A Space-Time DPG Method for Fluid Dynamics

by

Truman E. Ellis, M.S.

DISSERTATION PROPOSAL

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2014

Table of Contents

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 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. The stability characteristics of a broad

class of finite element methods can be analyzed according to the Ladyženskaja-Babuška-Brezzi condition. Leszek Demkowicz and Jay Gopalakrishnan first proposed the discontinuous Petrov-Galerkin method in 2009[18] in order to address stability issues for a very broad class of problems. The DPG method automatically satisfies stability criteria by construction which 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 order 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 streamwise upwind Petrov-Galerkin method while generalizing and improving it. The DPG method itself can be viewed as a generalization of least-squares finite elements or even of mixed methods.

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 DPG. 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.

1.2 Literature Review

We start this literature review by looking at various numerical methods that have been popular in the simulation of fluid mechanical 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. 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.

The presence of shocks in compressible Navier-Stokes simulations present 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. 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[34] 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[28]. 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 problems, and more general formulations had to be explored. Some of the early pioneers of finite elements for CFD include Oden, Zienkiewicz, Karniadakis, and Hughes[14].

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 (possible a function). The classical example of this technique is streamline upwind Petrov-Galerkin (SUPG) method for convection-diffusion using piecewise linear continuous finite elements[10]. 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 $b_{SUPG}(u, v) = l_{SUPG}$ 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 optimal convergence and stabilization 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[29]. 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[39]. 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 [3]. Of particular interest to our work in CFD is the paper by Cockburn and Shu[16] on DG for conservation laws. 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 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 piecewise discontinuous nature of DG allows for very simple h and p adaptivity and straightforward parallelization.

Like 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[9]. 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 dual 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[9].

The DG method has 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. It is 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 unknown 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. The other 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.

Hybridized Discontinuous Galerkin Methods The hybridized discontinuous Galerkin method was first introduced by Cockburn, Gopalakrishnan, and Lazarov[15] 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 coupling 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[30] 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 time domain. Some of the earliest proponents of this approach were Argyris and Scharpf[2], Fried[27], and Oden[37]. These techniques were built on the underlying concept of Hamilton's principle.

Space-time finite elements present an attractive way to handle meshes with moving boundaries. Lesoinne and Farhat[33] 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[47] 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 spacetime 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.* [32] then extended the method to compressible Navier-Stokes while Rhebergen *et al.* [40] developed the method for incompressible Navier-Stokes. Rhebergen and Cockburn[41] also developed a space-time HDG method for incompressible Navier-Stokes.

Tezduyar and Behr[45] 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[1] apply the procedure to compressible flows. Hughes and Stewart[31] develop a general space-time multiscale framework for deriving stabilized methods for transient phenomena.

The tent-pitcher algorithm of Üngör[46] 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.

1.2.3 DPG

The discontinuous Petrov-Galerkin finite element method was first proposed by Demkowicz and Gopalakrishnan in 2009[18, 19]. 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

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

with operator $B : U \to V'$ (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':

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

This 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 let's us represent our residual in V:

$$u_h = \underset{w_h \in U_h}{\arg\min} \frac{1}{2} \left\|R_V^{-1}(Bu - l)\right\|^2_{V'} .$$

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 solve 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* or *dual Petrov-Galerkin* method. 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 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[4] 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 conditions. 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.* [35] 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 ever 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 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 norms that are 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.

1.3 Goal

1.3.1 Organization of this Proposal

Chapter 2

Conservation in steady-state

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[38] 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.

There are also some mathematically attractive reasons to pursue local conservation. The Lax-Wendroff theorem guarantees that a convergent numerical solution to a system of hyperbolic conservation laws will converge to the correct 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 least squares methods seek to minimize the residual of the solution in the L^2 norm, while 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.* [6] accomplish local conservation by using a pointwise divergence free velocity space in the Stokes formulation. Chang and Nelson[13] developed the *restricted LSFEM*[13] 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 similar negative of transforming a minimization method to a saddle point problem. In the interest of crediting Chang and Nelson's restricted LSFEM, we call the following locally conservative DPG method the restricted DPG method (RDPG).

2.1 DPG is a minimum residual method

Roberts *et al.* presents a brief history and derivation of DPG with optimal test functions in [42]. We follow his derivation of the standard DPG method as a minimum residual method. Let U be the trial Hilbert space and V the test Hilbert space for a well-posed variational problem $b(u,v) = l(v)$. In operator form this is $Bu = l$, where $B : U \to V'$ and $\langle Bu, v\rangle = b(u,v)$. We seek to minimize the residual for the discrete space $U_h \subset U$:

$$u_h = \underset{u_h \in U_h}{\arg\min} \frac{1}{2} \|Bu_h - l\|_{V'}^2 \ . \tag{2.1}$$

Recalling that the Riesz operator $R_V : V \to V'$ is an isometry defined by

$$\langle R_V v, \delta v\rangle = (v, \delta v)_V \ , \quad \forall \delta v \in V,$$

we can use the Riesz inverse to minimize in the V-norm rather than its dual:

$$\frac{1}{2} \|Bu_h - l\|_{V'}^2 = \frac{1}{2} \left\|R_V^{-1}(Bu_h - l)\right\|_V^2 = \frac{1}{2} \left(R_V^{-1}(Bu_h - l), R_V^{-1}(Bu_h - l)\right)_V \ . \tag{2.2}$$

The first order optimality condition for (2.2) requires the Gâteaux derivative to be zero in all directions $\delta u \in U_h$, i.e.,

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

By definition of the Riesz operator, this is equivalent to

$$\left\langle Bu_h - l, R_V^{-1}B\delta u_h \right\rangle = 0 \quad \forall \delta u_h \in U_h \,. \tag{2.3}$$

Now, we can identify $v_{\delta u_h} \coloneqq R_V^{-1}B\delta u_h$ as the optimal test function for trial function δu_h. Define $T \coloneqq R_V^{-1}B : U_h \to V$ as the trial-to-test operator. Now we can rewrite (2.3) as

$$b(u_h, v_{\delta u_h}) = l(v_{\delta u_h}). \tag{2.4}$$

The DPG method then is to solve (2.4) with optimal test functions $v_{\delta u_h} \in V$ that solve the auxiliary problem

$$(v_{\delta u_h}, \delta v)_V = \langle R_V v_{\delta u_h}, \delta v \rangle = \langle B\delta u_h, \delta v \rangle = b(\delta u_h, \delta v) \quad \forall \delta v \in V. \tag{2.5}$$

Using a continuous test basis would result in a global solve for every optimal test function. Therefore DPG uses a discontinuous test basis which makes each solve element-local and much more computationally tractable. Of course, (2.5) still requires the inversion of the infinite-dimensional Riesz map, but approximating V by a finite dimensional space, V_h, which is of a higher polynomial degree than U_h (hence "enriched space") works well in practice.

No assumptions have been made so far on the definition of the inner product on V. In fact, proper choice of $(\cdot,\cdot)_V$ can make the difference between a solid DPG method and one that suffers from robustness issues.

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 *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:

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

which are developed more precisely in [42].

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

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

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:

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

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:

$$\begin{aligned} -(\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v)_K + \langle (\boldsymbol{\beta} u - \boldsymbol{\sigma}) \cdot \mathbf{n}, v \rangle_{\partial K} &= (f, v)_K \\ \frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau})_K + (u, \nabla \cdot \boldsymbol{\tau})_K - \langle u, \tau_n \rangle_{\partial K} &= 0\,. \end{aligned} \tag{2.6}$$

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 [20], linear elasticity [7], linear acoustics [23], Stokes [42]) 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[22] 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 (2.6) 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

$$\underbrace{-(\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v)_K + \left\langle \hat{t}, v \right\rangle_{\partial K} + \frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau})_K + (u, \nabla \cdot \boldsymbol{\tau})_K - \langle \hat{u}, \tau_n \rangle_{\partial K}}_{b(\mathbf{u},\mathbf{v})} = \underbrace{(f, v)_K}_{l(\mathbf{v})} \qquad \text{in } \Omega \tag{2.7}$$

$$\hat{u} = 0 \qquad \text{on } \Gamma \tag{2.8}$$

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

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 [24]: the ultra-weak variational formulation for convection-diffusion can be refactored as

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

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

$$\begin{aligned} \div \boldsymbol{\tau} - \boldsymbol{\beta} \cdot \nabla v &= u, \\ \frac{1}{\epsilon}\boldsymbol{\tau} + \nabla v &= \boldsymbol{\sigma}, \end{aligned}$$

then evaluating $b\left(\left(u, \boldsymbol{\sigma}, \widehat{u}, \widehat{f_n}\right), (\boldsymbol{\tau}, v)\right)$ 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 $\|v\|_V$, we have

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

where

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

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

$$\left(\|u\|^2_{L^2(\Omega)} + \|\boldsymbol{\sigma}\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}} \lesssim \left\|u, \boldsymbol{\sigma}, \widehat{u}, \widehat{f_n}\right\|_E . \tag{2.9}$$

