $V)$ defined by $b(u, v) = \langle Bu, v \rangle_{V' \times V}$. This gives the operator equation:

$$Bu = l \in V' .$$

We wish to minimize the residual $Bu - l$ in V':

$$u_h = \arg\min_{w_h \in U_h} \frac{1}{2} \|Bu - l\|^2_{V'} .$$

This is a very natural mathematical framework based soundly in functional analysis, but it is not yet a practical method as the V' norm is not especially tractable to work with. The insight is that since we are working with Hilbert spaces, we can use the Riesz representation theorem to find a complementary object in V rather than V'. Let $R_V : V \ni v \to (v, \cdot) \in V'$ be the Riesz map. Then the inverse Riesz map (which is an isometry) lets us represent our

residual in V:

$$u_h = \arg\min_{w_h \in U_h} \frac{1}{2} \left\| R_V^{-1}(Bu - l) \right\|_V^2 .$$

Taking the Gâteaux derivative to be zero in all directions $\delta u \in U_h$ gives,

$$\left(R_V^{-1}(Bu_h - l), R_V^{-1} B\delta u \right)_V = 0, \quad \forall \delta u \in U,$$

which by definition of the Riesz map is equivalent to

$$\left\langle Bu_h - l, R_V^{-1} B\delta u_h \right\rangle = 0 \quad \forall \delta u_h \in U_h \,,$$

with optimal test functions $v_{\delta u_h} \coloneqq R_V^{-1} B\delta u_h$ for each trial function δu_h. This gives a simple bilinear form

$$b(u_h, v_{\delta u_h}) = l(v_{\delta u_h}),$$

with $v_{\delta u_h} \in V$ that solves the auxiliary problem

$$(v_{\delta u_h}, \delta v)_V = \langle R_V v_{\delta u_h}, \delta v \rangle = \langle B\delta u_h, \delta v \rangle = b(\delta u_h, \delta v) \quad \forall \delta v \in V.$$

We might call this an *optimal Petrov-Galerkin*. We arrive at the same method by realizing the supremum in the inf-sup condition, motivating the *optimal* nomenclature. These optimal Petrov-Galerkin methods produce Hermitian, positive-definite stiffness matrices since

$$b(u_h, v_{\delta \boldsymbol{u}_h}) = (v_{u_h}, v_{\delta \boldsymbol{u}_h})_V = \overline{(v_{\delta \boldsymbol{u}_h}, v_{u_h})} = \overline{b(\delta u_h, v_{u_h})} \,.$$

We can calculate the energy norm (defined by $\|u\|_E \coloneqq \|Bu\|_{V'}$) of the Galerkin error without knowing the exact solution by using the residual:

$$\|u_h - u\|_E = \|B(u_h - u)\|_{V'} = \|Bu_h - l\|_{V'} = \left\| R_V^{-1}(Bu_h - l) \right\|_V ,$$

where we designate $R_V^{-1}(Bu_h - l)$ the *error representation function*. This has proven to be a very reliable *a-posteriori* error estimator for driving adaptivity.

Babuška's theorem[5] says that discrete stability and approximability imply convergence. That is, if M is the continuity constant for $b(u, v)$ which satisfies the discrete inf-sup condition with constant γ_h,

$$\sup_{v_h \in V_h} \frac{|b(u, v)|}{\|v_h\|_V} \geq \gamma_h \|u_h\|_U \ ,$$

then the Galerkin error satisfies the bound

$$\|u_h - u\|_U \leq \frac{M}{\gamma_h} \inf_{w_h \in U_h} \|w_h - u\|_U \ .$$

Optimal test functions realize the supremum in the discrete discrete inf-sup condition such that $\gamma_h \geq \gamma$, the infinite-dimensional inf-sup constant. If we then use the energy norm for $\|\cdot\|_U$, then $M = \gamma = 1$ and Babuška's estimate implies that the optimal Petrov-Galerkin method is the most stable Petrov-Galerkin method possible.

There are still many features of the method that are left to be decided, for example the U and V spaces. If V is taken to be a continuous space, then the auxiliary problem becomes global in scope, something that we would like to avoid. In order to ensure the auxiliary problem can be solved element-by-element, we take V to be discontinuous between elements. (Technically, V should also be infinite dimensional, but we have found it to be sufficient to use an "enriched" space of higher polynomial dimension than the trial space.) The downside to using discontinuous test functions is that it introduces new

interface unknowns. When the equations are integrated by parts over each element, the jump in test functions introduces new unknowns on the mesh skeleton that would have gone away with continuous test functions. Moro *et al.* [54] handle the flux unknowns with a numerical flux in the hybridized DPG method, but the standard DPG method treats these as new unknowns to be solved for. We still haven't specified our trial space U, but the rule is that for every integration by parts, a new skeleton unknown is introduced. Most DPG considerations break a second order PDE into a system of first order PDEs which introduces a trace unknown (from the constitutive law) and a flux unknown (from the conservation law), but Demkowicz and Gopalakrishnan also formulated a *primal DPG* method for second order equations that does not introduce a trace unknown. The overall number of interface unknowns in the primal DPG method is the same, however, since the solution is required to be H^1 conforming and the trace unknowns are essentially hidden here.

The final unresolved choice is what norm to apply to the V space. This is one of the most important factors in designing a robust DPG method as this norm needs to be inverted to solve for the optimal test functions. If the norm produces unresolved boundary layers in the auxiliary problem, then many of the attractive features of DPG may fall apart. But elimination of boundary layers in the auxiliary solve is not the only requirement at play. This choice also controls what norm the residual is minimized in. Often we want this norm to be equivalent to the L^2 norm. Fortunately, we have found that it is possible to design test norms such that the implied energy norm is provably

robust and equivalent to L^2 for convection-diffusion which serves as the most relevant model problem for our research. Norms for Navier-Stokes are derived by analogy to the convection-diffusion norm.

DPG has been successfully applied to a wide range of physical problems. Early work on the Poisson equation was published in [30]. Demkowicz *et al.* [34], Gopalakrishnan *et al.* [41], and Zitelli *et al.* [77] analyzed and solved the Helmholtz equation with DPG. DPG was applied to linear elasticity and plate problems in [10], [55], and [11]. A 2D Maxwell cloaking problem was solved with DPG in [37] and a 3D DPG theory for Maxwell was developed by Wieners and Wohlmuth[75]. DPG has been applied to various fluid problems including convection-diffusion[17, 19, 35, 36, 38], Stokes[38, 64], Burgers' equation[16], incompressible Navier-Stokes[65, 67], and compressible Navier-Stokes[18].

Camellia – A Library for Computing with DPG. DPG is a relatively young technology and has some fairly unique implementation requirements. In particular, the use of element interface unknowns, the computation of the optimal test functions, and the use of the error representation function to drive adaptivity are not common features in many finite element libraries. Nathan Roberts began work on the Camellia[66] library the summer of 2011 at Sandia National Laboratory. Jesse Chan and I soon followed as active contributors. Camellia is written in modern C++ on top of Trilinos[44] and supports distributed computation with MPI. It currently supports 1D, 2D,

and 3D meshes with line, quadrilateral, triangle, hexahedral, and tetrahedral elements as well as space-time in 1D and 2D and both $h-$ and $p-$adaptivity. Though Camellia has not undergone an official open source release yet and many of the features are still experimental, the source code is available at [68]. Every numerical result in this dissertation was computed using Camellia.

Chapter 2

Conservation in Steady-State

2.1 Motivation

We summarize some of our completed work on a locally conservative DPG formulation that was invented to address mass loss concerns for standard DPG. Locally conservative methods hold a special place for numerical analysts in the field of fluid dynamics. Perot[60] argues:

> Accuracy, stability, and consistency are the mathematical concepts that are typically used to analyze numerical methods for partial differential equations (PDEs). These important tools quantify how well the mathematics of a PDE is represented, but they fail to say anything about how well the physics of the system is represented by a particular numerical method. In practice, physical fidelity of a numerical solution can be just as important (perhaps even more important to a physicist) as these more traditional mathematical concepts. A numerical solution that violates the underlying physics (destroying mass or entropy, for example) is in many respects just as flawed as an unstable solution.

[0]This chapter is largely based on the journal article Locally Conservative Discontinuous Petrov-Galerkin Finite Elements for Fluid Problems which appeared in *Computers & Mathematics with Applications* Volume 68, Issue 11 in December 2014. Co-authors Jesse Chan and Leszek Demkowicz assisted with the mathematical proofs contained herein.

There are also some mathematically attractive reasons to pursue local conservation. The Lax-Wendroff theorem guarantees that a conservative numerical solution to a system of hyperbolic conservation laws will converge to a weak solution.

The discontinuous Petrov-Galerkin finite element method has been described as least squares finite elements with a twist. The key difference is that while least squares methods seek to minimize the residual of the solution in the L^2 norm, DPG seeks the minimization in a dual norm realized through the inverse Riesz map. Exact mass conservation has been an issue that has long plagued least squares finite elements. Several approaches have been used to try to adress this. Bochev *et al.* [8] accomplish local conservation by using a pointwise divergence free velocity space in the Stokes formulation. Chang and Nelson[20] developed the *restricted LSFEM*[20] by augmenting the least squares equations with Lagrange multipliers explicitly enforcing mass conservation element-wise. Our conservative formulation of DPG takes a similar approach and both methods share a similar negative of transforming a minimization method to a saddle point problem. In the interest of crediting Chang and Nelson's restricted LSFEM, we could call the following locally conservative DPG method the restricted DPG method, but we prefer to the term conservative DPG. Note that conservation is preserved with respect to fluxes rather than field variables as we explain later.

2.2 Element Conservative Convection-Diffusion

We now proceed to develop a locally conservative formulation of DPG for convection-diffusion type problems, but there are a few terms that we need to define first. If Ω is our problem domain, then we can partition it into finite elements K such that

$$\overline{\Omega} = \bigcup_K \bar{K}, \quad K \text{ open},$$

with corresponding external boundary Γ, *skeleton* Γ_h and *interior skeleton* Γ_h^0,

$$\Gamma_h := \bigcup_K \partial K \qquad \Gamma_h^0 := \Gamma_h - \Gamma.$$

We define broken Sobolev spaces element-wise:

$$\begin{aligned} H^1(\Omega_h) &:= \textstyle\prod_K H^1(K), \\ \boldsymbol{H}(\mathrm{div}, \Omega_h) &:= \textstyle\prod_K \boldsymbol{H}(\mathrm{div}, K). \end{aligned}$$

We also need the trace spaces:

$$\begin{aligned} H^{\frac{1}{2}}(\Gamma_h) := \Bigg\{ & \hat{v} = \{\hat{v}_K\} \in \prod_K H^{1/2}(\partial K) \ : \\ & \exists v \in H^1(\Omega) : v|_{\partial K} = \hat{v}_K \Bigg\}, \\ H^{-\frac{1}{2}}(\Gamma_h) := \Bigg\{ & \hat{\sigma}_n = \{\hat{\sigma}_{Kn}\} \in \prod_K H^{-1/2}(\partial K) \ : \\ & \exists \boldsymbol{\sigma} \in \boldsymbol{H}(\mathrm{div}, \Omega) : \hat{\sigma}_{Kn} = (\boldsymbol{\sigma} \cdot \boldsymbol{n})|_{\partial K} \Bigg\}, \end{aligned}$$

which are developed more precisely in [64].

2.2.1 Derivation

Now that we have briefly outlined the abstract DPG method, let us apply it to the convection-diffusion equation. The strong form of the steady convection-diffusion problem with homogeneous Dirichlet boundary conditions reads

$$\begin{cases} \nabla \cdot (\boldsymbol{\beta} u) - \epsilon \Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma \,, \end{cases}$$

where u is the property of interest, $\boldsymbol{\beta}$ is the convection vector, and f is the source term. Nonhomogeneous Dirichlet and Neumann boundary conditions are straightforward but would add technicality to the following discussion. Let us write this as an equivalent system of first order equations:

$$\begin{aligned} \nabla \cdot (\boldsymbol{\beta} u - \boldsymbol{\sigma}) &= f \\ \frac{1}{\epsilon} \boldsymbol{\sigma} - \nabla u &= \mathbf{0} \,. \end{aligned}$$

If we then multiply the first equation by some scalar test function v and the bottom equation by some vector-valued test function $\boldsymbol{\tau}$, we can integrate by parts over each element K:

$$\begin{aligned} -(\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v)_K + \langle (\boldsymbol{\beta} u - \boldsymbol{\sigma}) \cdot \mathbf{n}, v \rangle_{\partial K} &= (f, v)_K \\ \frac{1}{\epsilon} (\boldsymbol{\sigma}, \boldsymbol{\tau})_K + (u, \nabla \cdot \boldsymbol{\tau})_K - \langle u, \tau_n \rangle_{\partial K} &= 0 \,. \end{aligned} \tag{2.1}$$

The discontinuous Petrov-Galerkin method refers to the fact that we are using discontinuous optimal test functions that come from a space differing from the trial space. It does not specify our choice of trial space. Nevertheless, many versions of DPG in the literature (convection-diffusion [30], linear elasticity

[10], linear acoustics [34], Stokes [64]) associate DPG with the so-called "ultra-weak formulation." We will follow the same derivation for the convection-diffusion equation, but we emphasize that other formulations are available (in particular, the primal DPG[31] method presents an alternative with continuous trial functions). Thus, we seek field variables $u \in L^2(K)$ and $\boldsymbol{\sigma} \in \mathbf{L}^2(K)$. Mathematically, this leaves their traces on element boundaries undefined, and in a manner similar to the hybridized discontinuous Galerkin method, we define new unknowns for trace $\hat{u}$ and flux $\hat{t}$. Applying these definitions to (2.1) and adding the two equations together, we arrive at our desired variational problem.

Find $\boldsymbol{u} := (u, \boldsymbol{\sigma}, \hat{u}, \hat{t}) \in \boldsymbol{U} := L^2(\Omega_h) \times \boldsymbol{L}^2(\Omega_h) \times H^{1/2}(\Gamma_h) \times H^{-1/2}(\Gamma_h)$ such that

$$\begin{aligned} \underbrace{-(\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v)_K + \langle \hat{t}, v \rangle_{\partial K} + \frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau})_K + (u, \nabla \cdot \boldsymbol{\tau})_K - \langle \hat{u}, \tau_n \rangle_{\partial K}}_{b(\mathbf{u},\mathbf{v})} &= \underbrace{(f, v)_K}_{l(\mathbf{v})} \quad \text{in } \Omega \\ \hat{u} &= 0 \quad \text{on } \Gamma \end{aligned} \tag{2.2}$$

for all $\boldsymbol{v} := (v, \boldsymbol{\tau}) \in \boldsymbol{V} := H^1(\Omega_h) \times \boldsymbol{H}(\mathrm{div}, \Omega_h)$.

We note that for convection-diffusion problems we are particularly interested in designing a *robust* DPG method. Specifically, we are interested in designing methods whose behavior does not change as the diffusion parameter ϵ becomes very small. Naive Galerkin methods for convection-diffusion tend to suffer from a lack of robustness; specifically, the finite element error is bounded by a constant factor of the best approximation error, but the constant is often proportional to ϵ^{-1}. Our aim is to design a DPG method with this in mind.

We follow the methodology introduced by Heuer and Demkowicz in [36]: the ultra-weak variational formulation for convection-diffusion can be refactored as

$$b\left(\left(u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right),(v,\boldsymbol{\tau})\right)=\sum_{K\in\Omega_h}\left\langle\hat{t},v\right\rangle_{\partial K}+\left\langle\widehat{u},\tau_n\right\rangle_{\delta K}+\left(u,\nabla\cdot\boldsymbol{\tau}-\boldsymbol{\beta}\cdot\nabla v\right)_{L^2(K)}$$
$$+\left(\boldsymbol{\sigma},\frac{1}{\epsilon}\boldsymbol{\tau}+\nabla v\right)_{L^2(K)},$$

modulo application of boundary data. If we choose specific *conforming* test functions satisfying the adjoint equations

$$\nabla\cdot\boldsymbol{\tau}-\boldsymbol{\beta}\cdot\nabla v=u,$$
$$\frac{1}{\epsilon}\boldsymbol{\tau}+\nabla v=\boldsymbol{\sigma},$$

then evaluating $b\left(\left(u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right),(v,\boldsymbol{\tau})\right)$ at these specific test functions returns back $\|u\|^2+\|\boldsymbol{\sigma}\|^2$, the L^2 norm of our field variables. Multiplying and dividing through by the test norm $\|\boldsymbol{v}\|_V$, we have

$$\|u\|_{L^2}^2+\|\boldsymbol{\sigma}\|_{L^2}^2=b\left(\left(u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right),(v,\boldsymbol{\tau})\right)=\frac{b\left(\left(u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right),(v,\boldsymbol{\tau})\right)}{\|\boldsymbol{v}\|_V}\|\boldsymbol{v}\|_V$$
$$\leq\left\|u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right\|_E\|\boldsymbol{v}\|_V,$$

where

$$\left\|u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right\|_E=\sup_{v\in V\setminus\{0\}}\frac{b\left(\left(u,\boldsymbol{\sigma},\widehat{u},\hat{t}\right),(v,\boldsymbol{\tau})\right)}{\|\boldsymbol{v}\|_V}$$

is the DPG energy norm. If we can robustly bound the test norm $\|\boldsymbol{v}\|_V\lesssim\left(\|u\|_{L^2}^2+\|\boldsymbol{\sigma}\|_{L^2}^2\right)^{1/2}$ (i.e. derive a bound from above with a constant independent of ϵ), then we can divide through to get

$$\left(\|u\|_{L^2}^2+\|\boldsymbol{\sigma}\|_{L^2}^2\right)^{\frac{1}{2}}\lesssim\left\|(u,\boldsymbol{\sigma},\widehat{u},\hat{t})\right\|_E. \tag{2.3}$$

In other words, the energy norm in which DPG is optimal bounds the L^2 norm uniformly in epsilon. So, as we drive our energy error down to zero, we can expect that the L^2 error will also decrease regardless of ϵ.

We note that the construction of the test norm $\|\boldsymbol{v}\|_V$ for a robust DPG method depends on two things: the test norm, as well as the adjoint equation. In [36], the standard problem with Dirichlet conditions enforced over the entire boundary was considered; in [17], boundary conditions were chosen for the forward problem such that the induced adjoint problem was regularized and contained no strong boundary layers, allowing for the construction of a stronger test norm on V. We adopt a slight modification of the test norm introduced in [17] for numerical experiments here, which is motivated and explained in more detail in [18].

Having reviewed and laid the foundation for DPG methods, we can now formulate our conservative DPG scheme. Let $\boldsymbol{U}_h := U_h \times \boldsymbol{S}_h \times \hat{U}_h \times \hat{F}_h \subset L^2(\Omega_h) \times \boldsymbol{L}^2(\Omega_h) \times H^{\frac{1}{2}}(\Gamma_h) \times H^{-\frac{1}{2}}(\Gamma_h)$ be a finite-dimensional subspace, and let $\boldsymbol{u}_h := (u_h, \boldsymbol{\sigma}_h, \hat{u}_h, \hat{t}_h) \in \boldsymbol{U}_h$ be the group variable. The element conservative DPG scheme is derived from the Lagrangian:

$$L(\boldsymbol{u}_h, \lambda_K) = \frac{1}{2} \left\| R_V^{-1}(b(\boldsymbol{u}_h, \cdot) - (f, \cdot)) \right\|_{\boldsymbol{V}}^2 - \sum_K \lambda_K (b(\boldsymbol{u}_h, (1_K, \boldsymbol{0})) - l((1_K, \boldsymbol{0}))) \,, \tag{2.4}$$

where $(1_K, \boldsymbol{0})$ is the test function in which $v = 1$ on element K and 0 elsewhere and $\boldsymbol{\tau} = \boldsymbol{0}$ everywhere.

Taking the Gâteaux derivatives as before, we arrive at the following

system of equations:

$$\begin{cases} b(\boldsymbol{u}_h, T(\delta\boldsymbol{u}_h)) - \sum_K \lambda_K b(\delta\boldsymbol{u}_h, (1_K, \boldsymbol{0})) & = l(T(\delta\boldsymbol{u}_h)) \quad \forall \delta\boldsymbol{u}_h \in \boldsymbol{U}_h \\ b(\boldsymbol{u}_h, (1_K, \boldsymbol{0})) & = l((1_K, \boldsymbol{0})) \quad \forall K\,, \end{cases} \tag{2.5}$$

where $T := R_V^{-1} B : \boldsymbol{U}_h \to \boldsymbol{V}$ is the same trial-to-test operator as in the original formulation.

Denote $T(\delta\boldsymbol{u}_h) = (v_{\delta\boldsymbol{u}_h}, \boldsymbol{\tau}_{\delta\boldsymbol{u}_h}) \in H^1(\Omega_h) \times \boldsymbol{H}(\mathrm{div}, \Omega_h)$. Then, putting (2.5) into more concrete terms for convection-diffusion, we get:

$$\begin{cases} -(\boldsymbol{\beta}u - \boldsymbol{\sigma}, \nabla v_{\delta\boldsymbol{u}_h}) + \langle \hat{t}, v_{\delta\boldsymbol{u}_h} \rangle + \frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau}_{\delta\boldsymbol{u}_h}) + (u, \nabla \cdot \boldsymbol{\tau}_{\delta\boldsymbol{u}_h}) & \\ \qquad -\langle \hat{u}, \boldsymbol{\tau}_{\delta\boldsymbol{u}_h} \cdot \boldsymbol{n} \rangle - \sum_K \lambda_K (\delta\hat{t}, (1_K, \boldsymbol{0})) & = (f, v_{\delta\boldsymbol{u}_h}) \\ & \quad \forall \delta\boldsymbol{u}_h \in \boldsymbol{U}_h \\ \langle \hat{t}, (1_K, \boldsymbol{0}) \rangle & = (f, 1_K) \\ & \quad \forall K\,. \end{cases} \tag{2.6}$$

2.2.2 Stability Analysis

In the following analysis we neglect the error due to the approximation of optimal test functions. See [42] for a defense of this assumption. We follow the classical Brezzi's theory [12, 26] for an abstract mixed problem:

$$\begin{cases} \boldsymbol{u} \in \boldsymbol{U}, p \in Q & \\ a(\boldsymbol{u}, \boldsymbol{w}) + c(p, \boldsymbol{w}) & = l(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{U} \\ c(q, \boldsymbol{u}) & = g(q) \quad \forall q \in Q \end{cases} \tag{2.7}$$

where $\boldsymbol{U}, Q$ are Hilbert spaces, and a, c, l, g denote the appropriate bilinear and linear forms. Note that $a(\boldsymbol{u}, \boldsymbol{w}) = b(\boldsymbol{u}, T\boldsymbol{w}) = (T\boldsymbol{u}, T\boldsymbol{w})_V$ in the notation from the previous section.

Let the function $\boldsymbol{\psi}$ denote the $\boldsymbol{H}(\mathrm{div},\Omega)$ extension of flux $\hat{t}$ that realizes the minimum in the definition of the quotient (minimum energy extension) norm. The choice of norm for the Lagrange multipliers λ_K is implied by the quotient norm used for $H^{-1/2}(\Gamma_h)$ and continuity bound for form $c(p,\boldsymbol{w})$ representing the constraint:

$$\begin{aligned}
|c(\textstyle\sum_K \lambda_K(1_K,\mathbf{0}),(u,\boldsymbol{\sigma},\hat{u},\hat{t}))| &= |\textstyle\sum_K \lambda_K\langle\hat{t},1_K\rangle_{\partial K}| \\
&= |\textstyle\sum_K \lambda_K\langle v_n,1_K\rangle_{\partial K}| \\
&= |\textstyle\sum_K \lambda_K \int_K \mathrm{div}\boldsymbol{\psi}\, 1_K| \\
&\leq \textstyle\sum_K \lambda_K ||\mathrm{div}\boldsymbol{\psi}||_{L^2(K)}\mu(K)^{1/2} \\
&\leq (\textstyle\sum_K \mu(K)\lambda_K^2)^{1/2}\,(\textstyle\sum_K ||\mathrm{div}\boldsymbol{\psi}||^2_{L^2(K)})^{1/2} \\
&\leq \underbrace{\left(\sum_K \mu(K)\lambda_K^2\right)^{1/2}}_{=:||\boldsymbol{\lambda}||} ||\hat{t}||_{H^{-1/2}(\Gamma_h)} \\
&\leq \|\boldsymbol{\lambda}\|\,\|\boldsymbol{u}\|\,,
\end{aligned} \tag{2.8}$$

where $\mu(K)$ stands for the area (measure) of element K.

We proceed now with the discussion of the discrete inf-sup stability constants. We skip index h in the notation.

Inf Sup Condition relating spaces $\boldsymbol{U}$ and Q reads as follows:

$$\sup_{\boldsymbol{w}\in\boldsymbol{U}} \frac{|c(p,\boldsymbol{w})|}{||\boldsymbol{w}||_{\boldsymbol{U}}} \geq \beta||p||_Q\,. \tag{2.9}$$

Let

$$R\,:\,L^2(\Omega)\ni q\rightarrow \boldsymbol{\psi}\in \boldsymbol{H}(\mathrm{div},\Omega)\cap\boldsymbol{H}^1(\Omega)=\boldsymbol{H}^1(\Omega) \tag{2.10}$$

be the continuous right inverse of the divergence operator constructed by Costabel and McIntosh in [24]. Let $\boldsymbol{\psi}_h$ denote the classical, lowest order Raviart-Thomas (RT) interpolant of the function

$$\boldsymbol{\psi} = R(\sum_K \lambda_K 1_K) \,. \tag{2.11}$$

Note that $\mathrm{div}\boldsymbol{\psi}_h = \mathrm{div}\boldsymbol{\psi} = \lambda_K$ in element K.

Classical h-interpolation interpolation error estimates for the lowest error Raviart-Thomas elements and continuity of operator R imply the stability estimate:

$$\begin{aligned} ||\boldsymbol{\psi}_h|| &\leq ||\boldsymbol{\psi}_h - \boldsymbol{\psi}|| + ||\boldsymbol{\psi}|| \\ &\leq Ch||\boldsymbol{\psi}||_{H^1} + ||\boldsymbol{\psi}|| \\ &\leq C||\mathrm{div}\boldsymbol{\psi}|| = C(\textstyle\sum_K \mu(K)\lambda_K^2)^{1/2} \,. \end{aligned} \tag{2.12}$$

Above, C is a generic, mesh independent constant incorporating constant from the interpolation error estimate and the continuity constant of R. Let $\hat{t}$ be the trace of $\boldsymbol{\psi}_h$. We have then,

$$\sup_{\hat{t}\in H^{-1/2}(\Gamma_h)} \frac{|\sum_K \lambda_K \langle \hat{t}, 1_K\rangle_{\partial K}|}{||\hat{t}||_{H^{-1/2}(\Gamma_h)}} \geq \frac{|\sum_K \lambda_K \int_K \mathrm{div}\boldsymbol{\psi}_h \, 1_K|}{||\boldsymbol{\psi}_h||_{H(\mathrm{div},\Omega)}} \geq \frac{1}{C}(\sum_K \mu(K)\lambda_K^2)^{1/2} \,, \tag{2.13}$$

where C is the constant from stability estimate (2.12).

Notice that we have considered traces of lowest order Raviart-Thomas elements for the discretization of flux $\hat{t}$. The inf-sup condition for the lowest order RT spaces implies automatically the analogous condition for elements of arbitrary order; increasing the dimension of space U only makes the discrete inf-sup constant bigger.

Inf Sup in Kernel Condition is satisfied automatically due to the use of optimal test functions. First of all, we characterize the "kernel" space:

$$\begin{aligned} \boldsymbol{U}_0 := & \quad \{\boldsymbol{w} \in \boldsymbol{U} \,:\, c(q, \boldsymbol{w}) = 0 \quad \forall q \in Q\} \\ = & \quad \{(u, \boldsymbol{\sigma}, \hat{u}, \hat{t}) \,:\, \langle \hat{t}, 1_K \rangle_{\partial K} = 0 \quad \forall K\}\,. \end{aligned} \tag{2.14}$$

In other words, the kernel space contains only the equilibrated fluxes. With $\boldsymbol{u} \in \boldsymbol{U}_0$, we have then:

$$\begin{aligned} \sup_{\boldsymbol{w} \in \boldsymbol{U}_0} \frac{|a(\boldsymbol{u}, \boldsymbol{w})|}{||\boldsymbol{w}||_{\boldsymbol{U}}} \geq \frac{|b(\boldsymbol{u}, T\boldsymbol{u})|}{||\boldsymbol{u}||} = \frac{|b(\boldsymbol{u}, T\boldsymbol{u})|}{||T\boldsymbol{u}||} \frac{||T\boldsymbol{u}||}{||\boldsymbol{u}||} \\ = \sup_{(v,\boldsymbol{\tau})} \frac{|b((u, \boldsymbol{\sigma}, \hat{u}, \hat{t}), (v, \boldsymbol{\tau}))|}{||(v, \boldsymbol{\tau})||} \frac{||T\boldsymbol{u}||}{||\boldsymbol{u}||} \geq \gamma^2 ||(u, \boldsymbol{\sigma}, \hat{u}, \hat{t})||\,, \end{aligned} \tag{2.15}$$

where γ is the stability constant for the standard continuous DPG formulation. The first inequality follows as we plug in the definition for a and pick $\boldsymbol{w} = \boldsymbol{u}$. The second equality is trivial, while the next one follows by definition of the optimal test functions given through the trial-to-test operator T. The finally inequality springs from the fact that $\sup_{\boldsymbol{v}} \frac{|b(\boldsymbol{u}, \boldsymbol{v})|}{||\boldsymbol{v}||} \geq \gamma ||\boldsymbol{u}||$ and $||T\boldsymbol{u}||_V = ||R_V^{-1} B\boldsymbol{u}||_V = ||B\boldsymbol{u}||_{V'} \geq \gamma ||\boldsymbol{u}||$.

With both discrete inf-sup constants in place, we have the standard result: the FE error is bounded by the best approximation error in the constrained space. Notice that the exact Lagrange multipliers are zero, so the best approximation error involves only the solution $(u, \boldsymbol{\sigma}, \hat{u}, \hat{t})$.

2.2.2.1 Robustness Analysis

Recall the line of analysis leading to the construction of robust test norms allowing us to bound the L^2 error of the field variables by the energy

error, (2.3). With robust test norms, we have

$$\begin{aligned}(||u-u_h||^2+||\boldsymbol{\sigma}-\boldsymbol{\sigma}_h||^2)^{\frac{1}{2}} &\lesssim ||(u-u_h,\boldsymbol{\sigma}-\boldsymbol{\sigma}_h,\hat{u}-\hat{u}_h,\hat{t}-\hat{t}_h||_E \\ &= \inf_{(w_h,\boldsymbol{\varsigma}_h,\hat{w}_h,\hat{r}_h)} ||(u-w_h,\boldsymbol{\sigma}-\boldsymbol{\varsigma}_h,\hat{u}-\hat{w}_h,\hat{t}-\hat{r}_h||_E\,.\end{aligned} \tag{2.16}$$

The last equality follows from the fact that the DPG method delivers the best approximation error in the energy norm (minimizes the residual). This is no longer true for the conservative version. So, can we claim robustness in the sense of the inequality above for the conservative version as well?

One possible way to attack the problem is to switch to the energy norm in the Brezzi stability analysis. Dealing with the "inf-sup in kernel" condition is simple. Upon replacing the original norm of solution $\boldsymbol{u}$ with the energy norm, both constant γ and continuity constant become unity. In order to investigate the robustness of inf-sup constant β, we need to realize first what the energy norm of the flux $\hat{t}$ is. Given an element K, we solve for the optimal test functions corresponding to the flux $\hat{t}$,

$$\begin{cases} v_K \in H^1(K),\, \boldsymbol{\tau}_K \in \boldsymbol{H}(\mathrm{div},K) \\ ((v_K,\boldsymbol{\tau}_K),(\delta v,\delta\boldsymbol{\tau}))_V = \langle \hat{t},\delta v\rangle_{\partial K} \quad \forall \delta v \in H^1(K), \delta\boldsymbol{\tau}\in \boldsymbol{H}(\mathrm{div},K)\,. \end{cases} \tag{2.17}$$

The energy norm of $\hat{t}$ is then equal to

$$||\hat{t}||_E^2 = \sum_K ||(v_K,\boldsymbol{\tau}_K)||_V^2\,. \tag{2.18}$$

We need to establish sufficient conditions under which the inf-sup and continuity constants for the bilinear form representing the constraint are independent of viscosity ϵ.

Let us start with the inf-sup condition,

$$\sup_{\hat{t}} \frac{|\sum_K \lambda_K \langle \hat{t}, 1_K \rangle_{\partial K}|}{||\hat{t}||_E} \geq \beta \left(\sum_K \mu(K) \lambda_K^2 \right)^{1/2} . \tag{2.19}$$

As in the previous analysis, we select for $\hat{t}$ the trace of Raviart-Thomas interpolant $\boldsymbol{\psi}_h$ of $\boldsymbol{\psi} = R(\sum_K \lambda_K 1_K)$ where R is the right-inverse of the divergence operator constructed by Costabel and McIntosh. The only change compared with the previous analysis, is the evaluation of the norm of $\hat{t}_h$. For this, we need to solve the local problems:

$$\begin{aligned} ((v_K, \boldsymbol{\tau}_K), (\delta v, \delta \boldsymbol{\tau}))_V = \langle \hat{t}, \delta v \rangle_{\partial K} = \int_K \mathrm{div} \boldsymbol{\psi}_h \, \delta v = \int_K \mathrm{div} \boldsymbol{\psi} \, \delta v \\ = \int_K \lambda_K \delta v = \lambda_K (1_K, \delta v)_K \quad \forall \delta v \in H^1(K) \, \forall \delta \boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, K) . \end{aligned} \tag{2.20}$$

We need then an upper bound of the energy norm of $(v_h, \boldsymbol{\tau}_h)$:

$$\left(\sum_K ||(v_K, \boldsymbol{\tau}_K)||_V^2 \right)^{1/2} .$$

Substituting $(v_K, \boldsymbol{\tau}_K)$ for $(\delta v, \delta \boldsymbol{\tau})$ in (2.20), we get,

$$||(v_K, \boldsymbol{\tau}_K)||_V^2 = \lambda_K (1_K, v_K)_K . \tag{2.21}$$

If we have a robust stability estimate:

$$|(1_K, v_K)_K| \leq C \mu(K)^{1/2} ||(v_K, \boldsymbol{\tau}_K)||_V \tag{2.22}$$

(i.e. constant C is independent of ϵ), then

$$||(v_K, \boldsymbol{\tau}_K)||_V \leq C \mu(K)^{1/2} |\lambda_K| \tag{2.23}$$

and, eventually as needed,

$$\sum_K ||(v_K, \boldsymbol{\tau}_K)||_V^2 \leq C^2 \sum_K \mu(K)\lambda_K^2 \,, \tag{2.24}$$

which leads to the robust estimate of inf-sup constant β. For example, it is sufficient if

$$||v||_{L^2(K)} \leq ||(v_K, \boldsymbol{\tau}_K)||_V \,. \tag{2.25}$$

Notice that the stability analysis with the energy norm was, in a sense, easier than with the quotient norm. Only the divergence of the interpolant $\boldsymbol{\psi}_h$ enters (2.20) and it coincides with the divergence of $\boldsymbol{\psi}$.

We arrive at a similar situation in the continuity estimate of

$$\sum_K \lambda_K \langle \hat{t}, 1_K \rangle_{\partial K} \,.$$

Testing with $(1_K, \mathbf{0})$ in the local problem (2.17), we obtain,

$$((v, \boldsymbol{\tau}), (1_K, \mathbf{0}))_V = \langle \hat{t}, 1_K \rangle_{\partial K} \,. \tag{2.26}$$

If we have a robust estimate,

$$|((v, \boldsymbol{\tau}), (1_K, \mathbf{0}))_V| \leq C\mu(K)^{1/2} \, ||(v, \boldsymbol{\tau})||_V \,, \tag{2.27}$$

then

$$\begin{aligned} |\sum_K \lambda_K \langle \hat{t}, 1_K \rangle| &\leq C(\sum_K \mu(K)\lambda_K^2)^{1/2} \, (\sum_K ||(v, \boldsymbol{\tau})||_V^2)^{1/2} \\ &= C(\sum_K \mu(K)\lambda_K^2)^{1/2} ||\hat{t}||_E \leq C \, \|\boldsymbol{\lambda}\| \, \|\boldsymbol{u}\|_E \,, \end{aligned} \tag{2.28}$$

as needed.

For instance, condition (2.27) will be satisfied if the test inner product in (2.26) reduces to the L^2 term only,

$$((v, \boldsymbol{\tau}), (1_K, \mathbf{0}))_V = (v, 1_K)_{L^2(K)} \,. \tag{2.29}$$

With the robust stability and continuity constants for the mixed problem, the energy error of solution $(u, \boldsymbol{\sigma}, \hat{u}, \hat{t})$ (and Lagrange multipliers λ_K as well) is bounded robustly by the *best approximation error* of $(u, \boldsymbol{\sigma}, \hat{u}, \hat{t})$ measured in the energy norm. We arrive thus at the same situation as in the standard DPG method.

2.2.3 Robust Test Norms

The optimal test functions are determined by solving local problems determined by the choice of test norm. There are several options to consider. The graph norm [32] is one of the most natural norms to consider as it is derived directly from the adjoint of the problem supplemented with (possibly scaled) L^2 field terms to upgrade it from a semi-norm. Chan *et al.* [17] derived a more robust alternative norm for convection diffusion (dubbed the robust norm). We recently developed a modification of the robust norm that produces better results in the presence of singularities; for more details and motivation, see [18].

$$\begin{aligned} \|(v, \boldsymbol{\tau})\|^2_{V,K} \coloneqq \min\left\{\frac{1}{\epsilon}, \frac{1}{\mu(K)}\right\} \|\boldsymbol{\tau}\|^2_K + \|\nabla \cdot \boldsymbol{\tau} - \boldsymbol{\beta} \cdot \nabla v\|^2_K \\ + \|\boldsymbol{\beta} \cdot \nabla v\|^2_K + \epsilon \|\nabla v\|^2_K + \|v\|^2_K \,, \end{aligned} \tag{2.30}$$

where $||\cdot||_K$ signifies the L^2 norm over element K.

2.2.3.1 Adaptation for a Locally Conservative Formulation

With this choice of test norm, our local problem now becomes:

Find $v_{\delta \boldsymbol{u}_h} \in H^1(K)$, $\boldsymbol{\tau}_{\delta \boldsymbol{u}_h} \in \boldsymbol{H}(\mathrm{div}, K)$ such that:

$$\begin{aligned}\min\left\{\frac{1}{\epsilon}, \frac{1}{\mu(K)}\right\}(\boldsymbol{\tau}_{\delta \boldsymbol{u}_h}, \delta\boldsymbol{\tau})_K + (\nabla\cdot\boldsymbol{\tau}_{\delta \boldsymbol{u}_h} - \boldsymbol{\beta}\cdot\nabla v_{\delta \boldsymbol{u}_h}, \nabla\cdot\delta\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla\delta v)_K \\ + (\boldsymbol{\beta}\cdot\nabla v_{\delta \boldsymbol{u}_h}, \boldsymbol{\beta}\cdot\nabla\delta v)_K + \epsilon(\nabla v_{\delta \boldsymbol{u}_h}, \nabla\delta v)_K + \alpha(v_{\delta \boldsymbol{u}_h}, \delta v)_K = b(\delta\boldsymbol{u}_h, (\delta v, \delta\boldsymbol{\tau})) \\ \forall \delta v \in H^1(K),\, \delta\boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, K)\,, \qquad (2.31)\end{aligned}$$

where typically $\alpha = 1$.