In other words, the energy norm in which DPG is optimal bounds independently of ϵ the L^2 norm; 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 $\|v\|_V$ for a robust DPG method depends on two things: the test norm, as well as the adjoint equation. In [24], the standard problem with Dirichlet conditions enforced over the entire boundary was considered; in [11], 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 [11] for numerical experiments here, which is motivated and explained in more detail in

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:

$$L(\boldsymbol{u}_h, \lambda_k) = \frac{1}{2}\left\|R_V^{-1}(b(\boldsymbol{u}_h, \cdot) - (f, \cdot))\right\|^2_{\boldsymbol{V}} - \sum_K \lambda_K(b(\boldsymbol{u}_h, (1_K, \boldsymbol{0})) - l((1_K, \boldsymbol{0}))) , \tag{2.10}$$

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:

$$\begin{cases} b(\boldsymbol{u}_h, T(\delta\boldsymbol{u}_h)) - \sum_K \lambda_K b(\boldsymbol{u}_h, (1_K, \boldsymbol{0})) & = l(T(\delta\boldsymbol{u}_h)) \quad \forall \delta\boldsymbol{u}_h \in \boldsymbol{U}_h \\ b(\boldsymbol{u}_h, (1_K, \boldsymbol{0})) & = l((1_K, \boldsymbol{0})) \quad \forall K\,, \end{cases} \tag{2.11}$$

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 $T(\delta\boldsymbol{u}_h) = (v_{\delta\boldsymbol{u}_h}, \boldsymbol{\tau}_{\delta\boldsymbol{u}_h}) \in H^1(\Omega_h) \times \boldsymbol{H}(div, \Omega_h)$. Then, putting (2.11) into more concrete terms for convection-diffusion, we get:

$$\begin{cases} -(\boldsymbol{\beta} u - \boldsymbol{\sigma}, \nabla v_{\delta\boldsymbol{u}_h}) + \langle \hat{t}, v_{\delta\boldsymbol{u}_h} \rangle + \frac{1}{\epsilon}(\boldsymbol{\sigma}, \boldsymbol{\tau}_{\delta\boldsymbol{u}_h}) + (u, \nabla \cdot \boldsymbol{\tau}_{\delta\boldsymbol{u}_h}) - \langle \hat{u}, \boldsymbol{\tau}_{\delta\boldsymbol{u}_h} \cdot \boldsymbol{n} \rangle & \\ \qquad - \sum_K \lambda_K (\delta\hat{t}, (1_K, \boldsymbol{0})) & = (f, v_{\delta\boldsymbol{u}_h}) \quad \forall \delta\boldsymbol{u}_h \in \boldsymbol{U}_h \\ \langle \hat{t}, (1_K, \boldsymbol{0}) \rangle & = (f, 1_K) \quad \forall K\,. \end{cases} \tag{2.12}$$

2.2.2 Stability analysis

In the following analysis, we neglect the error due to the approximation of optimal test functions. We follow the classical Brezzi's theory [8, 25] for an abstract mixed problem:

$$\begin{cases} \boldsymbol{u} \in \boldsymbol{U}, p \in Q & \\ a(\boldsymbol{u}, \boldsymbol{w}) + c(p, \boldsymbol{w}) & = l(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \boldsymbol{U} \\ c(q, \boldsymbol{u}) & = g(q) \quad \forall q \in Q \end{cases} \tag{2.13}$$

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 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:

$$\begin{aligned}
|c(\textstyle\sum_K \lambda_K(1_K,\boldsymbol{0}),(u,\boldsymbol{\sigma},\hat{u},\hat{t}))| &= |\textstyle\sum_K \lambda_K \langle \hat{t}, 1_K\rangle_{\partial K}| \\
&= |\textstyle\sum_K \lambda_K \langle v_n, 1_K\rangle_{\partial K}| \\
&= |\textstyle\sum_K \lambda_K \int_K \mathrm{div}\boldsymbol{\psi}\, 1_K| \\
&\leq \textstyle\sum_K \lambda_K ||\mathrm{div}\boldsymbol{\psi}||_{L^2(K)} \mu(K)^{1/2} \\
&\leq (\textstyle\sum_K \mu(K)\lambda_K^2)^{1/2}\, (\textstyle\sum_K ||\mathrm{div}\boldsymbol{\psi}||^2_{L^2(K)})^{1/2} \\
&\leq \underbrace{\left(\sum_K \mu(K)\lambda_K^2\right)^{1/2}}_{=:||\boldsymbol{\lambda}||} ||\hat{t}||_{H^{-1/2}(\Gamma_h)}\,,
\end{aligned} \tag{2.14}$$

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:

$$\sup_{\boldsymbol{w}\in\boldsymbol{U}} \frac{|c(p,\boldsymbol{w})|}{||\boldsymbol{w}||_{\boldsymbol{U}}} \geq \beta ||p||_Q\,. \tag{2.15}$$

Let

$$R\,:\, L^2(\Omega) \ni q \rightarrow \boldsymbol{\psi} \in \boldsymbol{H}(\mathrm{div},\Omega)\cap \boldsymbol{H}^1(\Omega) = \boldsymbol{H}^1(\Omega) \tag{2.16}$$

be the continuous right inverse of the divergence operator constructed by Costabel and McIntosh in [17]. Let $\boldsymbol{\psi}_h$ denote the classical, lowest order Raviart-Thomas (RT) interpolant of function

$$\boldsymbol{\psi} = R(\sum_K \lambda_K 1_K)\,. \tag{2.17}$$

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:

$$\begin{aligned} ||\boldsymbol{\psi}_h|| &\leq ||\boldsymbol{\psi}_h - \boldsymbol{\psi}|| + ||\boldsymbol{\psi}|| \\ &\leq Ch||\boldsymbol{\psi}||_{H^1} + ||\boldsymbol{\psi}|| \\ &\leq C||\mathrm{div}\boldsymbol{\psi}|| = C(\textstyle\sum_K \mu(K)\lambda_K^2)^{1/2}\,. \end{aligned} \tag{2.18}$$

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

$$\sup_{\hat{t}\in H^{-1/2}(\Gamma_h)} \frac{|\sum_K \lambda_K \langle \hat{t}, 1_K\rangle_{\partial K}|}{||\hat{t}||_{H^{-1/2}(\Gamma_h)}} \geq \frac{|\sum_K \lambda_K \int_K \mathrm{div}\boldsymbol{\psi}_h\, 1_K|}{||\boldsymbol{\psi}_h||_{H(\mathrm{div},\Omega)}} \geq \frac{1}{C}(\sum_K \mu(K)\lambda_K^2)^{1/2}\,, \tag{2.19}$$

where C is the constant from stability estimate (2.18).

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:

$$\boldsymbol{U}_0 := \quad \{\boldsymbol{w} \in \boldsymbol{U} \,:\, c(q,\boldsymbol{w}) = 0 \quad \forall q \in Q\} \tag{2.20}$$

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

$$\sup_{\boldsymbol{w}\in\boldsymbol{U}_0} \frac{|a(\boldsymbol{u},\boldsymbol{w})|}{||\boldsymbol{w}||_{\boldsymbol{U}}} \geq \frac{|b(\boldsymbol{u},T\boldsymbol{u})|}{||\boldsymbol{u}||} = \frac{|b(\boldsymbol{u},T\boldsymbol{u})|}{||T\boldsymbol{u}||}\frac{||T\boldsymbol{u}||}{||\boldsymbol{u}||} = \sup_{(v,\boldsymbol{\tau})} \frac{|b((u,\boldsymbol{\sigma},\hat{u},\hat{t}),(v,\boldsymbol{\tau}))|}{||(v,\boldsymbol{\tau})||}\frac{||T\boldsymbol{u}||}{||\boldsymbol{u}||} \geq \gamma^2||(u,\boldsymbol{\sigma},\hat{u},\hat{t})||\,, \tag{2.21}$$

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 $||T\boldsymbol{u}||_V = ||R_V^{-1}B\boldsymbol{u}||_V = ||B\boldsymbol{u}||_{V'} \geq \gamma||u||$.