With a locally conservative formulation, we can take $\alpha = 0$ in local problem (2.31). The fact that the test functions will be determined then up to a constant does not matter, for $\hat{t}$ in equation $(2.6)_1$ is orthogonal to constants. Mathematically, we are dealing with equivalence classes of functions, but in order to obtain a single function that we can deal with numerically, we replace the alpha term with a zero mean scaling condition to obtain the new test norm,

$$\begin{aligned}\min\left\{\frac{1}{\epsilon}, \frac{1}{\mu(K)}\right\}(\boldsymbol{\tau}_{\delta \boldsymbol{u}_h}, \delta\boldsymbol{\tau})_K + (\nabla\cdot\boldsymbol{\tau}_{\delta \boldsymbol{u}_h} - \boldsymbol{\beta}\cdot\nabla v, \nabla\cdot\delta\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v)_K \qquad (2.32) \\ + (\boldsymbol{\beta}\cdot\nabla v_{\delta \boldsymbol{u}_h}, \boldsymbol{\beta}\cdot\nabla\delta v)_K + \epsilon(\nabla v_{\delta \boldsymbol{u}_h}, \nabla\delta v)_K + \frac{1}{\mu(K)}\int_K v_{\delta \boldsymbol{u}_h}\int_K \delta v\,,\end{aligned}$$

where the $\frac{1}{\mu(K)}$ coefficient is an arbitrary scaling condition that doesn't make a difference mathematically, but can affect the condition number of the actual solve. In practice, we use $\frac{1}{\mu(K)^2}$ since $\int_K v_{\delta \boldsymbol{u}_h}$ and $\int_K \delta v$ both scale like $\mu(K)$, but $\frac{1}{\mu(K)}$ is more convenient for the analysis in the next section. It is convenient to be able to take $\alpha = 0$ as we will see in some later numerical

experiments. We've noticed that this particularly helps with conditioning of the local problem as the mesh size decreases.

2.2.3.2 Verification of Robust Stability Estimate

In the robustness analysis in Section 2.2.2.1, we argued that if we have robust stability estimates:

$$(1_K, v_K) \leq C\mu(K)^{1/2}||(v, \boldsymbol{\tau})||_K \qquad \text{(2.22 revisted)}$$

and

$$|((v, \boldsymbol{\tau}), (1_K, \mathbf{0}))_V| \leq C\mu(K)^{1/2}\,||(v, \boldsymbol{\tau})||_V\,. \qquad \text{(2.27 revisted)}$$

then the conservative DPG method is robust.

We now proceed to show that the robust norms we are using satisfy this requirement. Consider the inner product from (2.31), with $\alpha = 1$. We wish to verify condition (2.22) with the norm derived from this inner product on the right hand side. By Cauchy-Schwarz

$$\int_K v \cdot 1 \leq \mu(K)^{1/2}\,\|v\|_{L^2(K)} \leq \mu(K)^{1/2}\,\|(v, \boldsymbol{\tau})\|_K\,, \qquad (2.33)$$

where $\|(v, \boldsymbol{\tau})\|_K$ is the norm derived from the inner product. Condition (2.27) comes out the same since

$$|((v, \boldsymbol{\tau}), (1_K, \mathbf{0}))| \leq \sum_K |(1_K, v_K)| \leq \sum_K \mu(K)^{1/2}\,\|(v, \boldsymbol{\tau})\|_K$$

element-wise.

Now we need to perform the same analysis for the modified inner product in (2.32). In this case, condition (2.22) follows even more naturally as

$$\int_K v \cdot 1 \leq \mu(K)^{1/2} \frac{1}{\mu(K)^{1/2}} \left| \int_K v \right| \leq \|(v, \boldsymbol{\tau})\|_K \ , \tag{2.34}$$

where $\|(v, \boldsymbol{\tau})\|$ now refers to the norm generated by inner product (2.32). Condition (2.27) follows by the same reasoning.

2.3 Application to Other Fluid Model Problems

Extension of these ideas to other fluid flow problems is relatively trivial. For the following problems, we just use the graph norm for the local problems.

2.3.1 Inviscid Burgers' Equation

We include the inviscid Burgers' equation in our suite of tests because, being a nonlinear hyperbolic conservation law, it falls under the scope of the Lax-Wendroff theorem. The inviscid Burger's equation is

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = f \, .$$

Define the space-time gradient: $\nabla_{xt} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial t}\right)^T$. We can now rewrite this as

$$\nabla_{xt} \cdot \begin{pmatrix} u^2/2 \\ u \end{pmatrix} = f \, .$$

Multiplying by a test function v, and integrating by parts:

$$-\left(\begin{pmatrix} u^2/2 \\ u \end{pmatrix}, \nabla_{xt} v \right) + \left\langle \hat{t}, v \right\rangle = (f, v) \ ,$$

where $\hat{t}$ is the trace of $\begin{pmatrix} u^2/2 \\ u \end{pmatrix} \cdot \mathbf{n}_{xt}$ on element boundaries, and $\mathbf{n}_{xt}$ is the space-time normal vector. As in convection-diffusion, local conservation implies that $\int_{\partial K} \hat{t} = \int_K f$ for all elements, K.

In order to solve this nonlinear problem, we linearize and do a simple Newton iteration until the solution converges. The linearized equation is

$$-\left(\begin{pmatrix} u \\ 1 \end{pmatrix} \Delta u, \nabla_{xt} v\right) + \langle \hat{t}, v \rangle = (f, v) + \left(\begin{pmatrix} u^2/2 \\ u \end{pmatrix}, \nabla_{xt} v\right) ,$$

where u is the previous solution iteration and Δu is the update. The results follow in Section 2.4.1.4.

2.3.2 Stokes Flow

We start with the VGP (velocity, gradient pressure) Stokes formulation:

$$\begin{aligned} \mu \Delta \boldsymbol{u} + \nabla p &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \,, \end{aligned}$$

where $\boldsymbol{u}$ is the velocity vector field. As a first order system of equations, this is

$$\begin{aligned} \frac{1}{\mu} \boldsymbol{\sigma} - \nabla \boldsymbol{u} &= 0 \\ \nabla \cdot \boldsymbol{\sigma} + \nabla p &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \,, \end{aligned}$$

where $\boldsymbol{\sigma}$ is a tensor valued stress field. Multiplying by test functions $\boldsymbol{\tau}$ (tensor valued), $\boldsymbol{v}$ (vector valued), and q (scalar valued), and integrating by parts:

$$\begin{aligned}\left(\frac{1}{\mu}\boldsymbol{\sigma},\boldsymbol{\tau}\right)+(\boldsymbol{u},\nabla\cdot\boldsymbol{\tau})-\langle\hat{\boldsymbol{u}},\boldsymbol{\tau}\cdot\mathbf{n}\rangle&=0\\ -(\boldsymbol{\sigma},\nabla\boldsymbol{v})-(p,\nabla\cdot\boldsymbol{v})+\left\langle\hat{\boldsymbol{t}},\boldsymbol{v}\right\rangle&=(\boldsymbol{f},\boldsymbol{v})\\ -(\boldsymbol{u},\nabla q)+\langle\hat{\boldsymbol{u}}\cdot\mathbf{n},q\rangle&=0\,,\end{aligned}$$

where $\hat{\boldsymbol{u}}$ is the trace of $\boldsymbol{u}$, and $\hat{\boldsymbol{t}}$ is the trace of $(\boldsymbol{\sigma}+p\mathbf{I})\cdot\mathbf{n}$. The solve for p is only unique up to a constant, so we also impose a zero mean condition, $\int_\Omega p=0$. Local conservation for Stokes flow means that over each element, $\int_K\hat{\boldsymbol{u}}\cdot\mathbf{n}=0$. Results follow in Sections 2.4.1.5 and 2.4.1.6.

2.4 Numerical Experiments

In 2.4.1 we define each numerical experiment, and in 2.4.2 we discuss the solution properties in general. We solve with second order field variables and flux (u, $\boldsymbol{\sigma}$, and $\hat{t}$), third order traces ($\hat{u}$), and fifth order test functions (v and $\boldsymbol{\tau}$).

We measure flux imbalance by looping over each element in the mesh and integrating the flux over each side and summing them together. We then integrate the source term over the volume of the element. The two should match each other, and the remainder is the flux imbalance. We get the net global flux imbalance by summing these quantities and taking the absolute value. The max local flux imbalance is the maximum absolute value of these flux imbalances.

2.4.1 Description of Problems

Unless otherwise noted, the problem domain is $\Omega = [0,1]^2$ and $f = 0$. Also note that unless otherwise noted, for all of the pseudo-color plots, blue corresponds to 0 and red to 1 with a linear scaling in between. Also, all convection-diffusion plots are of the field variable u. Inviscid Burgers' and Stokes results will be dealt with individually.

2.4.1.1 Eriksson-Johnson Model Problem

The Eriksson-Johnson problem is one of the few convection-diffusion problems with a known analytical solution. Take $\boldsymbol{\beta} = (1,0)^T$ and boundary conditions $\hat{t} = \boldsymbol{\beta} \cdot \mathbf{n} u_0$ when $\beta_n \leq 0$, where u_0 is the trace of the exact solution, and $\hat{u} = 0$ when $\beta_n > 0$. For $n = 1, 2, \cdots$, let $\lambda_n = n^2\pi^2\epsilon$, $r_n = \frac{1+\sqrt{1+4\epsilon\lambda_n}}{2\epsilon}$, and $s_n = \frac{1-\sqrt{1+4\epsilon\lambda_n}}{2\epsilon}$. The exact solution is

$$u(x,y) = C_0 + \sum_{n=1}^{\infty} C_n \frac{\exp(s_n(x-1)) - \exp(r_n(x-1))}{r_n \exp(-s_n) - s_n \exp(-s_n)} \cos(n\pi y)\,. \qquad (2.35)$$

The exact solution for $\epsilon = 10^{-2}$, $C_1 = 1$, and $C_{n\neq 1} = 0$ is shown in Figure 2.1. Error convergence and flux imbalance are shown in Figure 2.2 and Figure 2.3.

2.4.1.2 Vortex Problem

This problem models a mildly diffusive vortex convecting fluid in a circle. We deal with domain $\Omega = [-1,1]^2$, with $\epsilon = 10^{-4}$, and $\boldsymbol{\beta} = (-y, x)^T$. Note that $\boldsymbol{\beta} = \mathbf{0}$ at the domain center. We have an inflow boundary condition when $\boldsymbol{\beta} \cdot \mathbf{n} < 0$, in which case we set $\hat{t} = \boldsymbol{\beta} \cdot \mathbf{n} \cdot u_0$ where $u_0 = \frac{\sqrt{x^2+y^2}-1}{\sqrt{2}-1}$

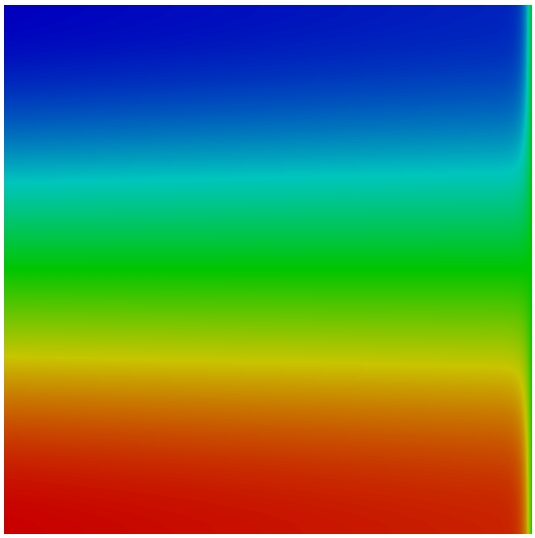


Figure 2.1: Erickson-Johnson exact solution

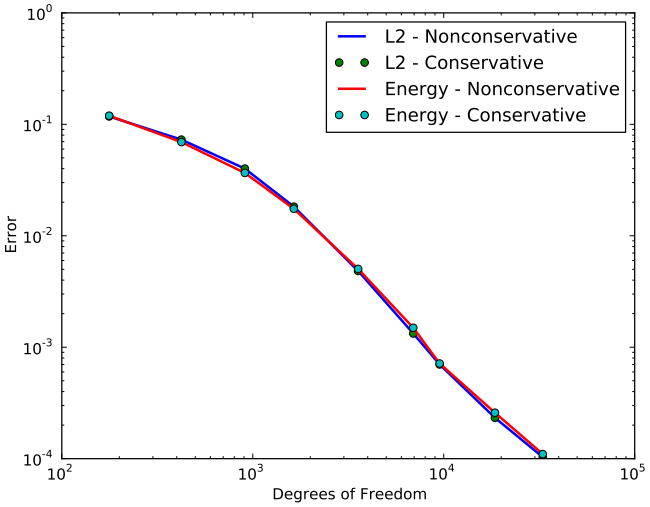


Figure 2.2: Error in Erickson-Johnson solutions

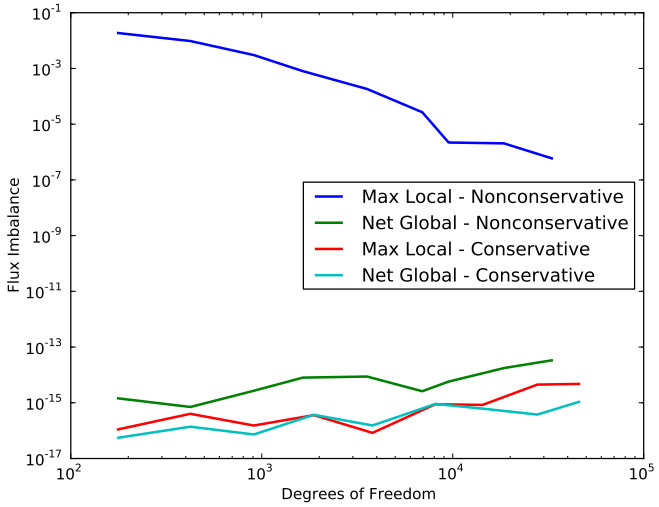
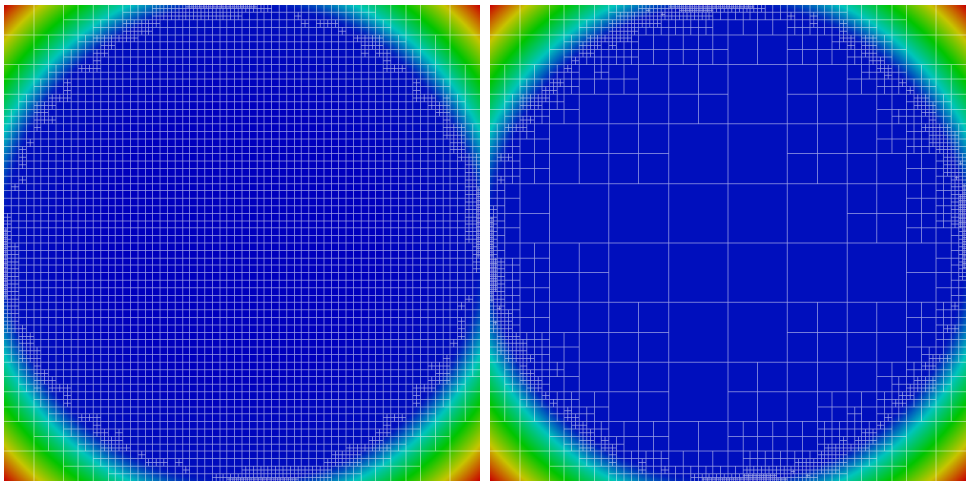


Figure 2.3: Flux imbalance in Erickson-Johnson solutions

which will vary from 0 at the center of boundary edges to 1 at corners. We don't enforce an outflow boundary. Results and flux imbalance are shown in Figure 2.4 and Figure 2.5.

2.4.1.3 Discontinuous Source Problem

Here, $\boldsymbol{\beta} = (0.5, 1)^T/\sqrt{1.25}$, and we have a discontinuous source term such that $f = 1$ when $y \geq 2x$ and $f = -1$ when $y < 2x$. We apply boundary conditions of $\hat{t} = 0$ on the inflow and $\hat{u} = 0$ on the outflow. Contrary to the other problems discussed, the solution for this problem does not range from zero to one. Rather, the colorbar in Figure 2.6 is scaled to $[-1.110, 0.889]$. Flux imbalance is shown in Figure 2.7.



(a) Nonconservative (b) Conservative

Figure 2.4: Vortex problem after 6 refinements

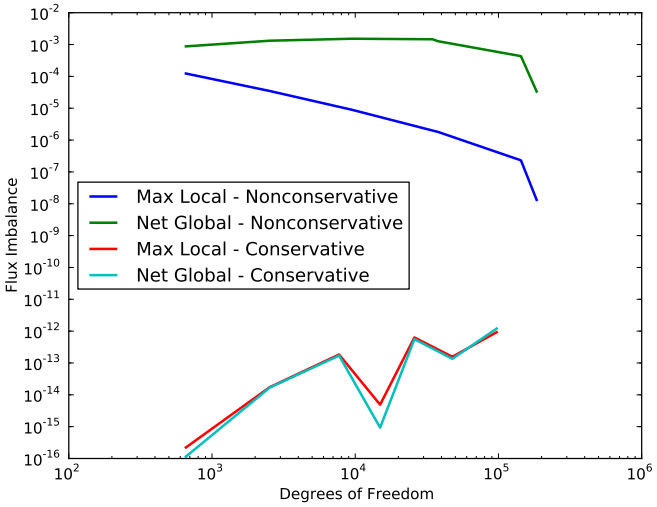
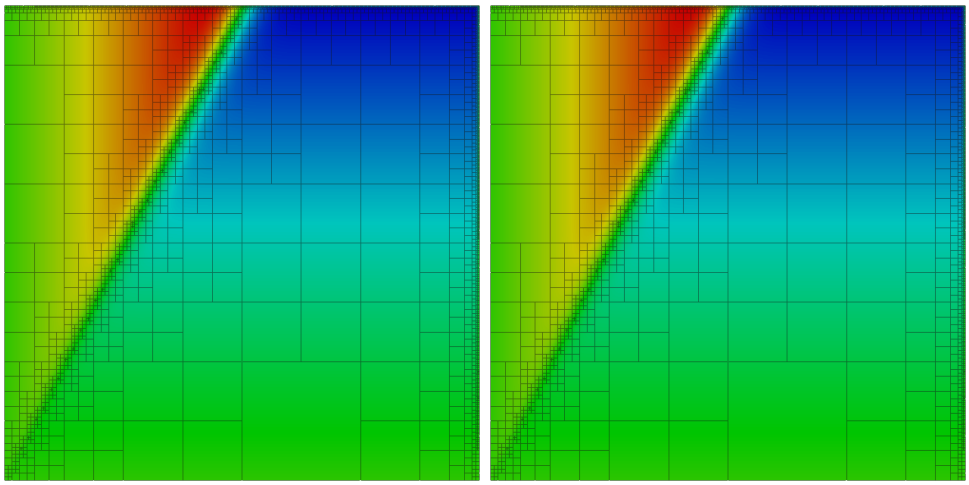


Figure 2.5: Flux imbalance in vortex solutions



(a) Nonconservative (b) Conservative

Figure 2.6: Discontinuous source problem after 8 refinements

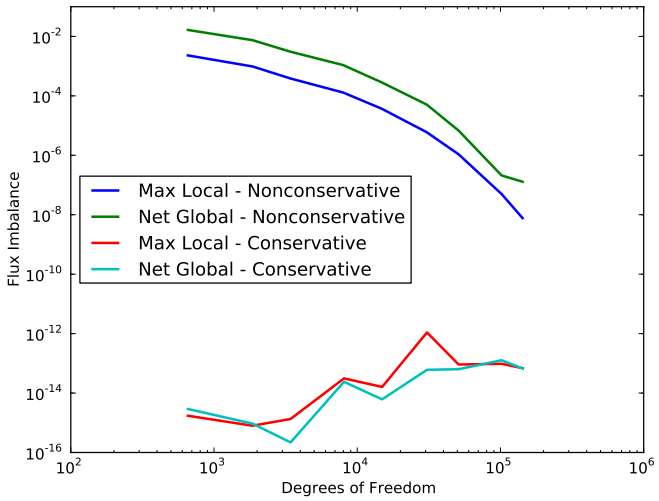


Figure 2.7: Flux imbalance in discontinuous source solutions

2.4.1.4 Inviscid Burgers' Equation

This is a standard test problem for Burgers' equation. The domain is a unit square. We assign boundary conditions $\hat{t} = -(1 - 2x)$ on the bottom, $\hat{t} = -1/2$ on the left, while $\hat{t} = 1/2$ on the right. Since this is a hyperbolic equation, there is no need to set a boundary condition on the top. Results and flux imbalance are shown in Figure 2.8 and Figure 2.9.

2.4.1.5 Stokes Flow Around a Cylinder

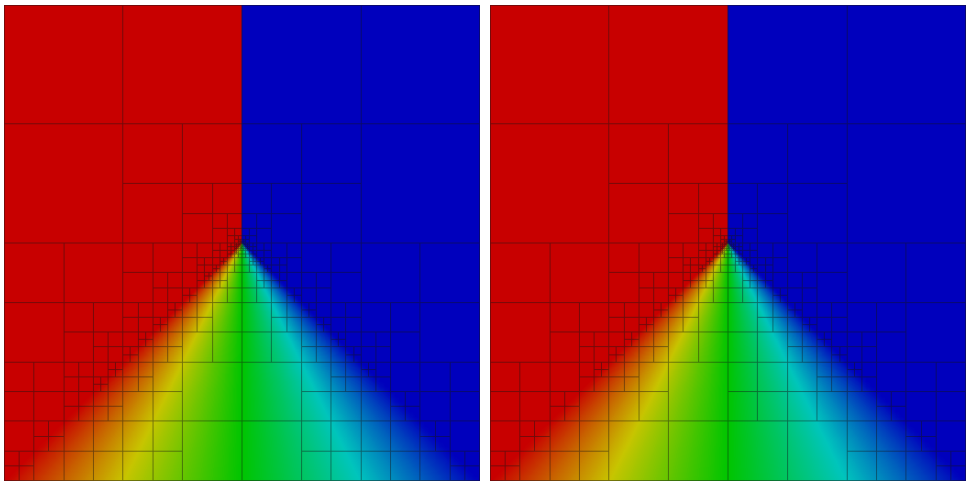
This is a common problem used to stress-test local conservation properties of least squares finite element methods. Since DPG can be viewed as a generalized least squares methods[32], we might expect it to struggle with this problem as well. The problem domain is detailed in Figure 2.10 with inlet and outlet velocity profiles

$$\boldsymbol{u}_{in} = \boldsymbol{u}_{out} = \begin{pmatrix} (1-y)(1+y) \\ 0 \end{pmatrix},$$

and zero flow on the cylinder and at the top and bottom walls. We use $\mu =$ with both Stokes problems and set velocity boundary conditions on $\hat{\boldsymbol{u}}$.

Bochev *et al.* [8] run this test with both $r = 0.6$ and $r = 0.9$; we repeat the same experiments with standard and conservative DPG methods starting from the very coarse meshes shown in Figure 2.11 while adaptively refining toward a resolved solution. The extreme pressure gradient in the $r = 0.9$ case obviously makes local conservation more challenging.

We measure mass loss more directly in these two Stokes problems. Be-



(a) Nonconservative (b) Conservative

Figure 2.8: Burgers' problem after 8 refinements

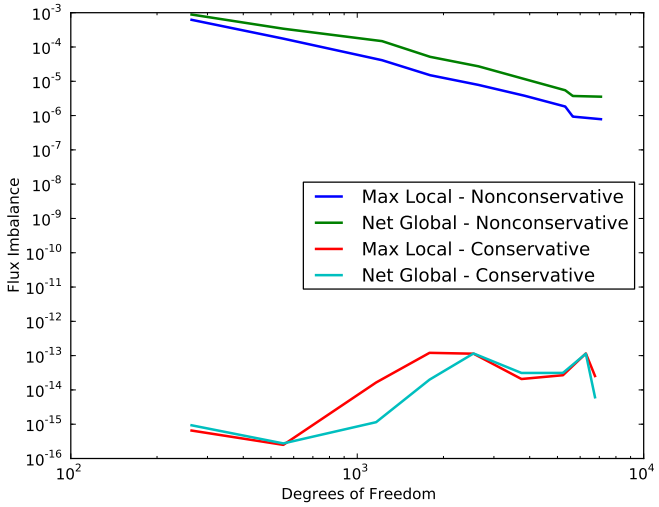


Figure 2.9: Flux imbalance in Burgers' solutions

cause fluid enters and leaves the domain only through the inlet and outlet boundaries, we should be able to integrate the mass flux over any cross-section of the mesh and get the same value. Unfortunately, it is not mathematically well-defined to take line integrals of our field variables which only live in L^2. We can however integrate the trace and flux variables over element boundaries. This carries the unfortunate limitation that we can only measure mass loss where there is a clear vertical mesh line. We therefore pick integration lines from the initial coarse mesh and measure the mass flux after each adaptive refinement step. The percent mass loss is thus

$$\%m_{loss} = \frac{\int_{\Gamma_{in}} \boldsymbol{u} \cdot \boldsymbol{n}_{in} d\ell - \int_S \boldsymbol{u} \cdot \boldsymbol{n}_S d\ell}{\int_{\Gamma_{in}} \boldsymbol{u} \cdot \boldsymbol{n}_{in} d\ell} \times 100,$$

where S is some vertical mesh line. Results and mass loss are shown in Figures 2.12 - 2.14.

2.4.1.6 Stokes Flow Over a Backward Facing Step

Similarly, least squares methods have historically performed very poorly when calculating Stokes flow over a backward facing step shown in Figure 2.15. The stress singularity at the reentrant corner seems to destroy local conservation. We assign parabolic inlet and outlet velocity boundary conditions

$$\boldsymbol{u}_{in} = \begin{pmatrix} 8(y-0.5)(1-y) \\ 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{u}_{out} = \begin{pmatrix} y(1-y) \\ 0 \end{pmatrix}$$

and zero velocity on all other boundaries. In this problem, we solve with fourth order field and flux variables, fifth order traces, and sixth order test functions. Results and mass loss are shown in Figure 2.4.1.6 and Figure 2.17.

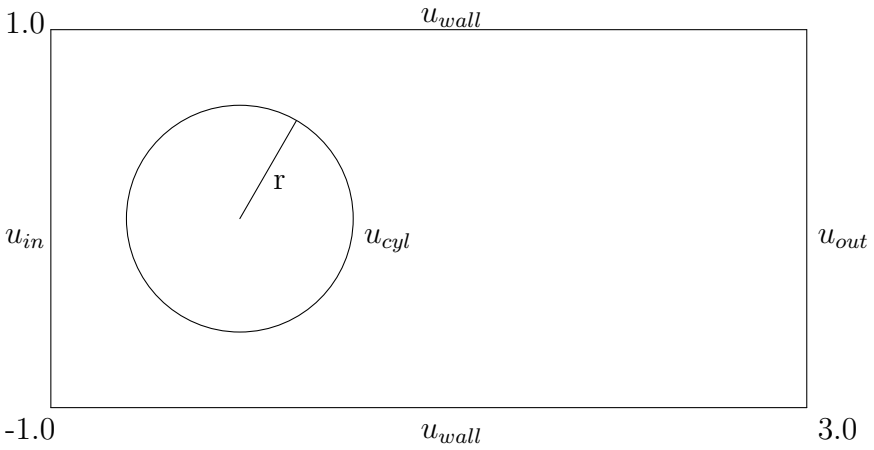
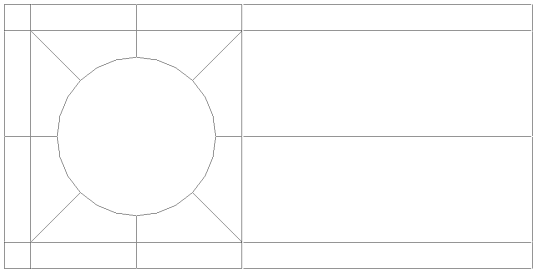
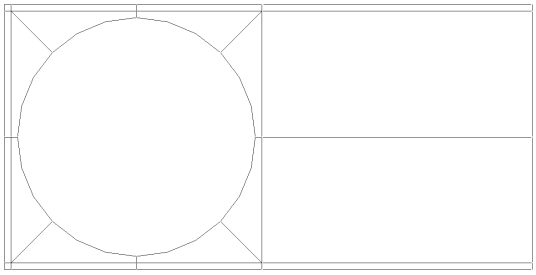


Figure 2.10: Stokes cylinder domain

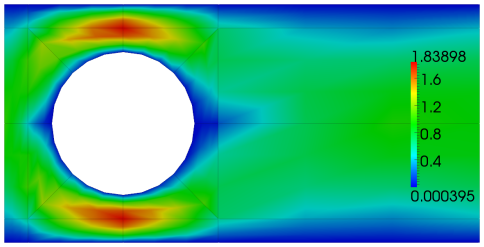


(a) Mesh for $r = 0.6$

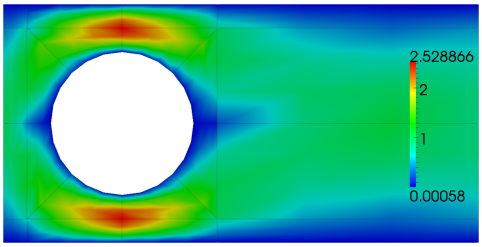


(b) Mesh for $r = 0.9$

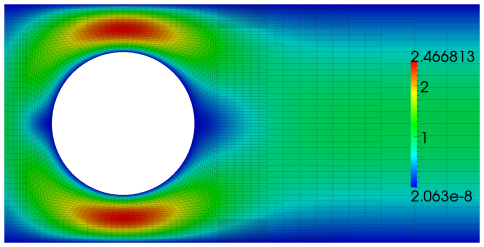
Figure 2.11: Initial mesh for Stokes flow over a cylinder



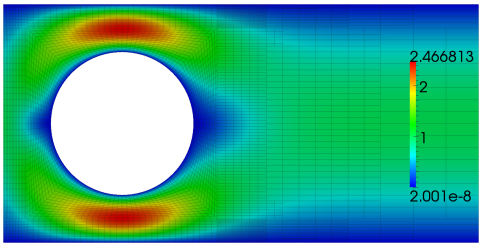
(a) Nonconservative on initial mesh with $r = 0.6$



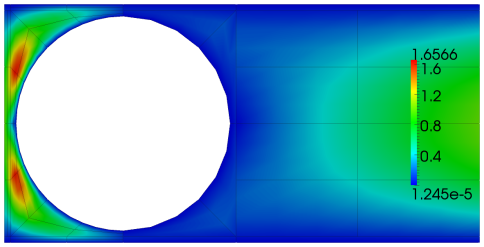
(b) Conservative on initial mesh with $r = 0.6$



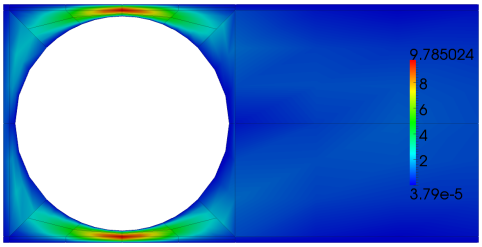
(c) Nonconservative after 6 refinements with $r = 0.6$



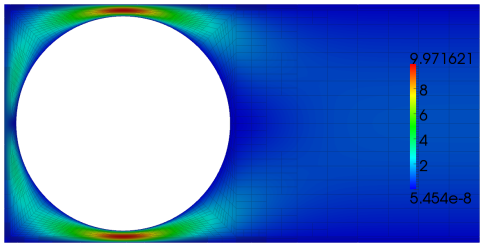
(d) Conservative after 6 refinements with $r = 0.6$



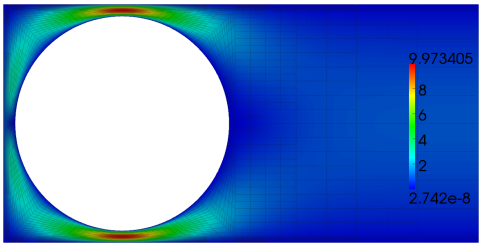
(e) Nonconservative after 1 refinement with $r = 0.9$



(f) Conservative after 1 refinement with $r = 0.9$

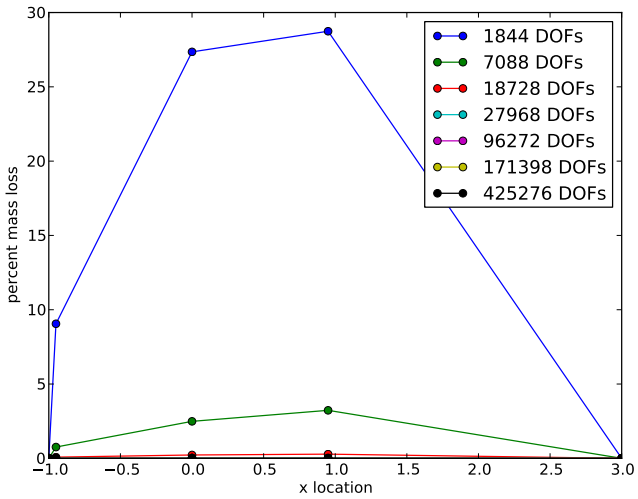


(g) Nonconservative after 6 refinements with $r = 0.9$

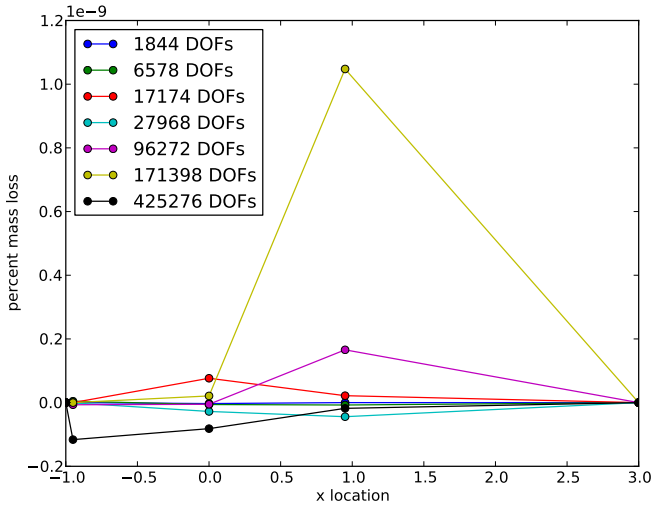


(h) Conservative after 6 refinements with $r = 0.9$

Figure 2.12: Stokes flow around a cylinder - velocity magnitude

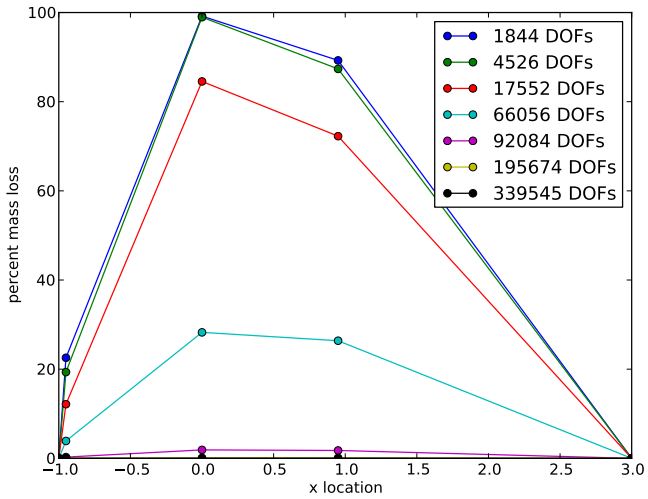


(a) Nonconservative

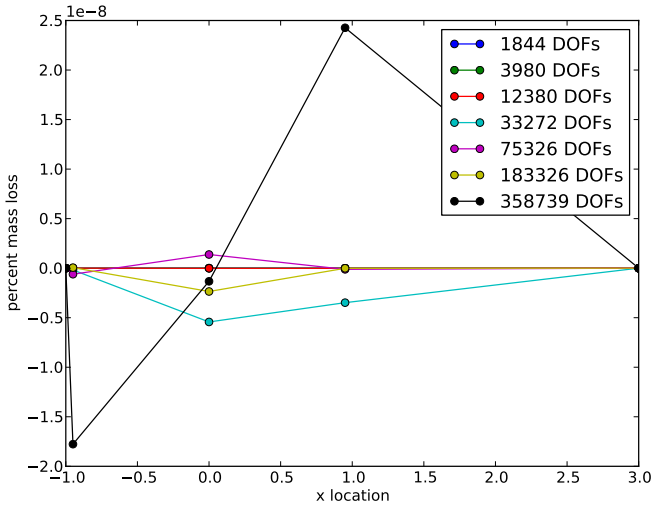


(b) Conservative

Figure 2.13: Mass loss in Stokes flow around a cylinder of radius 0.6



(a) Nonconservative



(b) Conservative

Figure 2.14: Mass loss in Stokes flow around a cylinder of radius 0.9

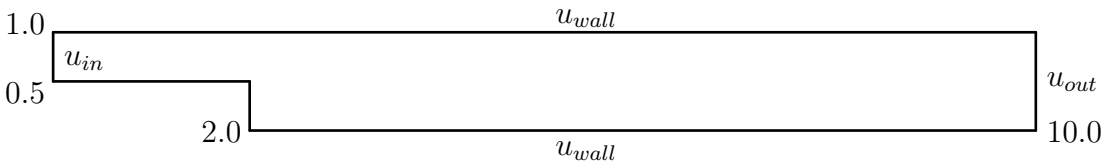


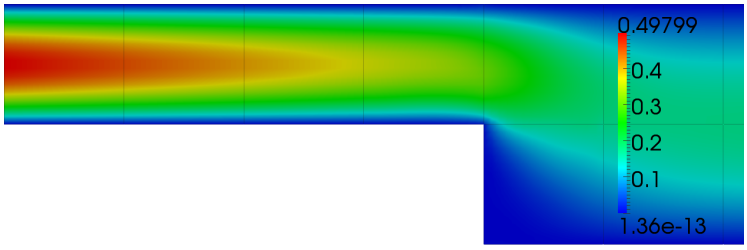
Figure 2.15: Stokes step domain

2.4.2 Analysis of Results

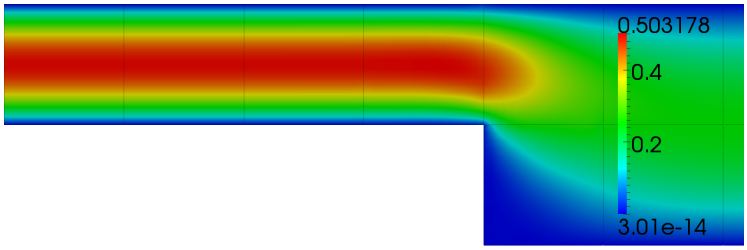
2.4.2.1 Convection-Diffusion Results

The general trend we observe from the results is that the solution quality of the standard and conservative formulations is nearly identical once sufficiently resolved.