With both discrete inf-sup constants in place, we have the standard result: the FE error is bounded by the best approximation error. Notice that the exact Lagrange multipliers are zero, so the best approximation error involves only 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, (2.9). With robust test norms, we have

$$\begin{aligned} \left(||u-u_h||^2 + ||\boldsymbol{\sigma}-\boldsymbol{\sigma}_h||^2\right)^{\frac{1}{2}} &\lesssim ||(u-u_h, \boldsymbol{\sigma}-\boldsymbol{\sigma}_h, \hat{u}-\hat{u}_h, \hat{t}-\hat{t}_h||_E \\ &= \inf_{(w_h,\boldsymbol{\varsigma}_h,\hat{w},\hat{r}_h)} ||(u-w_h, \boldsymbol{\sigma}-\boldsymbol{\varsigma}_h, \hat{u}-\hat{w}_h, \hat{t}-\hat{r}_h||_E \,. \end{aligned} \tag{2.22}$$

The last equality follows from the fact that DPG method delivers the best approximation error in the energy norm (minimizes the residual). This is no longer true for the restricted version. So, can we claim robustness in the sense of the inequality above for the restricted 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 flux $\hat{t}$ is. Given an element K, we solve for the optimal test functions corresponding to flux $\hat{t}$,

$$\begin{cases} v_K \in H^1(K),\, \boldsymbol{\tau}_K \in \boldsymbol{H}(\mathrm{div}, K) \\ ((v_K, \boldsymbol{\tau}_K), (\delta v, \delta\boldsymbol{\tau}))_V = \langle \hat{t}, \delta v\rangle_{\partial K} \quad \forall \delta v \in H^1(K), \delta\boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, K)\,. \end{cases} \tag{2.23}$$

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

$$||\hat{t}||_E^2 = \sum_K ||(v_K, \boldsymbol{\tau}_K)||_V^2 \,. \tag{2.24}$$

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,

$$\sup_{\hat{t}} \frac{|\sum_K \lambda_K \langle \hat{t}, 1_K \rangle|}{||\hat{t}||_E} \geq \beta \left(\sum_K \mu(K) \lambda_K^2 \right)^{1/2} . \tag{2.25}$$

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 norm of $\hat{t}_h$. For this, we need to solve the local problems:

$$\begin{aligned} ((v, \boldsymbol{\tau}), (\delta v, \delta \boldsymbol{\tau}))_V = \langle \hat{t}, \delta v \rangle_{\partial K} = \int_K \mathrm{div} \boldsymbol{\psi}_h \, \delta v = \int_K \mathrm{div} \boldsymbol{\psi} \, \delta v \\ = \int_K \lambda_K \delta v = \lambda_K (1_K, \delta v)_K \quad \forall \delta v \in H^1(K) \, \forall \delta \boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, K) \,. \end{aligned} \tag{2.26}$$

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

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

Substituting $(v, \boldsymbol{\tau})$ for $(\delta v, \delta \boldsymbol{\tau})$ in (2.26), we get,

$$||(v, \boldsymbol{\tau})||_V^2 = \lambda_K (1_K, v_K) \,. \tag{2.27}$$

If we have a robust stability estimate:

$$(1_K, v_K) \leq C \mu(K)^{1/2} ||(v, \boldsymbol{\tau})||_K \tag{2.28}$$

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

$$||(v,\boldsymbol{\tau})||_V \leq C\mu(K)^{1/2}|\lambda_K| \tag{2.29}$$

and, eventually as needed,

$$\sum_K ||(v,\boldsymbol{\tau})||_V^2 \leq C^2 \sum_K \mu(K)\lambda_K^2\,, \tag{2.30}$$

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

$$||v||_{L^2(K)} \leq ||(v,\boldsymbol{\tau})||_V\,. \tag{2.31}$$

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 (2.26) and it coincides with the divergence of $\boldsymbol{\psi}$.

We arrive at a similar situation in the continuity estimate of

$$\sum_K \lambda_K \langle \hat{t}, 1_K\rangle\,.$$

Testing with $(1_K, \mathbf{0})$ in the local problem (2.23), we obtain,

$$((v,\boldsymbol{\tau}),(1_K,\mathbf{0}))_V = \langle \hat{t}, 1_K\rangle_{\partial K}\,. \tag{2.32}$$

If we have a robust estimate,

$$|((v,\boldsymbol{\tau}),(1_K,\mathbf{0}))_V| \leq C\mu(K)^{1/2}\,||(v,\boldsymbol{\tau})||_V\,, \tag{2.33}$$

then

$$|\sum_K \lambda_K \langle \hat{t}, 1_K\rangle| \leq C(\sum_K \mu(K)\lambda_K^2)^{1/2}\,(\sum_K ||(v,\boldsymbol{\tau})||_V^2)^{1/2} = C(\sum_K \mu(K)\lambda_K^2)^{1/2}||\hat{t}||_E\,, \tag{2.34}$$

as needed.

For instance, condition (2.33) will be satisfied if the test inner product in (2.32) reduces to the L^2 term only,

$$((v,\boldsymbol{\tau}),(1_K,\mathbf{0}))_V = (v,1_K)_{L^2(K)}\,. \tag{2.35}$$

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 [21] 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.* [11] derived a more robust alternative norm for convection diffusion (dubbed the robust norm). We recently developed a modification of the robust norm that appears to produce better results in the presence of singularities; for full details see Jesse Chan's dissertation[12].

$$\|(v,\boldsymbol{\tau})\|^2_{V,K} \coloneqq \min\left\{\frac{1}{\epsilon},\frac{1}{\mu(K)}\right\}\|\boldsymbol{\tau}\|^2_K\,, + \|\div\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v\|^2_K \tag{2.36}$$

$$+ \|\boldsymbol{\beta}\cdot\nabla v\|^2_K + \epsilon\,\|\nabla v\|^2_K + \|v\|^2_K\,, \tag{2.37}$$

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:

$$\min\left\{\frac{1}{\epsilon}, \frac{1}{\mu(K)}\right\}(\boldsymbol{\tau}_{\delta \boldsymbol{u}_h}, \delta\boldsymbol{\tau})_K + (\nabla \cdot \boldsymbol{\tau}_{\delta \boldsymbol{u}_h} - \boldsymbol{\beta}\cdot\nabla v, \nabla\cdot\delta\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v)_K + (\boldsymbol{\beta}\cdot\nabla v_{\delta\boldsymbol{u}_h}, \boldsymbol{\beta}\cdot\nabla\delta v)_K$$
$$+ \epsilon(\nabla v_{\delta\boldsymbol{u}_h}, \nabla\delta v)_K + \alpha(v_{\delta\boldsymbol{u}_h}, \delta v)_K = b(\delta\boldsymbol{u}_h, (\delta v, \delta\boldsymbol{\tau})) \qquad \forall \delta v \in H^1(K),\ \delta\boldsymbol{\tau} \in \boldsymbol{H}(\mathrm{div}, K)\,, \tag{2.38}$$

where, typically, $\alpha = 1$.

With a locally conservative formulation, we can pass in local problem (2.38) with $\alpha \to 0$. The fact that the test functions will be determined then up to a constant does not matter, for $\delta\hat{t} \in \hat{F}_h^e$, equation $(2.12)_1$ 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,

$$\min\left\{\frac{1}{\epsilon}, \frac{1}{\mu(K)}\right\}(\boldsymbol{\tau}_{\delta \boldsymbol{u}_h}, \delta\boldsymbol{\tau})_K + (\nabla \cdot \boldsymbol{\tau}_{\delta \boldsymbol{u}_h} - \boldsymbol{\beta}\cdot\nabla v, \nabla\cdot\delta\boldsymbol{\tau} - \boldsymbol{\beta}\cdot\nabla v)_K \tag{2.39}$$
$$+ (\boldsymbol{\beta}\cdot\nabla v_{\delta\boldsymbol{u}_h}, \boldsymbol{\beta}\cdot\nabla\delta v)_K + \epsilon(\nabla v_{\delta\boldsymbol{u}_h}, \nabla\delta v)_K + \frac{1}{\mu(K)}\int_K v_{\delta\boldsymbol{u}_h}\int_K \delta v\,,$$

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 more the analysis in the next section. It is convenient to be able to take $\alpha \to 0$ as we will see in some later numerical experiments.

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:

$$(1_K, v_K) \leq C\mu(K)^{1/2}||(v, \boldsymbol{\tau})||_K \tag{2.28 revisted}$$

and

$$|((v, \boldsymbol{\tau}), (1_K, \mathbf{0}))_V| \leq C\mu(K)^{1/2}\,||(v, \boldsymbol{\tau})||_V\,. \tag{2.33 revisted}$$

then the restricted DPG method is robust.

We now proceed to show that the robust norms we are using satisfy this requirement. Consider the inner product from (2.38), with $\alpha = 1$. We wish to verify condition (2.28) with the norm derived from this inner product on the right hand side. By Cauchy-Schwarz

$$\int_K v \cdot 1 \leq \mu(K)^{1/2} \|v\|_{L^2(K)} \leq \mu(K)^{1/2} \|(v, \boldsymbol{\tau})\|_K \ , \tag{2.40}$$

where $\|(v, \boldsymbol{\tau})\|_K$ is the norm derived from the inner product. Condition (2.33) comes out the same since

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

element-wise.

Now we wish the perform the same analysis for the modified inner product in (2.39). In this case, condition (2.28) follows even more naturally as

$$\int_K v \cdot 1 \leq \mu(K)^{1/2} \frac{1}{\mu(K)^{1/2}} \left| \int_K v \right| \leq \|(v, \boldsymbol{\tau})\|_K \ , \tag{2.41}$$

where $\|(v, \boldsymbol{\tau})\|$ now refers to the norm generated by inner product (2.39). Condition (2.33) follows by the same reasoning.

2.3 Numerical Experiments

We now demonstrate the effectiveness of the restricted DPG method with some numerical experiments. Flux imbalance is measured 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.