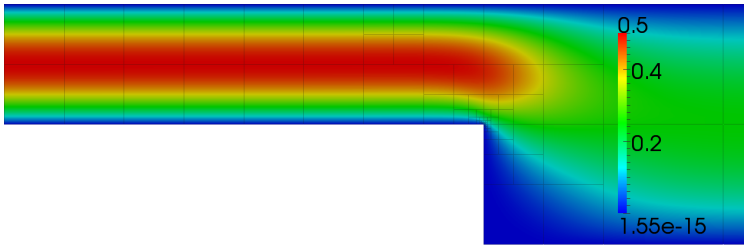
It is clear when comparing the refinement patterns that the two methods appear to calculate slightly different error representation functions (which determine which elements to adaptively refine). Standard DPG minimizes the error in the energy norm, but the Lagrange multipliers in the conservative formulation shift the solution slightly, so we should see somewhat higher error and different elements will get chosen for refinement. The choice of test norm also plays into this calculation of the error representation function. As discussed earlier, the conservative formulation allows us to throw away the L^2 term on v. The inclusion of this term required certain assumptions on $\boldsymbol{\beta}$ [17] that break down for the vortex problem, where $|\boldsymbol{\beta}| \to 0$ in the center of the domain. Here, we see the standard method needlessly refines in the center of the domain where the solution is constant. The conservative scheme is



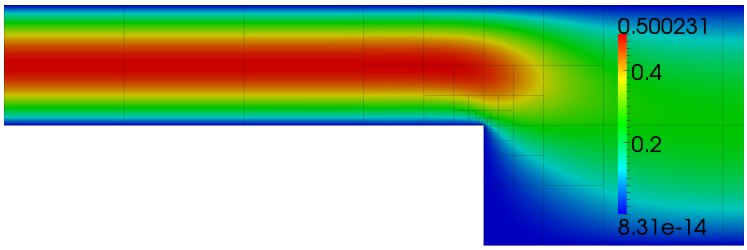
(a) Nonconservative on initial mesh



(b) Conservative on initial mesh

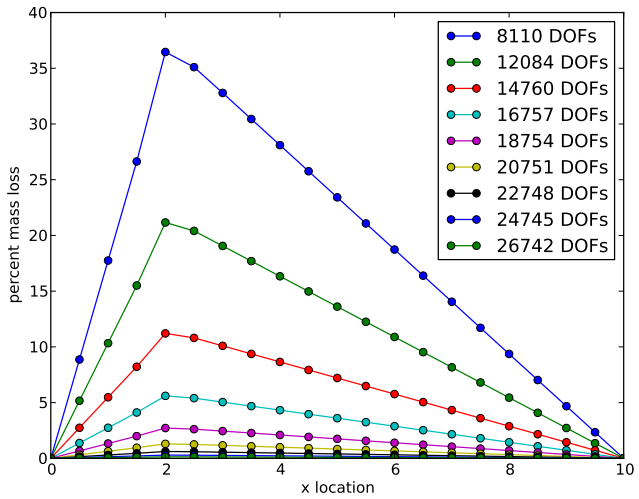


(c) Nonconservative after 8 refinement steps

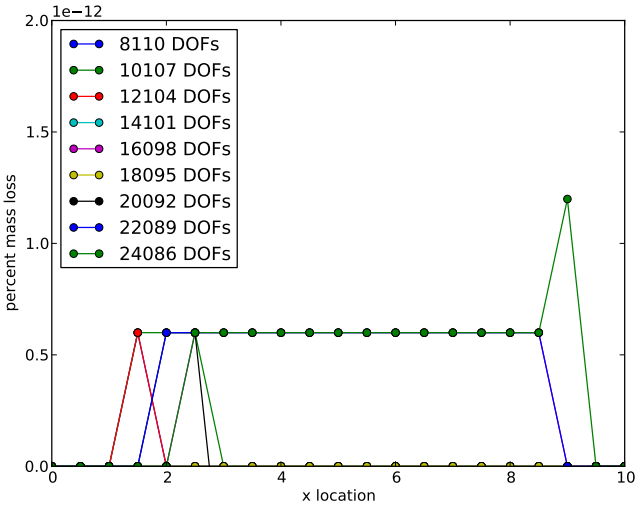


(d) Conservative after 8 refinement steps

Figure 2.16: Stokes backward facing step - velocity magnitude



(a) Nonconservative



(b) Conservative

Figure 2.17: Mass loss in Stokes backward facing step

more discerning about refinements and focuses them where solution features are changing. In general, though, both methods appear to follow very similar refinement patterns.

It should not come as a surprise that the standard and conservative solutions match each other so closely. The conservative formulation enforces local conservation more strictly, but if we examine the flux imbalance plots, the standard DPG formulation is nearly conservative on its own – and appears to become more conservative with refinement. The flux imbalance of the conservative methods appears to bounce around close to the machine epsilon (plus a few orders of magnitude). The level of enforcement appears to creep up with more degrees of freedom, indicating possible accruement of numerical error.

2.4.2.2 Burgers' Results

Standard and conservative DPG perform nearly identically for the inviscid Burgers' problem. It is obvious that the Lax-Wendroff condition of local conservation is a sufficient, but not necessary condition for numerical solutions to hyperbolic conservation laws. We see the same behavior with the flux imbalance plots that was so common with convection-diffusion.

2.4.2.3 Stokes Results

The two Stokes problems are the first ones we encounter that stress the local conservation property of standard DPG. With a cylinder radius of 0.6, standard DPG loses nearly 30% of the mass post-cylinder, but quickly recovers

most of that with further refinement. As we increase the cylinder radius to 0.9, the problem only exacerbates. Nearly 100% of the mass is lost in the constricted region on coarse meshes. It takes a much higher level of resolution to recover the mass loss. The stress singularity at the reentrant corner of the backward facing step causes issues for standard DPG on coarse meshes. It seems that the error in approximating the singularity outweighs the error of missed mass conservation. If we focus refinements at the singularity, the error eventually drops far enough for the method to become nearly conservative. The small amount of mass loss for the conservative method is clearly due to accumulation of floating point error.

The most significant benefit of enforcing local conservation for these problems is that it allows us to recover the essential flow features with much coarser meshes. On the $r = 0.6$ cylinder problem, the peak velocity magnitude of the conservative solution is fairly close on the coarsest mesh, while the nonconservative solution severely underpredicts the peak. With the $r = 0.9$ cylinder, this problem is only worse. After just one adaptive refinement, the conservative solution nails the peak velocity. The nonconservative solution is completely useless at this point. We see the same thing with the backward facing step problem. The conservative solution preserves qualitative features even on the coarsest mesh, while standard DPG requires far higher resolution to achieve a similar solution.

Chapter 3

Robust DPG Methods for Transient Convection-Diffusion

3.1 Introduction

The process of developing robust DPG methods for steady convection-diffusion was explored in [17, 36]. In the sense, the main challenge is to come up with a correct test norm. The residual is measured in the dual test norm, and the DPG method minimizes the residual. The residual can be interpreted as a special *energy norm*. In other words, the DPG method delivers an orthogonal projection in the energy norm. The task is especially challenging for singular perturbation problems. Given a trial norm, we strive to determine a quasi-optimal test norm such that the corresponding energy norm is robustly equivalent to the trial norm of choice. An additional difficulty comes from the fact that the optimal test functions should be easily approximated with a simple enrichment strategy. For convection dominated diffusion, this means that the test functions should not develop boundary layers. The task of determining the quasi optimal test norm (we call it a *robust test norm* leads

[0]This chapter is largely based on the journal article Robust DPG Methods for Transient Convection-Diffusion which appeared as ICES Report 15-21 in 2015. Co-authors Jesse Chan and Leszek Demkowicz assisted with the mathematical proofs contained herein.

then to a stability analysis for the adjoint equation which is the subject of this chapter. For a more general discussion on the subject, see [33]. In this chapter, two new robust norms are derived and numerical verifications of the theory are presented.

3.2 Transient Convection-Diffusion

In order to better illustrate choice of the U and V spaces, we introduce the transient convection-diffusion problem. Consider spatial domain Ω and corresponding space-time domain $Q = \Omega \times [0, T]$ with boundary $\Gamma = \Gamma_- \cup \Gamma_+ \cup \Gamma_0 \cup \Gamma_T$ where Γ_- is the inflow boundary ($\boldsymbol{\beta} \cdot \boldsymbol{n}_x < 0$, where $\boldsymbol{\beta}$ is the convection vector and $\boldsymbol{n}_x$ is the outward spatial normal), Γ_+ is the outflow boundary ($\boldsymbol{\beta} \cdot \boldsymbol{n}_x \geq 0$), Γ_0 is the initial time boundary, and Γ_T is the final time boundary. Let $\Gamma_h := \bigcup \partial K$ denote the entire mesh skeleton, where ∂K denotes the boundary of element K. Γ_{h_x} denotes any parts of the skeleton with a nonzero spatial normal and Γ_{h_t} have a nonzero temporal normal.

The transient convection-diffusion equation is

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\beta} u) - \epsilon \Delta u = f \,,$$

where u is the quantity of interest, often interpreted to be a concentration of some quantity, ϵ is the diffusion coefficient, and f is the source term.

We apply flux boundary conditions on the inflow and trace boundary

conditions on the outflow

$$\begin{aligned} \operatorname{tr}\left(\boldsymbol{\beta}\cdot u-\epsilon\nabla u\right)\cdot\boldsymbol{n}_x &= t_- \quad \text{on} \quad \Gamma_- \\ \operatorname{tr}\left(u\right) &= u_+ \quad \text{on} \quad \Gamma_+ \\ \operatorname{tr}\left(u\right) &= u_0 \quad \text{on} \quad \Gamma_0\,. \end{aligned}$$

We note that Dirichlet boundary conditions also induce Dirichlet boundary conditions for the adjoint problem. Since the direction of convection is reversed for the adjoint convection-diffusion problem, this results in boundary layer adjoint solutions, which must be controlled using special weighted norms [36, 69]. However, since the convection-diffusion operator is not self-adjoint, the Cauchy inflow boundary condition induces a Neumann boundary condition for the adjoint problem. As a result, the adjoint solution does not contain boundary layers, simplifying the construction of a robust DPG method.

3.2.1 Relevant Sobolev Spaces

We begin by defining operators $\nabla_{xt}u := \begin{pmatrix} \nabla u \\ \frac{\partial u}{\partial t} \end{pmatrix}$ and $\nabla_{xt}\cdot\boldsymbol{u} := \nabla\cdot\boldsymbol{u}_x + \frac{\partial u_t}{\partial t}$, where $\boldsymbol{u} = (\boldsymbol{u}_x, u_t)$. We will need the following Sobolev spaces defined on our space-time domain.

$$\begin{aligned} H^1(Q) &= \left\{u \in L^2\left(Q\right) \,:\, \nabla u \in \mathbf{L}^2\left(Q\right)\right\} \\ H^1_{xt}(Q) &= \left\{u \in L^2\left(Q\right) \,:\, \nabla_{xt} u \in \mathbf{L}^2\left(Q\right)\right\} \\ \boldsymbol{H}(\operatorname{div}, Q) &= \left\{\boldsymbol{\sigma} \in \mathbf{L}^2\left(Q\right) \,:\, \nabla\cdot\boldsymbol{\sigma} \in L^2\left(Q\right)\right\} \\ \boldsymbol{H}(\operatorname{div}_{xt}, Q) &= \left\{\boldsymbol{\sigma} \in \mathbf{L}^2\left(Q\right) \,:\, \nabla_{xt}\cdot\boldsymbol{\sigma} \in L^2\left(Q\right)\right\} \end{aligned}$$

We will also need the corresponding broken Sobolev spaces.

$$
\begin{aligned}
H^1(Q_h) &= \left\{u \in L^2(Q) \,:\, u|_K \in H^1(K),\, K \in Q_h\right\} &&= \prod_{K\in Q_h} H^1(K) \\
H^1_{xt}(Q_h) &= \left\{u \in L^2(Q) \,:\, u|_K \in H^1_{xt}(K),\, K \in Q_h\right\} &&= \prod_{K\in Q_h} H^1_{xt}(K) \\
\boldsymbol{H}(\mathrm{div}, Q_h) &= \left\{\boldsymbol{\sigma} \in \mathbf{L}^2(Q) \,:\, u|_K \in \boldsymbol{H}(\mathrm{div}, K),\, K \in Q_h\right\} &&= \prod_{K\in Q_h} \boldsymbol{H}(\mathrm{div}, K) \\
\boldsymbol{H}(\mathrm{div}_{xt}, Q_h) &= \left\{\boldsymbol{\sigma} \in \mathbf{L}^2(Q) \,:\, u|_K \in \boldsymbol{H}(\mathrm{div}_{xt}, K),\, K \in Q_h\right\} &&= \prod_{K\in Q_h} \boldsymbol{H}(\mathrm{div}_{xt}, K)
\end{aligned}
$$

Consider the following trace operators:

$$
\begin{aligned}
\mathrm{tr}^K_{\mathrm{grad}} u &= u|_{\partial K_x} && u \in H^1(K) \\
\mathrm{tr}^K_{\mathrm{div}_{xt}} \boldsymbol{\sigma} &= \boldsymbol{\sigma}|_{\partial K_{xt}} \cdot \boldsymbol{n}_{K_{xt}} && \boldsymbol{\sigma} \in \boldsymbol{H}(\mathrm{div}_{xt}, K)
\end{aligned}
$$

where ∂K_x refers to spatial faces of element K, ∂K_{xt} to the full space-time boundary, and $\boldsymbol{n}_{K_{xt}}$ is the unit outward normal on ∂K_{xt}. The operators tr_{grad} and $\mathrm{tr}_{\mathrm{div}_{xt}}$ perform the same operation element by element to produce the linear maps

$$
\begin{aligned}
\mathrm{tr}_{\mathrm{grad}} \,:\, H^1(Q_h) &\to \prod_{K\in Q_h} H^{1/2}(\partial K_x) \\
\mathrm{tr}_{\mathrm{div}_{xt}} \,:\, \boldsymbol{H}(\mathrm{div}_{xt}, Q_h) &\to \prod_{K\in Q_h} H^{-1/2}(\partial K_{xt})
\end{aligned}
$$

Finally, we define spaces of interface functions. In order that our functions be single valued, we use the following definitions.

$$
\begin{aligned}
H^{1/2}(\Gamma_{h_x}) &= \mathrm{tr}_{\mathrm{grad}} H^1(Q)\,, \\
H^{-1/2}_{xt}(\Gamma_h) &= \mathrm{tr}_{\mathrm{div}_{xt}} H(\mathrm{div}_{xt}, Q)\,.
\end{aligned}
$$

For more details on broken and trace Sobolev spaces, see [15].

3.2.2 Variational Formulations

There are many possible manipulations that could be performed before arriving at a variational formulation. We begin by reformulating the problem in terms of the first order system:

$$\begin{aligned} \frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u &= 0 \\ \nabla_{xt} \cdot \begin{pmatrix} \boldsymbol{\beta} u - \boldsymbol{\sigma} \\ u \end{pmatrix} &= f \,. \end{aligned} \tag{3.1}$$

Multiplying (3.1) by test functions $\boldsymbol{\tau} \in \mathbf{L}^2(Q)$ and $v \in L^2(Q)$, we obtain the following "trivial" variational formulation equivalent to the strong form:

$$\begin{array}{lll} u \in H^1_{xt}(Q) & u = u_+ & \text{on } \Gamma_+ \\ & u = u_0 & \text{on } \Gamma_0 \\ \boldsymbol{\sigma} \in \boldsymbol{H}(\mathrm{div}, Q) & (\boldsymbol{\beta} u - \epsilon \nabla u) \cdot \boldsymbol{n} = t_- & \text{on } \Gamma_- \\ \left(\frac{1}{\epsilon}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) - (\nabla u, \boldsymbol{\tau}) & = 0 & \forall \boldsymbol{\tau} \in \mathbf{L}^2(Q) \\ \left(\nabla_{xt} \cdot \begin{pmatrix} \boldsymbol{\beta} u - \boldsymbol{\sigma} \\ u \end{pmatrix}, v\right) & = f & \forall v \in L^2(Q) \,. \end{array} \tag{3.2}$$

We can now choose either to relax (integrate by parts and build in the boundary conditions) or strongly enforce each equation. The steady state case and resulting options are explored and analyzed in further detail in [27] and are termed the trivial formulation (don't relax anything), the classical formulation (relax the second equation), the mixed formulation (relax the first equation), and the ultra-weak formulation (relax both equations). The stability constants for the four formulations are related, but the functional

settings and norms of convergence change. Early DPG work emphasized the ultra-weak formulation since in many ways it was the easiest to analyze, though recently the classical formulation has been under very active consideration. In the interests of simpler analysis, we focus on the ultra-weak formulation in this chapter.

$$
\begin{aligned}
& u \in L^2(Q)\,,\ \boldsymbol{\sigma} \in \mathbf{L}^2(Q) \\
& \left(\frac{1}{\epsilon}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) + (u, \nabla\cdot\boldsymbol{\tau}) \qquad = 0 \quad \forall \boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, Q) \,:\, \boldsymbol{\tau}\cdot\boldsymbol{n}_x = 0 \text{ on } \Gamma_- \\
& -\left(\begin{pmatrix} \boldsymbol{\beta}u - \boldsymbol{\sigma} \\ u \end{pmatrix}, \nabla_{xt} v\right) \ = f \quad \forall v \in H^1_{xt}(Q) \,:\, v = 0 \text{ on } \Gamma_+ \cup \Gamma_0\,,
\end{aligned}
\tag{3.3}
$$

We can remove the conditions on the test functions by introducing trace unknowns

$$
\begin{aligned}
\hat{u} &= \mathrm{tr}(u) && \text{on } \partial Q_x \\
\hat{t} &= \mathrm{tr}\begin{pmatrix} \boldsymbol{\beta}u - \boldsymbol{\sigma} \\ u \end{pmatrix} \cdot \boldsymbol{n}_{xt} && \text{on } \partial Q_{xt}\,.
\end{aligned}
$$

Our new ultra-weak formulation with conforming test functions is

$$
\begin{aligned}
& u \in L^2(Q)\,,\ \boldsymbol{\sigma} \in \mathbf{L}^2(Q) \\
& \hat{u} \in H^{1/2}(\partial Q_x)\,, && \hat{u} = u_+ && \text{on } \Gamma_+ \\
& \hat{t} \in H^{-1/2}_{xt}(\partial Q)\,, && \hat{t} = t_- \quad \text{on } \Gamma_-\,, \quad \hat{t} = -u_0 \text{ on } \Gamma_0 \\
& \left(\frac{1}{\epsilon}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) + (u, \nabla\cdot\boldsymbol{\tau}) - \langle \hat{u}, \boldsymbol{\tau}\cdot\boldsymbol{n}_x\rangle \quad = 0 && && \forall \boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, Q) \\
& -\left(\begin{pmatrix} \boldsymbol{\beta}u - \boldsymbol{\sigma} \\ u \end{pmatrix}, \nabla_{xt} v\right) + \langle \hat{t}, v\rangle \quad = f && && \forall v \in H^1_{xt}(Q)\,.
\end{aligned}
\tag{3.4}
$$

3.2.3 Broken Test Functions

One of the key insights that led to the development of the DPG framework was the process of breaking test functions, that is testing with functions from larger broken Sobolev spaces, replacing $H^1_{xt}(Q)$ with $H^1_{xt}(Q_h)$ and $\boldsymbol{H}(\mathrm{div}, Q)$ with $\boldsymbol{H}(\mathrm{div}, Q_h)$. Discretizing such spaces is much simpler than standard spaces which require enforcement of global continuity conditions. The cost of introducing broken spaces is that we have to extend our interface unknowns $\hat{u}$ and $\hat{t}$ to live on the mesh skeleton. Our ultra-weak formulation with broken test functions looks like

$$
\begin{aligned}
& u \in L^2(Q)\,,\ \boldsymbol{\sigma} \in \mathbf{L}^2(Q) \\
& \hat{u} \in H^{1/2}(\Gamma_{h_x})\,, && \hat{u} = u_+ && \text{on } \Gamma_+ \\
& \hat{t} \in H^{-1/2}_{xt}(\Gamma_h)\,, && \hat{t} = t_- \quad \text{on } \Gamma_-\,, \quad \hat{t} = -u_0 \text{ on } \Gamma_0 \\
& \left(\frac{1}{\epsilon}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) + (u, \nabla\cdot\boldsymbol{\tau}) - \langle \hat{u}, \boldsymbol{\tau}\cdot\boldsymbol{n}_x\rangle && = 0 && \forall \boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, Q_h) \\
& -\left(\begin{pmatrix} \boldsymbol{\beta} u - \boldsymbol{\sigma} \\ u \end{pmatrix}, \nabla_{xt} v\right) + \langle \hat{t}, v\rangle && = f && \forall v \in H^1_{xt}(Q_h)\,.
\end{aligned}
\tag{3.5}
$$

The main consequence of breaking test functions is that it reduces the cost of solving for optimal test functions from a global solve to an embarrassingly parallel solve element-by-element. Now that we've derived a suitable variational formulation, we are left with the task of selecting a test norm with which to compute our optimal test functions.

3.3 Robust Test Norms

The final unresolved choice is what norm to apply to the V space. This is one of the most important factors in designing a robust DPG method as the corresponding Riesz operator needs to be inverted to solve for the optimal test functions. If the norm produces unresolved boundary layers in the auxiliary problem, then many of the attractive features of DPG may fall apart. This is the primary emphasis of this chapter. The problem of constructing stable test norms for steady convection-diffusion was addressed in [17, 36]. In this chapter, we extend that work to transient convection-diffusion in space-time.

We define a robust test norm such that the L^2 norm of the solution is bounded by the energy norm of the solution with a constant independent of ϵ. We can rewrite any ultra-weak formulation with broken test functions as the following bilinear form with group variables:

$$b\left(\left(u,\hat{u}\right),v\right) = \left(u, A^*v\right)_{L^2} + \left\langle \widehat{u}, [\![v]\!] \right\rangle_{\Gamma_h}$$

where A^* represents the adjoint. In the case of convection-diffusion, $u := \{u, \boldsymbol{\sigma}\}$, $\hat{u} := \left\{\hat{u}, \hat{t}\right\}$, $v := \{v, \boldsymbol{\tau}\}$.

Note that for conforming v^* satisfying $A^*v^* = u$

$$\begin{aligned}\left\|u\right\|_{L^2}^2 = b(u,v^*) &= \frac{b(u,v^*)}{\left\|v^*\right\|_V}\left\|v^*\right\|_V \\ &\leq \sup_{v^*\neq 0}\frac{\left|b(u,v^*)\right|}{\left\|v^*\right\|}\left\|v^*\right\| = \left\|u\right\|_E\left\|v^*\right\|_V .\end{aligned}$$

This defines a necessary condition for robustness, namely that

$$\left\|v^*\right\|_V \lesssim \left\|u\right\|_{L^2} . \tag{3.6}$$

If this condition is satisfied, then we get our final result:

$$\|u\|_{L^2} \lesssim \|u\|_E \ .$$

So far, we've assumed that our finite set of optimal test functions are assembled from an infinite dimensional space. In practice, we have found it to be sufficient to use an "enriched" space of higher polynomial dimension than the trial space [42]. This adds an additional requirement when assembling a robust test norm, namely that our optimal test functions should be adequately representable within this enriched space. We illustrate this point by considering three norms which satisfy the above conditions for 1D steady convection-diffusion. The graph norm is $\left(\|A^*v\|_{L^2}^2 + \|v\|_{L^2}^2\right)^{\frac{1}{2}}$:

$$\|(v,\boldsymbol{\tau})\|^2 = \|\nabla\cdot\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v\|^2 + \left\|\frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v\right\|^2 + \|v\|^2 + \|\boldsymbol{\tau}\|^2 \ .$$

Remark 3.3.1. *In the DPG technology, the test norm must be* localizable*, i.e.,*

$$\|v\|_V^2 = \sum_K \|v\|_{V(K)}^2$$

where $\|v\|_{V(K)}$ denotes a test norm (and not just a seminorm) for the element test space. In practice this means the addition of properly scaled L^2-terms. Without those terms, we could not invert the Riesz operator on the element level. Addition of the L^2 terms does not necessarily contradict the robustness of the norm, see the discussion in [33] on bounded below operators. An alternate strategy was explored in the previous chapter where we enforce the element conservation property by securing the presence of a constant function in the

element test space. The residual is then minimized only over the orthogonal complement to the constants which eliminates the need for adding the L^2-term to the test norm.

The robust norm was derived in [17]:

$$\begin{aligned}\|(v,\boldsymbol{\tau})\|^2 = \|\boldsymbol{\beta}\cdot\nabla v\|^2 + \epsilon\,\|\nabla v\|^2 + \min\left(\frac{\epsilon}{h^2},1\right)\|v\|^2 \\ + \|\nabla\cdot\boldsymbol{\tau}\|^2 + \min\left(\frac{1}{h^2},\frac{1}{\epsilon}\right)\|\boldsymbol{\tau}\|^2\ .\end{aligned}$$

The case for the coupled robust norm was made in [18]:

$$\begin{aligned}\|(v,\boldsymbol{\tau})\|^2 = \|\boldsymbol{\beta}\cdot\nabla v\|^2 + \epsilon\,\|\nabla v\|^2 + \min\left(\frac{\epsilon}{h^2},1\right)\|v\|^2 \\ + \|\nabla\cdot\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v\|^2 + \min\left(\frac{1}{h^2},\frac{1}{\epsilon}\right)\|\boldsymbol{\tau}\|^2\ .\end{aligned}$$

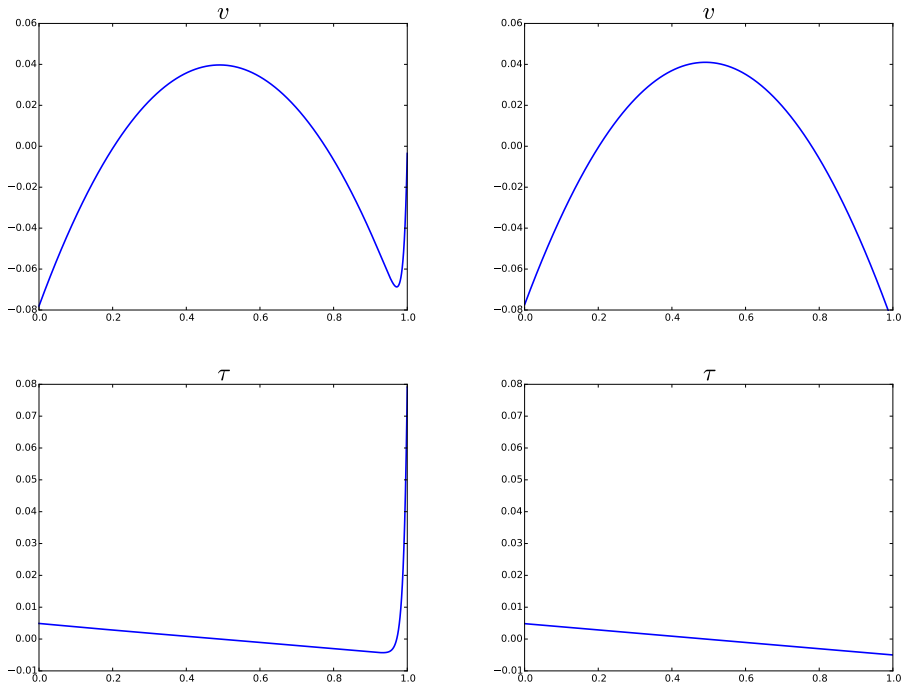
The argument for the coupled norm was that in certain cases we noticed pollution of u from errors in $\boldsymbol{\sigma}$, for example at singularities in $\boldsymbol{\sigma}$, u also exhibited degraded quality with the robust norm. The coupled robust norm seemed to relax this behavior, i.e. errors in u appear more independent of errors in $\boldsymbol{\sigma}$.

The bilinear form and test norm define a mapping from input trial functions to an optimal test function:

$$T = R_V^{-1}B\ :\ U\to V\ .$$

In Figures 3.1 - 3.3, we plot the optimal test functions produced given $\epsilon = 10^{-2}$, a representative trial function $u = x - \frac{1}{2}$, and either the graph norm, the robust norm, or the coupled robust norm. Note that the optimal test functions will

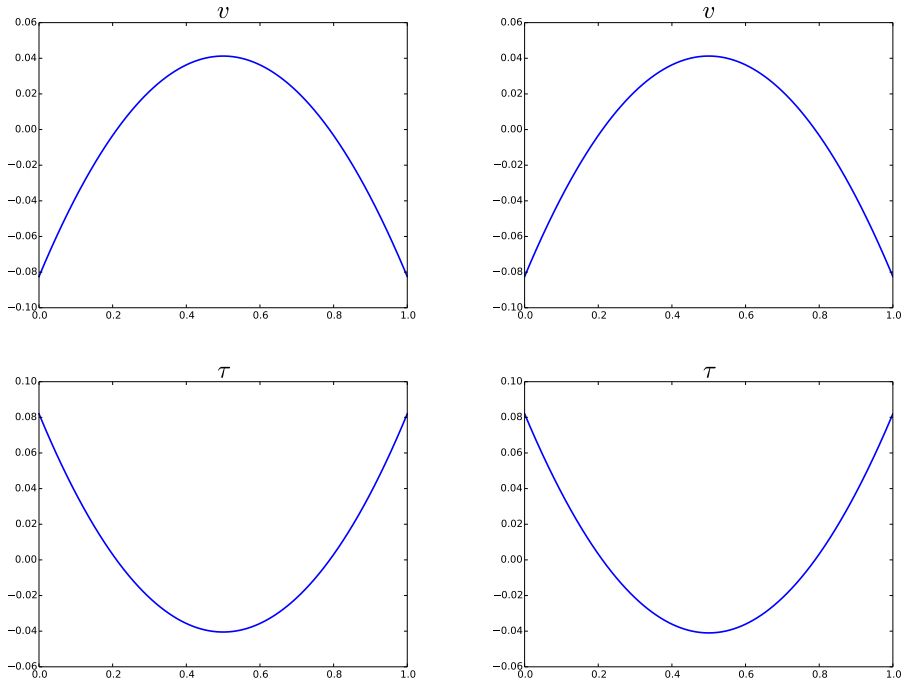
be different for any other trial function. In the left column, we see the fully resolved *ideal* optimal test function that DPG theory relies on. On the right, we see the approximated optimal test function using a enriched cubic test space.



(a) Ideal (b) Approximated

Figure 3.1: Graph norm optimal test functions for $u = x - \frac{1}{2}$

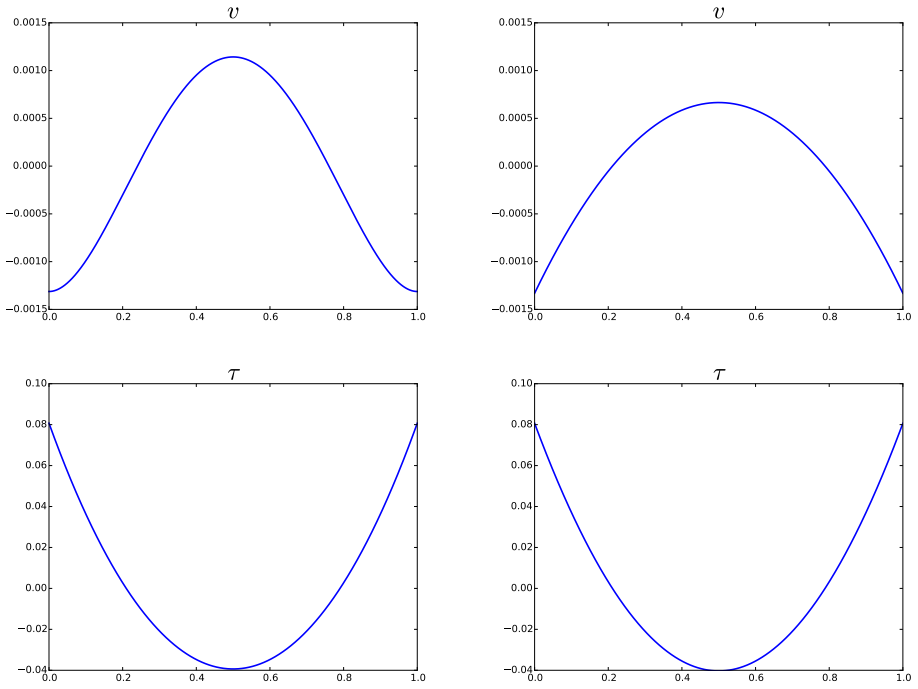
Mathematically, the graph norm satisfies the necessary condition to be a robust norm, but the ideal optimal test functions contain strong boundary layers which can not be realistically approximated with the provided enriched space. If the approximated optimal test functions can not come sufficiently close to the ideal, then the whole DPG theory falls apart. See [42] for more



(a) Ideal (b) Approximated

Figure 3.2: Robust norm optimal test functions for $u = x - \frac{1}{2}$

discussion. This provides an additional condition on a test norm before we can truly call it robust: the ideal test functions must be adequately representable within the provided enriched space. This ultimately comes down to an analysis of the relative magnitudes of individual terms within the test norm, usually attempting to bound reactive or convective terms by diffusive terms. The coupled robust norm satisfies condition (3.6) and also produces relatively smooth optimal test functions that can be sufficiently approximated with a cubic polynomial space. Niemi *et al.* attempted to approximate boundary layers in optimal shape functions with Shishkin meshes [56, 57].



(a) Ideal (b) Approximated

Figure 3.3: Coupled robust norm optimal test functions for $u = x - \frac{1}{2}$

3.3.1 Application to Transient Convection-Diffusion

Now we present the analysis leading to two robust norms for transient convection-diffusion. Consider the problem with homogeneous boundary con-

ditions

$$\begin{aligned}
\frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u &= 0 \\
\frac{\partial u}{\partial t} + \boldsymbol{\beta}\cdot\nabla u - \nabla\cdot\boldsymbol{\sigma} &= f \\
\beta_n u - \epsilon\frac{\partial u}{\partial n} &= 0 \text{ on } \Gamma_- \\
u &= 0 \text{ on } \Gamma_+ \\
u &= u_0 \text{ on } \Gamma_0.
\end{aligned}$$

Let $\tilde{\boldsymbol{\beta}} := \begin{pmatrix} \boldsymbol{\beta} \\ 1 \end{pmatrix}$, then we can rewrite this as

$$\begin{aligned}
\frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u &= 0 \\
\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt} u - \nabla\cdot\boldsymbol{\sigma} &= f \\
\beta_n u - \epsilon\frac{\partial u}{\partial n} &= 0 \text{ on } \Gamma_- \\
u &= 0 \text{ on } \Gamma_+ \\
u &= u_0 \text{ on } \Gamma_0.
\end{aligned}$$

The adjoint operator A^* is given by

$$A^*(v,\boldsymbol{\tau}) = \left(\frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v, -\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt} v + \nabla\cdot\boldsymbol{\tau}\right).$$

We decompose now the continuous adjoint problem

$$A^*(v,\boldsymbol{\tau}) = (\boldsymbol{f}, g)$$

into two cases a continuous part with forcing term g

$$\begin{aligned}
\frac{1}{\epsilon}\boldsymbol{\tau}_1 + \nabla v_1 &= 0 \\
-\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \nabla \cdot \boldsymbol{\tau}_1 &= g \\
\boldsymbol{\tau}_1 \cdot \boldsymbol{n}_x &= 0 \text{ on } \Gamma_- \\
v_1 &= 0 \text{ on } \Gamma_+ \\
v_1 &= 0 \text{ on } \Gamma_T \,,
\end{aligned}$$

and a continuous part with forcing $\boldsymbol{f}$

$$\begin{aligned}
\frac{1}{\epsilon}\boldsymbol{\tau}_2 + \nabla v_2 &= \boldsymbol{f} \\
-\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_2 + \nabla \cdot \boldsymbol{\tau}_2 &= 0 \\
\boldsymbol{\tau}_2 \cdot \boldsymbol{n}_x &= 0 \text{ on } \Gamma_- \\
v_2 &= 0 \text{ on } \Gamma_+ \\
v_2 &= 0 \text{ on } \Gamma_T \,.
\end{aligned}$$

(The boundary conditions can be derived by taking the ultra-weak formulation and choosing boundary conditions such that the temporal flux and spatial flux terms $\langle \widehat{u}, [\![\tau_n]\!] \rangle_{\Gamma_{out}}$ and $\langle \hat{t}_n, [\![v]\!] \rangle_{\Gamma_{in}}$ are zero.)

We can then derive that the test norms

$$\begin{aligned}
\|(v,\boldsymbol{\tau})\|_{V,K}^2 \coloneqq \left\| \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v \right\|_K^2 + \epsilon \|\nabla v\|_K^2 + \|v\|_K^2 \\
+ \|\nabla \cdot \boldsymbol{\tau}\|_K^2 + \frac{1}{\epsilon}\|\boldsymbol{\tau}\|_K^2 \,,
\end{aligned} \tag{3.7}$$

and

$$\|(v,\boldsymbol{\tau})\|_{V,K}^2 \coloneqq \left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|_K^2 + \epsilon\left\|\nabla v\right\|_K^2 + \|v\|_K^2 + \left\|\nabla\cdot\boldsymbol{\tau} - \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|_K^2 + \frac{1}{\epsilon}\left\|\boldsymbol{\tau}\right\|_K^2 , \tag{3.8}$$

respectively designated the *robust* test norm and the *coupled robust* test norm, provide the necessary bound $\|v^*\|_V \lesssim \|u\|_{L^2(Q)}$.

Remark 3.3.2. *We haven't developed a mathematical theory for it, but we've also had numerical success with a norm that we've dubbed the NSDecoupled norm because we first stumbled on it during experiments with compressible Navier-Stokes:*

$$\|(v,\boldsymbol{\tau})\|_{V,K}^2 \coloneqq \left\|\boldsymbol{\beta}\cdot\nabla v + \frac{\partial v}{\partial t}\right\|_K^2 + \|\nabla v\|_K^2 + \|v\|_K^2 + \|\nabla\cdot\boldsymbol{\tau}\|_K^2 + \frac{1}{h^2}\left\|\boldsymbol{\tau}\right\|_K^2 .$$

We mention it because it appeared to be the most successful in simulations of the moving piston problem in 5.4.3.

In the following lemmas we establish the following bounds:

- Bound on $\|(v_1,\boldsymbol{\tau}_1)\|_V$. Lemma 3.3.2 gives $\left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\| \leq \|g\|$. Since $\nabla\cdot\boldsymbol{\tau}_1 = g + \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1$,

$$\|\nabla\cdot\boldsymbol{\tau}_1\| \leq \|g\| + \left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\| \leq 2\|g\| .$$

  Or, the fact that $\nabla\cdot\boldsymbol{\tau} - \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1 = g$ clearly gives

$$\left\|\nabla\cdot\boldsymbol{\tau} - \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\| = \|g\| .$$

Lemma 3.3.1 gives $\|v_1\|^2 + \epsilon \|\nabla v_1\|^2 \leq \|g\|^2$. Since $\epsilon^{1/2}\nabla v_1 = -\epsilon^{-1/2}\boldsymbol{\tau}_1$,

$$\frac{1}{\epsilon}\|\boldsymbol{\tau}_1\|^2 \leq \|g\|^2 .$$

Thus, all $(v_1, \boldsymbol{\tau}_1)$ terms in (3.7) and (3.8) are accounted for, guaranteeing at least robust control of u.

- Bound on $\|(v_2, \boldsymbol{\tau}_2)\|_V$. The fact that $\nabla \cdot \boldsymbol{\tau} - \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v = 0$ clearly gives

$$\left\| \nabla \cdot \boldsymbol{\tau} - \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_2 \right\| = 0 \leq \|\boldsymbol{f}\| .$$

Lemma 3.3.1 gives $\|v_2\|^2 + \epsilon \|\nabla v_2\|^2 \leq \epsilon \|\boldsymbol{f}\|^2$. Since $\epsilon^{1/2}\nabla v_2 = \boldsymbol{f} - \epsilon^{-1/2}\boldsymbol{\tau}_2$,

$$\frac{1}{\epsilon}\|\boldsymbol{\tau}_2\|^2 \leq (1+\epsilon)\|\boldsymbol{f}\|^2 .$$

We have not been able to develop bounds on $\left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_2\right\|$ and $\|\nabla \cdot \boldsymbol{\tau}\|$ which means that we can not guarantee robust control of $\boldsymbol{\sigma}$ with with provided test norms.

We proceed now with the technical estimates.

Lemma 3.3.1. *For the duration of this lemma, let $v := v_1 + v_2$. Assuming the advection field $\boldsymbol{\beta}$ is incompressible, i.e. $\nabla \cdot \boldsymbol{\beta} = 0$,*

$$\|v\|^2 + \epsilon \|\nabla v\|^2 \leq \|g\|^2 + \epsilon \|\boldsymbol{f}\|^2 .$$

Proof. Define $w = e^t v$ and note that $\frac{\partial w}{\partial t} = \left(\frac{\partial v}{\partial t} + v\right) e^t$ while all spatial derivatives go through. Multiplying the adjoint by w and integrating over Q gives

$$-\int_Q \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v w - \epsilon \Delta v w = \int_Q g w - \epsilon \int_Q \nabla \cdot \boldsymbol{f} w$$

or

$$-\int_Q e^t v\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v-\epsilon\int_Q e^t v\Delta v=\int_Q e^t gv-\epsilon\int_Q e^t v\nabla\cdot\boldsymbol{f}$$

Integrating by parts:

$$\begin{aligned}\int_Q \nabla_{xt}\cdot\left(e^t\tilde{\boldsymbol{\beta}}v\right)v-\int_\Gamma e^t\tilde{\boldsymbol{\beta}}\cdot\boldsymbol{n}v^2+\epsilon\int_Q e^t\nabla v\cdot\nabla v-\epsilon\int_{\Gamma_x}e^t v\cdot\nabla v\cdot\boldsymbol{n}_x\\ =\int_Q e^t gv+\epsilon\int_Q e^t\nabla v\cdot\boldsymbol{f}-\epsilon\int_{\Gamma_x}e^t v\boldsymbol{f}\cdot\boldsymbol{n}_x\end{aligned}$$

Note that $\nabla_{xt}\cdot e^t v\tilde{\boldsymbol{\beta}}=e^t(\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v+v)$ if $\nabla\cdot\boldsymbol{\beta}=0$. Moving some terms to the right hand side, we get

$$\begin{aligned}&\int_Q e^t v^2+\int_Q\epsilon e^t\nabla v\cdot\nabla v\\ &\qquad=\int_Q e^t gv+\epsilon\int_Q e^t\nabla v\cdot\boldsymbol{f}-\epsilon\int_{\Gamma_x}e^t v\boldsymbol{f}\cdot\boldsymbol{n}_x\\ &\qquad\quad-\int_Q e^t\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}vv\quad+\int_\Gamma e^t\tilde{\boldsymbol{\beta}}\cdot\boldsymbol{n}v^2\quad+\epsilon\int_{\Gamma_x}e^t v\cdot\nabla v\cdot\boldsymbol{n}_x\end{aligned}$$

Note that $1 \leq \|e^t\|_\infty = e^T$. Then

$$
\begin{aligned}
&\|v\|^2 + \epsilon \|\nabla v\|^2 \\
&\quad \leq e^T \left(\int_Q gv + \epsilon \int_Q \nabla v \cdot \boldsymbol{f} - \epsilon \int_{\Gamma_-} v \underbrace{\boldsymbol{f} \cdot \boldsymbol{n}_x}_{=\cancel{\tau_n}^{\,0} \frac{\partial v}{\partial \boldsymbol{n}_x}} - \epsilon \int_{\Gamma_+} \underbrace{v}_{=0} \boldsymbol{f} \cdot \boldsymbol{n}_x \right. \\
&\qquad \left. - \int_Q \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v v + \int_\Gamma \tilde{\boldsymbol{\beta}} \cdot \boldsymbol{n} v^2 + \epsilon \int_{\Gamma_-} v \cdot \nabla v \cdot \boldsymbol{n}_x + \epsilon \int_{\Gamma_+} \underbrace{v}_{=0} \frac{\partial v}{\partial \boldsymbol{n}_x} \right) \\
&\qquad \text{Note: boundary conditions give } \boldsymbol{\tau}_n = 0 \text{ on } \Gamma_- \text{ and } v = 0 \text{ on } \Gamma_+ \\
&\quad = e^T \left(\int_Q gv + \epsilon \int_Q \nabla v \cdot \boldsymbol{f} \cancel{- \epsilon \int_{\Gamma_-} v \frac{\partial v}{\partial \boldsymbol{n}_x} + \epsilon \int_{\Gamma_x} v \frac{\partial v}{\partial \boldsymbol{n}_x}} \right. \\
&\qquad \left. - \frac{1}{2} \int_Q \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v^2 + \int_\Gamma \tilde{\boldsymbol{\beta}} \cdot \boldsymbol{n} v^2 \right) \\
&\qquad \text{Note: } \Gamma_x = \Gamma_- \cup \Gamma_+ \text{ and } v = 0 \text{ on } \Gamma_- \\
&\quad = e^T \left(\int_Q gv + \epsilon \int_Q \nabla v \cdot \boldsymbol{f} + \frac{1}{2} \int_Q \cancel{\nabla_{xt} \cdot \tilde{\boldsymbol{\beta}} v^2}^{\,0} - \frac{1}{2} \int_\Gamma \tilde{\boldsymbol{\beta}} \cdot \boldsymbol{n} v^2 + \int_\Gamma \tilde{\boldsymbol{\beta}} \cdot \boldsymbol{n} v^2 \right) \\
&\qquad \text{Note: Integration by parts of } -\frac{1}{2} \int_Q \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v^2 \text{ and } \nabla \cdot \boldsymbol{\beta} = 0 \\
&\quad = e^T \left(\int_Q gv + \epsilon \int_Q \nabla v \cdot \boldsymbol{f} \right. \\
&\qquad \left. + \frac{1}{2} \left(\int_{\Gamma_0} \underbrace{-v^2}_{\leq 0} + \int_{\Gamma_T} \cancel{v^2}^{\,0} + \int_{\Gamma_-} \underbrace{\boldsymbol{\beta} \cdot \boldsymbol{n}_x v^2}_{\leq 0} + \int_{\Gamma_+} \boldsymbol{\beta} \cdot \boldsymbol{n}_x \cancel{v^2}^{\,0} \right) \right) \\
&\qquad \text{Note: Split boundary term into components, } v = 0 \text{ on } \Gamma_+ \text{ and } \Gamma_T \\
&\quad \leq e^T \left(\int_Q gv + \epsilon \int_Q \nabla v \cdot \boldsymbol{f} \right) \\
&\quad \leq e^T \left(\frac{\|g\|^2}{2} + \epsilon \frac{\|\boldsymbol{f}\|^2}{2} + \frac{\|v\|^2}{2} + \epsilon \frac{\|\nabla v\|^2}{2} \right) . \\
&\qquad \text{Note: Young's inequality}
\end{aligned}
$$

□

Lemma 3.3.2. *If* $\left\|\nabla\boldsymbol{\beta} - \frac{1}{2}\nabla\cdot\boldsymbol{\beta}\boldsymbol{I}\right\|_{L^\infty} \leq C_{\boldsymbol{\beta}}$ *we can bound*

$$\left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\| \lesssim \|g\|\,.$$

Proof. Multiply $-\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1 = g - \nabla\cdot\boldsymbol{\tau}_1$ by $-\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1$ and integrate over Q to get

$$\left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\|^2 = -\int_Q g\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1 + \int_Q \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla\cdot\boldsymbol{\tau}_1\,. \tag{3.9}$$

Note that

$$\frac{1}{\epsilon}\int_Q \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla\cdot\boldsymbol{\tau}_1 = -\int_Q \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla\cdot\nabla v_1$$

Note: $\boldsymbol{\tau}_1 = \epsilon\nabla v_1$

$$= -\int_{\Gamma_x}\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla v_1\cdot\boldsymbol{n}_x + \int_Q \nabla(\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1)\cdot\nabla v_1$$

Note: Integration by parts

$$= -\int_{\Gamma_x}\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla v_1\cdot\boldsymbol{n}_x + \int_Q (\nabla\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1)\cdot\nabla v_1 + \int_Q \tilde{\boldsymbol{\beta}}\cdot\nabla\nabla_{xt}v_1\cdot\nabla v_1$$

$$= -\int_{\Gamma_x}\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla v_1\cdot\boldsymbol{n}_x + \int_Q (\nabla\boldsymbol{\beta}\cdot\nabla v_1)\cdot\nabla v_1 + \frac{1}{2}\int_Q \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}(\nabla v_1\cdot\nabla v_1)$$

Note: $\nabla\nabla_{xt}v_1\cdot\nabla v_1 = \nabla_{xt}\nabla v_1\cdot\nabla v_1 = \frac{1}{2}\nabla_{xt}(\nabla v_1\cdot\nabla v_1)$

$$= -\int_{\Gamma_x}\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla v_1\cdot\boldsymbol{n}_x + \int_Q (\nabla\boldsymbol{\beta}\cdot\nabla v_1)\cdot\nabla v_1 + \frac{1}{2}\int_\Gamma \tilde{\boldsymbol{\beta}}\cdot\boldsymbol{n}(\nabla v_1\cdot\nabla v_1) - \frac{1}{2}\int_Q \nabla_{xt}\cdot\tilde{\boldsymbol{\beta}}(\nabla v_1\cdot\nabla v_1)$$

Note: Integration by parts

$$= -\int_{\Gamma_x}\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla v_1\cdot\boldsymbol{n}_x + \int_Q (\nabla\boldsymbol{\beta}\cdot\nabla v_1)\cdot\nabla v_1 + \frac{1}{2}\int_\Gamma \tilde{\boldsymbol{\beta}}\cdot\boldsymbol{n}(\nabla v_1\cdot\nabla v_1) - \frac{1}{2}\int_Q \nabla\cdot\boldsymbol{\beta}(\nabla v_1\cdot\nabla v_1)$$

Note: $\nabla_{xt}\cdot\tilde{\boldsymbol{\beta}} = \nabla\cdot\boldsymbol{\beta}$

$$= -\int_{\Gamma_x}\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\nabla v_1\cdot\boldsymbol{n}_x + \frac{1}{2}\int_\Gamma \tilde{\boldsymbol{\beta}}\cdot\boldsymbol{n}(\nabla v_1\cdot\nabla v_1) + \int_Q \nabla v_1(\nabla\boldsymbol{\beta} - \frac{1}{2}\nabla\cdot\boldsymbol{\beta}\boldsymbol{I})\nabla v_1\,.$$

Note: $(\nabla\boldsymbol{\beta}\cdot\nabla v_1)\cdot\nabla v_1 - \frac{1}{2}\nabla\cdot\boldsymbol{\beta}(\nabla v_1\cdot\nabla v_1) = \nabla v_1(\nabla\boldsymbol{\beta} - \frac{1}{2}\nabla\cdot\boldsymbol{\beta}\boldsymbol{I})\nabla v_1$

Plugging this into (3.9), we get

$$\begin{aligned}
\left\|\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1\right\|^2 &= -\int_Q g\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \epsilon \int_Q \nabla v_1 (\nabla \boldsymbol{\beta} - \frac{1}{2}\nabla \cdot \boldsymbol{\beta}\boldsymbol{I}) \nabla v_1 \\
&\quad - \epsilon \int_{\Gamma_x} \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 \nabla v_1 \cdot \boldsymbol{n}_x + \frac{\epsilon}{2} \int_\Gamma \tilde{\boldsymbol{\beta}} \cdot \boldsymbol{n} (\nabla v_1 \cdot \nabla v_1) \\
&= -\int_Q g\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \epsilon \int_Q \nabla v_1 (\nabla \boldsymbol{\beta} - \frac{1}{2}\nabla \cdot \boldsymbol{\beta}\boldsymbol{I}) \nabla v_1 \\
&\quad - \epsilon \int_{\Gamma_-} \tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 \underbrace{\nabla v_1 \cdot \boldsymbol{n}_x}_{=0} - \epsilon \int_{\Gamma_+} \left(\underbrace{\frac{\partial v_1}{\partial t}}_{=0} + \boldsymbol{\beta} \cdot \nabla v_1 \right) \nabla v_1 \cdot \boldsymbol{n}_x \\
&\qquad \text{Note: } \nabla v_1 \cdot \boldsymbol{n}_x = \tau_{1n} = 0 \text{ on } \Gamma_-,\ v_1 = 0 \text{ on } \Gamma_+ \\
&\quad + \frac{\epsilon}{2} \int_{\Gamma_-} \underbrace{\boldsymbol{\beta} \cdot \boldsymbol{n}_x}_{<0} (\nabla v_1 \cdot \nabla v_1) + \frac{\epsilon}{2} \int_{\Gamma_+} \boldsymbol{\beta} \cdot \boldsymbol{n}_x (\nabla v_1 \cdot \nabla v_1) \\
&\quad + \frac{\epsilon}{2} \int_{\Gamma_0} \underbrace{n_t}_{<0} (\nabla v_1 \cdot \nabla v_1) + \frac{\epsilon}{2} \int_{\Gamma_T} n_t \underbrace{(\nabla v_1 \cdot \nabla v_1)}_{=0} \\
&\qquad \text{Note: } v_1 = 0 \text{ on } \Gamma_T \\
&\leq -\int_Q g\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \epsilon \int_Q \nabla v_1 (\nabla \boldsymbol{\beta} - \frac{1}{2}\nabla \cdot \boldsymbol{\beta}\boldsymbol{I}) \nabla v_1 \\
&\quad + \epsilon \int_{\Gamma_+} \left(-\frac{\partial v_1}{\partial \boldsymbol{n}_x} \boldsymbol{\beta} + \frac{1}{2} \boldsymbol{\beta} \cdot \boldsymbol{n}_x \nabla v_1 \right) \cdot \nabla v_1 \\
&\qquad \text{Note: Dropped negative terms from RHS} \\
&= -\int_Q g\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \epsilon \int_Q \nabla v_1 (\nabla \boldsymbol{\beta} - \frac{1}{2}\nabla \cdot \boldsymbol{\beta}\boldsymbol{I}) \nabla v_1 \\
&\quad + \epsilon \int_{\Gamma_+} \left(-\frac{\partial v_1}{\partial \boldsymbol{n}_x} \boldsymbol{\beta} + \frac{1}{2} \boldsymbol{\beta} \cdot \boldsymbol{n}_x \frac{\partial v_1}{\partial \boldsymbol{n}_x} \boldsymbol{n}_x \right) \cdot \frac{\partial v_1}{\partial \boldsymbol{n}_x} \boldsymbol{n}_x \\
&\qquad \text{Note: } \nabla v_1 \cdot \nabla v_1 = \nabla v_1 \cdot \nabla v_1 \boldsymbol{n}_x \cdot \boldsymbol{n}_x = (\nabla v_1 \cdot \boldsymbol{n}_x \boldsymbol{n}_x) \cdot (\nabla v_1 \cdot \boldsymbol{n}_x \boldsymbol{n}_x) \\
&= -\int_Q g\tilde{\boldsymbol{\beta}} \cdot \nabla_{xt} v_1 + \epsilon \int_Q \nabla v_1 (\nabla \boldsymbol{\beta} - \frac{1}{2}\nabla \cdot \boldsymbol{\beta}\boldsymbol{I}) \nabla v_1 \,. \\
&\quad \underbrace{-\frac{\epsilon}{2} \int_{\Gamma_+} \left(\frac{\partial v_1}{\partial \boldsymbol{n}_x} \right)^2 \boldsymbol{\beta} \cdot \boldsymbol{n}_x}_{<0}
\end{aligned}$$

$$
\begin{aligned}
&\leq -\int_Q g\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1+\epsilon\int_Q\nabla v_1(\nabla\boldsymbol{\beta}-\frac{1}{2}\nabla\cdot\boldsymbol{\beta}\boldsymbol{I})\nabla v_1\\
&\leq \frac{\|g\|^2}{2}+\frac{\left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\|^2}{2}+\epsilon\int_Q\nabla v_1(\nabla\boldsymbol{\beta}-\frac{1}{2}\nabla\cdot\boldsymbol{\beta}\boldsymbol{I})\nabla v_1\\
&\quad\text{Note: Young's inequality}\\
&\leq \frac{\|g\|^2}{2}+\frac{\left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\|^2}{2}+\epsilon C_{\boldsymbol{\beta}}\left\|\nabla v_1\right\|^2\\
&\quad\text{Note: Assumption on }\boldsymbol{\beta}\\
&\leq \left(\frac{1}{2}+C_{\boldsymbol{\beta}}\right)\|g\|^2+\frac{\left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v_1\right\|^2}{2}\,.
\end{aligned}
$$

□

In conclusion, with either robust test norm, we can claim the following stability result,

$$
\begin{aligned}
\|u-u_h\| &\lesssim \|(u,\boldsymbol{\sigma},\hat{u},\hat{t})-(u_h,\boldsymbol{\sigma}_h,\hat{u}_h,\hat{t}_h)\|_E\\
&= \inf_{(u_h,\boldsymbol{\sigma}_h,\hat{u}_h,\hat{t}_h)}\|(u,\boldsymbol{\sigma},\hat{u},\hat{t})-(u_h,\boldsymbol{\sigma}_h,\hat{u}_h,\hat{t}_h)\|_E\,.
\end{aligned}
$$

Notice that, contrary to the steady-state case, we have not been able to secure a robust L^2 bound for the stress. The best approximation error in the energy norm can be estimated locally, i.e. element-wise, see [17, 36]. This leads to an ultimate, final h estimate but not necessarily with robust constants. The loss of robustness in the best approximation error estimate is the consequence of rescaling the L^2-terms to avoid boundary layers in the optimal test functions. However, similarly to the steady-state case, with refinements, the mesh-dependent L^2-terms converge to the optimal ones so we hope to regain

robustness in the limit. We do not attempt to analyze the best approximation error in this contribution and restrict ourselves to numerical experiments only.

3.4 Numerical Tests

The norms given in (3.7) and (3.8) are robust, but the reaction (0^{th} order) terms still dominate the diffusion terms which produces boundary layers in optimal test functions and prohibits their resolution with a simple enrichment strategy. We can mitigate this by introducing mesh-dependent norms:

$$\begin{aligned}\|(v,\boldsymbol{\tau})\|_{V,K}^2 \coloneqq & \left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|_K^2 + \epsilon\left\|\nabla v\right\|_K^2 + \min\left(\frac{\epsilon}{h^2},1\right)\|v\|_K^2 \\ & + \|\nabla\cdot\boldsymbol{\tau}\|_K^2 + \min\left(\frac{1}{\epsilon},\frac{1}{h^2}\right)\|\boldsymbol{\tau}\|_K^2 \ ,\end{aligned} \tag{3.10}$$

and

$$\begin{aligned}\|(v,\boldsymbol{\tau})\|_{V,K}^2 \coloneqq & \left\|\tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|_K^2 + \epsilon\left\|\nabla v\right\|_K^2 + \min\left(\frac{\epsilon}{h^2},1\right)\|v\|_K^2 \\ & + \left\|\nabla\cdot\boldsymbol{\tau} - \tilde{\boldsymbol{\beta}}\cdot\nabla_{xt}v\right\|_K^2 + \min\left(\frac{1}{\epsilon},\frac{1}{h^2}\right)\|\boldsymbol{\tau}\|_K^2 \ .\end{aligned} \tag{3.11}$$

Note that any version of (3.7) and (3.8) with smaller coefficients also satisfies the criteria for robustness. The mesh dependent coefficients were chosen in an attempt to balance the relative size of "reaction" terms like $\|v\|$ which scale like h^d with "diffusive" terms like $\epsilon\|\nabla v\|$ which scale like h^{d-2}. This is also the mechanism by which we avoid creating sharp boundary layers in our optimal test functions – by correctly balancing reactive and diffusive terms. In the following numerical experiments, we compute with these mesh dependent norms.

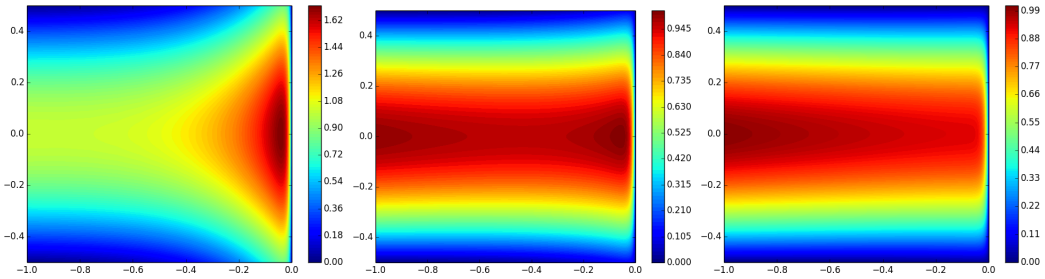
We verify robust convergence of our transient coupled robust norm on an analytical solution (shown in Figure 3.4) that decays to a steady state Eriksson-Johnson problem:

$$u = \exp(-lt)\left[\exp(\lambda_1 x) - \exp(\lambda_2 x)\right] + \cos(\pi y)\frac{\exp(s_1 x) - \exp(r_1 x)}{\exp(-s_1) - \exp(-r_1)},$$

where $l = 4$, $\lambda_{1,2} = \frac{-1 \pm \sqrt{1-4\epsilon l}}{-2\epsilon}$, $r_1 = \frac{1+\sqrt{1+4\pi^2\epsilon^2}}{2\epsilon}$, and $s_1 = \frac{1-\sqrt{1+4\pi^2\epsilon^2}}{2\epsilon}$. The problem domain is $[-1, 0] \times [-0.5, 0.5]$ and $\boldsymbol{\beta} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. We show robustness for $\epsilon = 10^{-2}$, 10^{-4}, 10^{-6}, 10^{-8} for linear, quadratic, and quartic polynomial trial functions. Flux boundary conditions were applied based on the exact solution at $x = -1$ and $t = 0$ while trace boundary conditions were set at $y = -0.5$, $y = 0.5$, and $x = 0$. An adaptive solve was undertaken using a greedy refinement strategy in which any elements with at least 20% of the energy error of highest energy error element were refined at each step. See [35] for details on adaptivity within the DPG context.

In the plot legends, L^2 indicates $\left(\|u - u_{\text{exact}}\|_L^2 + \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_{\text{exact}}\|_{L^2}\right)^{\frac{1}{2}}$ while V^* indicates the energy error reported by the method. Despite a lack of guaranteed control $\boldsymbol{\sigma}$ by norms (3.10) and (3.11), $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_{\text{exact}}\|_{L^2}$ is included in the L^2 error computation and does appear to be under control in the problems considered here. When plotted in isolation, the L^2 error in $\boldsymbol{\sigma}$ was usually orders of magnitude smaller than $\|u - u_{\text{exact}}\|_{L^2}$.

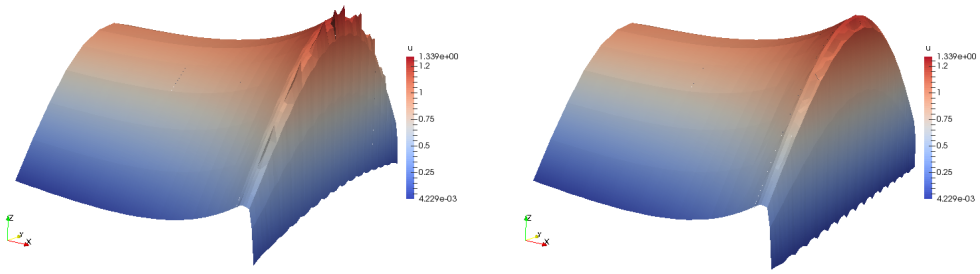
We provide surface plots of temporal slices of the solution at $t = 0.2$ for the two norms with $\epsilon = 10^{-2}$, and $p = 2$ after 4 adaptive refinements. The results conform to our previous experience with steady convection-diffusion



(a) $t = 0.0$ (b) $t = 0.5$ (c) $t = 1.0$

Figure 3.4: Transient Eriksson-Johnson solution

where the coupled robust norm tends to produce smoother results in regions with sharp gradients.

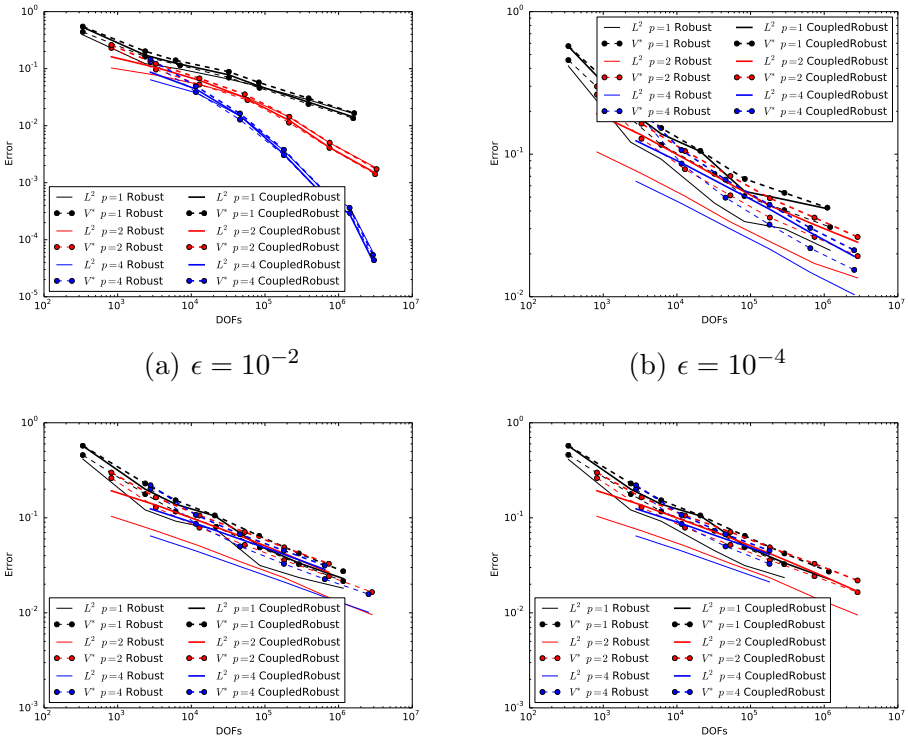


(a) Robust norm (b) Coupled robust norm

Figure 3.5: u at $t = 0.2$ for $\epsilon = 10^{-2}$ and $p = 2$ after 4 adaptive refinements

3.5 Summary

As expected, convergence of the energy error appears to be a reliable predictor of convergence of the L^2 error. This relation is especially tight for moderate values of ϵ. We've developed two robust test norms for transient



(a) $\epsilon = 10^{-2}$ (b) $\epsilon = 10^{-4}$

(c) $\epsilon = 10^{-6}$ (d) $\epsilon = 10^{-8}$

Figure 3.6: Convergence to analytical solution

convection-diffusion, though neither one guarantees robust control over $\boldsymbol{\sigma}$ as we had with their steady analogs.

Chapter 4

Space-Time DPG for Incompressible Navier-Stokes

DPG for steady incompressible Navier-Stokes was studied by Roberts in [65]. We choose a variational formulation more consistent with our work on transient convection-diffusion where the fluxes are related to the conservation law. The equations are trivial for one spatial dimension so space-time incompressible Navier-Stokes requires either 3D or 4D solves. We derive a space-time DPG formulation for spatially 2D Navier-Stokes and show preliminary convergence results for the Taylor-Green vortex problem.

4.1 Nonlinear Form

The 2D incompressible Navier-Stokes equations are:

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u} - \nu \nabla \boldsymbol{u} + p\mathbb{I}) &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0\,, \end{aligned}$$

where $\boldsymbol{u}$ is the velocity, p is the pressure, ν is the kinematic viscosity, and $\boldsymbol{f}$ contains any momentum source terms. As a first order system in space-time

divergence form, this is

$$\begin{aligned} \frac{1}{\nu}\mathbb{D} - \nabla \boldsymbol{u} &= 0 \\ \nabla_{xt} \cdot \begin{pmatrix} \boldsymbol{u} \otimes \boldsymbol{u} - \mathbb{D} + p\mathbb{I} \\ \boldsymbol{u} \end{pmatrix} &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0\,. \end{aligned}$$

Multiplying by test functions $\mathbb{S} \in \mathbb{H}(\mathrm{div}, Q)$, $\boldsymbol{v} \in \boldsymbol{H}^1_{xt}(Q)$, $q \in H^1(Q)$, and integrating by parts, we get

$$\begin{aligned} \left(\frac{1}{\nu}\mathbb{D}, \mathbb{S}\right) + (\boldsymbol{u}, \nabla \cdot \mathbb{S}) - \langle \hat{\boldsymbol{u}}, \mathbb{S} \cdot \boldsymbol{n}_x \rangle &= 0 \\ -\left(\begin{pmatrix} \boldsymbol{u} \otimes \boldsymbol{u} - \mathbb{D} + p\mathbb{I} \\ \boldsymbol{u} \end{pmatrix}, \nabla_{xt} \boldsymbol{v}\right) + \left\langle \hat{\boldsymbol{t}}, \boldsymbol{v} \right\rangle &= (\boldsymbol{f}, \boldsymbol{v}) \\ -(\boldsymbol{u}, \nabla q) + \langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, q \rangle &= 0\,, \end{aligned}$$

where $\mathbb{D} \in \mathbb{L}^2(Q)$, $\boldsymbol{u} \in \mathbf{L}^2(Q)$, $p \in L^2(Q)$, and

$$\begin{aligned} \hat{\boldsymbol{u}} &= \mathrm{tr}(\boldsymbol{u}) && \in \boldsymbol{H}^{1/2}(\Gamma_{h_x}) \\ \hat{\boldsymbol{t}} &= \mathrm{tr}\left(\boldsymbol{u} \otimes \boldsymbol{u} - \mathbb{D} + p\mathbb{I}\right) \cdot \boldsymbol{n}_x + \mathrm{tr}(\boldsymbol{u}) \cdot n_t && \in \boldsymbol{H}^{-1/2}_{xt}(\Gamma_h)\,. \end{aligned}$$

4.2 Linearization

We split our residual into volume and trace terms:

$$R(\boldsymbol{u}, p, \mathbb{D}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{t}}) = R(\boldsymbol{u}, p, \mathbb{D}) + R(\hat{\boldsymbol{u}}, \hat{\boldsymbol{t}})\,.$$

where

$$R(\boldsymbol{u}, p, \mathbb{D}) = \left(\frac{1}{\nu}\mathbb{D}, \mathbb{S}\right) + (\boldsymbol{u}, \nabla \cdot \mathbb{S}) - \left(\begin{pmatrix} \boldsymbol{u} \otimes \boldsymbol{u} - \mathbb{D} + p\mathbb{I} \\ \boldsymbol{u} \end{pmatrix}, \nabla_{xt}\boldsymbol{v}\right) - (\boldsymbol{f}, \boldsymbol{v}) - (\boldsymbol{u}, \nabla q)$$

and

$$R(\hat{\boldsymbol{u}}, \hat{\boldsymbol{t}}) = -\langle \hat{\boldsymbol{u}}, \mathbb{S} \cdot \boldsymbol{n} \rangle + \langle \hat{\boldsymbol{t}}, \boldsymbol{v} \rangle + \langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, q \rangle \ .$$

Note that $R(\hat{\boldsymbol{u}}, \hat{\boldsymbol{t}})$ is already linear, so we only need to linearize terms dependent on the volume variables. Let $\{\boldsymbol{u}, p, \mathbb{D}\} = \left\{\tilde{\boldsymbol{u}}, \tilde{p}, \tilde{\mathbb{D}}\right\} + \{\Delta\boldsymbol{u}, \Delta p, \Delta\mathbb{D}\}$, where $\{\tilde{\boldsymbol{u}}, \tilde{p}, \mathbb{D}\}$ is the previous solution in the Newton iteration and $\{\Delta\boldsymbol{u}, \Delta p, \mathbb{D}\}$ is the update. We linearize about $\left\{\tilde{\boldsymbol{u}}, \tilde{p}, \tilde{\mathbb{D}}\right\}$ so that our linear problem becomes

$$\frac{\partial R(\tilde{\boldsymbol{u}}, \tilde{p}, \tilde{\mathbb{D}})}{\partial(\boldsymbol{u}, p, \mathbb{D})} \begin{pmatrix} \Delta\boldsymbol{u} \\ \Delta p \\ \Delta\mathbb{D} \end{pmatrix} + R(\hat{\boldsymbol{u}}, \hat{\boldsymbol{t}}) = -R(\tilde{\boldsymbol{u}}, \tilde{p}, \tilde{\mathbb{D}})$$

where

$$\frac{\partial R(\tilde{\boldsymbol{u}}, \tilde{p}, \tilde{\mathbb{D}})}{\partial(\boldsymbol{u}, p, \mathbb{D})} \begin{pmatrix} \Delta\boldsymbol{u} \\ \Delta p \\ \Delta\mathbb{D} \end{pmatrix} = \left(\frac{1}{\nu}\Delta\mathbb{D}, \mathbb{S}\right) + (\Delta\boldsymbol{u}, \nabla \cdot \mathbb{S}) - \left(\begin{pmatrix} \Delta\boldsymbol{u} \otimes \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{u}} \otimes \Delta\boldsymbol{u} - \Delta\mathbb{D} + \Delta p\mathbb{I} \\ \Delta\boldsymbol{u} \end{pmatrix}, \nabla_{xt}\boldsymbol{v}\right) - (\Delta\boldsymbol{u}, \nabla q) \ .$$

Note that for the steady state case, the pressure is only uniquely defined up to a constant, so in order to obtain a unique solution it is sufficient to set

either a zero mean condition or to constrain the pressure to a certain value at a point. In the transient case, pressure is unique up to any arbitrary function of t. This issue disappears for problems with boundary conditions on $\hat{\boldsymbol{t}}$ as the definition of the flux contains a pressure term, but for problems with pure $\hat{\boldsymbol{u}}$ boundary conditions, we choose a spatial point and constrain the pressure to a specific value at that point for all time.

4.3 Robust Test Norms

We develop test norms for the incompressible Navier-Stokes equations by drawing analogies to our robust norms for transient convection-diffusion. If we group the test terms according to their interaction with trial variables, the left hand side of the convection-diffusion bilinear form looks like:

$$(\boldsymbol{\sigma}, \frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v) + (u, \nabla \cdot \boldsymbol{\tau} - \boldsymbol{\beta} \cdot \nabla v - \frac{\partial v}{\partial t}) \, .$$

Doing the same thing for incompressible Navier-Stokes yields:

$$\left(\Delta \mathbb{D}, \frac{1}{\nu}\mathbb{S} + \nabla \boldsymbol{v}\right) + \left(\Delta \boldsymbol{u}, \nabla \cdot \mathbb{S} - \nabla q - \left(\tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{v} + \tilde{\boldsymbol{u}} \cdot (\nabla \boldsymbol{v})^T + \frac{\partial \boldsymbol{v}}{\partial t}\right)\right) + (p, -\nabla \cdot \boldsymbol{v}) \, .$$

Recall the robust (3.10) and coupled robust (3.11) test norms:

$$\|(v, \boldsymbol{\tau})\|^2_{V,K} \coloneqq \left\| \boldsymbol{\beta} \cdot \nabla v + \frac{\partial v}{\partial t} \right\|^2_K + \epsilon \left\| \nabla v \right\|^2_K + \min\left(\frac{\epsilon}{h^2}, 1\right) \|v\|^2_K + \|\nabla \cdot \boldsymbol{\tau}\|^2_K + \min\left(\frac{1}{\epsilon}, \frac{1}{h^2}\right) \|\boldsymbol{\tau}\|^2_K \, ,$$

and

$$\|(v,\boldsymbol{\tau})\|_{V,K}^2 \coloneqq \left\|\boldsymbol{\beta}\cdot\nabla v + \frac{\partial v}{\partial t}\right\|_K^2 + \epsilon\|\nabla v\|_K^2 + \min\left(\frac{\epsilon}{h^2},1\right)\|v\|_K^2 + \left\|\nabla\cdot\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v - \frac{\partial v}{\partial t}\right\|_K^2 + \min\left(\frac{1}{\epsilon},\frac{1}{h^2}\right)\|\boldsymbol{\tau}\|_K^2 \ .$$

This leads us to define the respective norms for incompressible Navier-Stokes:

$$\begin{aligned}\|(\boldsymbol{v},\mathbb{D},q)\|_{V,K}^2 \coloneqq &\left\|\tilde{\boldsymbol{u}}\cdot\nabla\boldsymbol{v} + \tilde{\boldsymbol{u}}\cdot(\nabla\boldsymbol{v})^T + \frac{\partial\boldsymbol{v}}{\partial t}\right\|_K^2 + \nu\|\nabla\boldsymbol{v}\|_K^2 + \min\left(\frac{\nu}{h^2},1\right)\|\boldsymbol{v}\|_K^2 \\ &+ \|\nabla\cdot\mathbb{S} - \nabla q\|_K^2 \\ &+ \min\left(\frac{1}{\nu},\frac{1}{h^2}\right)\|\mathbb{S}\|_K^2 + \|\nabla\cdot\boldsymbol{v}\|_K^2 + \|q\|_K^2 \ ,\end{aligned}$$

and

$$\begin{aligned}\|(\boldsymbol{v},\mathbb{D},q)\|_{V,K}^2 \coloneqq &\left\|\tilde{\boldsymbol{u}}\cdot\nabla\boldsymbol{v} + \tilde{\boldsymbol{u}}\cdot(\nabla\boldsymbol{v})^T + \frac{\partial\boldsymbol{v}}{\partial t}\right\|_K^2 + \nu\|\nabla\boldsymbol{v}\|_K^2 + \min\left(\frac{\nu}{h^2},1\right)\|\boldsymbol{v}\|_K^2 \\ &+ \left\|\nabla\cdot\mathbb{S} - \nabla q - \left(\tilde{\boldsymbol{u}}\cdot\nabla\boldsymbol{v} + \tilde{\boldsymbol{u}}\cdot(\nabla\boldsymbol{v})^T + \frac{\partial\boldsymbol{v}}{\partial t}\right)\right\|_K^2 \\ &+ \min\left(\frac{1}{\nu},\frac{1}{h^2}\right)\|\mathbb{S}\|_K^2 + \|\nabla\cdot\boldsymbol{v}\|_K^2 + \|q\|_K^2 \ .\end{aligned}$$

4.4 Numerical Experiments

4.4.1 Taylor-Green Vortex

The problem domain is $[0,2\pi]\times[0,2\pi]$ with a final time of π and the analytical solution is

$$\boldsymbol{u} = e^{-\frac{2}{Re}t}\begin{pmatrix}\sin x\cos y\\ -\cos x\sin y\end{pmatrix} \ ,$$

a vector plot of which is shown in Figure 4.1. We apply spatial boundary conditions on $\hat{\boldsymbol{u}}$ and at $t=0$ we apply boundary conditions on $\hat{\boldsymbol{t}}$ according to

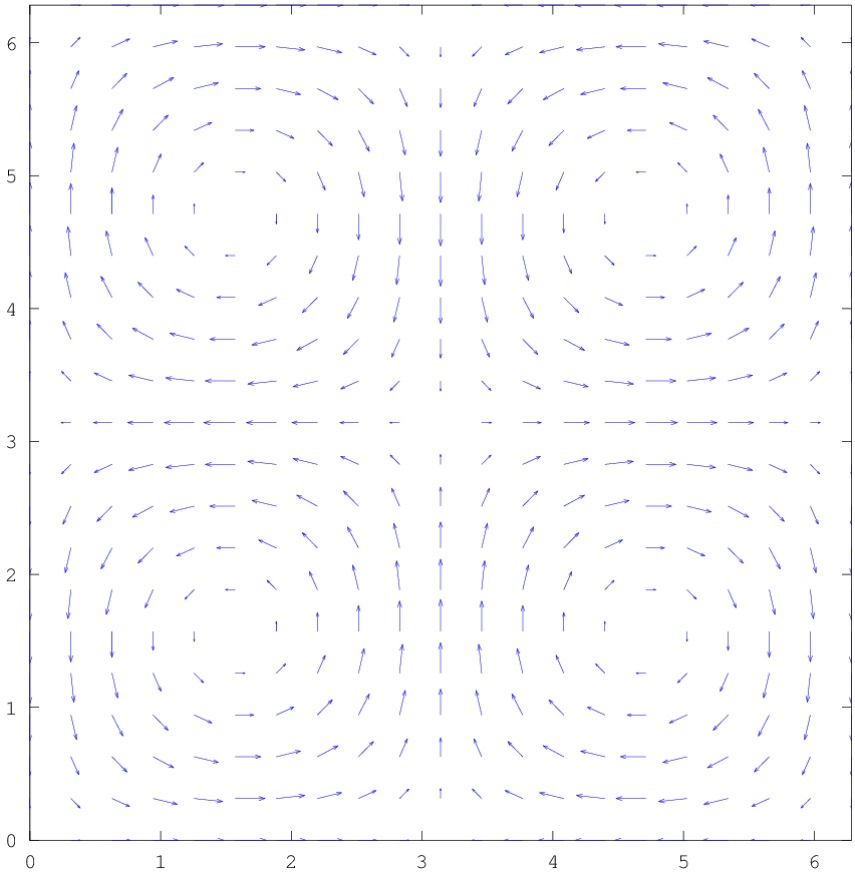


Figure 4.1: Taylor-Green vortex

the initial conditions. Plots of L^2 and energy (V^*) error for various polynomial orders and Reynolds numbers are shown in Figure 4.2 for the coupled robust norm. As expected from our results for convection-diffusion, energy error and L^2 error follow each other. An attempt at solving transient flow over a cylinder is outlined in Appendix D.1.2.1.