We measure mass loss much more directly in the Stokes problem. Because 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

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

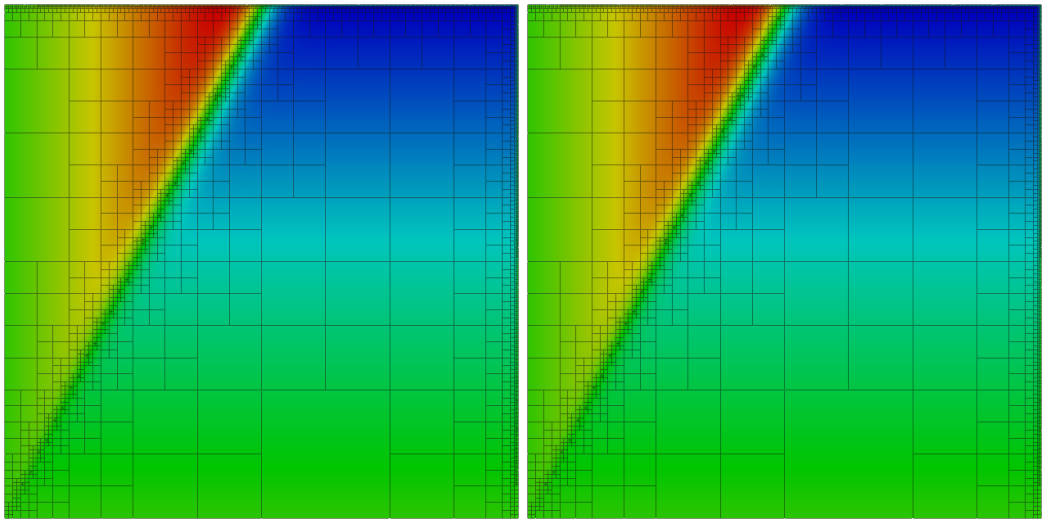
where S is some vertical mesh line.

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 $\boldsymbol{\tau}$).

2.3.1 Convection-Diffusion

Extensive results for convection-diffusion can be found in [26], but we select one representative example for brevity. 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. The colorbar in Figure 2.1 is scaled to $[-1.110, 0.889]$ where red indicates the maximum value and blue the minimum.

We can see that the converged solutions look nearly identical whether or not we enforce local conservation. From the flux imbalance plots, we can see that standard DPG starts with somewhat noticeable flux imbalance, but it converges to a conservative solution with refinement.



(a) Nonconservative (b) Conservative

Figure 2.1: Discontinuous source problem after 8 refinements

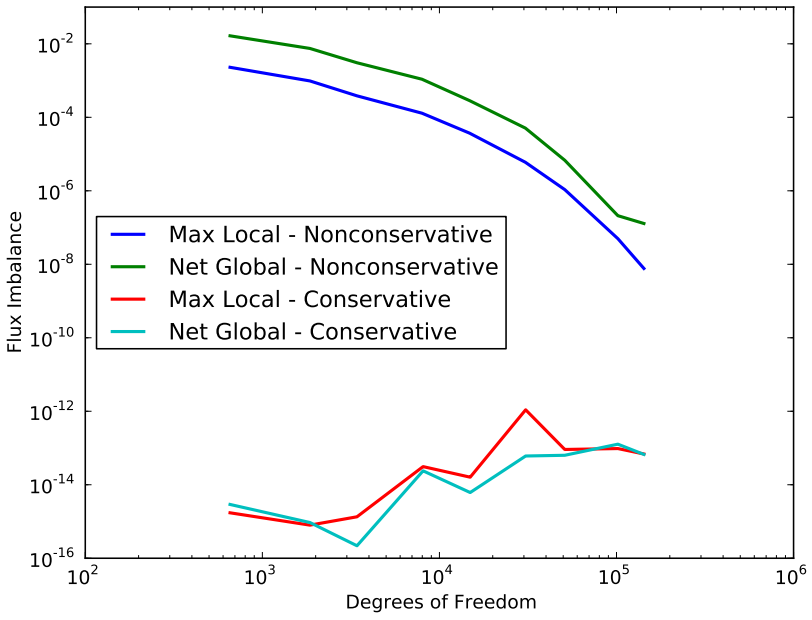


Figure 2.2: Flux imbalance in discontinuous source solutions

Restricted DPG flux loss bounces around near machine precision as expected. It appears to grow slightly with number of degrees of freedom, but that is just accumulation of numerical error.

2.3.2 Inviscid Burgers' Equation

Extending local conservation to other fluid model problems is very straight-forward. We only need to identify what terms need to be conserved and constrain them via Lagrange multipliers. 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

$$\nabla_{xt} \cdot \begin{pmatrix} u^2/2 \\ u \end{pmatrix} = 0\,.$$

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

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

where $\hat{t} := \mathrm{tr}\left(\begin{pmatrix} u^2/2 \\ u \end{pmatrix}\right) \cdot \mathbf{n}_{xt}$ 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. For the locally conservative version, we need $\int_{\partial K} \hat{t} = \int_K f$ for all elements K in the mesh.

As a numerical test, we try a standard Burgers' shock problem. 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.

Examining the results, we see the exact same kind of results that we saw for convection-diffusion, indicating that hyperbolic nonlinear conservation laws present no problem to the method.

2.3.3 Stokes Flow

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

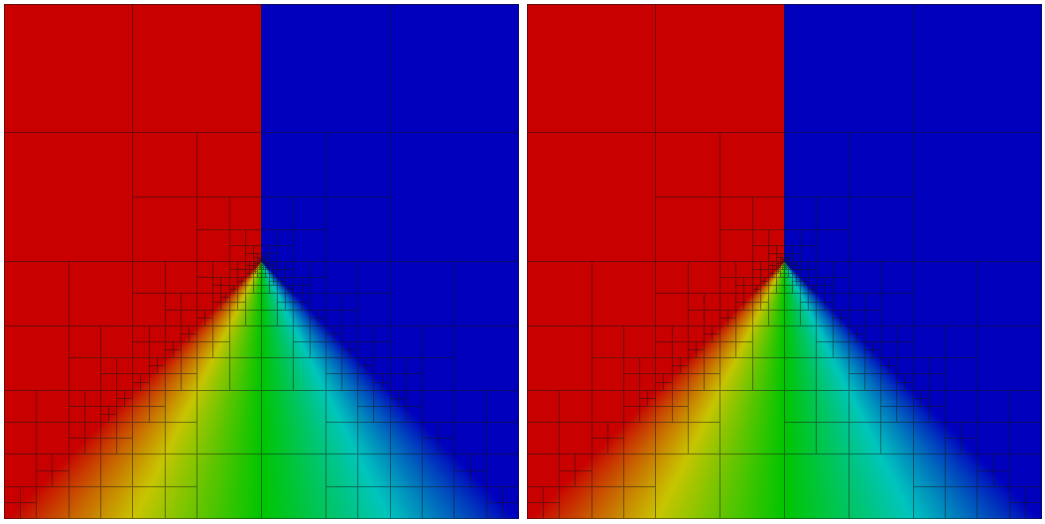
$$\begin{aligned} \mu \Delta \boldsymbol{u} + \nabla p &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \,, \end{aligned}$$

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:

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



(a) Nonconservative (b) Conservative

Figure 2.3: Burgers' problem after 8 refinements

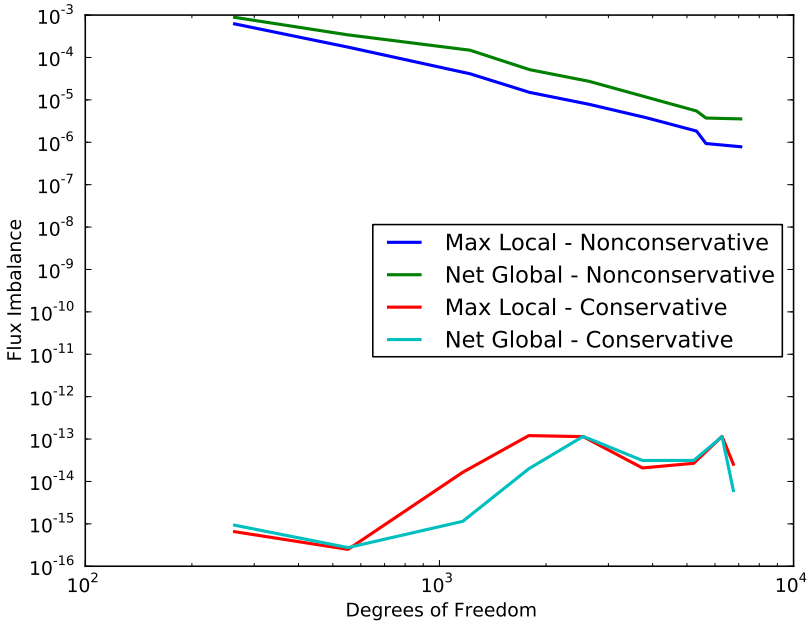


Figure 2.4: Flux imbalance in Burgers' solutions

where $\hat{\boldsymbol{u}} := \mathrm{tr}(\boldsymbol{u})$, and $\hat{\boldsymbol{t}} := \mathrm{tr}\,(\boldsymbol{\sigma} + p\mathbf{I}) \cdot \boldsymbol{n}$. 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, $\int_K \hat{\boldsymbol{u}} \cdot \mathbf{n} = 0$.