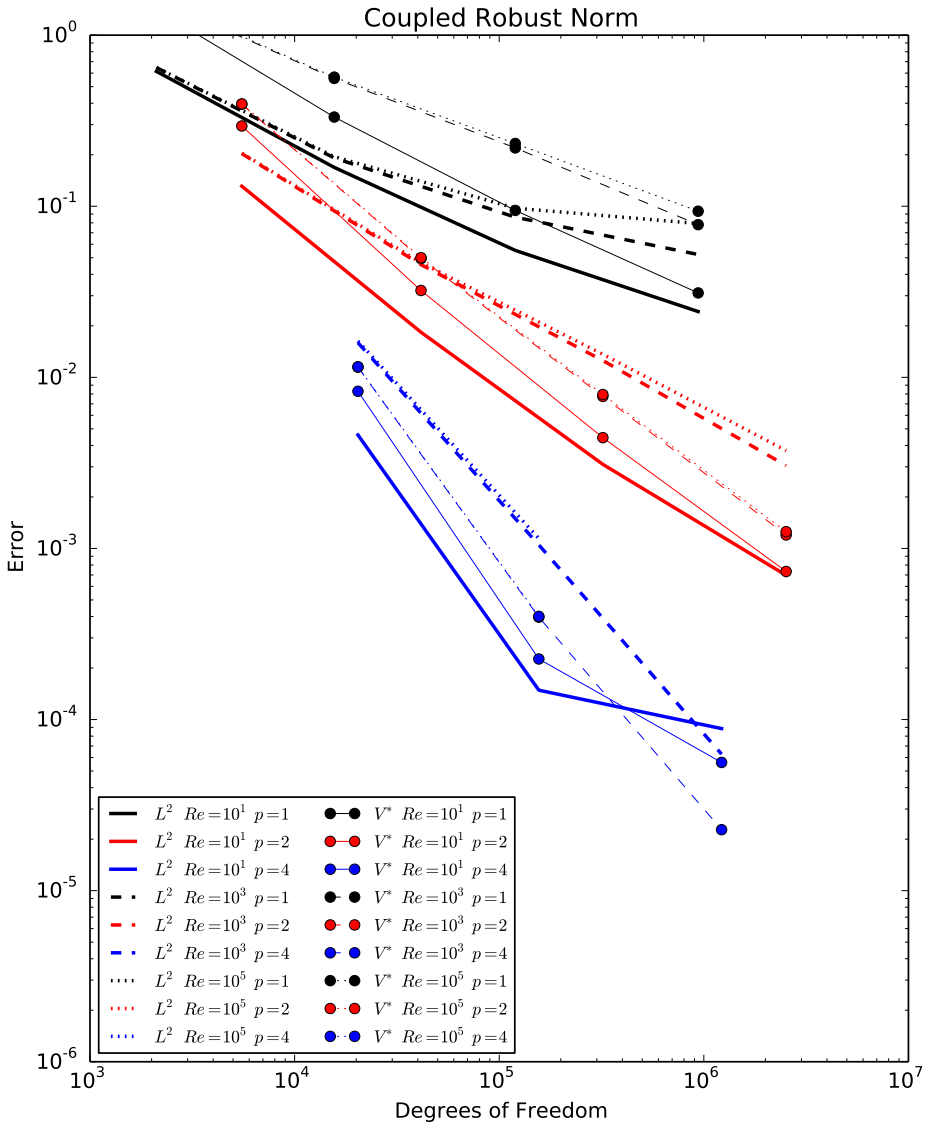


Figure 4.2: Convergence to Taylor-Green analytical solution

Chapter 5

Space-Time DPG for Compressible Navier-Stokes

DPG for steady compressible Navier-Stokes was studied by Jesse Chan in [18]. He observed that a pseudo-time stepping technique was necessary to get the Gauss-Newton solve to converge to a quality solution. This suggested that space-time approach which naturally includes the transient terms might achieve such results with a simpler Newton iteration.

We derive an ultra-weak space-time divergence formulation of the transient compressible Navier-Stokes equations, linearizing and developing robust test norms in a similar manner as was done in the previous chapter. We focus our numerical results on shock tube problems for which analytical solutions are known for the inviscid Euler equations. Despite the absence of any sophisticated shock capturing, we are able to resolve the shocks with adaptivity and produce some decent numerical results.

5.1 Nonlinear Form

The compressible Navier-Stokes equations are

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\boldsymbol{u}\\ \rho e_0\end{bmatrix}+\nabla\cdot\begin{bmatrix}\rho\boldsymbol{u}\\ \rho\boldsymbol{u}\otimes\boldsymbol{u}+p\mathbb{I}-\mathbb{T}\\ \rho\boldsymbol{u}e_0+\boldsymbol{u}p+\boldsymbol{q}-\boldsymbol{u}\cdot\mathbb{T}\end{bmatrix}=\begin{bmatrix}f_c\\ \boldsymbol{f}_m\\ f_e\end{bmatrix}, \tag{5.1}$$

where ρ is the density, $\boldsymbol{u}$ is the velocity, p is the pressure, $\mathbb{I}$ is the identity matrix, $\mathbb{T}$ is the deviatoric stress tensor or viscous stress, e_0 is the total energy, $\boldsymbol{q}$ is the heat flux, and f_c, $\boldsymbol{f}_m$, and f_e are the source terms for the continuity, momentum, and energy equations, respectively. Assuming Stokes hypothesis that $\lambda=-\frac{2}{3}\mu$,

$$\mathbb{T}=2\mu\mathbb{S}^*=2\mu\left[\frac{1}{2}\left(\nabla\boldsymbol{u}+(\nabla\boldsymbol{u})^T\right)-\frac{1}{3}\nabla\cdot\boldsymbol{u}\mathbb{I}\right],$$

where $\mathbb{S}^*$ is the trace-less viscous strain rate tensor. As we are using Navier-Stokes as a stand-in for the Euler equations, it is sufficient to use a constant μ rather than something more physical like Sutherland's formula. In order to work with standard finite element spaces, we introduce a new variable $\mathbb{D}=\mu\nabla\boldsymbol{u}$, so that $\mathbb{T}=\left(\mathbb{D}+\mathbb{D}^T-\frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right)$. The heat flux is given by Fourier's law:

$$\boldsymbol{q}=-C_p\frac{\mu}{Pr}\nabla T,$$

where C_p is the specific heat at constant pressure and Pr is the laminar Prandtl number: $Pr:=\frac{C_p\mu}{\lambda}$. We need to close these equations with an equation of state. An ideal gas assumption gives

$$\gamma:=\frac{C_p}{C_v},\quad p=\rho RT,\quad e=C_vT,\quad C_p-C_v=R,$$

where γ is the ratio of specific heats, C_v is the specific heat at constant volume, R is the gas constant, e is the internal energy, T is the temperature, and γ, C_p, C_v, and R are constant properties of the fluid. The total specific energy is defined by

$$e_0 = e + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\,.$$

We can write our first order system of equations in space-time as follows:

$$\frac{1}{\mu}\mathbb{D} - \nabla\boldsymbol{u} = 0 \tag{5.2a}$$

$$\frac{Pr}{C_p\mu}\boldsymbol{q} + \nabla T = 0 \tag{5.2b}$$

$$\nabla_{xt}\cdot\begin{pmatrix}\rho\boldsymbol{u}\\ \rho\end{pmatrix} = f_c \tag{5.2c}$$

$$\nabla_{xt}\cdot\begin{pmatrix}\rho\boldsymbol{u}\otimes\boldsymbol{u} + \rho RT\mathbb{I} - \left(\mathbb{D}+\mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right)\\ \rho\boldsymbol{u}\end{pmatrix} = \boldsymbol{f}_m \tag{5.2d}$$

$$\nabla_{xt}\cdot\begin{pmatrix}\rho\boldsymbol{u}\left(C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \boldsymbol{u}\rho RT + \boldsymbol{q} - \boldsymbol{u}\cdot\left(\mathbb{D}+\mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right)\\ \rho\left(C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right)\end{pmatrix} = f_e\,, \tag{5.2e}$$

where our solution variables are ρ, $\boldsymbol{u}$, T, $\mathbb{D}$, and $\boldsymbol{q}$, each in a scalar, vector, or tensor version of $L^2(Q)$.

We can simplify the following discussion by introducing the following

notation. The conserved quantities for each equation are:

$$\begin{aligned} C_c &:= \rho \\ \boldsymbol{C}_m &:= \rho \boldsymbol{u} \\ C_e &:= \rho(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u})\,, \end{aligned}$$

while the Euler fluxes are:

$$\begin{aligned} \boldsymbol{F}_c &:= \rho \boldsymbol{u} \\ \mathbb{F}_m &:= \rho \boldsymbol{u}\otimes\boldsymbol{u} + \rho R T \mathbb{I} \\ \boldsymbol{F}_e &:= \rho \boldsymbol{u}\left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \boldsymbol{u}\rho R T\,, \end{aligned}$$

and the viscous fluxes are:

$$\begin{aligned} \boldsymbol{K}_c &:= \boldsymbol{0} \\ \mathbb{K}_m &:= \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\ \boldsymbol{K}_e &:= -\boldsymbol{q} + \boldsymbol{u}\cdot\left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right)\,. \end{aligned}$$

The constitutive terms are:

$$\begin{aligned} \mathbb{M}_{\mathbb{D}} &:= \mathbb{D} \\ \boldsymbol{M}_{\boldsymbol{q}} &:= \frac{Pr}{C_p}\boldsymbol{q}\,, \end{aligned}$$

and the constitutive relations are:

$$\begin{aligned} \boldsymbol{G}_{\mathbb{D}} &:= \boldsymbol{u} \\ G_{\boldsymbol{q}} &:= -T\,. \end{aligned}$$

Multiplying (5.2) by test functions $\mathbb{S} \in \mathbb{H}(\mathrm{div}, Q)$, $\boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, Q)$, $v_c \in H^1_{xt}(Q)$, $\boldsymbol{v}_m \in \mathbf{H}^1_{xt}(Q)$, $v_e \in H^1_{xt}(Q)$ and integrating by parts, we get

$$\left(\frac{1}{\mu}\mathbb{M}_{\mathbb{D}}, \mathbb{S}\right) + (\boldsymbol{G}_{\mathbb{D}}, \nabla \cdot \mathbb{S}) - \langle \hat{\boldsymbol{u}}, \mathbb{S}\boldsymbol{n}_x \rangle = 0 \tag{5.3a}$$

$$\left(\frac{1}{\mu}\boldsymbol{M}_{\boldsymbol{q}}, \boldsymbol{\tau}\right) + (G_{\boldsymbol{q}}, \nabla \cdot \boldsymbol{\tau}) + \left\langle \hat{T}, \tau_n \right\rangle = 0 \tag{5.3b}$$

$$-\left(\begin{pmatrix} \boldsymbol{F}_c - \mathbb{K}_c \\ C_c \end{pmatrix}, \nabla_{xt} v_c\right) + \langle \hat{t}_c, v_c \rangle = (f_c, v_c) \tag{5.3c}$$

$$-\left(\begin{pmatrix} \mathbb{F}_m - \mathbb{K}_m \\ \boldsymbol{C}_m \end{pmatrix}, \nabla_{xt} \boldsymbol{v}_m\right) + \langle \hat{\boldsymbol{t}}_m, \boldsymbol{v}_m \rangle = (\boldsymbol{f}_m, \boldsymbol{v}_m) \tag{5.3d}$$

$$-\left(\begin{pmatrix} \boldsymbol{F}_e - \boldsymbol{K}_e \\ C_e \end{pmatrix}, \nabla_{xt} v_e\right) + \langle \hat{t}_e, v_e \rangle = (f_e, v_e)\ , \tag{5.3e}$$

where

$$\begin{aligned}
\hat{\boldsymbol{u}} &= \mathrm{tr}(\boldsymbol{u}) && \in \boldsymbol{H}^{1/2}(\Gamma_{h_x}) \\
\hat{T} &= \mathrm{tr}(T) && \in H^{1/2}(\Gamma_{h_x}) \\
\hat{t}_c &= \mathrm{tr}\left(\boldsymbol{F}_c - \boldsymbol{K}_c\right) \cdot \boldsymbol{n}_x + \mathrm{tr}\left(C_c\right) n_t && \in H^{-1/2}_{xt}(\Gamma_h) \\
\hat{\boldsymbol{t}}_m &= \mathrm{tr}\left(\mathbb{F}_m - \mathbb{K}_m\right) \cdot \boldsymbol{n}_x + \mathrm{tr}\left(\boldsymbol{C}_m\right) n_t && \in \boldsymbol{H}^{-1/2}_{xt}(\Gamma_h) \\
\hat{t}_e &= \mathrm{tr}\left(\boldsymbol{F}_e - \boldsymbol{K}_e\right) \cdot \boldsymbol{n}_x + \mathrm{tr}\left(C_e\right) n_t && \in H^{-1/2}_{xt}(\Gamma_h)\,.
\end{aligned}$$

We can further simplify this by introducing group terms and group variables:

$$
\begin{aligned}
C &:= \{C_c\,,\, \boldsymbol{C}_m\,,\, C_e\} \\
F &:= \{\boldsymbol{F}_c\,,\, \mathbb{F}_m\,,\, \boldsymbol{F}_e\} \\
K &:= \{\boldsymbol{K}_c\,,\, \mathbb{K}_m\,,\, \boldsymbol{K}_e\} \\
M &:= \{\mathbb{M}_{\mathbb{D}}\,,\, \boldsymbol{M}_{\boldsymbol{q}}\} \\
G &:= \{\boldsymbol{G}_{\mathbb{D}}\,,\, G_{\boldsymbol{q}}\} \\
f &:= \{f_c\,,\, \boldsymbol{f}_m\,,\, f_e\} \\
W &:= \{\rho\,,\, \boldsymbol{u}\,,\, T\} \\
\hat{W} &:= \left\{\hat{\boldsymbol{u}}\,,\, -\hat{T}\right\} \\
\Sigma &:= \{\mathbb{D}, \boldsymbol{q}\} \\
\hat{t} &:= \left\{\hat{t}_e\,,\, \hat{\boldsymbol{t}}_m,\,,\, \hat{t}_e\right\} \\
\Psi &:= \{\mathbb{S}\,,\, \boldsymbol{\tau}\} \\
V &:= \{v_c\,,\, \boldsymbol{v}_m,\,,\, v_e\}\ .
\end{aligned}
$$

Our final nonlinear variational formulation looks very similar to what we had for convection-diffusion:

$$
\begin{aligned}
&\left(\frac{1}{\mu}M, \Psi\right) + (G, \nabla\cdot\Psi) - \left\langle \hat{W}, \Psi\cdot\boldsymbol{n}_x\right\rangle = 0 \\
&\quad - \left(\begin{pmatrix} F-K \\ C \end{pmatrix}, \nabla_{xt} V\right) + \left\langle \hat{t}, V\right\rangle = (f, V)\ .
\end{aligned}
$$

With appropriate change of variables, we could use this same form to consider a solution in terms of either conservation variables or entropy variables, a topic we briefly consider in Appendix B.

5.2 Linearization

We again begin by splitting our residual into trace and volume terms:

$$R(W,\Psi,\hat{W},\hat{t}) = R(W,\Psi) + R(\hat{W},\hat{t})\,,$$

where

$$R(W,\Psi) = \left(\frac{1}{\mu}M,\Psi\right) + (G,\nabla\cdot\Psi) - \left(\begin{pmatrix} F-K \\ C \end{pmatrix},\nabla_{xt}V\right) - (f,V)\ ,$$

and

$$R(\hat{W},\hat{t}) = -\left\langle \hat{W},\Psi\cdot\boldsymbol{n}_x\right\rangle + \left\langle \hat{t},V\right\rangle\ .$$

Again $R(\hat{W},\hat{t})$ is already linear, so we only need to linearize terms dependent on W. Let $\{W,\Psi\} = \{\tilde{W},\tilde{\Psi}\} + \{\Delta W,\Delta\Psi\}$, where $\{\tilde{W},\tilde{\Psi}\}$ is the previous solution in a Newton iteration and $\{\Delta W,\Delta\Psi\}$ is the update. We linearize about $\{\tilde{W},\tilde{\Psi}\}$ so that our linear problem becomes

$$\frac{\partial R(\tilde{W},\tilde{\Psi})}{\partial\{W,\Psi\}}\begin{pmatrix}\Delta W\\ \Delta\Psi\end{pmatrix} + R(\hat{W},\hat{t}) = -R(\tilde{W},\tilde{\Psi})\,,$$

with unknowns ΔW, $\Delta\Psi$, $\hat{W}$, and $\hat{t}$. The full definitions for these linearized terms can be found in Appendix B.

5.3 Robust Test Norms

The adjoint equations are:

$$\frac{1}{\mu}M^*(\Psi) + K^*(\nabla V) = \begin{pmatrix}\frac{1}{\mu}\mathbb{M}^*_{\mathbb{D}}(\mathbb{S})\\ \frac{1}{\mu}\boldsymbol{M}^*_{\boldsymbol{q}}(\boldsymbol{\tau})\end{pmatrix} + \begin{pmatrix}\mathbb{K}^*_{\mathbb{D}}(\nabla V)\\ \boldsymbol{K}^*_{\boldsymbol{q}}(\nabla V)\end{pmatrix}$$

$$-\begin{pmatrix}F^*\\ C^*\end{pmatrix}(\nabla_{xt}V) + G^*(\nabla\Psi) = -\begin{pmatrix}F^*_c(\nabla V) + C^*_c(V_{,t})\\ \boldsymbol{F}^*_m(\nabla V) + \boldsymbol{C}^*_m(V_{,t})\\ F^*_e(\nabla V) + C^*_e(V_{,t})\end{pmatrix} + \begin{pmatrix}G^*_c(\nabla\Psi)\\ \boldsymbol{G}^*_m(\nabla\Psi)\\ G^*_e(\nabla\Psi)\end{pmatrix}\ ,$$

where these terms can be developed by analyzing the bilinear form and grouping terms according to trial variable:

$$
\begin{aligned}
M^*_{\mathbb{D}}\mathbb{S} &= \mathbb{S} \\
M^*_{\boldsymbol{q}}\boldsymbol{\tau} &= \frac{Pr}{C_p}\boldsymbol{\tau} \\
K^*_{\mathbb{D}}\nabla V &= \nabla \boldsymbol{v}_m + (\nabla \boldsymbol{v}_m)^T - \frac{2}{3}\nabla \cdot \boldsymbol{v}_m \mathbb{I} \\
&\quad + \tilde{\boldsymbol{u}} \otimes \nabla v_e + (\tilde{\boldsymbol{u}} \otimes \nabla v_e)^T - \frac{2}{3}\tilde{\boldsymbol{u}} \cdot \nabla v_e \mathbb{I} \\
K^*_{\boldsymbol{q}}\nabla V &= -\nabla v_e \\
\boldsymbol{F}^*_c \cdot \nabla V &= \tilde{\boldsymbol{u}} \cdot \nabla v_c + \tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} : \nabla \boldsymbol{v}_m + R\tilde{T}\nabla \cdot \boldsymbol{v}_m + C_v\tilde{T}\tilde{\boldsymbol{u}} \cdot \nabla v_e \\
&\quad + \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} \cdot \nabla v_e + R\tilde{T}\tilde{\boldsymbol{u}} \cdot \nabla v_e \\
\boldsymbol{C}^*_c \cdot V_{,t} &= v_{c,t} + \tilde{\boldsymbol{u}} \cdot \boldsymbol{v}_{m,t} + (C_v\tilde{T} + \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}})v_{e,t} \\
\boldsymbol{F}^*_m \cdot \nabla \boldsymbol{v}_m &= \tilde{\rho}\nabla v_c + (\nabla \boldsymbol{v}_m + (\nabla \boldsymbol{v}_m)^T)\tilde{\rho}\tilde{\boldsymbol{u}} + C_v\tilde{T}\tilde{\rho}\nabla v_e \\
&\quad + \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\nabla v_e + \tilde{\rho}\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} \cdot \nabla v_e + R\tilde{T}\tilde{\rho}\nabla v_e \\
&\quad - \tilde{\mathbb{D}}\nabla v_e - (\tilde{\mathbb{D}})^T\nabla v_e + \frac{2}{3}\operatorname{tr}(\tilde{\mathbb{D}})\nabla v_e \\
\boldsymbol{C}^*_m \cdot V_{,t} &= \tilde{\rho}\boldsymbol{v}_{m,t} + \tilde{\rho}\tilde{\boldsymbol{u}}v_{e,t} \\
\boldsymbol{F}^*_e \cdot \nabla V &= R\tilde{\rho}\nabla \cdot \boldsymbol{v}_m + C_v\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \nabla v_e + R\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \nabla v_e \\
\boldsymbol{C}^*_e \cdot V_{,t} &= C_v\tilde{\rho}v_{e,t} \\
\boldsymbol{G}^*_c\nabla\Psi &= 0 \\
\boldsymbol{G}^*_m\nabla\Psi &= \nabla \cdot \mathbb{S} \\
\boldsymbol{G}^*_e\nabla\Psi &= -\nabla \cdot \boldsymbol{\tau}\,.
\end{aligned}
$$

We develop the analogous robust norm:

$$\|(V,\Psi)\|_{V,K}^2 \coloneqq \|F^* + C^*\|_K^2 + \mu \|K^*\|_K^2 + \min\left(\frac{\mu}{h^2}, 1\right) \|V\|_K^2 + \|G^*\|_K^2 + \min\left(\frac{1}{\mu}, \frac{1}{h^2}\right) \|M^*\|_K^2 \ ,$$

coupled robust norm:

$$\|(V,\Psi)\|_{V,K}^2 \coloneqq \|F^* + C^*\|_K^2 + \mu \|K^*\|_K^2 + \min\left(\frac{\mu}{h^2}, 1\right) \|V\|_K^2 + \|G^* - F^* - C^*\|_K^2 + \min\left(\frac{1}{\mu}, \frac{1}{h^2}\right) \|M^*\|_K^2 \ ,$$

and NSDecoupled norm:

$$\|(V,\Psi)\|_{V,K}^2 \coloneqq \|F^* + C^*\|_K^2 + \|K^*\|_K^2 + \|V\|_K^2 + \|G^*\|_K^2 + \frac{1}{h^2} \|M^*\|_K^2 \ .$$

5.4 Numerical Experiments

We consider three 1D test problems as verification[1]. The Sod shock tube, Noh implosion, and piston problem all have analytical solutions derived based on an inviscid flow assumption (Euler's equations). However, in the absence of viscosity, Euler's equations can have multiple solutions and most

[1]We attempted the 2D analog of the Noh problem and decay to steady state of supersonic flow over a flat plate but our naive shock capturing strategy did not work very well with these 3D space-time solves. For 2D Noh, the Newton iterations immediately took the density negative. Attempts to correct this by scaling back the Newton update to enforce positivity of density only resulted in a nonconvergent Newton iteration. Carrying on with negative density eventually caused the iterations to diverge. Initial flat plate results were slightly more encouraging, but we ran into serious scaling issues explored in Appendix D and were unable to sufficiently resolve any solution features to obtain publishable results.

numerical methods introduce a certain amount of artificial viscosity in order to select a unique solution. Such schemes usually require the artificial viscosity to scale in some sense with mesh size so that they can effectively handle shocks. We run our simulations without any artificial viscosity, but in order to get a well-posed problem, we do introduce a small amount of physical viscosity. We apply a continuation in viscosity trick in order to achieve cleaner refinement patterns, setting

$$\mu = \max\left(\mu_{\text{fine}},\, \min\left(\mu_{\text{coarse}},\, \frac{1}{2^{r+k}}\right)\right) ,$$

where μ_{fine} is the final viscosity we want, μ_{coarse} is the desired viscosity on coarse meshes, r is the refinement number and k is a problem dependent parameter that determines how rapidly μ ramps down to μ_{fine}. Essentially we are just simulating low viscosity Navier-Stokes as a stand-in for the unsolvable pure Euler equations.

5.4.1 Sod Shock Tube

The Sod shock tube problem was developed by Gary Sod in 1978[71], and has proven to be a popular problem for verification of compressible Navier-Stokes and Euler solvers. It serves to verify that a numerical method can effectively handle a rarefaction wave, material discontinuity, and shock wave all in one domain. The domain of interest is a shock tube of length 1 with a material interface in the middle. The material on the left has initial conditions of $(\rho_L, p_L, u_L) = (1, 1, 0)$ while the material on the right has $(\rho_R, p_R, u_R) = (0.125, 0.1, 0)$; both materials have $\gamma = 1.4$. At $t = 0$ the interface between

the materials is broken, and shock wave propagates into the right material, while a rarefaction wave moves left. The analytical solution is self-similar, but it is common to take $t = 0.2$ as a final time. At this time the shock wave and rarefaction waves have not hit the boundaries, so it is sufficient to set boundary conditions corresponding to the initial conditions. In our case, we set $\hat{t}_c = \hat{t}_e = 0$ on the left and right boundaries, $\hat{t}_m = -\rho_L R T_L$ on the left, and $\hat{t}_m = \rho_R R T_R$ on the right, while the fluxes are set equal to the discontinuous initial conditions on the $t = 0$ boundary. No boundary condition is required on the $t = 0.2$ boundary since the equations are hyperbolic in time. We solve this with $p = 2$, $\Delta p = 2$, and one continuous time slab starting with only 4 space-time elements. It is possible to solve this problem by setting $\mu_{\text{fine}} = \mu_{\text{coarse}} = 10^{-4}$, but we get cleaner refinement patterns by setting $\mu_{\text{coarse}} = 100$ and $k = 4$.

The results are plotted in Figures 5.1 - 5.3 for three different refinement levels: the initial coarse mesh, 6 adaptive refinements, and 12 refinements. The coarsest mesh is obviously not sufficient to resolve the features of the flow, but it is at least somewhat representative of the exact solution. We see significant overshoots and undershoots as we start to pick up on the shock, but these die away as we resolve to the viscous length scale. The contact discontinuity is never fully resolved because the energy error never registers strongly enough to drive further refinement. We predict that once the shock is sufficiently resolved, this would be the next priority for the refinement strategy.

5.4.2 Noh Implosion

The Noh implosion problem[58] is another standard test for Euler solvers. The initial conditions are of an ideal gas with $\gamma = 5/3$, zero pressure, uniform initial density of 1, and uniform velocity toward the center of the domain. An infinitely strong shock propagates outward at a speed of 1/3. For 1D flow, the post shock density jumps to 4. The domain is $[-1, 0] \times [0, 1]$. We apply boundary conditions $\hat{t}_c = \hat{t}_m = -1$, $\hat{t}_e = 0$ on the left boundary, symmetry conditions $\hat{u} = \hat{t}_c = \hat{t}_e = 0$ on the right boundary, and flux conditions on $t = 0$ according to the initial conditions. We solve with $p = 1$, $\Delta p = 2$, $\mu_{\text{fine}} = 10^3$, $\mu_{\text{coarse}} = 10$, and $k = 0$. The continuation in viscosity strategy makes a significant difference keeping the refinement pattern clean on this problem. If we jump straight to the final viscosity, we get a lot of spurious shock behavior on coarse meshes which eventually go away, but leave a lot of unnecessary refinements.

The results for the initial mesh, an intermediate mesh, and the final mesh are plotted in Figure 5.4. We see an unnecessary refinement pattern that appears in the 10th refinement mesh. We hypothesize that this might be related to poor resolution of the error representation function in these parts of the domain. One notable feature of the final solution is that we don't see a drop in the density near the symmetry boundary. This phenomena is known as wall heating and, though unphysical, appears to be nearly universal in simulations of this problem. We don't perfectly match the solution, there are some wiggles at the shock front that could be resolved better, but the fact

that we don't see any noticeable wall heating is significant.

5.4.3 Piston Problem

In the piston problem, we have a compressible gas with $\gamma = 5/3$ initially at rest with zero pressure. At $t = 0$, the left wall of the domain (initially $[0, 1]$) starts moving inward at a velocity of 1. This triggers a shock which precedes the moving piston and collides with the stationary right wall at $t = 0.8$. The initial density is 1, but jumps to 4 after the first shock, and 10 after the second. By the final time of $t = 0.85$ the second shock has traversed half the remaining distance from the right wall to the piston. The symmetry conditions from the Noh problem are applied on the right wall. The left boundary has normal $n_{xt} = (-\sqrt{2}, \sqrt{2})$ which means that fluxes at our disposal are:

$$\begin{aligned}
\hat{t}_c &= \sqrt{2}(-\rho u + \rho) \\
\hat{t}_m &= \sqrt{2}(-\rho u^2 - \rho RT + \rho u) \\
\hat{t}_e &= \sqrt{2}(-\rho u(C_v T + \frac{1}{2}u^2) - u\rho RT + \rho(C_v T + \frac{1}{2}u^2)) \,,
\end{aligned}$$

and since $u = 1$ at the left wall

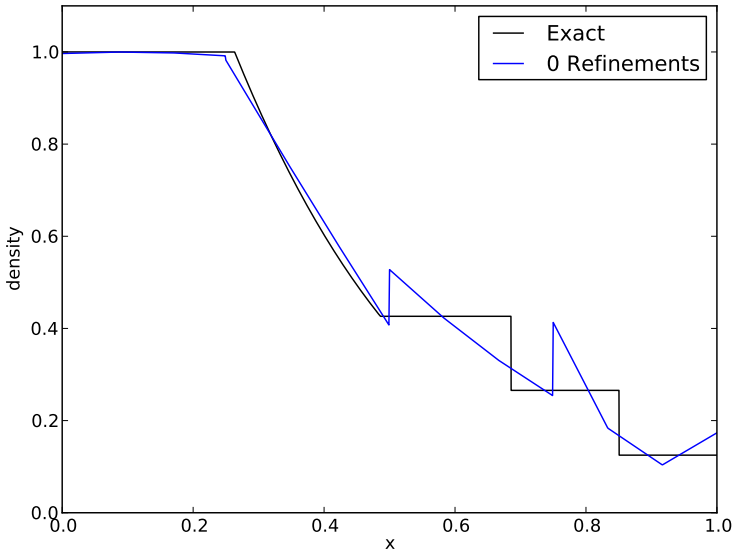
$$\begin{aligned}
\hat{t}_c &= 0 \\
\hat{t}_m &= -\sqrt{2}\rho RT \\
\hat{t}_e &= -\sqrt{2}\rho RT \,.
\end{aligned}$$

Therefore we set the following boundary conditions at the left boundary: $\hat{u} = 1$, $\hat{t}_c = 0$, and $\hat{t}_m - \hat{t}_e = 0$ implemented as a penalty condition. We

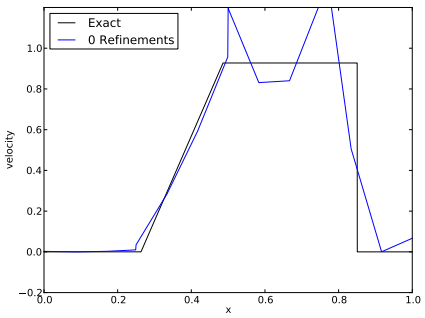
solve using $p = 2$, $\Delta p = 2$, a fixed $\mu = 100$, and an initial 4×4 space-time mesh. Unfortunately, the robust and coupled robust norms did not produce the cleanest solutions on this problem, and we were forced to use the NS-Decoupled norm which has less mathematical justification but seems to work very well on shock problems. Final results and mesh are shown in Figure 5.5 and Figure 5.6.

5.5 Summary of Compressible Results

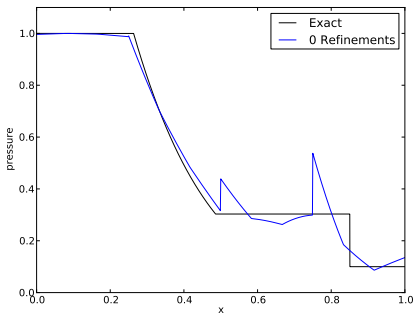
The chief strength of the DPG method is in its stability and adaptivity properties. It makes no claims of being a robust technique for handling shocks and in fact we ran into a lot of shock related difficulties in arriving at these solutions. The continuation in viscosity strategy, though avoidable, was an attempt at mitigating these challenges. What is notable is that we were able to initialize each simulation from very coarse meshes and adaptively resolve the solution features.



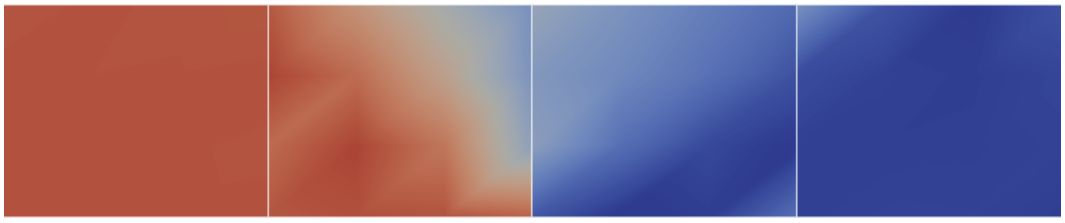
(a) Density



(b) Velocity

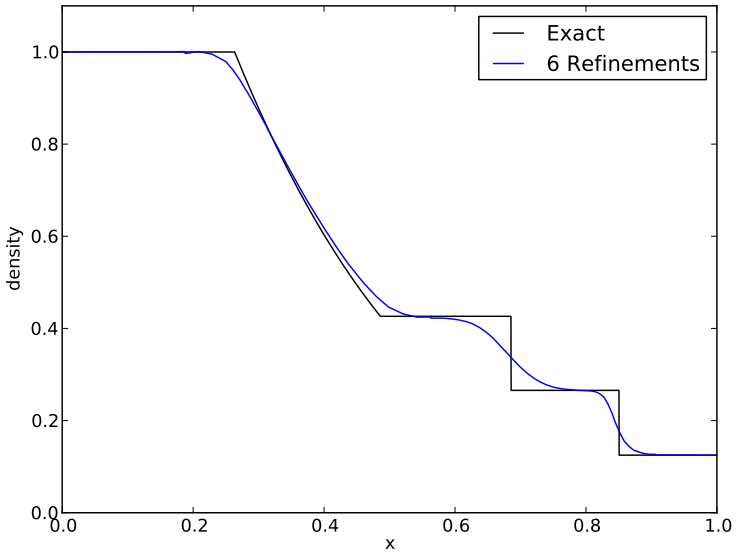


(c) Pressure

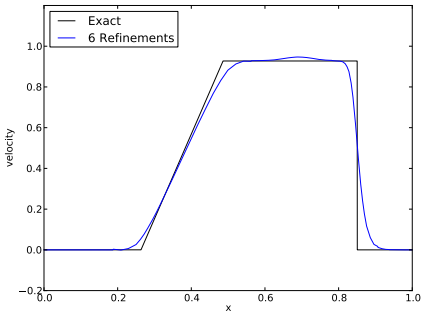


(d) Mesh

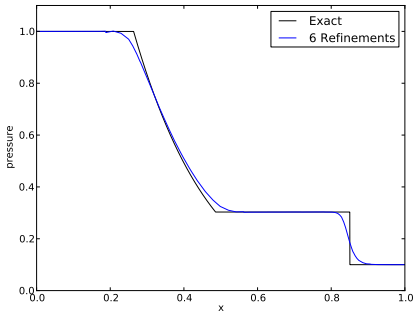
Figure 5.1: Sod solution with robust norm, initial mesh



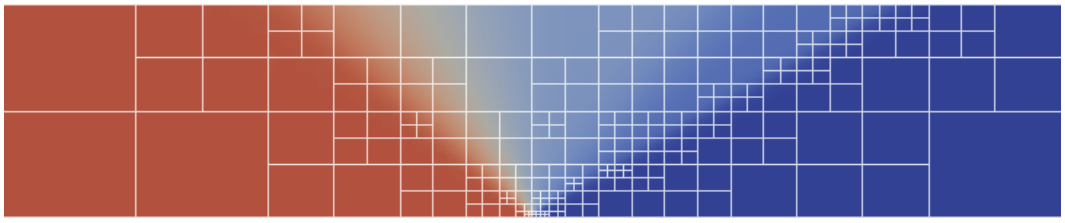
(a) Density



(b) Velocity

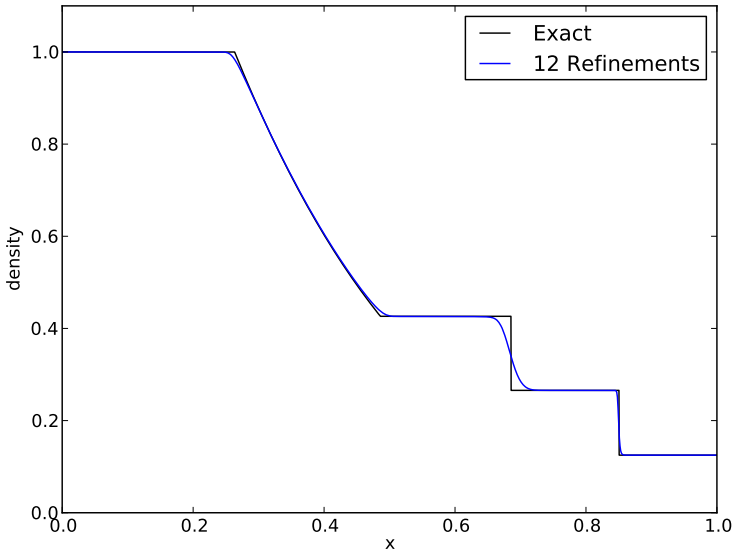


(c) Pressure

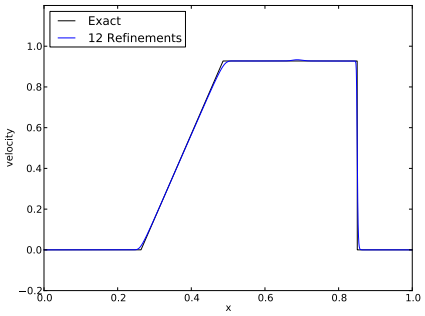


(d) Mesh

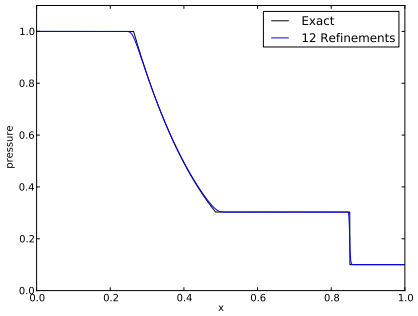
Figure 5.2: Sod solution with robust norm, 6th refinement



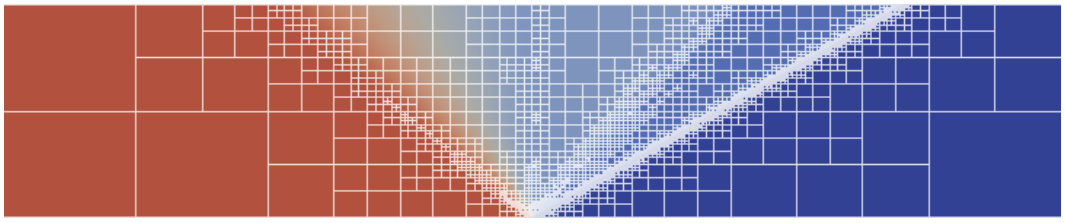
(a) Density



(b) Velocity

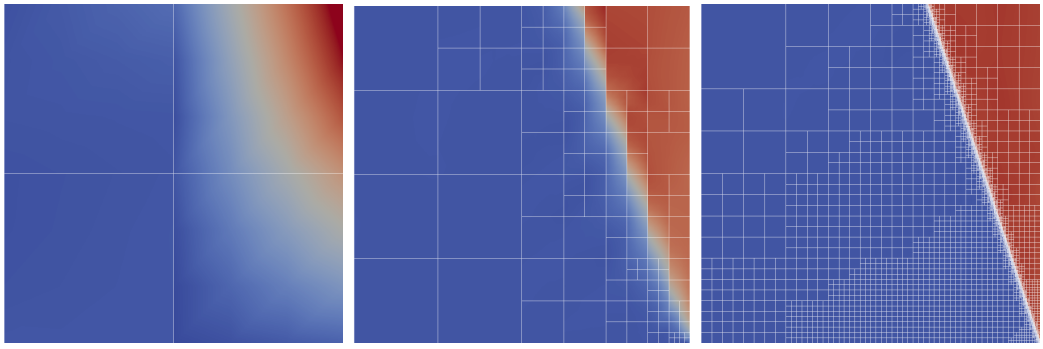


(c) Pressure

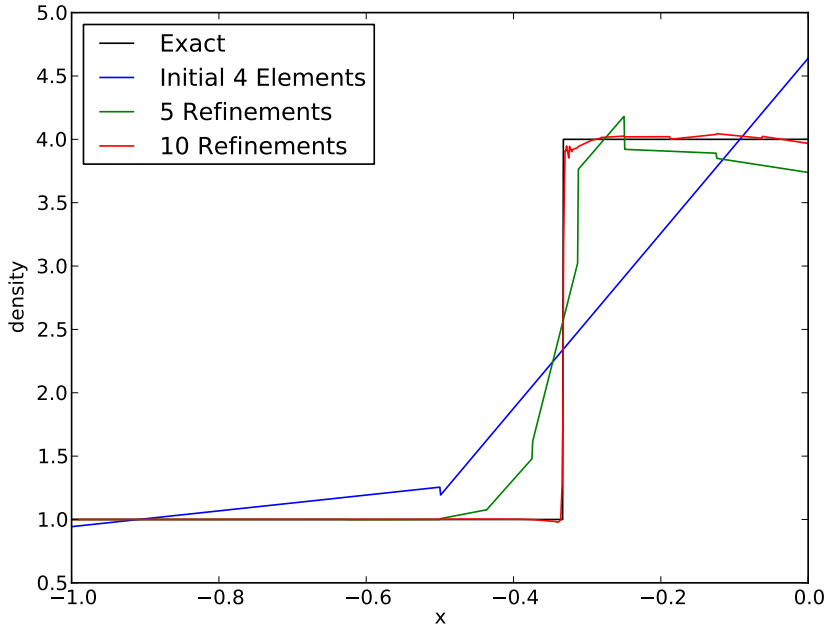


(d) Mesh

Figure 5.3: Sod solution with robust norm, 12th refinement

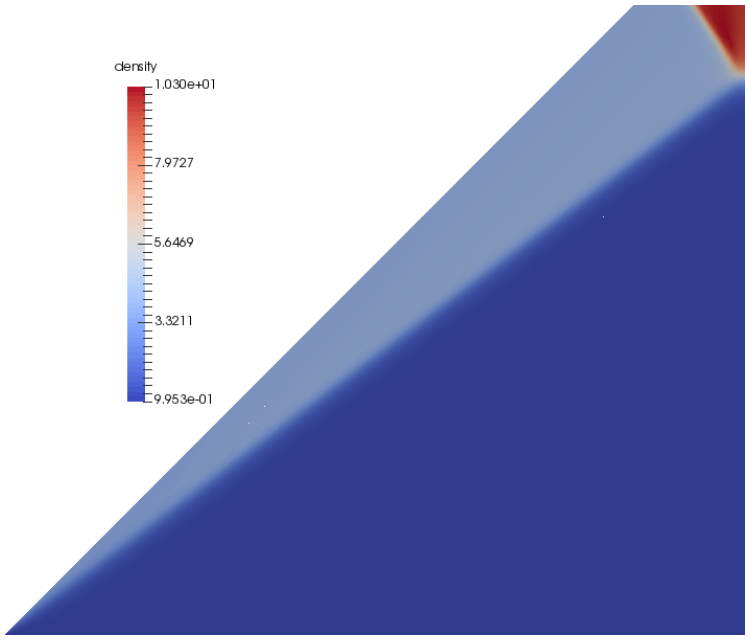


(a) Initial Mesh (b) After 5 refinements (c) After 10 refinements

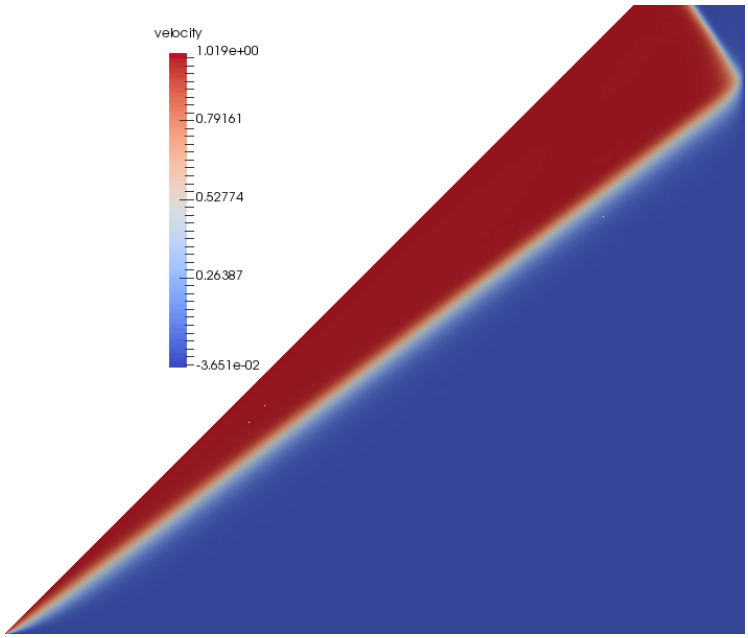


(d) Density at final time

Figure 5.4: Noh solution with robust norm



(a) Density



(b) Velocity

Figure 5.5: Piston solution with NSDecoupled norm after 8 adaptive refinements

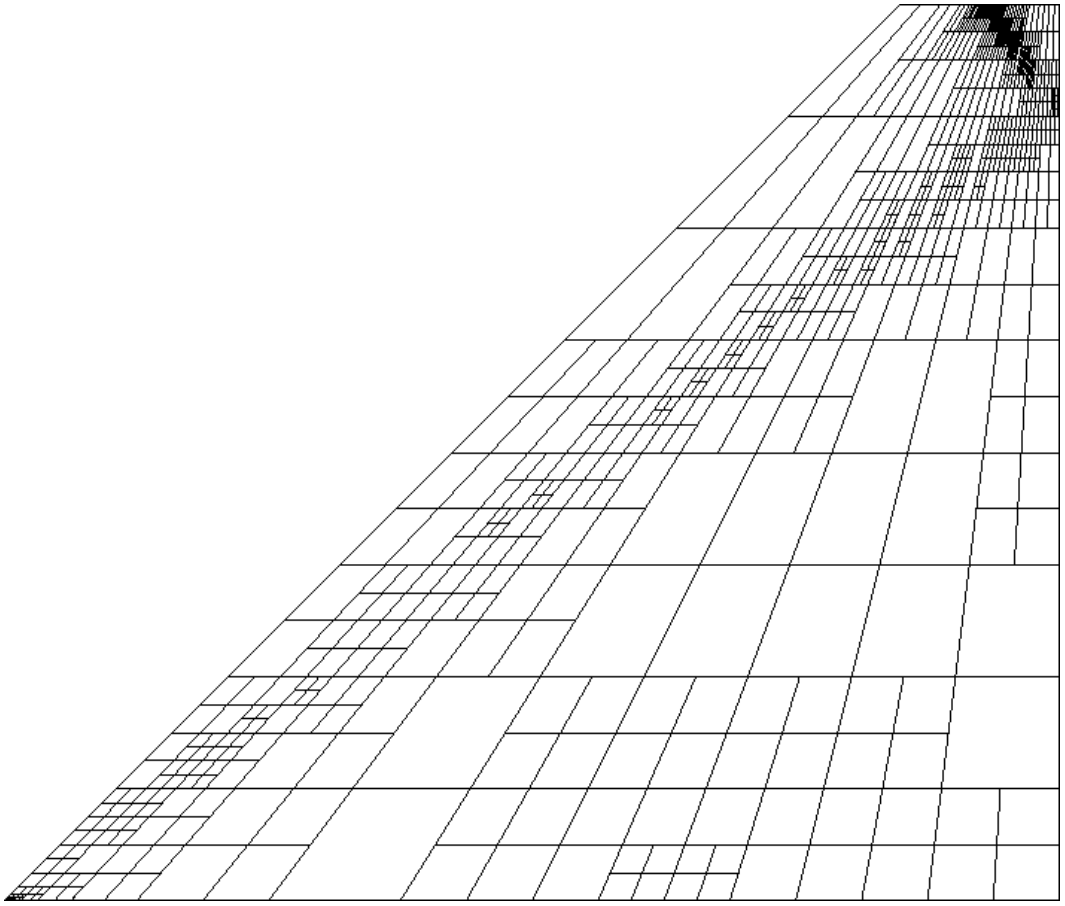


Figure 5.6: Piston mesh with NSDecoupled norm after 8 adaptive refinements

Chapter 6

Conclusions and Future Directions

The goal of this work has been to develop a proof of concept for a space-time discontinuous Petrov-Galerkin finite element method with applications to fluid flow applications. Chapter 1 provided motivations for applying DPG to transient fluid problems and explored some of the alternatives in the field. Local conservation is an important property to computational fluid dynamics practitioners. In Chapter 2 we developed a variant of DPG that is locally conservative through the addition of Lagrange multipliers to the system. This locally conservative DPG method was proved to be stable and robust and shown to dramatically improve coarse mesh numerical results on several test problems.

In Chapter 3 we develop a theory for space-time DPG applied to convection-diffusion type problems. We use an ultra-weak formulation where the conservation equation is placed in a space-time divergence form. This allows us to define physically meaningful fluxes related to conservation principles and eases the transition to Navier-Stokes. We propose new test norms for space-time convection-diffusion and prove that they provide near optimal convergence of the primary variable. Numerical results confirm the theory,

showing that the energy norm in which the solution is optimal robustly bounds the L^2 norm.

Chapter 4 and Chapter 5 develop space-time DPG methods for transient incompressible and compressible Navier-Stokes by drawing analogies to transient convection-diffusion. This includes the analogous ultra-weak formulations in space-time divergence form and robust test norms. Numerical verifications of the theory show the expected behavior.

Several side projects are explored in the appendices. An implicit Runge-Kutta time stepping strategy for DPG is described in Appendix A and shown to converge at the expected rates. We developed space-time DPG implementations of compressible Navier-Stokes under three popular variable transformations in Appendix B. Physically meaningful test norms for compressible Navier-Stokes inspired by entropy were then proposed in Appendix C, though numerical experiments seemed to prefer the standard non-entropy scaled test norms.

6.1 Accomplishments

On the theoretical side, I have developed and proven robustness of both locally conservative and space-time discontinuous Petrov-Galerkin finite element methods for convection-diffusion problems. This included the development of robust test norms for both of these formulations. I also used the concept of entropy to derive new test norms for compressible Navier-Stokes such that the residual is minimized in a physically consistent way.

On the numerical and computational side, I confirmed numerically the robustness of my test norms for locally conservative and space-time DPG. I also demonstrated convergence of space-time DPG for incompressible Navier-Stokes and obtained various shock tube results for compressible Navier-Stokes. I implemented space-time DPG for various variable transformations of the compressible Navier-Stokes equations and compared the numerical results. Within the primitive variable formulation, I implemented entropy scaled test norms and compared to the standard test norms inspired by convection-diffusion. I also implemented an ESDIRK (explicit first step singly diagonal implicit Runge-Kutta) time stepping strategy for DPG. Finally, I've been an active contributor to the parallel *hp*-adaptive DPG code base Camellia[66] from which all of these results were generated.

This dissertation includes applications of both locally conservative and space-time DPG to problems in convection-diffusion, Burgers' equation, Stokes flow, incompressible Navier-Stokes, and compressible Navier-Stokes. Of particular note are simulations of Stokes flow over a cylinder and a backward facing step, incompressible Navier-Stokes simulations of Taylor-Green vortices, and several shock tube simulations of compressible Navier-Stokes including a problem with a moving boundary.

6.2 Future Work

This work was really a proof of concept and much work remains in order to make this a competitive numerical method for transient fluid flow

problems.

6.2.1 Improve Scaling

The most pressing issue before pursuing further work on space-time DPG is to improve the scaling of our global solve. Past explorations of DPG were primarily focused on two dimensional solves. In space-time this two spatial dimensions requires a full 3D solve. Much to our chagrin after working to implement a 3D adaptive code, we discovered that our global solvers did not scale nearly as well as we expected on these higher dimensional problems. We further explore this issue and some possible solutions in Appendix D.

6.2.2 Shock Capturing

The strength of DPG lie in its stability and adaptivity properties. Shock capturing for the Euler and compressible Navier-Stokes equations has more to do with limiting Gibbs phenomenon of overshoots and undershoots around shocks on meshes coarser than the viscous length scale. As such, if we were serious about applying DPG to shock problems, we would want to augment it with some sort of shock capturing strategy, preferably a consistent one that reduces to the original equations in the limit as we fully resolve solution features.

Another possible solution that we've begun exploring is the development of DPG for non-Hilbert L^p Banach spaces. Gibbs phenomenon is well known to be less pronounced in L^1 spaces than the L^2 spaces at the foun-

dation of most finite element theory. The downside to this approach is that any finite element theory built around Hilbert spaces is no longer applicable and previously linear problems like convection-diffusion become nonlinear in non-Hilbert spaces.

6.2.3 More Extensive 2D Results

With the implementation of the previous two topics, we open the door to many more interesting 2D transient problems. The issue of scaling prevented us from producing meaningful results for unsteady incompressible flow over a cylinder as we originally planned. This would also allow us to consider classical problems like vortex shedding off of an oscillating airfoil. It would be worthwhile to see if our lack of wall heating on the 1D Noh problem carries over to the 2D case as well. In the current state of things, undershoots around shocks cause the density to dip negative which causes the equations to be ill posed for the next Newton iterate. If we perform a line search on the Newton update to keep density positive, the line search drops below 10^{-6}, effectively stalling the Newton iteration. We believe that shock capturing could regularize the solution and allow us to converge to a solution.

6.2.4 Anisotropic Refinements

Anisotropic refinements in space-time are a necessary first step in order to make time slabs a more attractive option, a point that is illustrated in Appendix D. Jesse Chan developed an anisotropic refinement strategy for 2D

computations in Camellia, but this process gets significantly more difficult in 3D or higher space-time meshes.

6.2.5 3D Results

We've implemented space-time as a tensor product of a spatial mesh and a temporal line. In theory this means that 3D space-time shouldn't be significantly more complicated to implement, but we expect the costs to blow up even more than they did from 2D to 3D, as we would now be performing 4D global solves. Additionally, the mesh partitioning libraries we leverage to distribute elements across processors are not set up to handle 4D meshes. The pursuit of 3D problems would force us to fundamentally rethink how we implement space-time DPG.

Appendices

Appendix A

Implicit Time Stepping with DPG

The proposed research into space-time DPG does not imply that DPG is incompatible with other time integration techniques. We did spend some time exploring popular alternatives such as some ESDIRK (explicit first step singly diagonal implicit Runge-Kutta) methods before we ultimately concluded that a space-time formulation might more naturally fit with our adaptive techniques. In this chapter, we briefly outline some of our exploratory work on implicit time integrators with DPG.

There are two different ways of coupling a spatial solver and a temporal solver. The *method of lines* first discretizes the spatial variables, which converts the original initial-boundary-value problem into a system of ordinary differential equations (ODEs) which are then discretized in time. It is unclear whether this approach is possible for DPG since the semi-discrete residual is not well defined and DPG is a minimum residual method. The alternative, sometimes called the *method of discretization in time* or *Rothe's method* reverses the order of discretization. The first temporal discretization converts the problem into a sequence of boundary-value (-like) problems. In this case, it is possible to build a DPG method since it is much clearer how to define a

residual. It is worth noting that spatial and temporal discretization in general do not commute[70].

Finally, there is the choice between explicit and implicit time-stepping versions of the method of discretization in time. We wish to solve the system

$$\frac{\partial U}{\partial t} + f(U) = 0\,.$$

It is not immediately clear how one could perform explicit time-stepping with DPG since an explicit system has $f(U)$ on the right hand side, but the DPG traces and fluxes are included in the $f(U)$ term and thus need to be solved for. So moving forward, we focus on implicit techniques which also have superior stability properties.

A.1 Backward Euler

The simplest implicit time stepping method would be backward Euler, for which we get the following system to solve at each time step n:

$$\frac{U^n}{\Delta t} + f(U^n) = \frac{U^{n-1}}{\Delta t}\,, \tag{A.1}$$

where U^{n-1} is known data from the previous time step, and Δt is the time step. In general, $f(U^n)$ could be nonlinear, in which case we define a residual

$$R(U^n) = \frac{U^n}{\Delta t} + f(U^n) - \frac{U^{n-1}}{\Delta t}\,. \tag{A.2}$$

Given an approximate solution $\tilde{U}^n$, we wish to solve for an increment ΔU such that $U^n = \tilde{U}^n + \Delta U$ is a better approximation of the true solution.

Approximating $R(U^n) = 0$ by $R(\tilde{U}^n) + R'(\tilde{U}^n)\Delta U = 0$, where $R'(\tilde{U}^n)$ is the Jacobian of R at $\tilde{U}^n$, we obtain a linear equation

$$\frac{\Delta U}{\Delta t} + f'(\tilde{U})\Delta U = \frac{U_n}{\Delta t} - \frac{\tilde{U}}{\Delta t} - f(\tilde{U})\,. \tag{A.3}$$

Note that $f(\tilde{U})$ only contains terms that had to be linearized. In general, we do not need to linearize our flux and trace terms in DPG, and hence those terms are excluded from $f(\tilde{U})$.

A.2 ESDIRK

After a literature search, ESDIRK time stepping schemes were identified as a potentially attractive high order time integration technique to couple with DPG. From an implementation point of view, ESDIRK schemes are much simpler to implement than full implicit Runge-Kutta schemes since each stage may be computed in sequence rather than as a fully coupled system. This cuts down on the number of unknowns to keep track of, reducing memory requirements. The "explicit first stage" is completely trivial, requiring no work at all. This reduces a formally s-stage scheme to $s - 1$ stages of actual computational work. Finally, the final stage coincides with the desired value at the nth time step, eliminating the need to have a final reconstruction step. A 6

stage ESDIRK algorithm has the following Butcher tableau:

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & a_{10} & a_{11} & 0 & 0 & 0 & 0 \\ c_2 & a_{20} & a_{21} & a_{22} & 0 & 0 & 0 \\ c_3 & a_{30} & a_{31} & a_{32} & a_{33} & 0 & 0 \\ c_4 & a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & 0 \\ c_5 & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\ \hline & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{array} .$$

From a stability point of view, ESDIRK schemes provide both A-stability and L-stability. The more classical backwards differentiation formula are not A-stable above second order. ESDIRK schemes enforce a "stiffly accurate" assumption that $a_{sj} = b_j$ which makes the solution at the next time step U^n independent of any explicit process within the integration step. There is also precedence for using ESDIRK schemes with fluid dynamics simulations (see [7], where ESDIRK schemes were found to be more efficient than BDF schemes for laminar flow over a cylinder).

A.2.1 ESDIRK with DPG

For an s stage ESDIRK scheme, we solve a series of equations for $k = 0, \cdots, s-1$

$$\frac{U^k}{a_{kk}\Delta t} + f(U^k) = \frac{U_n}{a_{kk}\Delta t} - \sum_{j=0}^{k-1} \frac{a_{kj}}{a_{kk}} f(U^j) .$$

From the first equation we see that $U^0 = U_n$. And we have that $U_{n+1} = U^s$. For a nonlinear system, define residual

$$R(U^k) = \frac{U^k}{a_{kk}\Delta t} + f(U^k) - \frac{U_n}{a_{kk}\Delta t} + \sum_{j=0}^{k-1} \frac{a_{kj}}{a_{kk}} f(U^j)$$

Utilizing the same linearization as above, we arrive at our linearized system

$$\frac{\Delta U}{a_{kk}\Delta t} + f'(\tilde{U}^k)\Delta U = \frac{U_n}{a_{kk}\Delta t} - \frac{\tilde{U}^k}{a_{kk}\Delta t} - f(\tilde{U}^k) - \sum_{j=0}^{k-1} \frac{a_{kj}}{a_{kk}} f(U^j)\,, \qquad \text{(A.4)}$$

which is to be solved iteratively at each stage until $R(\tilde{U}^k)$ is smaller than some tolerance. Note that contrary to the $f(\tilde{U})$ term which comes from the linearization and excludes flux and trace terms, $f(U^j)$ will need to keep the flux and trace terms from the DPG bilinear form. It is worth noting that terms necessary to construct $f(U^0)$ might not available from the initial condition because they include traces and fluxes. It is certainly possible to initialize the fluxes and traces for the initial condition, but it is not quite as convenient as setting the field variables. Thus in the following numerical experiment, we kick start the simulation with a backward Euler solve on a time step one thousandth the size of requested time step before switching fully to the ESDIRK scheme.

A.2.2 Case Study: 2D Burgers' Equation

We consider the 2D Burger's equations and accompanying problem outlined in [76]. The 2D Burgers' equations are:

$$\begin{aligned} \frac{\partial u_1}{\partial t} + u_1\frac{\partial u_1}{\partial x} + u_2\frac{\partial u_1}{\partial y} - \frac{1}{R}\Delta u_1 = 0 \\ \frac{\partial u_2}{\partial t} + u_1\frac{\partial u_2}{\partial x} + u_2\frac{\partial u_2}{\partial y} - \frac{1}{R}\Delta u_2 = 0\,, \end{aligned} \qquad \text{(A.5)}$$

where R is the effective Reynolds number.

A.2.2.1 DPG Formulation

As a first order system, this is

$$\begin{aligned} R\boldsymbol{\sigma}_1 - \nabla u_1 &= 0 \\ R\boldsymbol{\sigma}_2 - \nabla u_2 &= 0 \\ \frac{\partial u_1}{\partial t} + R\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_1 - \nabla \cdot \boldsymbol{\sigma}_1 &= 0 \\ \frac{\partial u_2}{\partial t} + R\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_2 - \nabla \cdot \boldsymbol{\sigma}_2 &= 0\,. \end{aligned} \tag{A.6}$$

Multiplying by test functions $\boldsymbol{\tau}_1$, $\boldsymbol{\tau}_2$, v_1, v_2, and integrating by parts:

$$\begin{aligned} (R\boldsymbol{\sigma}_1, \boldsymbol{\tau}_1) + (u_1, \nabla \cdot \boldsymbol{\tau}_1) - \langle \hat{u}_1, \tau_{1n} \rangle &= 0 \\ (R\boldsymbol{\sigma}_2, \boldsymbol{\tau}_2) + (u_2, \nabla \cdot \boldsymbol{\tau}_2) - \langle \hat{u}_2, \tau_{2n} \rangle &= 0 \\ \left(\frac{\partial u_1}{\partial t}, v_1\right) + \left(R\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_1, v_1\right) + (\boldsymbol{\sigma}_1, \nabla v_1) - \left\langle \hat{t}_1, v_1 \right\rangle &= 0 \\ \left(\frac{\partial u_2}{\partial t}, v_2\right) + \left(R\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_2, v_2\right) + (\boldsymbol{\sigma}_2, \nabla v_2) - \left\langle \hat{t}_2, v_2 \right\rangle &= 0\,, \end{aligned} \tag{A.7}$$

where it is clear that $v_1, v_2 \in H^1(K)$, and $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \boldsymbol{H}(\mathrm{div}, K)$. In order to plug this into (A.4), we need to identify $f(U^j)$, $f(\tilde{U})$, and $f'(\tilde{U})\Delta U$. We can identify $f(U^j)$ as the sum of the left hand terms in (A.7) at Runge-Kutta stage j, and $f(\tilde{U})$ is the same thing except for the boundary terms in angle brackets evaluated at the previous nonlinear iteration. Finally, $f'(\tilde{U})\Delta U$ is simply the

linearization around $\tilde{U}$:

$$
\begin{aligned}
&(R\Delta\boldsymbol{\sigma}_1, \boldsymbol{\tau}_1) + (\Delta u_1, \nabla\cdot\boldsymbol{\tau}_1) - \langle \hat{u}_1, \tau_{1n}\rangle + \\
&(R\Delta\boldsymbol{\sigma}_2, \boldsymbol{\tau}_2) + (\Delta u_2, \nabla\cdot\boldsymbol{\tau}_2) - \langle \hat{u}_2, \tau_{2n}\rangle + \\
&\left(R\begin{pmatrix}\tilde{u}_1\\ \tilde{u}_2\end{pmatrix}\cdot\Delta\boldsymbol{\sigma}_1, v_1\right) + \left(R\begin{pmatrix}\Delta u_1\\ \Delta u_2\end{pmatrix}\cdot\tilde{\boldsymbol{\sigma}}_1, v_1\right) + (\Delta\boldsymbol{\sigma}_1, \nabla v_1) - \left\langle \hat{t}_1, v_1\right\rangle + \\
&\left(R\begin{pmatrix}\tilde{u}_1\\ \tilde{u}_2\end{pmatrix}\cdot\Delta\boldsymbol{\sigma}_2, v_2\right) + \left(R\begin{pmatrix}\Delta u_1\\ \Delta u_2\end{pmatrix}\cdot\tilde{\boldsymbol{\sigma}}_2, v_2\right) + (\Delta\boldsymbol{\sigma}_2, \nabla v_2) - \left\langle \hat{t}_2, v_2\right\rangle ,
\end{aligned}
\tag{A.8}
$$

where the fluxes and traces are simply solved for at each nonlinear iteration rather than updated like the field variables. Now that we have identified the various pieces, we can just plug this system into (A.4) and time step toward a transient solution.

A.2.2.2 Numerical Example

An exact solution to the 2D Burgers' equations is[76]

$$
\begin{aligned}
u_1(x,y,t) &= \frac{3}{4} - \frac{1}{4(1+e^{R(-t-4x+4y)/32})}\\
u_2(x,y,t) &= \frac{3}{4} + \frac{1}{4(1+e^{R(-t-4x+4y)/32})}\,.
\end{aligned}
\tag{A.9}
$$

We solve on a unit square domain from $t = 0$ to 0.5 with initial condition given by (A.9) at $t = 0$ and boundary conditions that evolve with the exact solution. We use a 6 stage ESDIRK scheme (which should be 4th order accurate) with the time step equal to the mesh size. We also use a 4th order accurate DPG scheme for the spatial solve at each Runge-Kutta stage. If our temporal and spatial schemes are implemented correctly, we should expect overall 4th

order convergence. And, in fact, we do achieve the desired convergence rate according to Figure A.1.

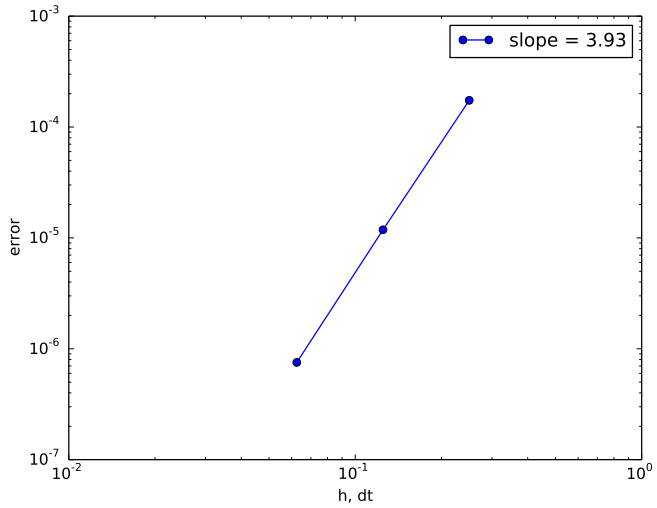


Figure A.1: L^2 convergence of u_1 and u_2 for the 2D Burgers' equation

Appendix B

Comparison of Primitive, Conservation, and Entropy Variables for Compressible Navier-Stokes

In this appendix we discuss some work we did exploring a comparison between three formulations of the compressible Navier-Stokes equations: primitive variables, conservation variables, and entropy variables. Primitive variables are the natural, physically intuitive variables in which the Navier-Stokes equations are usually presented: density, velocity, and temperature. Conservation variables are popular as they simplify time stepping algorithms. The independent variables are density, momentum, and total energy. Entropy variables were proposed by Tom Hughes in [46] and are selected such that the stiffness matrix in a Bubnov-Galerkin finite element discretization is symmetric. However the independent variables do not correspond to any intuitive physical quantity and the resulting equations are the most nonlinear of the three. Recalling the definitions from Chapter 5, we define the necessary linear and nonlinear terms that fit within that framework.

B.1 Primitive Variables

We begin by recalling the definitions for primitive variables:

$$
\begin{aligned}
C_c &:= \rho \\
\boldsymbol{C}_m &:= \rho \boldsymbol{u} \\
C_e &:= \rho(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}) \\
\boldsymbol{F}_c &:= \rho \boldsymbol{u} \\
\mathbb{F}_m &:= \rho \boldsymbol{u} \otimes \boldsymbol{u} + \rho R T \mathbb{I} \\
\boldsymbol{F}_e &:= \rho \boldsymbol{u} \left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u} \right) + \boldsymbol{u}\rho R T \\
\boldsymbol{K}_c &:= \boldsymbol{0} \\
\mathbb{K}_m &:= \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I} \right) \\
\boldsymbol{K}_e &:= -\boldsymbol{q} + \boldsymbol{u}\cdot\left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I} \right) \\
\mathbb{M}_{\mathbb{D}} &:= \mathbb{D} \\
\boldsymbol{M}_{\boldsymbol{q}} &:= \frac{Pr}{C_p}\boldsymbol{q} \\
\boldsymbol{G}_{\mathbb{D}} &:= \boldsymbol{u} \\
G_{\boldsymbol{q}} &:= -T\,.
\end{aligned}
$$

B.1.1 Linearized Terms

Let $W = \{\rho, \boldsymbol{u}, T\}$. The linearized terms are:

$$\frac{\partial C_c(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := \Delta\rho$$

$$\frac{\partial \boldsymbol{C}_m(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := \Delta\rho\tilde{\boldsymbol{u}} + \tilde{\rho}\Delta\boldsymbol{u}$$

$$\frac{\partial C_e(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := C_v\Delta\rho\tilde{T} + C_v\tilde{\rho}\Delta T + \frac{1}{2}\left(\Delta\rho\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}} + \tilde{\rho}\Delta\boldsymbol{u}\cdot\tilde{\boldsymbol{u}} + \tilde{\rho}\tilde{\boldsymbol{u}}\cdot\Delta\boldsymbol{u}\right)$$

$$\frac{\partial \boldsymbol{F}_c(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := \Delta\rho\tilde{\boldsymbol{u}} + \tilde{\rho}\Delta\boldsymbol{u}$$

$$\frac{\partial \mathbb{F}_m(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := \Delta\rho\tilde{\boldsymbol{u}}\otimes\tilde{\boldsymbol{u}} + \tilde{\rho}\Delta\boldsymbol{u}\otimes\tilde{\boldsymbol{u}} + \tilde{\rho}\tilde{\boldsymbol{u}}\otimes\Delta\boldsymbol{u} + R\left(\Delta\rho\tilde{T} + \tilde{\rho}\Delta T\right)\mathbb{I}$$

$$\begin{aligned}\frac{\partial \boldsymbol{F}_e(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := {} & C_v\Delta\rho\tilde{\boldsymbol{u}}\tilde{T} + C_v\tilde{\rho}\Delta\boldsymbol{u}\tilde{T} + C_v\tilde{\rho}\tilde{\boldsymbol{u}}\Delta T \\ & + \frac{1}{2}\Delta\rho\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}} + \frac{1}{2}\tilde{\rho}\Delta\boldsymbol{u}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}} + \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\Delta\boldsymbol{u}\cdot\tilde{\boldsymbol{u}} + \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}\cdot\Delta\boldsymbol{u} \\ & + R\Delta\boldsymbol{u}\tilde{\rho}\tilde{T} + R\tilde{\boldsymbol{u}}\Delta\rho\tilde{T} + R\tilde{\boldsymbol{u}}\tilde{\rho}\Delta T\end{aligned}$$

$$\frac{\partial \boldsymbol{K}_c(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := \boldsymbol{0}$$

$$\frac{\partial \mathbb{K}_m(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := \left(\Delta\mathbb{D} + \Delta\mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\Delta\mathbb{D})\mathbb{I}\right)$$

$$\begin{aligned}\frac{\partial \boldsymbol{K}_e(\tilde{W}, \tilde{\Psi})}{\partial\{W, \Psi\}} \begin{pmatrix} \Delta W \\ \Delta\Psi \end{pmatrix} := {} & -\Delta\boldsymbol{q} + \Delta\boldsymbol{u}\cdot\left(\tilde{\mathbb{D}} + \tilde{\mathbb{D}}^T - \frac{2}{3}\operatorname{tr}(\tilde{\mathbb{D}})\mathbb{I}\right) \\ & + \tilde{\boldsymbol{u}}\cdot\left(\Delta\mathbb{D} + \Delta\mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\Delta\mathbb{D})\mathbb{I}\right)\end{aligned}$$

$$
\begin{aligned}
\frac{\partial \mathbb{M}_{\mathbb{D}}(\tilde{W},\tilde{\Psi})}{\partial\{W,\Psi\}}\begin{pmatrix}\Delta W\\ \Delta\Psi\end{pmatrix} &:= \Delta\mathbb{D}\\
\frac{\partial \boldsymbol{M}_{\boldsymbol{q}}(\tilde{W},\tilde{\Psi})}{\partial\{W,\Psi\}}\begin{pmatrix}\Delta W\\ \Delta\Psi\end{pmatrix} &:= \frac{Pr}{C_p}\Delta\boldsymbol{q}\\
\frac{\partial \boldsymbol{G}_{\mathbb{D}}(\tilde{W},\tilde{\Psi})}{\partial\{W,\Psi\}}\begin{pmatrix}\Delta W\\ \Delta\Psi\end{pmatrix} &:= \Delta\boldsymbol{u}\\
\frac{\partial G_{\boldsymbol{q}}(\tilde{W},\tilde{\Psi})}{\partial\{W,\Psi\}}\begin{pmatrix}\Delta W\\ \Delta\Psi\end{pmatrix} &:= -\Delta T\,.
\end{aligned}
$$

B.2 Conservation Variables

The definition of conservation variables is as follows:

$$
\begin{aligned}
\rho &= \rho\\
\boldsymbol{m} &= \rho\boldsymbol{u}\\
E &= \rho\left(C_vT+\frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right)\,.
\end{aligned}
$$

This gives us new definitions for our nonlinear terms:

$$
\begin{aligned}
C_c &:= \rho \\
\boldsymbol{C}_m &:= \boldsymbol{m} \\
C_e &:= E \\
\boldsymbol{F}_c &:= \boldsymbol{m} \\
\mathbb{F}_m &= \frac{\boldsymbol{m} \otimes \boldsymbol{m}}{\rho} + (\gamma - 1)\left(E - \frac{\boldsymbol{m} \cdot \boldsymbol{m}}{2\rho}\right)\mathbb{I} \\
\boldsymbol{F}_e &= \gamma E \frac{\boldsymbol{m}}{\rho} - (\gamma - 1)\frac{\boldsymbol{m} \cdot \boldsymbol{m}}{2\rho^2}\boldsymbol{m} \\
\boldsymbol{K}_c &:= \boldsymbol{0} \\
\mathbb{K}_m &:= \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\
\boldsymbol{K}_e &:= -\boldsymbol{q} + \frac{\boldsymbol{m}}{\rho} \cdot \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\
\mathbb{M}_{\mathbb{D}} &:= \mathbb{D} \\
\boldsymbol{M}_{\boldsymbol{q}} &:= \frac{Pr}{C_p}\boldsymbol{q} \\
\boldsymbol{G}_{\mathbb{D}} &:= \frac{\boldsymbol{m}}{\rho} \\
G_{\boldsymbol{q}} &:= -\left(\frac{E - \frac{1}{2\rho}\boldsymbol{m} \cdot \boldsymbol{m}}{C_v \rho}\right).
\end{aligned}
$$

B.2.1 Linearized Terms

Let $U = \{\rho, \boldsymbol{m}, E\}$. After linearizing, we get the following:

$$\frac{\partial C_c(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \Delta\rho$$

$$\frac{\partial \boldsymbol{C}_m(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \Delta\boldsymbol{m}$$

$$\frac{\partial C_e(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \Delta E$$

$$\frac{\partial \boldsymbol{F}_c(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \Delta\boldsymbol{m}$$

$$\begin{aligned}\frac{\partial \mathbb{F}_m(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} :=\ & \frac{\Delta\boldsymbol{m}\otimes\tilde{\boldsymbol{m}}}{\tilde{\rho}} + \frac{\tilde{\boldsymbol{m}}\otimes\Delta\boldsymbol{m}}{\tilde{\rho}} - \frac{\tilde{\boldsymbol{m}}\otimes\tilde{\boldsymbol{m}}}{\tilde{\rho}^2}\Delta\rho \\ & + (\gamma-1)\left(\Delta E - \frac{\Delta\boldsymbol{m}\cdot\tilde{\boldsymbol{m}}}{2\tilde{\rho}} - \frac{\tilde{\boldsymbol{m}}\cdot\Delta\boldsymbol{m}}{2\tilde{\rho}} + \frac{\tilde{\boldsymbol{m}}\cdot\tilde{\boldsymbol{m}}}{2\tilde{\rho}^2}\Delta\rho\right)\mathbb{I}\end{aligned}$$

$$\begin{aligned}\frac{\partial \boldsymbol{F}_e(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} :=\ & \gamma\left(\Delta E\frac{\tilde{\boldsymbol{m}}}{\tilde{\rho}} + \tilde{E}\frac{\Delta\boldsymbol{m}}{\tilde{\rho}} - \tilde{E}\frac{\tilde{\boldsymbol{m}}}{\tilde{\rho}^2}\Delta\rho\right) \\ & + (\gamma-1)\left(-\frac{\Delta\boldsymbol{m}\tilde{\boldsymbol{m}}\cdot\tilde{\boldsymbol{m}}}{2\tilde{\rho}^2} - \frac{\tilde{\boldsymbol{m}}\Delta\boldsymbol{m}\cdot\tilde{\boldsymbol{m}}}{2\tilde{\rho}^2}\right. \\ & \left. -\frac{\tilde{\boldsymbol{m}}\tilde{\boldsymbol{m}}\cdot\Delta\boldsymbol{m}}{2\tilde{\rho}^2} + \frac{\tilde{\boldsymbol{m}}\tilde{\boldsymbol{m}}\cdot\tilde{\boldsymbol{m}}}{\tilde{\rho}^3}\Delta\rho\right)\end{aligned}$$

$$\frac{\partial \boldsymbol{K}_c(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \boldsymbol{0}$$

$$\frac{\partial \mathbb{K}_m(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \left(\Delta\mathbb{D} + \Delta\mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\Delta\mathbb{D})\mathbb{I}\right)$$

$$\begin{aligned}\frac{\partial \boldsymbol{K}_e(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} :=\ & -\Delta\boldsymbol{q} + \left(\frac{\Delta\boldsymbol{m}}{\tilde{\rho}} - \frac{\tilde{\boldsymbol{m}}}{\tilde{\rho}^2}\Delta\rho\right)\cdot\left(\tilde{\mathbb{D}} + \tilde{\mathbb{D}}^T - \frac{2}{3}\operatorname{tr}(\tilde{\mathbb{D}})\mathbb{I}\right) \\ & + \frac{\tilde{\boldsymbol{m}}}{\tilde{\rho}}\cdot\left(\Delta\mathbb{D} + \Delta\mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\Delta\mathbb{D})\mathbb{I}\right)\end{aligned}$$

$$\frac{\partial \mathbb{M}_{\mathbb{D}}(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \Delta\mathbb{D}$$

$$\frac{\partial \boldsymbol{M}_{\boldsymbol{q}}(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \frac{Pr}{C_p}\Delta\boldsymbol{q}$$

$$\frac{\partial \boldsymbol{G}_{\mathbb{D}}(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := \frac{\Delta\boldsymbol{m}}{\tilde{\rho}} - \frac{\tilde{\boldsymbol{m}}}{\tilde{\rho}^2}\Delta\rho$$

$$\frac{\partial G_{\boldsymbol{q}}(\tilde{U},\tilde{\Psi})}{\partial\{U,\Psi\}}\begin{pmatrix}\Delta U\\ \Delta\Psi\end{pmatrix} := -\left(\frac{\Delta E - \frac{1}{2\tilde{\rho}}\Delta\boldsymbol{m}\cdot\tilde{\boldsymbol{m}} - \frac{1}{2\tilde{\rho}}\tilde{\boldsymbol{m}}\cdot\Delta\boldsymbol{m} + \frac{1}{2\tilde{\rho}^2}\tilde{\boldsymbol{m}}\cdot\tilde{\boldsymbol{m}}\Delta\rho}{C_v\tilde{\rho}} - \frac{\tilde{E} - \frac{1}{2\tilde{\rho}}\tilde{\boldsymbol{m}}\cdot\tilde{\boldsymbol{m}}}{C_v\tilde{\rho}^2}\Delta\rho\right).$$