We present a numerical experiment of Stokes flow around a cylinder, but [26] also includes results for Stokes flow over a backward facing step. 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[21], we might expect it to struggle with this problem as well. The problem domain is detailed in Figure 2.5 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.* [6] run this test with both $r = 0.6$ and $r = 0.9$; we repeat the same experiments with standard and restricted DPG methods starting from the very coarse meshes shown in Figure 2.6 while adaptively refining toward a resolved solution. The extreme pressure gradient in the $r = 0.9$ case obviously makes local conservation more challenging.

The Stokes problem is the first one we encounter that stresses 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 small amount of mass loss for the restricted method is clearly due to accumulation of floating point error.

The most significant benefit of enforcing local conservation for this problem 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 exacerbated. After just one adaptive refinement, the conservative solution nails the peak velocity. The nonconservative solution is completely useless at this point.

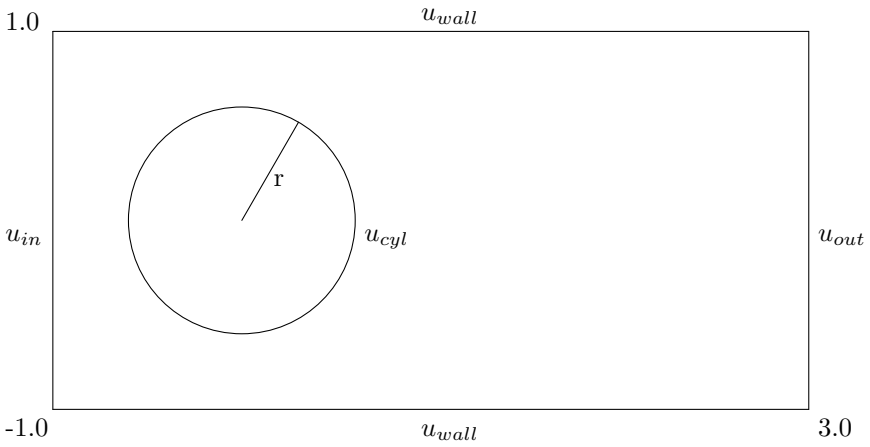
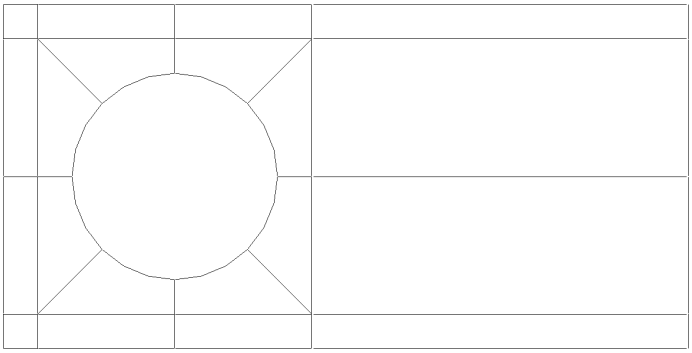
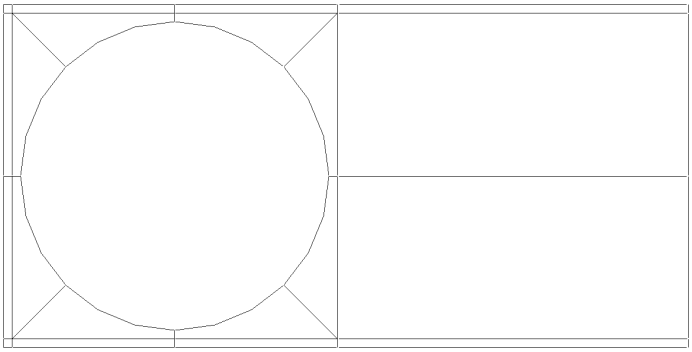


Figure 2.5: Stokes cylinder domain

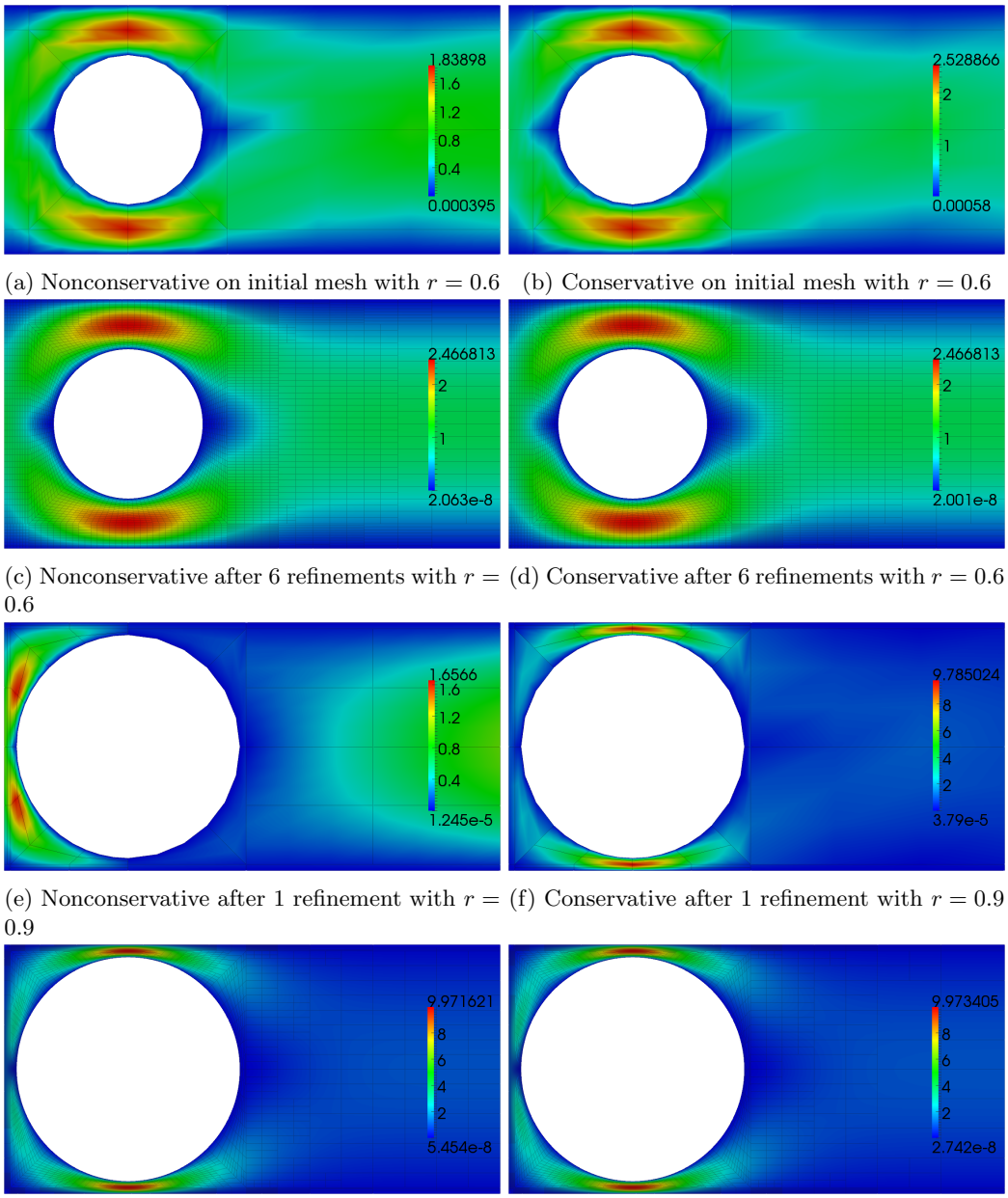


(a) Mesh for $r = 0.6$



(b) Mesh for $r = 0.9$

Figure 2.6: 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 $r = 0.6$

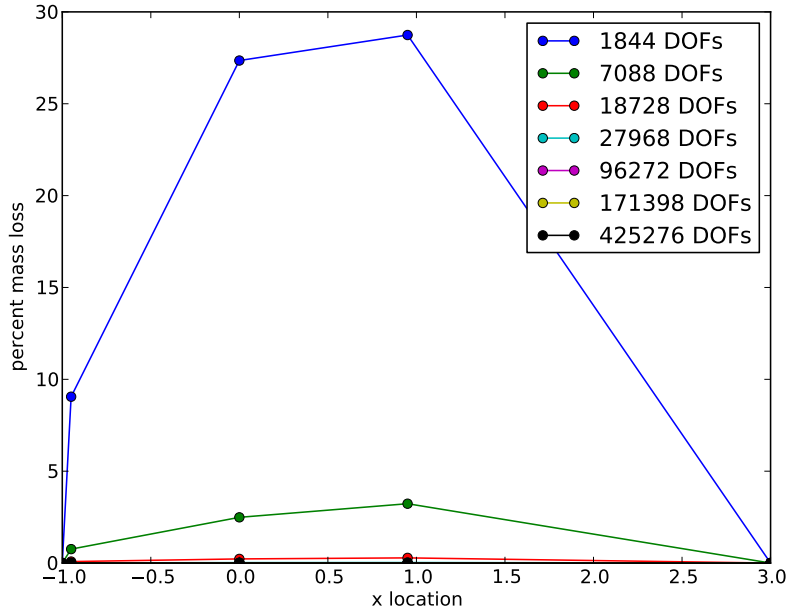
(e) Nonconservative after 1 refinement with $r = 0.9$

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

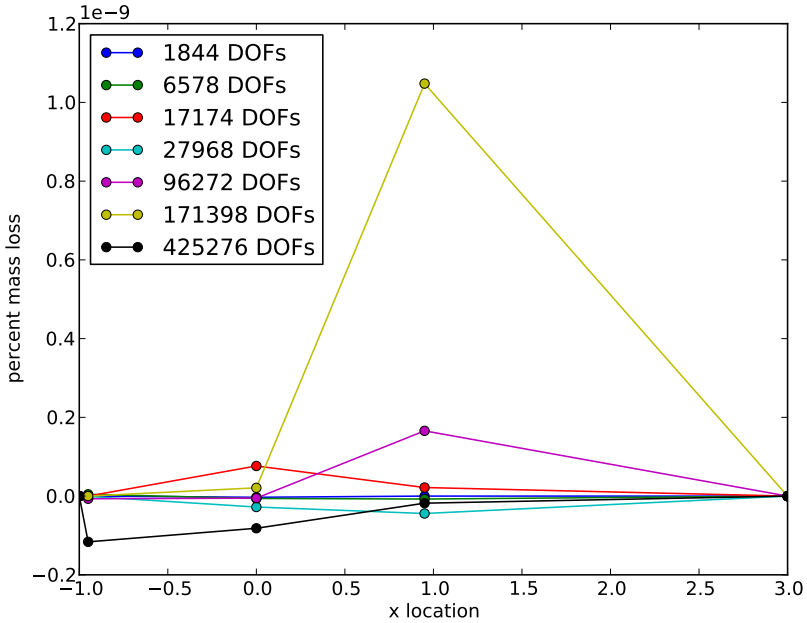
(g) Nonconservative after 6 refinements with $r = 0.9$

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

Figure 2.7: Stokes flow around a cylinder - velocity magnitude

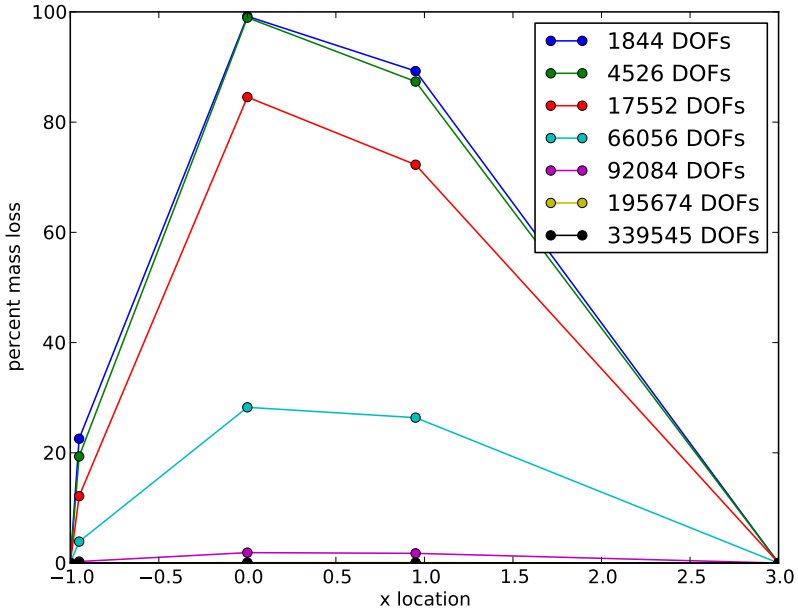


(a) Nonconservative

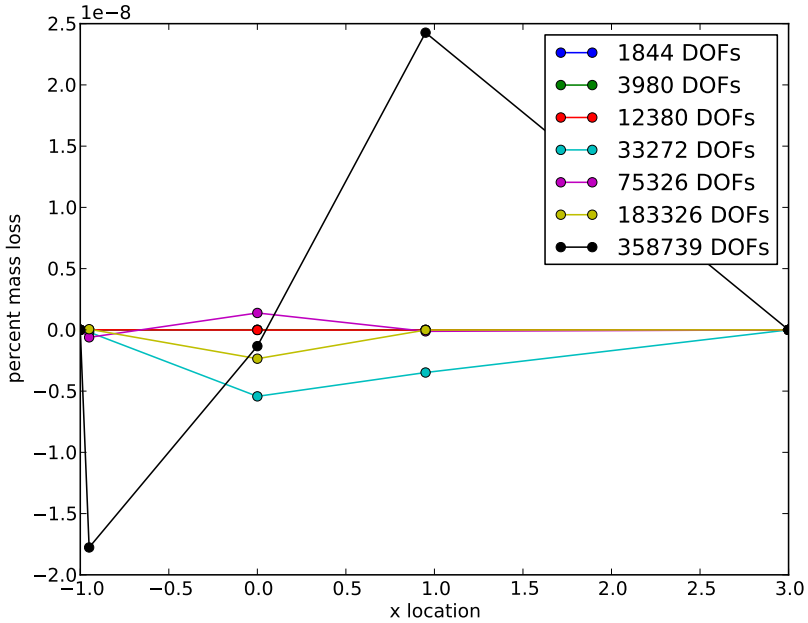


(b) Conservative

Figure 2.8: Mass loss in Stokes flow around a cylinder of radius 0.6



(a) Nonconservative



(b) Conservative

Figure 2.9: Mass loss in Stokes flow around a cylinder of radius 0.9

Chapter 3

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.

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.

3.1 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:

$$\frac{U^n}{\Delta t} + f(U^n) = \frac{U^{n-1}}{\Delta t}\,, \tag{3.1}$$

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

$$R(U^n) = \frac{U^n}{\Delta t} + f(U^n) - \frac{U^{n-1}}{\Delta t}\,. \tag{3.2}$$

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

$$\frac{\Delta U}{\Delta t} + f'(\tilde{U})\Delta U = \frac{U_n}{\Delta t} - \frac{\tilde{U}}{\Delta t} - f(\tilde{U})\,. \tag{3.3}$$

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 $f(\tilde{U})$.

3.2 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 [5], where ESDIRK schemes were found to be more efficient than BDF schemes for laminar flow over a cylinder).

3.2.1 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

$$\frac{\Delta U}{a_{kk}\Delta t} + f'(\tilde{U}^k)\Delta U = \frac{U_n}{a_{kk}\Delta t} - \frac{\tilde{U}^k}{a_{kk}\Delta t} - f(\tilde{U}^k) - \sum_{j=0}^{k-1} \frac{a_{kj}}{a_{kk}} f(U^j) , \tag{3.4}$$

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.

3.2.2 Case Study: 2D Burgers' Equation

We consider the 2D Burger's equations and accompanying problem outlined in [48]. The 2D Burgers' equations are:

$$\begin{aligned} \frac{\partial u_1}{\partial t} + u_1 \frac{\partial u_1}{\partial x} + u_2 \frac{\partial u_1}{\partial y} - \frac{1}{R}\Delta u_1 = 0 \\ \frac{\partial u_2}{\partial t} + u_1 \frac{\partial u_2}{\partial x} + u_2 \frac{\partial u_2}{\partial y} - \frac{1}{R}\Delta u_2 = 0\,, \end{aligned} \tag{3.5}$$

where R is the effective Reynolds number.