B.3 Entropy Variables

Now we wish to do a change of variables to entropy variables:

$$V_c = \frac{-E + (E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m})\left(\gamma + 1 - \ln\left[\frac{(\gamma-1)(E-\frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m})}{\rho^\gamma}\right]\right)}{E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m}}$$

$$\boldsymbol{V}_m = \frac{\boldsymbol{m}}{E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m}}$$

$$V_e = \frac{-\rho}{E - \frac{1}{2\rho}\boldsymbol{m}\cdot\boldsymbol{m}},$$

with reverse mapping:

$$\rho = -\alpha V_e$$

$$\boldsymbol{m} = \alpha\boldsymbol{V}_m$$

$$E = \alpha\left(1 - \frac{1}{2V_e}\boldsymbol{V}_m\cdot\boldsymbol{V}_m\right),$$

where

$$\alpha(V_c, \boldsymbol{V}_m, V_e) = \left[\frac{\gamma - 1}{(-V_e)^\gamma}\right]^{\frac{1}{\gamma-1}} \exp\left[\frac{-\gamma + V_c - \frac{1}{2V_e}\boldsymbol{V}_m \cdot \boldsymbol{V}_m}{\gamma - 1}\right].$$

The nonlinear terms are:

$$\begin{aligned}
C_c &:= -\alpha V_e \\
\boldsymbol{C}_m &:= \alpha \boldsymbol{V}_m \\
C_e &:= \alpha\left(1 - \frac{1}{2V_e}\boldsymbol{V}_m \cdot \boldsymbol{V}_m\right) \\
\boldsymbol{F}_c &= \alpha \boldsymbol{V}_m \\
\mathbb{F}_m &= \alpha\left(-\frac{\boldsymbol{V}_m \otimes \boldsymbol{V}_m}{V_e} + (\gamma - 1)\mathbb{I}\right) \\
\boldsymbol{F}_e &= \alpha\frac{\boldsymbol{V}_m}{V_e}\left(\frac{1}{2V_e}\boldsymbol{V}_m \cdot \boldsymbol{V}_m - \gamma\right) \\
\boldsymbol{K}_c &:= \boldsymbol{0} \\
\mathbb{K}_m &:= \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\
\boldsymbol{K}_e &:= -\boldsymbol{q} + \frac{\boldsymbol{V}_m}{V_e} \cdot \left(\mathbb{D} + \mathbb{D}^T - \frac{2}{3}\operatorname{tr}(\mathbb{D})\mathbb{I}\right) \\
\mathbb{M}_{\mathbb{D}} &:= \mathbb{D} \\
\boldsymbol{M}_{\boldsymbol{q}} &:= \frac{Pr}{C_p}\boldsymbol{q} \\
\boldsymbol{G}_{\mathbb{D}} &:= -\frac{\boldsymbol{V}_m}{V_e} \\
G_{\boldsymbol{q}} &:= \frac{1}{C_v V_e}.
\end{aligned}$$

B.3.1 Linearized Terms

Let $V = \{V_c, \boldsymbol{V}_m, V_e\}$. And the linearized terms for entropy variables are:

$$
\begin{aligned}
\frac{\partial C_c(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= -\tilde{V}_e\frac{\partial\alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} - \alpha(\tilde{V},\tilde{\Psi})\Delta V_e\\
\frac{\partial \boldsymbol{C}_m(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \tilde{\boldsymbol{V}}_m\frac{\partial\alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} + \alpha(\tilde{V},\tilde{\Psi})\Delta\boldsymbol{V}_m\Delta\rho\tilde{\boldsymbol{u}} + \tilde{\rho}\Delta\boldsymbol{u}\\
\frac{\partial C_e(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \left(1-\frac{1}{2\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m\right)\frac{\partial\alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix}\\
&\quad - \alpha(\tilde{V},\tilde{\Psi})\frac{1}{\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\Delta\boldsymbol{V}_m + \alpha(\tilde{V},\tilde{\Psi})\frac{1}{2\tilde{V}_e^2}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m\Delta V_e\\
\frac{\partial \boldsymbol{F}_c(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \tilde{\boldsymbol{V}}_m\frac{\partial\alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} + \alpha(\tilde{V},\tilde{\Psi})\Delta\boldsymbol{V}_m\\
\frac{\partial \mathbb{F}_m(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \left(-\frac{\tilde{\boldsymbol{V}}_m\otimes\tilde{\boldsymbol{V}}_m}{\tilde{V}_e} + (\gamma-1)\mathbb{I}\right)\frac{\partial\alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix}\\
&\quad + \alpha(\tilde{V},\tilde{\Psi})\left(-\frac{\Delta\boldsymbol{V}_m\otimes\tilde{\boldsymbol{V}}_m}{\tilde{V}_e} - \frac{\tilde{\boldsymbol{V}}_m\otimes\Delta\boldsymbol{V}_m}{\tilde{V}_e} + \frac{\tilde{\boldsymbol{V}}_m\otimes\tilde{\boldsymbol{V}}_m}{\tilde{V}_e^2}\Delta V_e\right)\\
\frac{\partial \boldsymbol{F}_e(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \frac{\tilde{\boldsymbol{V}}_m}{\tilde{V}_e}\left(\frac{1}{2\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m-\gamma\right)\frac{\partial\alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix}\\
&\quad + \alpha(\tilde{V},\tilde{\Psi})\left(\frac{\Delta\boldsymbol{V}_m}{\tilde{V}_e}\left(\frac{1}{2\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m-\gamma\right)\right.\\
&\qquad -\frac{\tilde{\boldsymbol{V}}_m}{V_e^2}\left(\frac{1}{2\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m-\gamma\right)\Delta V_e\\
&\qquad \left.+\frac{\tilde{\boldsymbol{V}}_m}{\tilde{V}_e}\left(\frac{1}{\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\Delta\boldsymbol{V}_m - \frac{1}{2\tilde{V}_e^2}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m\Delta V_e\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \boldsymbol{K}_c(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \mathbf{0}\\
\frac{\partial \mathbb{K}_m(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \left(\Delta\mathbb{D}+\Delta\mathbb{D}^T-\frac{2}{3}\operatorname{tr}(\Delta\mathbb{D})\mathbb{I}\right)\\
\frac{\partial \boldsymbol{K}_e(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= -\Delta\boldsymbol{q}+\left(\frac{\Delta\boldsymbol{V}_m}{\tilde{V}_e}-\frac{\tilde{\boldsymbol{V}}_m}{\tilde{V}_e^2}\Delta V_e\right)\cdot\left(\tilde{\mathbb{D}}+\tilde{\mathbb{D}}^T-\frac{2}{3}\operatorname{tr}(\tilde{\mathbb{D}})\mathbb{I}\right)\\
&\quad+\frac{\tilde{\boldsymbol{V}}_m}{\tilde{V}_e}\cdot\left(\Delta\mathbb{D}+\Delta\mathbb{D}^T-\frac{2}{3}\operatorname{tr}(\Delta\mathbb{D})\mathbb{I}\right)\\
\frac{\partial \mathbb{M}_{\mathbb{D}}(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \Delta\mathbb{D}\\
\frac{\partial \boldsymbol{M}_{\boldsymbol{q}}(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= \frac{Pr}{C_p}\Delta\boldsymbol{q}\\
\frac{\partial \boldsymbol{G}_{\mathbb{D}}(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= -\left(\frac{\Delta\boldsymbol{V}_m}{\tilde{V}_e}-\frac{\tilde{\boldsymbol{V}}_m}{\tilde{V}_e^2}\Delta V_e\right)\\
\frac{\partial G_{\boldsymbol{q}}(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &:= -\frac{1}{C_v V_e^2}\Delta V_e\\
\frac{\partial \alpha(\tilde{V},\tilde{\Psi})}{\partial\{V,\Psi\}}\begin{pmatrix}\Delta V\\ \Delta\Psi\end{pmatrix} &= \left[\frac{\gamma-1}{(-\tilde{V}_e)^{\gamma}}\right]^{-1}\gamma(-\tilde{V}_e)^{-(\gamma+1)}\alpha(\tilde{V},\tilde{\Psi})\Delta V_e\\
&\quad+\frac{\alpha(\tilde{V},\tilde{\Psi})}{\gamma-1}\left(\Delta V_c-\frac{1}{\tilde{V}_e}\tilde{\boldsymbol{V}}_m\cdot\Delta\boldsymbol{V}_m+\frac{1}{2\tilde{V}_e^2}\tilde{\boldsymbol{V}}_m\cdot\tilde{\boldsymbol{V}}_m\Delta V_e\right).
\end{aligned}
$$

B.4 Numerical Experiments

We perform a couple numerical experiments to compare the different formulations. In Chapter 5 we used a incrementally decreased μ with every refinement step as this approach was found to produce cleaner refinement patterns; here we hold μ constant for each problem to show that it is still

possible to arrive at a converged solution, but we end up with a less desirable final refinement pattern.

B.4.1 Sod Shock Tube

We repeat the Sod shock tube problem described in Chapter 5 with $\mu = 10^{-5}$, $p = 2$, $\Delta p = 2$, and the NSDecoupled norm. We omit plots of velocity and pressure as they don't really contribute anything new to the comparisons. Comparing Figures B.1 - B.3, it seems that primitive and conservation variables are of similar quality, at least by the eyeball norm. Entropy variables, on the other hand, suffer from much more extreme overshoots and undershoots compared to the other formulations.

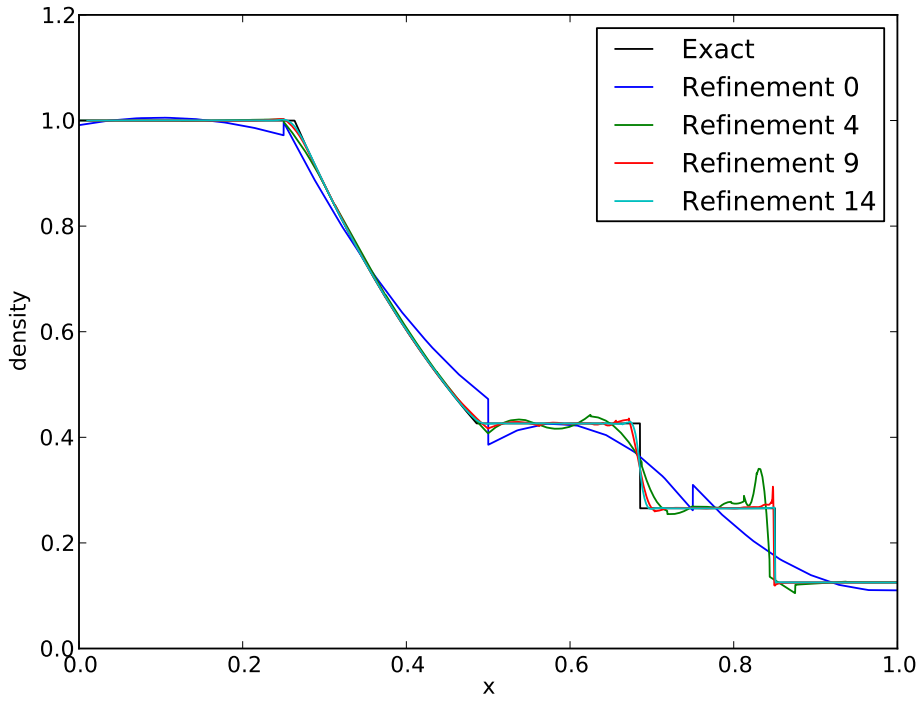
B.4.2 Noh Implosion

We repeat the Noh problem from before with $\mu = 10^{-3}$, $p = 2$, $\Delta p = 2$, and the NSDecoupled norm. In Chapter 5 we simulated a half domain with a symmetry boundary condition at the origin; here we compute the full domain. The other difference is that this simulation was computed as a series of four time slabs rather than as one monolithic computation. This means that the $[0, \frac{1}{4}]$ time slab was computed for 8 adaptive refinement steps then the final solution was projected onto the $[\frac{1}{4}, \frac{1}{2}]$ time slab as an initial condition. This was repeated until we arrived at the $[\frac{3}{4}, 1]$ time slab, where the density traces in Figure B.4 are taken. We see more unwanted refinements in this computation compared to Chapter 5 due to the spurious shock patterns that develop on

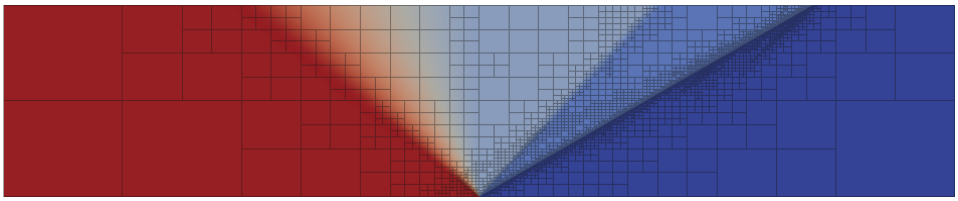
coarse meshes. We are not able to compare the entropy formulation for this problem since the initial conditions contain infinities under this formulation. Again, primitive and conservation variables produce similar results.

B.5 Conclusion

The conclusion then is that since DPG already produces a symmetric, positive-definite stiffness matrix, there is no reason to prefer entropy variables. The choice between primitive and conservation variables depends on which one is easier to implement as they will both give similar results. We decided to stick with primitive variables as they were slightly simpler and less nonlinear.

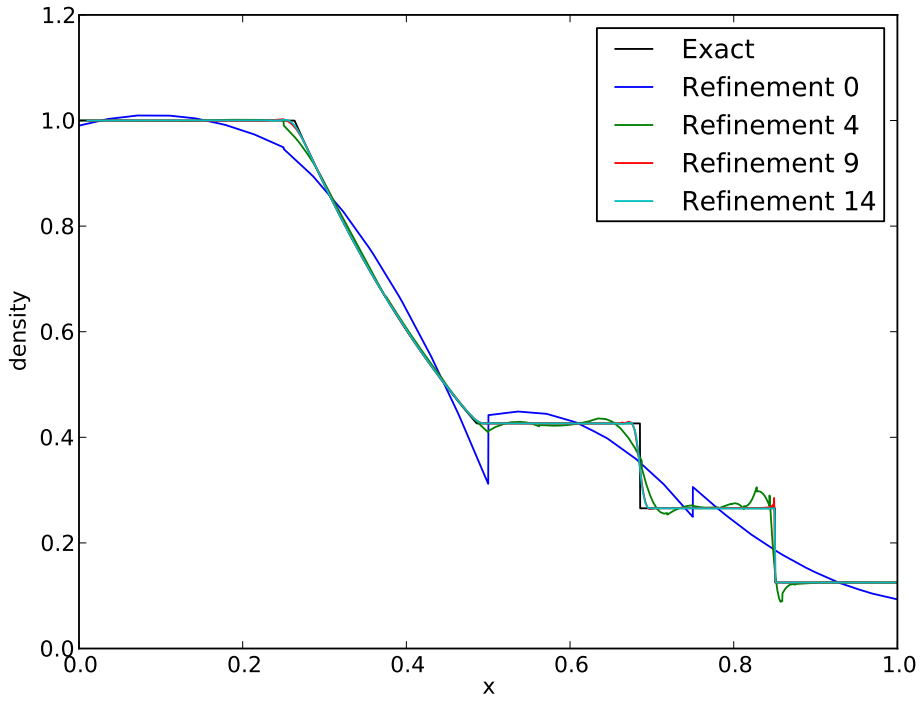


(a) Density

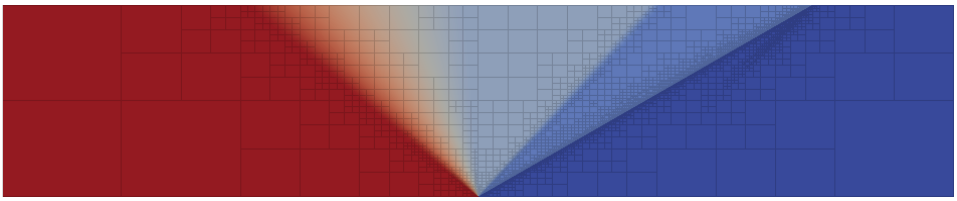


(b) Final mesh colored by ρ

Figure B.1: Sod problem with primitive variables

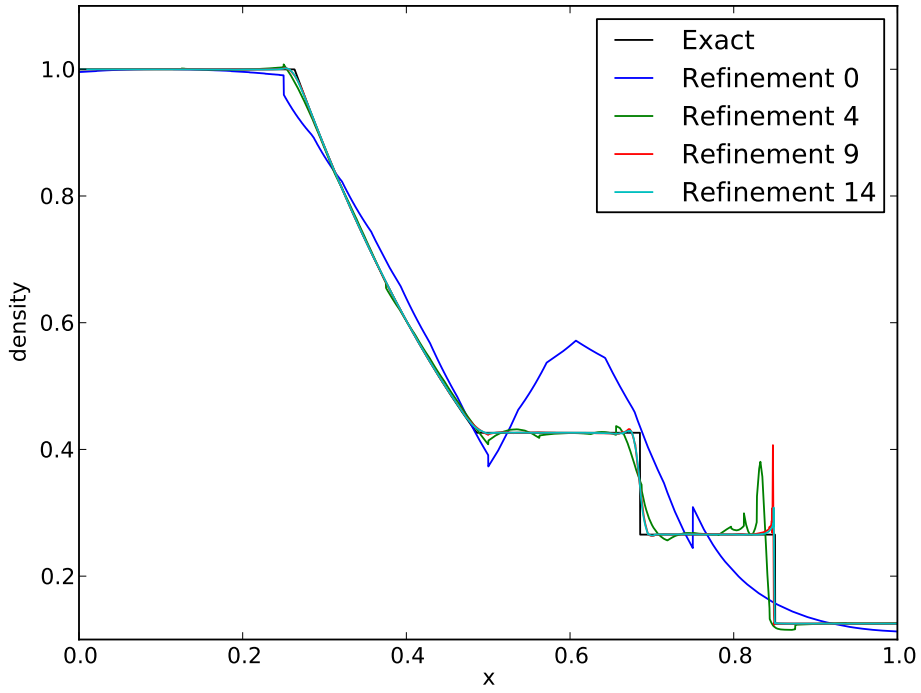


(a) Density

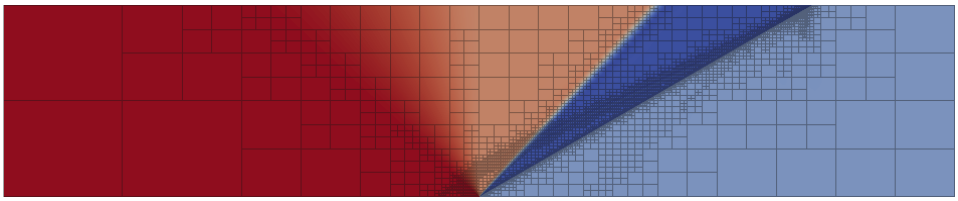


(b) Final mesh colored by ρ

Figure B.2: Sod problem with conservation variables



(a) Density



(b) Final mesh colored by V_c

Figure B.3: Sod problem with entropy variables

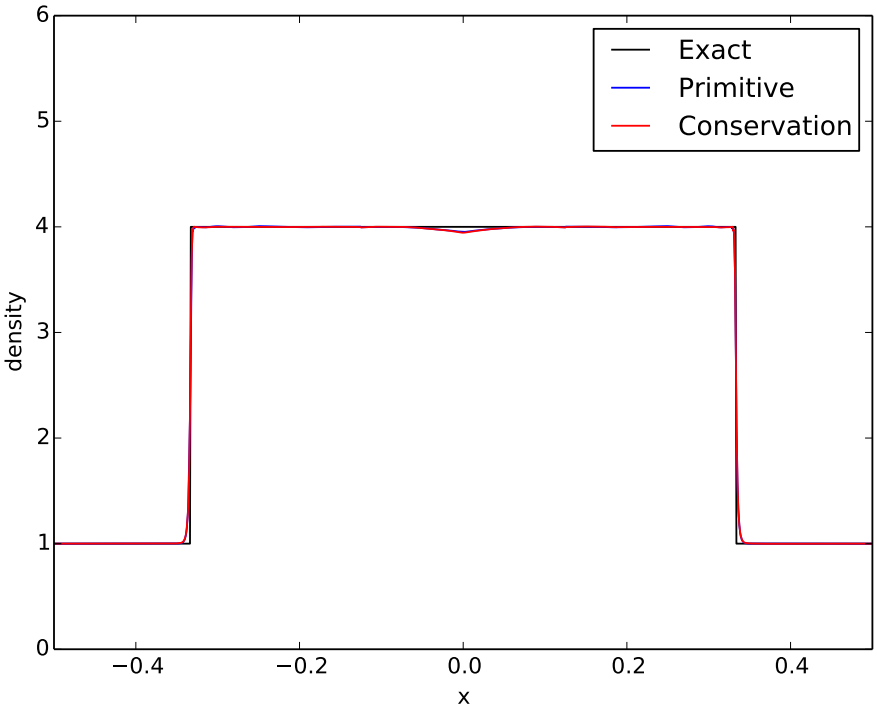
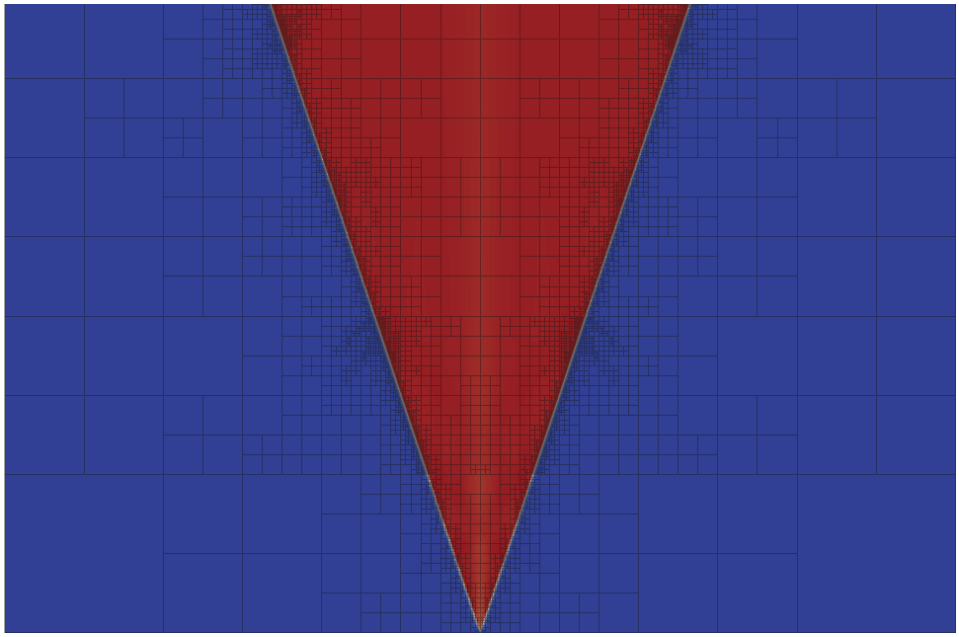
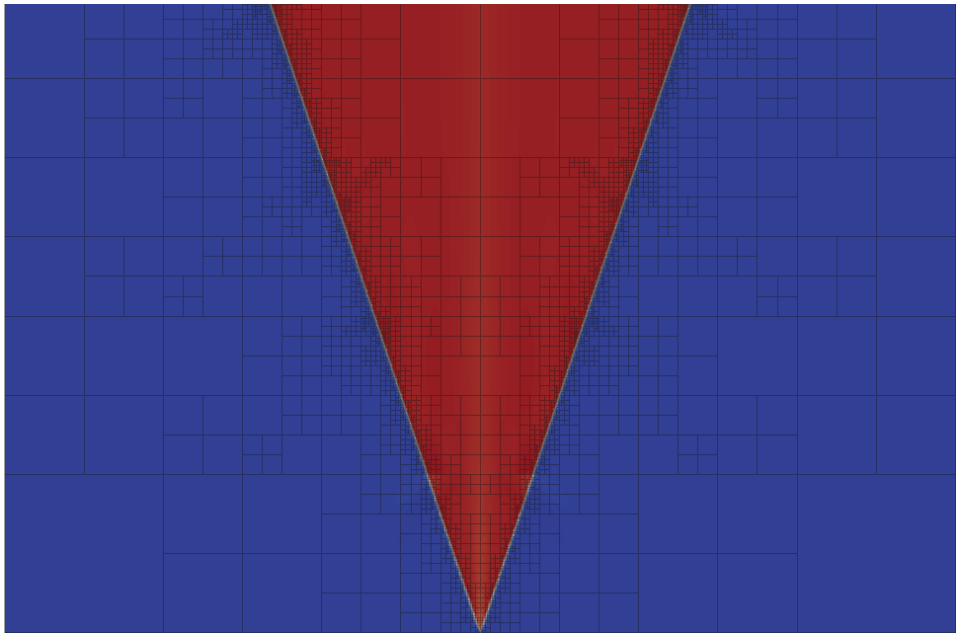


Figure B.4: Density at final time



(a) Final mesh with primitive variables



(b) Final mesh with conservation variables

Figure B.5: Noh meshes colored by ρ

Appendix C

Entropy Norms for Compressible Navier-Stokes

C.1 Motivation

From the previous appendix, let W, U, and V denote the set of primitive, conservation, and entropy variables respectively. It is well known that the entropy function

$$H = -\rho \log(p\rho^{-\gamma}) \,.$$

provides a natural residual for the system of equations. The Hessian of H is known as the symmetrizer of the Navier-Stokes system: $A_0 = H_{,UU}$. The inner product $(U, A_0 U)$ provides a natural measure (metric) for the Euler equations. By definition of the entropy variables (see [46]) $V_{,U} = H_{,UU}$, where

$$V_{,U}(U) = \begin{bmatrix} \frac{4\gamma\rho^2 E^2 - 4\gamma\rho E \boldsymbol{m}\cdot\boldsymbol{m} + (1+\gamma)(\boldsymbol{m}\cdot\boldsymbol{m})^2}{\rho(\boldsymbol{m}\cdot\boldsymbol{m}-2\rho E)^2} & -\frac{2\boldsymbol{m}\boldsymbol{m}\cdot\boldsymbol{m}}{(\boldsymbol{m}\cdot\boldsymbol{m}-2\rho E)^2} & -\frac{4\rho(\rho E - \boldsymbol{m}\cdot\boldsymbol{m})}{(\boldsymbol{m}\cdot\boldsymbol{m}-2\rho E)^2} \\ & \frac{2\rho(2\rho E + \boldsymbol{m}\cdot\boldsymbol{m})}{(\boldsymbol{m}\cdot\boldsymbol{m}-2\rho E)^2} & -\frac{4\rho^2\boldsymbol{m}}{(\boldsymbol{m}\cdot\boldsymbol{m}-2\rho E)^2} \\ Symm. & & \frac{4\rho^3}{(\boldsymbol{m}\cdot\boldsymbol{m}-2\rho E)^2} \end{bmatrix} .$$

Since our previous comparison of Navier-Stokes formulations showed no strong reason to prefer anything over primitive variables, we will choose to work with primitive variables in this appendix. As such, we need to perform a change of variables to find the symmetrizer for the set of primitive variables:

$U = U_{,W}W$. Our entropy metric is then

$$(U_{,W}W, V_{,U}U_{,W}W) = \left(W, U_{,W}^T V_{,U}U_{,W}W\right)$$

Then

$$U_{,W} = \begin{bmatrix} 1 & 0 & 0 \\ \boldsymbol{u} & \rho & 0 \\ C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u} & \rho\boldsymbol{u} & C_v\rho \end{bmatrix}$$

where $V_{,U}$ in primitive variables is

$$V_{,U}(W) = \begin{bmatrix} \frac{\gamma}{\rho} + \frac{(\boldsymbol{u}\cdot\boldsymbol{u})^2}{4\rho C_v^2 T^2} & -\frac{\frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\boldsymbol{u}}{\rho C_v^2 T^2} & -\frac{(C_vT - \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u})}{\rho C_v^2 T^2} \\ & \frac{C_vT + \boldsymbol{u}\cdot\boldsymbol{u}}{\rho C_v^2 T^2} & -\frac{\boldsymbol{u}}{\rho C_v^2 T^2} \\ Symm. & & \frac{1}{\rho C_v^2 T^2} \end{bmatrix}$$

and

$$A_0(W) = U_{,W}^T V_{,U} U_{,W} = \begin{bmatrix} \frac{\gamma-1}{\rho} & 0 & 0 \\ 0 & \frac{\rho}{C_vT} & 0 \\ 0 & 0 & \frac{\rho}{T^2} \end{bmatrix}.$$

As a check, $(W, A_0(W)W)$ has consistent units of density.

C.2 Entropy Scaled Test Norms

We repeat the argument to develop the necessary condition for a robust norm, but where we replace the bound on $\|u\|$ with $\left\|A_0^{\frac{1}{2}}u\right\|$. Let u represent all volume variables, $\hat{u}$ all interface variables, and v all test variables. We can write our ultra-weak formulation as

$$b\left((u,\hat{u}),v\right) = (u, A^*v)_{L^2} + \langle\widehat{u}, [\![v]\!]\rangle_{\Gamma_h},$$

where A^* represents the adjoint. For conforming v^* satisfying $A^*v^* = A_0 u$:

$$\begin{aligned}\left\|A_0^{\frac{1}{2}}u\right\|^2 &= \frac{b(u,v^*)}{\|v^*\|_V}\|v^*\|_V \\ &\leq \sup_{v^*\neq 0}\frac{|b(u,v^*)|}{\|v^*\|}\|v^*\| = \|u\|_E\|v^*\|_V\ .\end{aligned}$$

This defines a necessary condition for robustness, namely that

$$\|v^*\|_V \lesssim \left\|A_0^{\frac{1}{2}}u\right\|_{L^2}\ . \tag{C.1}$$

If this condition is satisfied, then we get our final result:

$$\left\|A_0^{\frac{1}{2}}u\right\|_{L^2} \lesssim \|u\|_E\ .$$

We begin by loading our compressible Navier-Stokes adjoint equations with $A_0 W$:

$$\begin{aligned}\frac{1}{\mu}M^*(\Psi) + K^*(\nabla V) &= 0 \\ -\begin{pmatrix} F^* \\ C^* \end{pmatrix}(\nabla_{xt}V) + G^*(\nabla\Psi) &= A_0 W\ .\end{aligned}$$

Without proof, we suggest the existence of analogous lemmas 3.3.1 and 3.3.2 for this case, namely that there exist bounds

$$\left\|A_0^{-\frac{1}{2}}V\right\|^2 + \mu\left\|A_0^{-\frac{1}{2}}\nabla V\right\|^2 \leq \left\|A_0^{\frac{1}{2}}W\right\|^2 \tag{C.2}$$

$$\left\|A_0^{-\frac{1}{2}}\begin{pmatrix} F^* \\ C^* \end{pmatrix}(\nabla_{xt}V)\right\| \lesssim \left\|A_0^{\frac{1}{2}}W\right\|\ . \tag{C.3}$$

These would hypothetically be derived by substituting the first adjoint equation into the second then multiplying both sides by

$$A_0^{-\frac{1}{2}}e^t V$$

and

$$-A_0^{-\frac{1}{2}} \begin{pmatrix} F^* \\ C^* \end{pmatrix} (\nabla_{xt} V),$$

respectively for each desired bound, then integrating over Q and following similar manipulations as were done in said lemmas. Assuming the existence of said bounds, the analogous entropy scaled robust and coupled robust norms for compressible Navier-Stokes would be

$$\begin{aligned} \|(V,\Psi)\|_{V,K}^2 \coloneqq & \left\| A_0^{-\frac{1}{2}} (F^* + C^*) \right\|_K^2 + \mu \left\| A_0^{-\frac{1}{2}} K^* \right\|_K^2 + \min\left(\frac{\mu}{h^2}, 1\right) \left\| A_0^{-\frac{1}{2}} V \right\|_K^2 \\ & + \left\| A_0^{-\frac{1}{2}} G^* \right\|_K^2 + \min\left(\frac{1}{\mu}, \frac{1}{h^2}\right) \left\| A_0^{-\frac{1}{2}} M^* \right\|_K^2, \end{aligned}$$

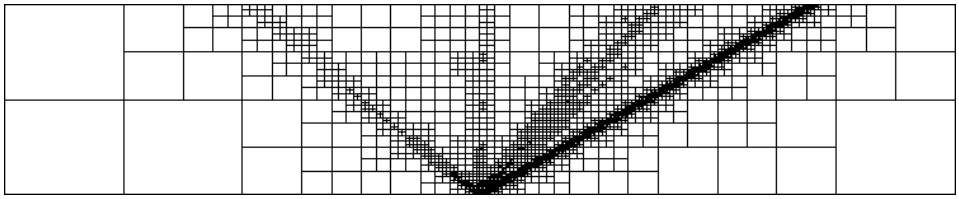
and

$$\begin{aligned} \|(V,\Psi)\|_{V,K}^2 \coloneqq & \left\| A_0^{-\frac{1}{2}} (F^* + C^*) \right\|_K^2 + \mu \left\| A_0^{-\frac{1}{2}} K^* \right\|_K^2 + \min\left(\frac{\mu}{h^2}, 1\right) \left\| A_0^{-\frac{1}{2}} V \right\|_K^2 \\ & + \left\| A_0^{-\frac{1}{2}} (G^* - F^* - C^*) \right\|_K^2 + \min\left(\frac{1}{\mu}, \frac{1}{h^2}\right) \left\| A_0^{-\frac{1}{2}} M^* \right\|_K^2. \end{aligned}$$

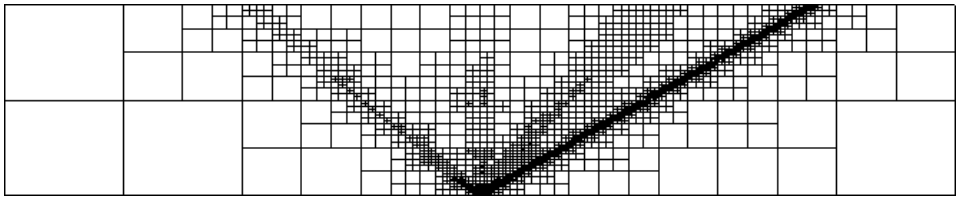
Note that in practice, ρ and T may get very close to 0 which can make the Gram matrix for the test space inner product singular. In order to avoid this, we bound the ρ and T terms in A_0 such that they are always greater than or equal to 0.01.

We attempted two comparisons of the robust norm and the entropy scaled robust norm. The results for the Sod shock tube are very comparable, but the Newton iterations failed to converge on the Noh problem. We chalk this up as an interesting mathematical investigation, but a little disappointing numerically. Besides, the equations have already been nondimensionalized,

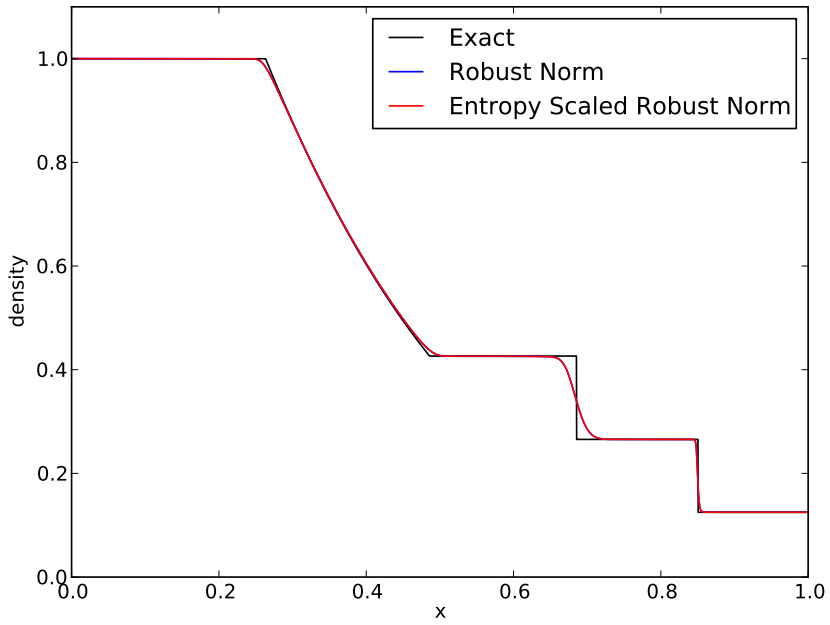
so it seems slightly superfluous to additionally use the concept of entropy to develop consistent test norms.



(a) Final mesh with robust norm



(b) Final mesh with entropy scaled robust norm



(c) Density at final time

Figure C.1: Sod solution after 12 refinements

Appendix D

Scaling Issues

D.1 Global Solvers

The one challenge which we most significantly underestimated before undertaking this work was how our solver would scale on these space-time problems. Preliminary 1D results (2D in space-time), were computable with standard direct solvers, but as we moved to 2D (3D in space-time), direct solvers proved to be a major bottleneck to larger solves, not least of which because they tend to take up more memory than iterative solvers. Fortunately, my collaborator Nathan Roberts at Argonne National Lab has been implementing flexible multigrid strategies within Camellia. Unfortunately, multigrid is well known to perform poorly on convection-dominated diffusion problems. In fact, we can easily construct a case for convection-diffusion with $\epsilon = 10^{-2}$ on a 64×64 mesh solved with Camellia's default multigrid strategy outlined below that exhibits the convergence history in Figure D.1 for the iterative solve.