3.2.2.1 DPG Formulation

As a first order system, this is

$$\begin{aligned} R\boldsymbol{\sigma}_1 - \nabla u_1 = 0 \\ R\boldsymbol{\sigma}_2 - \nabla u_2 = 0 \\ \frac{\partial u_1}{\partial t} + R \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_1 - \nabla \cdot \boldsymbol{\sigma}_1 = 0 \\ \frac{\partial u_2}{\partial t} + R \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_2 - \nabla \cdot \boldsymbol{\sigma}_2 = 0\,. \end{aligned} \tag{3.6}$$

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

$$\begin{aligned}
(R\boldsymbol{\sigma}_1, \boldsymbol{\tau}_1) + (u_1, \nabla \cdot \boldsymbol{\tau}_1) - \langle \hat{u}_1, \tau_{1n} \rangle &= 0 \\
(R\boldsymbol{\sigma}_2, \boldsymbol{\tau}_2) + (u_2, \nabla \cdot \boldsymbol{\tau}_2) - \langle \hat{u}_2, \tau_{2n} \rangle &= 0 \\
\left(\frac{\partial u_1}{\partial t}, v_1 \right) + \left(R \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_1, v_1 \right) + (\boldsymbol{\sigma}_1, \nabla v_1) - \left\langle \hat{t}_1, v_1 \right\rangle &= 0 \\
\left(\frac{\partial u_2}{\partial t}, v_2 \right) + \left(R \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \boldsymbol{\sigma}_2, v_2 \right) + (\boldsymbol{\sigma}_2, \nabla v_2) - \left\langle \hat{t}_2, v_2 \right\rangle &= 0 \,,
\end{aligned} \tag{3.7}$$

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 (3.4), 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 (3.7) 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 $\tilde{U}$:

$$\begin{aligned}
&(R\Delta\boldsymbol{\sigma}_1, \boldsymbol{\tau}_1) + (\Delta u_1, \nabla \cdot \boldsymbol{\tau}_1) - \langle \hat{u}_1, \tau_{1n} \rangle + \\
&(R\Delta\boldsymbol{\sigma}_2, \boldsymbol{\tau}_2) + (\Delta u_2, \nabla \cdot \boldsymbol{\tau}_2) - \langle \hat{u}_2, \tau_{2n} \rangle + \\
&\left(R \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} \cdot \Delta\boldsymbol{\sigma}_1, v_1 \right) + \left(R \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix} \cdot \tilde{\boldsymbol{\sigma}}_1, v_1 \right) + (\Delta\boldsymbol{\sigma}_1, \nabla v_1) - \left\langle \hat{t}_1, v_1 \right\rangle + \\
&\left(R \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} \cdot \Delta\boldsymbol{\sigma}_2, v_2 \right) + \left(R \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix} \cdot \tilde{\boldsymbol{\sigma}}_2, v_2 \right) + (\Delta\boldsymbol{\sigma}_2, \nabla v_2) - \left\langle \hat{t}_2, v_2 \right\rangle \,,
\end{aligned} \tag{3.8}$$

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 (3.4) and time step toward a transient solution.

3.2.2.2 Numerical Example

An exact solution to the 2D Burgers' equations is

$$\begin{aligned}
u_1(x, y, t) &= \frac{3}{4} - \frac{1}{4(1 + e^{R(-t-4x+4y)/32})} \\
u_2(x, y, t) &= \frac{3}{4} + \frac{1}{4(1 + e^{R(-t-4x+4y)/32})} \,.
\end{aligned} \tag{3.9}$$

We solve on a unit square domain from $t = 0$ to 0.5 with initial condition given by (3.9) 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 3.1.

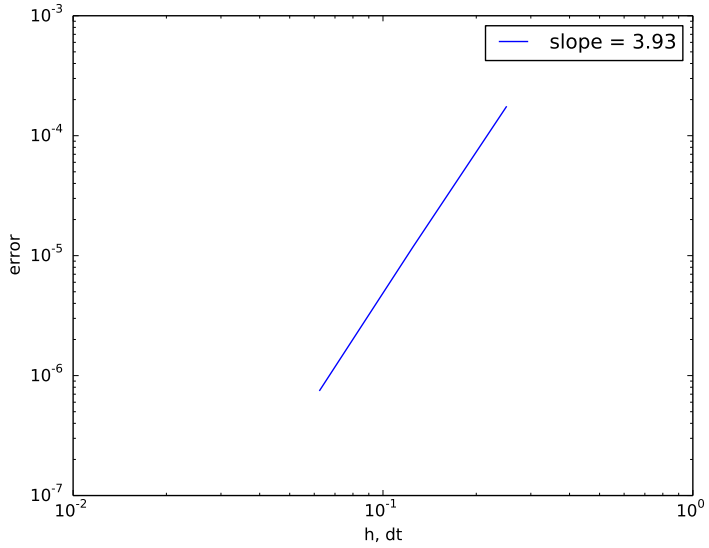


Figure 3.1: L^2 convergence of u_1 and u_2 for the 2D Burgers' equation

Chapter 4

Space-time DPG

We summarize some completed work on space-time DPG. At the time of writing, Camellia does not officially support space-time computations, but we can fake it for 1D spatial problems by pretending the y-direction is time. Complications arise when the PDE under consideration is parabolic (i.e. contains second derivatives in space, but only first derivatives in time). Mathematically, this leaves traces undefined on element edges without a spatial normal component. Practically, this means that I had to hack the Camellia code in order to support these "spatial traces". We say that the code was "hacked" to indicate that we modified the code in an "ugly" manner in order to obtain the following results, but plan is to do this according to better software practices in the proposed work, since the current implementation is not very maintainable.

All of the following problems (unless otherwise noted) are run with the graph norm which is simply defined from the adjoint of the system supplemented with L^2 terms to upgrade it to a full norm.

4.1 Heat equation

The simplest space-time problem we can consider where the the spatial and temporal dimensions are treated differently is the heat equation. We start with a general n-dimensional spatial derivation and later simplify to spatially 1D with a few numerical experiments.

4.1.1 Derivation

Let $\Omega(t) \subset \mathbb{R}^d$ be the spatial domain with boundary $\partial\Omega$. The heat equation is

$$\frac{\partial u}{\partial t} - \mu \Delta u = f\,, \quad \boldsymbol{x} \in \Omega\,,\ t \in (t_0, T) \tag{4.1}$$

where u is unknown heat, ϵ is the diffusion scale, f is the source term, t_0 is the start time, and T is the final time. Let $Q \subset \mathbb{R}^{d+1}$ denote the full space-time domain which is then tessellated into space-time elements K.

The second order formulation of the heat equation is really just a composition of Fourier's law and conservation of energy:

$$\begin{aligned} \boldsymbol{\sigma} - \epsilon \nabla u &= 0 \\ \frac{\partial u}{\partial t} - \nabla \cdot \boldsymbol{\sigma} &= f\,, \end{aligned} \tag{4.2}$$

where $\boldsymbol{\sigma}$ is the heat flux. The key insight that we will use over and over in the following problems is that we can rewrite our conservation equation in terms of a space-time divergence operator: $\nabla_{xt} \cdot () := \nabla \cdot () + \frac{\partial()}{\partial t}$. Our new system is then

$$\begin{aligned} \frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u &= 0 \\ \nabla_{xt} \cdot \begin{pmatrix} -\boldsymbol{\sigma} \\ u \end{pmatrix} &= f\,. \end{aligned} \tag{4.3}$$

We now proceed with the standard DPG practice and multiply by test functions $\boldsymbol{\tau}$ and v and integrate by parts over each space-time element K:

$$\begin{aligned} \left(\frac{1}{\epsilon}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) + (u, \nabla \cdot \boldsymbol{\tau}) - \langle \hat{u}, \boldsymbol{\tau} \cdot \boldsymbol{n}_x \rangle &= 0 \\ -\left(\begin{pmatrix} -\boldsymbol{\sigma} \\ u \end{pmatrix}, \nabla_{xt} v\right) + \langle \hat{t}, v \rangle &= f\,, \end{aligned} \tag{4.4}$$

where

$$\hat{u} := \mathrm{tr}(u)$$

$$\hat{t} := \mathrm{tr}(-\boldsymbol{\sigma}) \cdot \boldsymbol{n}_x + \mathrm{tr}(u) \cdot n_t$$

are new unknowns that live on the mesh skeleton introduced by the integration by parts. Note that the constitutive law was only integrated by parts over spatial dimensions, which means that "spatial trace" $\hat{u}$ only exists on mesh boundaries with a nonzero spatial normal component. On the other hand, flux $\hat{t}$ exists on all mesh boundaries, but changes nature between pure spatial and temporal edges while taking on a mixed nature on slanted boundaries. We illustrate the support of these skeleton variables in Figure 4.1.

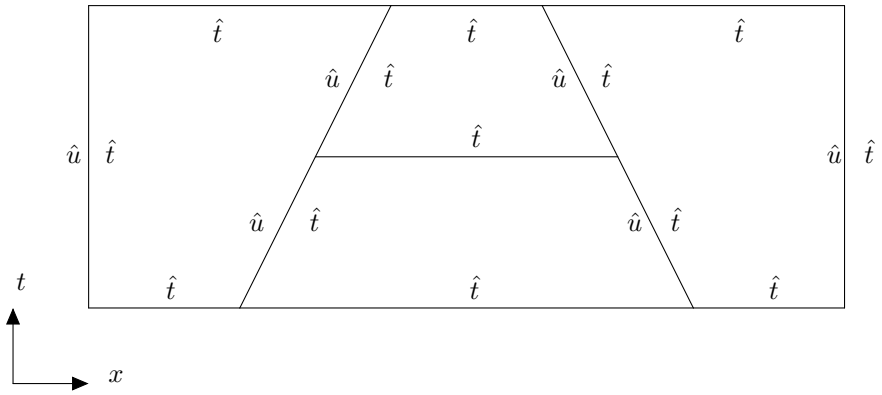


Figure 4.1: Support of flux and spatial trace variables

4.1.2 Problems considered

If we consider a domain $\Omega = [0, 1]^2$ with an initial condition of $u = \cos(2\pi x)$ with zero flux conditions at the boundaries, the exact solution is

$$u = \cos(2\pi x)e^{-4*\pi^2\epsilon t}\,.$$

We ran this with $\epsilon = 10^{-2}$ on a sequence of uniform meshes and $p = 2$ for the field representation of u. We were able to achieve the expected third order convergence as demonstrated in Figure 4.2.