The details of this simulation aren't important, the point is that it is fairly trivial to contrive a test problem where multigrid performs very poorly for convection-diffusion. This behavior appears to be especially bad on uniform meshes and seems to be somewhat mitigated on adaptive meshes. Ideally

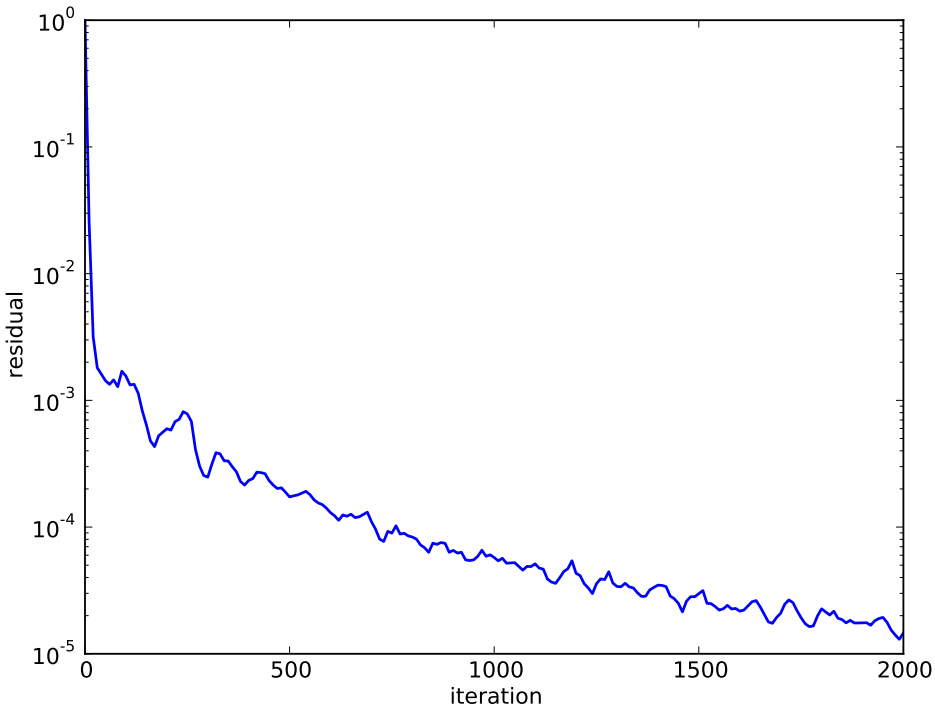


Figure D.1: Residual convergence for a simple convection-diffusion problem

we would like to implement a line smoother which is known to improve performance on convection-dominated diffusion problems, but that is outside the scope of this thesis.

D.1.1 Overview of Multigrid in Camellia

Conjugate gradient is a natural choice for iteratively solving DPG problems because they are always symmetric (Hermitian) positive definite. However a good preconditioner is necessary for efficiency. Nathan Roberts, implemented a geometric multigrid preconditioner that has allowed us to solve

larger problems than we could with direct solvers. I've served as more of a user and tester of the multigrid strategies than as a developer, so I'll only briefly describe an overview of the strategy we settled on that were used for the simulations in this thesis.

After exploring the various options of additive or multiplicative two-cycle, V-cycle, W-cycle, or full multigrid, we settled on a multiplicative V-cycle strategy. We've chosen to employ an overlapping additive Schwarz smoother. In constructing the mesh hierarchy for the multigrid, going from a high order fine mesh, we first start with p-coarsening followed by h-coarsening. More details on multigrid within Camellia will appear in an upcoming technical report by Nathan Roberts.

D.1.2 Scaling on Test Problems

Both space-time and multigrid are fairly recent, experimental features within Camellia and the combination of the two has not scaled as well as we initially expected. In the following tables we illustrate the ballooning cost of these space-time solves for 2D incompressible Navier-Stokes. A 2D space-time solver was implemented for compressible Navier-Stokes as well, but the scaling issues illustrated here for incompressible Navier-Stokes were significantly worse in the presence of shocks. Despite significant effort, we were not able to obtain publishable results for any 2D shock problems.

D.1.2.1 Incompressible Flow Over a Cylinder

Table D.1 refers to a space-time solve of transient flow over a flat plate. Listed times are in seconds. The domain is $[-3, 9] \times [-4.5, 4.5]$ with a 0.5 radius cylinder in at the origin and a final time of 4. The Reynolds number is 100, the flow is initialized to the solution of potential flow over a cylinder. Velocity conditions are applied to the inflow, zero slip to the cylinder, and zero traction to every other boundary. The initial mesh has 80 space-time elements and with quadratic trial functions has 31304 DOFs and looks like Figure D.2. After 4 adaptive refinements, the problem is up to 11742 elements, 4144674 DOFs, and looks like Figure D.3. This problem was excluded from the main set of incompressible results in Chapter 4 because we don't achieve nearly enough resolution to observe any interesting flow features.

The cost per solve increases dramatically with every adaptive refinement step. We compare three runs done on the Lonestar system at the Texas Advanced Computing Center. In the first, we use 1 node with 24 processors and then compare this to 4 nodes with 96 processors and 32 nodes with 768 total processors. Strong scaling results are computed relative to the previous solve with ideal values being $4\times$ and $8\times$ for the 4 node and 32 node runs, respectively. It is clear that increasing the number of processors does accelerate the solve, but we are not very close to the ideal speedup. We hypothesize that load balancing on this problem is sub-optimal as not every processor has to deal with curvilinear element computations around the cylinder. With 768 processors, it takes more than 2 hours to complete 10 Newton iterations on

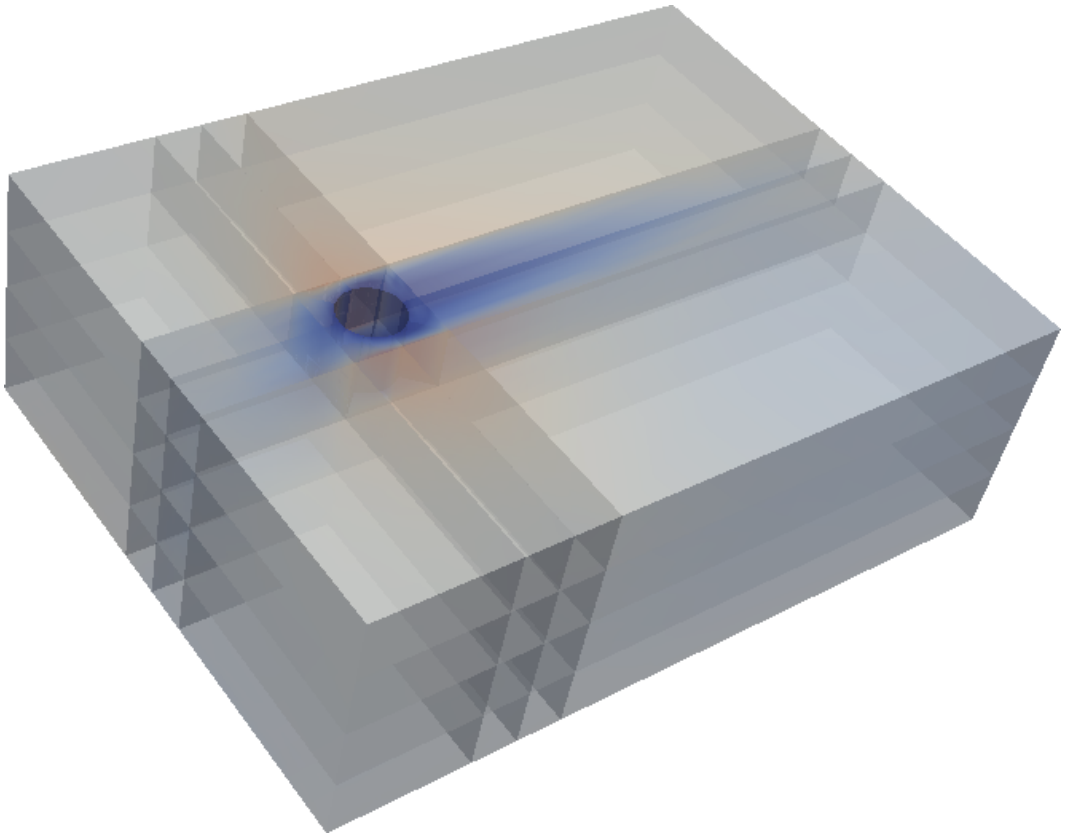


Figure D.2: Initial mesh for cylinder problem colored by velocity magnitude

the fourth refinement step with just over 4 million DOFs. We estimate it would take about 6 refinement steps before we start resolving the viscous flow features.

D.1.2.2 Taylor-Green Vortex

We also consider the Taylor-Green vortex problem described in Chapter 4. The timings for the case of $Re = 1000$ and $p = 2$ are shown in Table D.2. We see better scaling for this problem as there are not any curvilinear elements

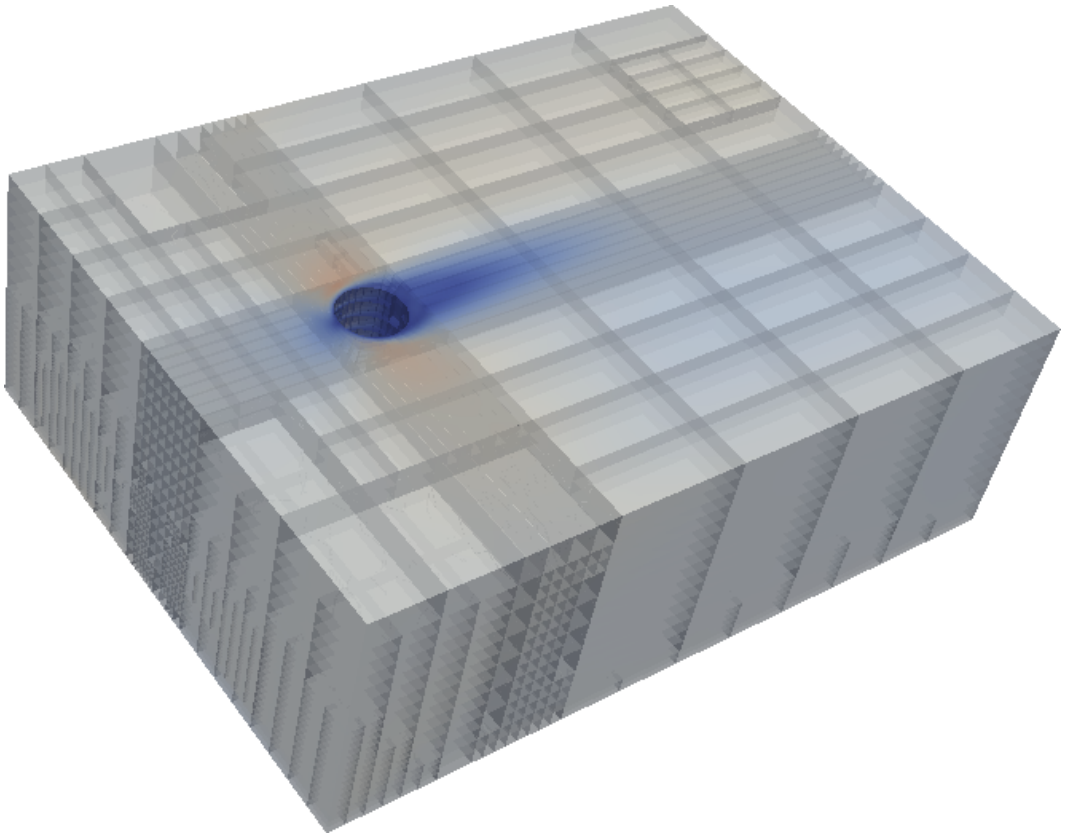


Figure D.3: Fourth adaptive mesh for cylinder problem colored by velocity magnitude

Table D.1: Solve time for transient flow over a cylinder

			1 Node	4 Nodes		32 Nodes	
Ref	Elems	DOFs	Time	Time	Scaling vs 1	Time	Scaling vs 4
0	80	31304	1772	453	3.91	451	1.01
1	605	225908	8190	3574	2.29	717	4.98
2	3013	1081598	32008	12076	2.65	2648	4.56
3	9726	3429384		28744		6319	4.54
4	11742	4144674				8510	

to deal with but the time to solve still blows up considerable with every refinement step.

Table D.2: Solve time for the Taylor-Green vortex

			1 Node	4 Nodes	
Ref	Elems	DOFs	Time	Time	Scaling vs 1
0	60	21302	331.0	140.6	2.35
1	312	108410	945.2	290.6	3.25
2	2020	691834	4880.2	1363.5	3.58
3	9244	3043024		6171.6	

D.2 The Question of Space-Time Slabs

Here we briefly explore the benefits of splitting a computation into space-time slabs under the following assumptions.

1. The maximum required spatial resolution is much finer than the required temporal resolution.

2. Regions requiring high spatial resolution are concentrated in relatively compact parts of the domain.

3. Only isotropic refinements are permitted.

4. The number of time slabs is a power of 2.

The first and second conditions are representative of the boundary layer and shock problems considered in this thesis. The third condition is necessary

as Camellia does not currently support anisotropic refinements in space-time. The fourth assumption simplifies the analysis and is at least representative of a common sense time slab strategy.

Our test case is a steady boundary layer problem with exact solution

$$u = 1 - e^{\frac{x}{\epsilon}}$$

solved on a space-time domain $[-1, 0] \times [0, 1]$. We choose this problem because it is easy to analyze the optimal refinement strategy, but it should be possible to generalize this analysis to more complicated patterns. The optimal refinement pattern (while $h > \epsilon$) just keeps refining toward the right side of the domain. We consider three possible time slab strategies and illustrate each with the same spatial resolution around the boundary layer. The first is to solve the problem as a single space-time slab starting with a single element. This is represented in Figure D.4. The second strategy is to split the domain into a sequence of time slabs each starting with a single space-time element, represented in Figure D.5. The third is to uniformly pre-refine each time slab slab so that it has as many spatial elements as the total number of time slabs, represented in Figure D.6. Theoretically we could design more optimal initial meshes for each time slab, but that would require *a priori* knowledge of the location of solution features.

In each strategy, we wish to refine until we reach a desired spatial resolution of the boundary layer; the figures show a resolution of $h = 1/16$. We can now count the total number of elements for each approach. Let N be

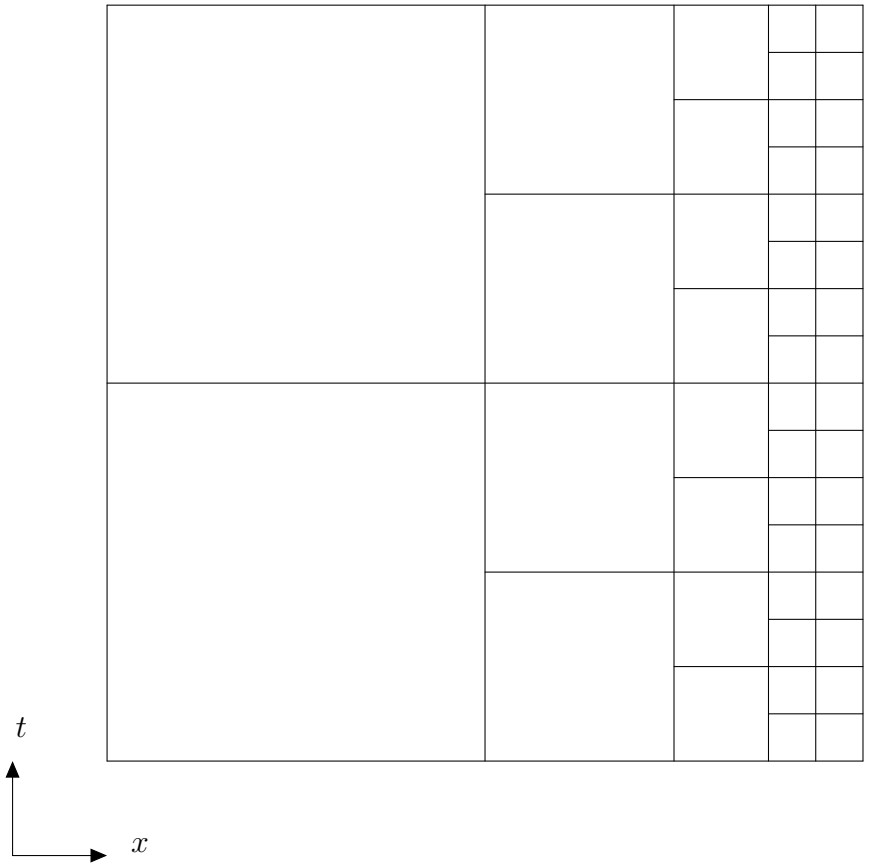


Figure D.4: First time slab strategy

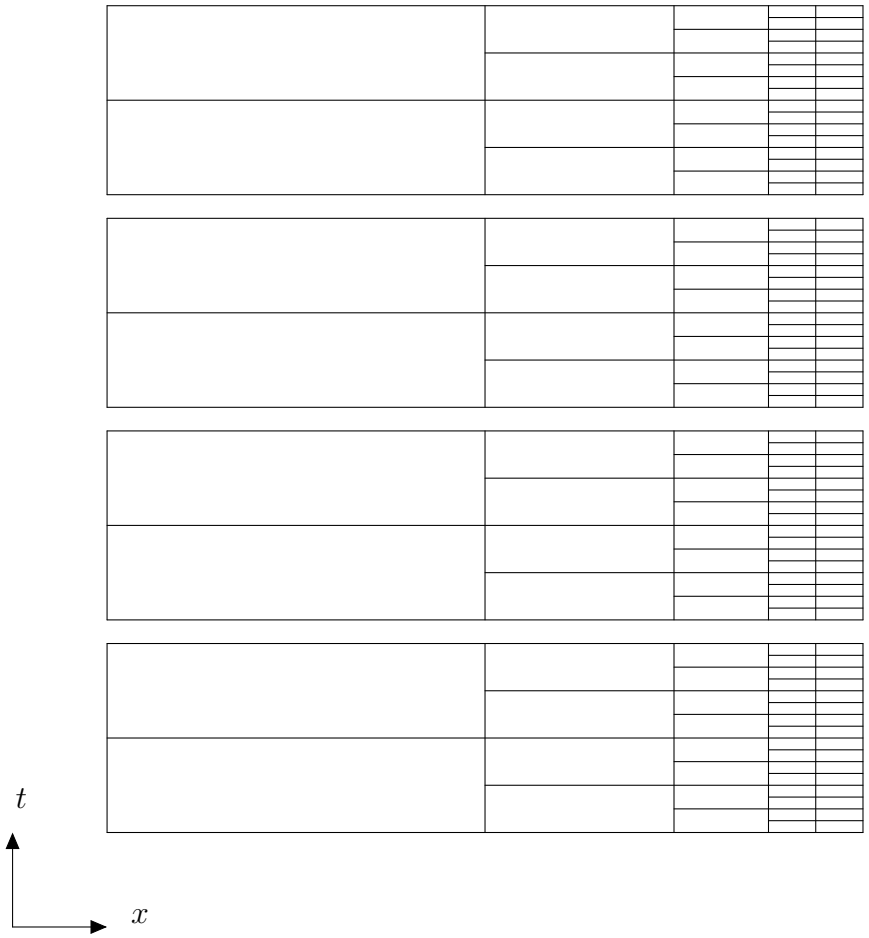


Figure D.5: Second time slab strategy

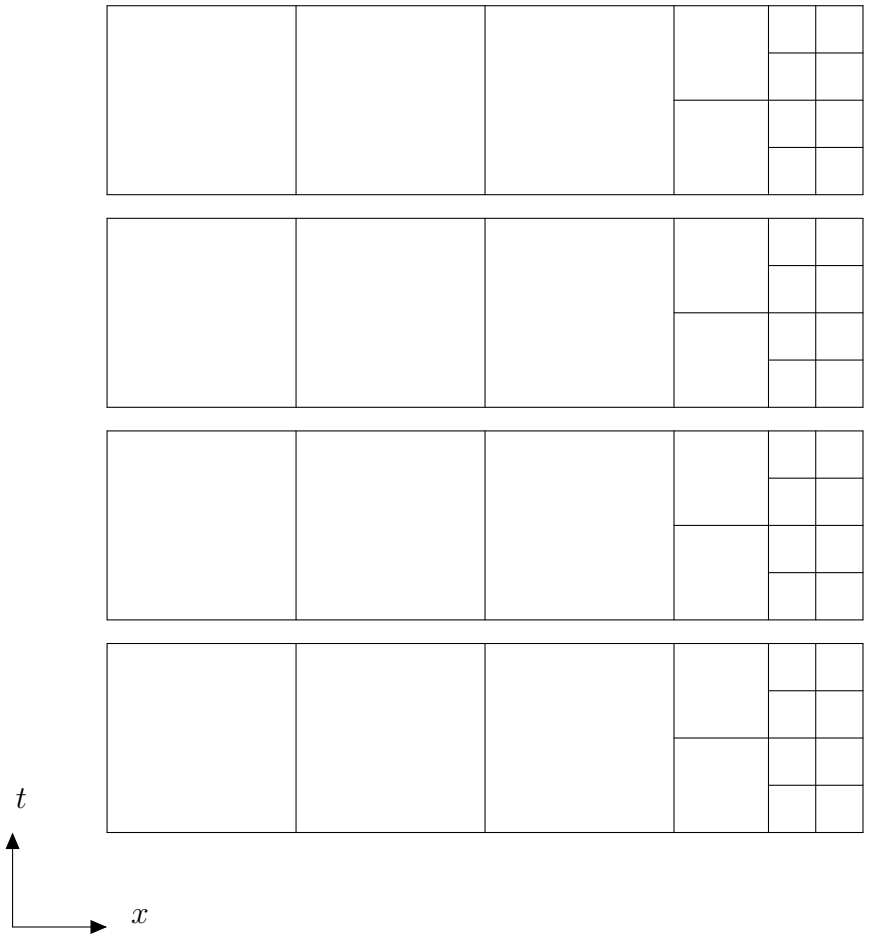


Figure D.6: Third time slab strategy

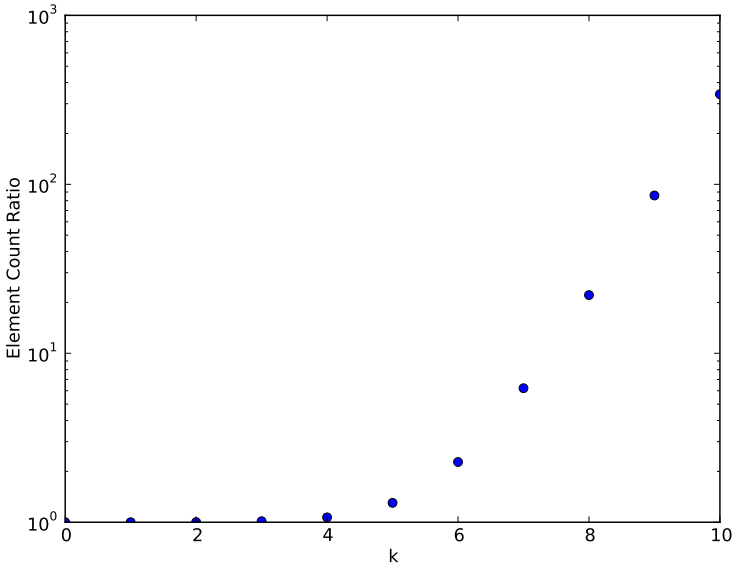


Figure D.7: Ratio of total element counts E_{tot3}/E_{tot1}

the total number of refinements to achieve the desired spatial resolution, i.e. $h = \frac{1}{2^N}$ for the smallest mesh elements. Let 2^k be the number of time slabs in approaches 2 and 3. The first strategy has a final mesh of $E_{tot1} = 2^N + \sum_{r=1}^{N} 2^r$ elements. The second approach has the same number of elements *per time slab* and is thus not an attractive alternative (at least without anisotropic refinements). The third approach has $E_{slab3} = 2^k - 1 + 2^{N-k} + \sum_{r=1}^{N-k} 2^r$ elements in each time slab, or $E_{tot3} = 2^k \cdot E_{slab3}$. Obviously, the total number of elements summed over every time slab will be higher for this approach, (as demonstrated in Figure D.7) but each individual time slab will have fewer elements than the first approach.

There are two possible reasons we might want to use approach 3 over

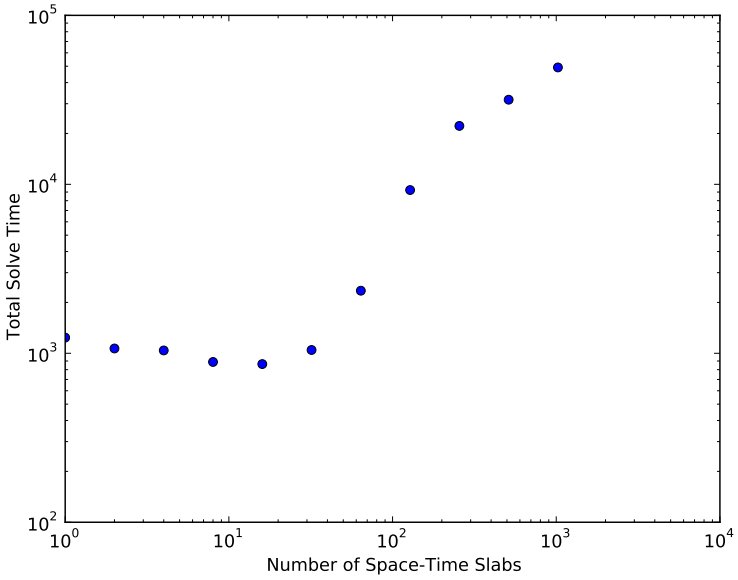


Figure D.8: Total solve time using strategy 3

approach 1. The first is speed, if the sum of the solve times for each individual time slab is less than the solve time for a single solve done with approach 1, this might be an attractive option. In fact, for this test problem, we can directly compute this for various numbers of time slabs. For the sake of comparison, the solve time is defined to be the total time to solve all time slabs while adaptively refining to a resolution of $h = 1/2^{10}$ with the default geometric multigrid settings in Camellia (discussed above). We plot these results in Figure D.8. There does appear to be a sweet spot for this problem at 16 time slabs, but the potential speedup alone isn't enough to justify the more complicated implementation.

A more compelling reason has to do with memory. It is possible that

for certain problems we might consider the solution of the entire space-time domain might require more memory than is available. By splitting the solve into smaller time slabs, you could mitigate produce smaller global solves that do fit into memory. So far, the memory constraint has not been a significant concern for the problems under consideration here, so we opted to stick with the simplest approach, the first strategy.

Bibliography

[1] R. Abedi, B. Petracovici, and R.B. Haber. A space-time discontinuous Galerkin method for linearized elastodynamics with element-wise momentum balance. *Comput. Methods in Appl. Mech. Eng.*, 195(2528):3247 – 3273, 2006.

[2] S.K. Aliabadi and T.E. Tezduyar. Space-time finite element computation of compressible flows involving moving boundaries and interfaces. *Comput. Methods in Appl. Mech. Eng.*, 107(12):209 – 223, 1993.

[3] J.H. Argyris and D.W. Scharpf. Finite elements in time and space. *Nucl. Eng. Des.*, 10(4):456 – 464, 1969.

[4] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, May 2001.

[5] I. Babuška. Error-bounds for finite element method. *Numer. Math*, 16, 1970/1971.

[6] C. Bajer and C. Bonthoux. State-of-the-art in the space-time method. *Shock Vib. Dig.*, 23:3 – 9, May 1991.

[7] H. Bijl, M.H. Carpenter, V.N. Vatsa, and C.A. Kennedy. Implicit time integration schemes for the unsteady compressible NavierStokes equations: Laminar flow. *J. Comp. Phys.*, 179(1):313–329, June 2002.

[8] P. Bochev, J. Lai, and L. Olson. A locally conservative, discontinuous least-squares finite element method for the Stokes equations. *Int. J. Numer. Methods Fluids*, 68:782–804, 2010.

[9] C. Bottasso, S. Micheletti, and R. Sacco. A multiscale formulation of the discontinuous Petrov-Galerkin method for advective-diffusive problems. *Comput. Methods in Appl. Mech. Eng.*, 194:2819–2838, 2005.

[10] J. Bramwell, L.F. Demkowicz, J. Gopalakrishnan, and W. Qiu. A locking-free *hp* DPG method for linear elasticity with symmetric stresses. *Numer. Math.*, 122(4):671–707, 2012.

[11] J. Bramwell, L.F. Demkowicz, and W. Qiu. Solution of dual–mixed elasticity equations using Arnold–Falk–Winther element and discontinuous Petrov Galerkin method. A comparison. Technical Report 23, ICES, 2010.

[12] F. Brezzi. On the existence, uniqueness, and approximation of saddle point problems arising from Lagrangian multipliers. *R.A.I.R.O., Anal. Numér.*, 2:129–151, 1974.

[13] F. Brezzi, B. Cockburn, L.D. Marini, and E. Sli. Stabilization mechanisms in discontinuous Galerkin finite element methods. *Comput. Meth-*

ods in Appl. Mech. Eng., 195(2528):3293 – 3310, 2006.

[14] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.*, pages 199–259, September 1990.

[15] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan. Breaking spaces and forms for the DPG method and applications including Maxwell equations. *Comput. Math. Appl.*, 2015. revised version under review.

[16] J. Chan, L. Demkowicz, and R. Moser. A DPG method for steady viscous compressible flow. *Comput. Fluids*, 98(0):69 – 90, 2014. 12th US-NCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics - A special issue dedicated to the 80th birthday of Professor Antony Jameson.

[17] J. Chan, N. Heuer, T. Bui-Thanh, and L. Demkowicz. A robust DPG method for convection-dominated diffusion problems II: Adjoint boundary conditions and mesh-dependent test norms. *Comp. Math. Appl.*, 67(4):771 – 795, 2014.

[18] J.L. Chan. *A DPG Method for Convection-Diffusion Problems.* PhD thesis, University of Texas at Austin, 2013.

[19] J.L. Chan, J. Gopalakrishnan, and L.F. Demkowicz. Global properties of DPG test spaces for convection-diffusion problems. Technical report,

ICES, 2013.

[20] C.L. Chang and J.J. Nelson. Least-squares finite element method for the Stokes problem with zero residual of mass conservation. *SIAM J. Num. Anal.*, 34:480–489, 1997.

[21] T.J. Chung. *Computational Fluid Dynamics*. Cambridge University Press, 1st edition, 2002.

[22] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 47(2):1319–1365, February 2009.

[23] B. Cockburn and C. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comp. Phys.*, 141(2):199 – 224, 1998.

[24] M. Costabel and A. McIntosh. On Bogovskiĭ and regularized Poincaré integral operators for de Rham complexes on Lipschitz domains. *Math. Z.*, 265(2):297–320, 2010.

[25] W. Dahmen, C. Huang, C. Schwab, and G. Welper. Adaptive Petrov-Galerkin methods for first order transport equations. *SIAM J. Numer. Anal.*, 50(5):2420–2445, 2012.

[26] L.F. Demkowicz. Babuška $\leftrightarrow$ Brezzi? Technical report, ICES, 2006.

[27] L.F. Demkowicz. Various variational formulations and closed range theorem. Technical Report 15-03, ICES, January 2015.

[28] L.F. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. Part I: The transport equation. *Comput. Methods Appl. Mech. Engrg.*, 2009.

[29] L.F. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. Part II: Optimal test functions. *Numer. Meth. Part. D. E.*, 2010.

[30] L.F. Demkowicz and J. Gopalakrishnan. Analysis of the DPG method for the Poisson equation. *SIAM J. Numer. Anal.*, 49(5):1788–1809, September 2011.

[31] L.F. Demkowicz and J. Gopalakrishnan. A primal DPG method without a first order reformulation. *Comp. Math. Appl.*, 66:1058–1064, 2013.

[32] L.F. Demkowicz and J. Gopalakrishnan. *Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations (eds. X. Feng, O. Karakashian, Y. Xing)*, volume 157, chapter An Overview of the DPG Method, pages 149–180. IMA Volumes in Mathematics and its Applications, 2014.

[33] L.F. Demkowicz and J. Gopalakrishnan. Discontinuous Petrov-Galerkin (DPG) method. Technical Report 15-20, ICES, December 2015.

[34] L.F. Demkowicz, J. Gopalakrishnan, I. Muga, and J. Zitelli. Wavenumber explicit analysis of a DPG method for the multidimensional Helmholtz equation. *Comput. Methods in Appl. Mech. Eng.*, 213216(0):126 – 138, 2012.

[35] L.F. Demkowicz, J. Gopalakrishnan, and A.H. Niemi. A class of discontinuous Petrov-Galerkin methods. Part III: Adaptivity. *Appl. Numer. Math.*, 62(4):396–427, April 2012.

[36] L.F. Demkowicz and N. Heuer. Robust DPG method for convection-dominated diffusion problems. *SIAM J. Numer. Anal.*, 51(5):1514–2537, 2013.

[37] L.F. Demkowicz and J. Li. Numerical simulations of cloaking problems using a DPG method. *Comput. Mech.*, 51(5):661–672, 2013.

[38] T.E. Ellis, L.F. Demkowicz, and J.L. Chan. Locally conservative discontinuous Petrov-Galerkin finite elements for fluid problems. *Comp. Math. Appl.*, 68(11):1530 – 1549, 2014.

[39] R.D. Falgout, S. Friedhoff, Tz.V. Kolev, S.P. MacLachlan, and J.B. Schroder. Parallel time integration with multigrid. *SIAM J. Sci. Comput.*, 36(6):C635C661, 2014.

[40] I. Fried. Finite-element analysis of time-dependent phenomena. *AIAA J.*, 7(6):1170–1173, 1969.

[41] J. Gopalakrishnan, I. Muga, and N. Olivares. Dispersive and dissipative errors in the DPG method with scaled norms for Helmholtz equation. *SIAM J. Sci. Comput.*, 36(1):A20–A39, 2014.

[42] J. Gopalakrishnan and W. Qiu. An analysis of the practical DPG method. *Math. Comp.*, 83(286):537–552, March 2014.

[43] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comp. Phys.*, 131(1):3 – 47, 1997.

[44] M.A. Heroux, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long, R.P. Pawlowski, E.T. Phipps, A.G. Salinger, H.K. Thornquist, R.S. Tuminaro, J.M. Willenbring, A. Williams, and K.S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

[45] T.J.R. Hughes, G.R. Feijo, L. Mazzei, and J.-B. Quincy. The variational multiscale method – a paradigm for computational mechanics. *Comput. Methods in Appl. Mech. Eng.*, 166(1 - 2):3 – 24, 1998.

[46] T.J.R. Hughes, L.P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Comput. Methods Appl. Mech. Engrg.*, 54:223–234, 1986.

[47] T.J.R. Hughes and G.M. Hulbert. Space-time finite element methods for elastodynamics: Formulations and error estimates. *Comput. Methods in Appl. Mech. Eng.*, 66(3):339 – 363, 1988.

[48] T.J.R. Hughes and J.R. Stewart. A space-time formulation for multiscale phenomena. *J. Comput. Appl. Math.*, 74(12):217 – 229, 1996.

[49] Z. Kaczkowski. The method of finite space-time elements in dynamics of structures. *J. Tech. Phys.*, 16(1):69 – 84, 1975.

[50] C.M. Klaij, J.J.W. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comp. Phys.*, 217(2):589 – 611, 2006.

[51] Lawrence Livermore National Laboratory. XBraid: Parallel Time Integration with Multigrid, 2016. `http://computation.llnl.gov/projects/parallel-time-integration-multigrid`.

[52] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput. Methods in Appl. Mech. Eng.*, 134(1 - 2):71 – 90, 1996.

[53] X. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comp. Phys.*, 115(1):200 – 212, 1994.

[54] D. Moro, N.C. Nguyen, and J. Peraire. A hybridized discontinuous Petrov-Galerkin scheme for scalar conservation laws. *Int. J. Num. Meth. Eng.*, 2011.

[55] A.H. Niemi, J.A. Bramwell, and L.F. Demkowicz. Discontinuous Petrov-Galerkin method with optimal test functions for thin-body problems in solid mechanics. *Comput. Methods in Appl. Mech. Eng.*, 200(9-12):1291–1300, February 2011.

[56] A.H. Niemi, N.O. Collier, and V.M. Calo. Automatically stable discontinuous Petrov-Galerkin methods for stationary transport problems: Quasi-optimal test space norm. *Comput. Math. Appl.*, 66(10):2096–2113, December 2013.

[57] A.H. Niemi, N.O. Collier, and V.M. Calo. Discontinuous Petrov-Galerkin method based on the optimal test space norm for steady transport problems in one space dimension. *J. Comput. Sci.*, 4(3):157–163, 2013.

[58] W.F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *J. Comp. Phys.*, 72(1):78 – 120, 1987.

[59] J.T. Oden. A general theory of finite elements. II. Applications. *Int. J. Numer. Meth. Eng.*, 1(3):247–259, 1969.

[60] J.B. Perot. Discrete conservation properties of unstructured mesh schemes. *Annu. Rev. Fluid Mech.*, 43:299–318, 2011.

[61] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos National Laboratory, 1973.

[62] S. Rhebergen and B. Cockburn. A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. *J. Comp. Phys.*, 231(11):4185 – 4204, 2012.

[63] S. Rhebergen, B. Cockburn, and J.J.W. Van Der Vegt. A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 233:339–358, January 2013.

[64] N. Roberts, T. Bui-Thanh, and L. Demkowicz. The DPG method for the Stokes problem. *Comp. Math. Appl.*, 67(4):966 – 995, 2014.

[65] N.V. Roberts. *A Discontinuous Petrov-Galerkin Methodology for Incompressible Flow Problems.* PhD thesis, University of Texas at Austin, 2013.

[66] N.V. Roberts. Camellia: A software framework for discontinuous Petrov-Galerkin methods. *Comp. Math. Appl.*, 68(11):1581 – 1604, 2014.

[67] N.V. Roberts, L.F. Demkowicz, and R.D. Moser. A discontinuous Petrov-Galerkin methodology for adaptive solutions to the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 301:456 – 483, 2015.

[68] N.V. Roberts, T.E. Ellis, and J.L. Chan. Camellia: A Software Toolbox for Discountinuous Petrov-Galerkin (DPG) Methods, 2016. `https://github.com/CamelliaDPG/Camellia`.

[69] H. G. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations*, volume 24 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2nd edition, 2008.

[70] A. Safjan, L. Demkowicz, and J.T. Oden. Adaptive finite element methods for hyperbolic systems with application to transient acoustics. *Int. J. Numer. Meth. Eng.*, 32:677–707, September 1991.

[71] G.A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comp. Phys.*, 27(1):1 – 31, 1978.

[72] T.E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests. *Comput. Methods in Appl. Mech. Eng.*, 94(3):339 – 351, 1992.

[73] A. Ün̈gör. Tent-Pitcher: A meshing algorithm for space-time discontinuous Galerkin methods. pages 111–122. 9th Internat. Meshing Roundtable, 2000.

[74] J.J.W. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. General formulation. *J. Comp. Phys.*, 182(2):546 – 585, 2002.

[75] Ch. Wieners and B. Wohlmuth. Robust operator estimates. Technical report, Oberwolfach Reports, 2013.

[76] H. Zhu, H. Shu, and M. Ding. Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method. *Comp. Math. Appl.*, 60(3):840–848, August 2010.

[77] J. Zitelli, I. Muga, L.F. Demkowicz, J. Gopalakrishnan, D. Pardo, and V. Calo. A class of discontinuous Petrov-Galerkin methods. Part IV: Wave propagation problems. *J. Comp. Phys.*, 230:2406–2432, 2011.

Vita

Truman Ellis received Bachelor of Science and Master of Science degrees from California Polytechnic State University in 2010. In the fall of 2010 he began a doctoral program in Computational Science, Engineering, and Mathematics at the University of Texas at Austin under the supervision of Drs. Leszek Demkowicz and Robert Moser. During his graduate career, he completed four summers of research at Lawrence Livermore National Laboratory under the supervision of Drs. Tzanio Kolev and Robert Rieben developing a high order curvilinear finite element solver for shock hydrodynamics. Upon completion of his doctoral degree, he will work as a postdoctoral researcher at Sandia National Laboratory.

Email: truman.e.ellis@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.