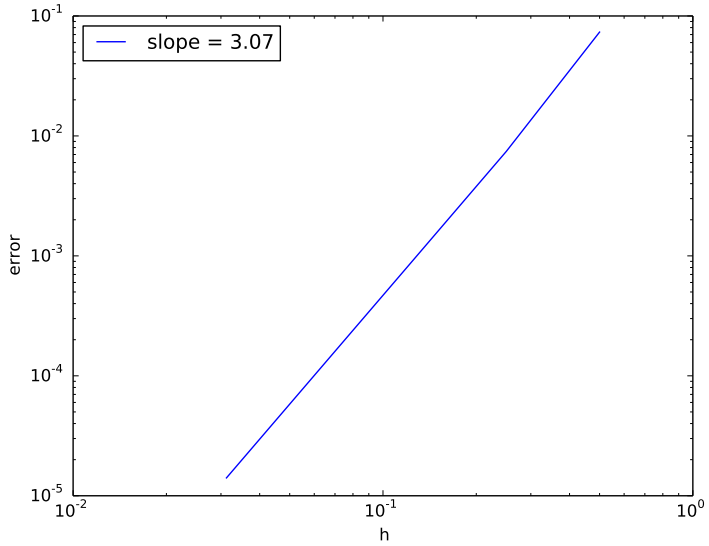


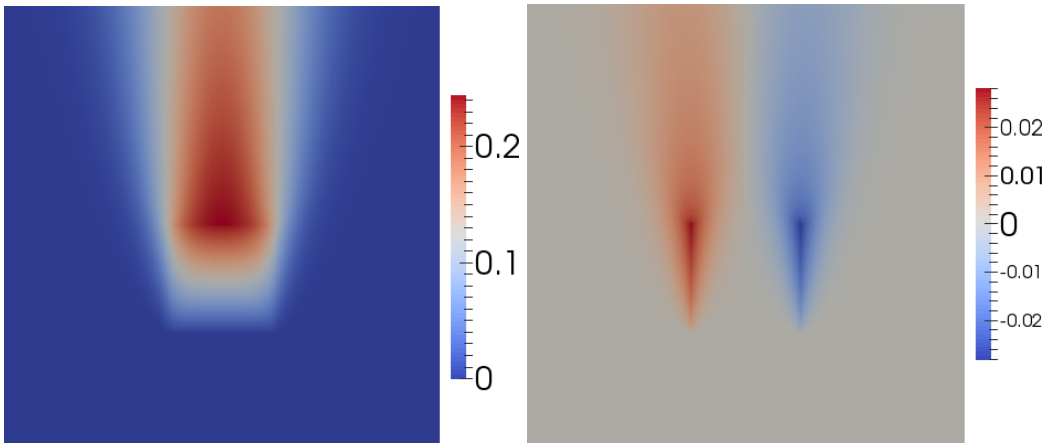
Figure 4.2: L^2 convergence of u for the space-time heat equation

In order to demonstrate local space-time adaptivity we consider one more problem for the heat equation. On the same domain, and with the same boundary conditions as the previous example, we let the initial heat distribution be zero. Then between $t = 0.25$ and $t = 0.5$ we turn on a pulse source term of one on $0.375 \leq x \leq 0.625$. Starting from an initial mesh of $4x4$, we adaptively refine four times and obtain the results in Figure 4.3. Notice that $\hat{u}$ in Figure 4.3c only lives on vertical edges as was discussed earlier. Also notice that the full mesh shown in Figure 4.3d automatically adapts spatially and temporally to where features are rapidly changing.

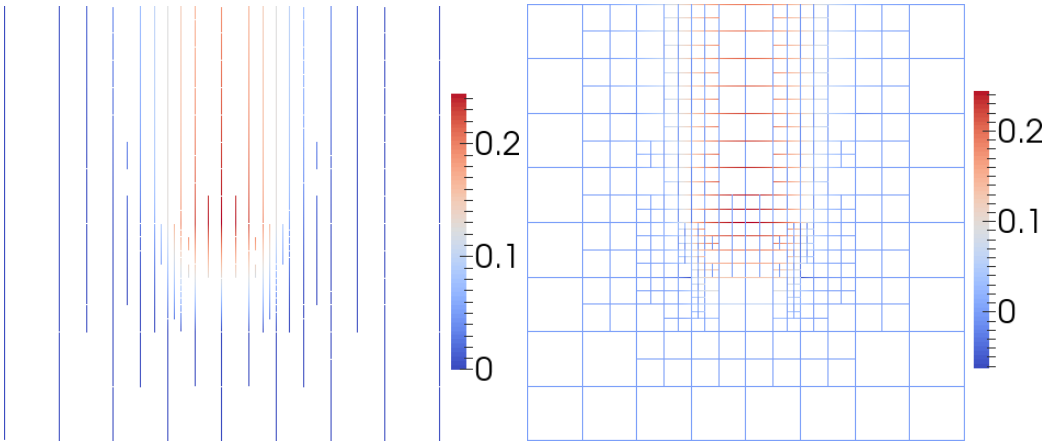
4.2 Convection-Diffusion

Transient convection-diffusion is identical to the heat equation with the addition of a convective term:

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



(a) u (b) σ



(c) $\hat{u}$ (d) $\hat{t}$

Figure 4.3: Pulsed space-time heat problem after 4 refinements

The d-dimensional transient convection-diffusion equations could be viewed as a $d+1$ steady convection-diffusion problem with zero diffusion in the time direction.

4.2.1 Derivation

As a first order system in space-time, this is

$$\begin{aligned} \frac{1}{\epsilon}\boldsymbol{\sigma} - \nabla u &= 0 \\ \nabla_{xt} \cdot \begin{pmatrix} \boldsymbol{\beta} u - \boldsymbol{\sigma} \\ u \end{pmatrix} &= f \,. \end{aligned} \tag{4.5}$$

Multiplying (4.5) by test functions, and integrating by parts over each element, we obtain the following bilinear form:

$$\begin{aligned} \left(\frac{1}{\epsilon}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) + (u, \nabla \cdot \boldsymbol{\tau}) - \langle \hat{u}, \tau_n \rangle &= 0 \\ -\left(\begin{pmatrix} \boldsymbol{\beta} u - \boldsymbol{\sigma} \\ u \end{pmatrix}, \nabla_{xt} v\right) + \langle \hat{t}, v \rangle &= f \,, \end{aligned} \tag{4.6}$$

where now $\hat{t} = \mathrm{tr}\,(\boldsymbol{\beta} u - \boldsymbol{\sigma}) \cdot \boldsymbol{n}_x + \mathrm{tr}(u) \cdot n_t$, and $\hat{u}$ is as before. Our test functions, $\boldsymbol{\tau}$ and v live in the same spaces as for the heat equation.

4.2.2 Problems considered

Since space-time convection-diffusion is identical the heat equation with the addition of a convective term, we only pursue one numerical experiment to demonstrate that everything works as expected. This problem is inspired by the previous heat problem with the spatial domain extended to prevent the convected heat from impinging on the right wall. It might be interesting to impose a zero boundary condition on $\hat{u}$ and watch a boundary layer build up on the right wall, but instead we enforce a zero flux condition and content ourselves with the inner layer that forms around the source pulse. This is an arbitrary requirement necessitated by the "hackish" nature of this code. We haven't taken the time to allow enforcement of Dirichlet boundary conditions on spatial fluxes,

and since the code in this state was intended to be short-lived, it doesn't make sense to invest too heavily in adding features. Thus we enforce zero flux conditions on both walls as before. For this problem, the domain extends from $[0, 1.5] \times [0, 1]$ with the pulse occurring at $[0.25, 0.5] \times [0.25, 0.5]$.

4.3 Transient Compressible Navier-Stokes

We make a large jump from convection-diffusion to the compressible Navier-Stokes equations. The following discussion holds in any dimension, but the provided results are only for spatially 1D flows. The compressible Navier-Stokes equations are

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho \boldsymbol{u} \\ \rho e_0 \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \boldsymbol{u} \\ \rho \boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{I} - \mathbb{D} \\ \rho \boldsymbol{u} e_0 + \boldsymbol{u} p + \boldsymbol{q} - \boldsymbol{u} \cdot \mathbb{D} \end{bmatrix} = \begin{bmatrix} f_c \\ \boldsymbol{f}_m \\ f_e \end{bmatrix}, \tag{4.7}$$

where ρ is the density, $\boldsymbol{u}$ is the velocity, p is the pressure, $\boldsymbol{I}$ is the identity matrix, $\mathbb{D}$ 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 $\lambda = -\frac{2}{3}\mu$,

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

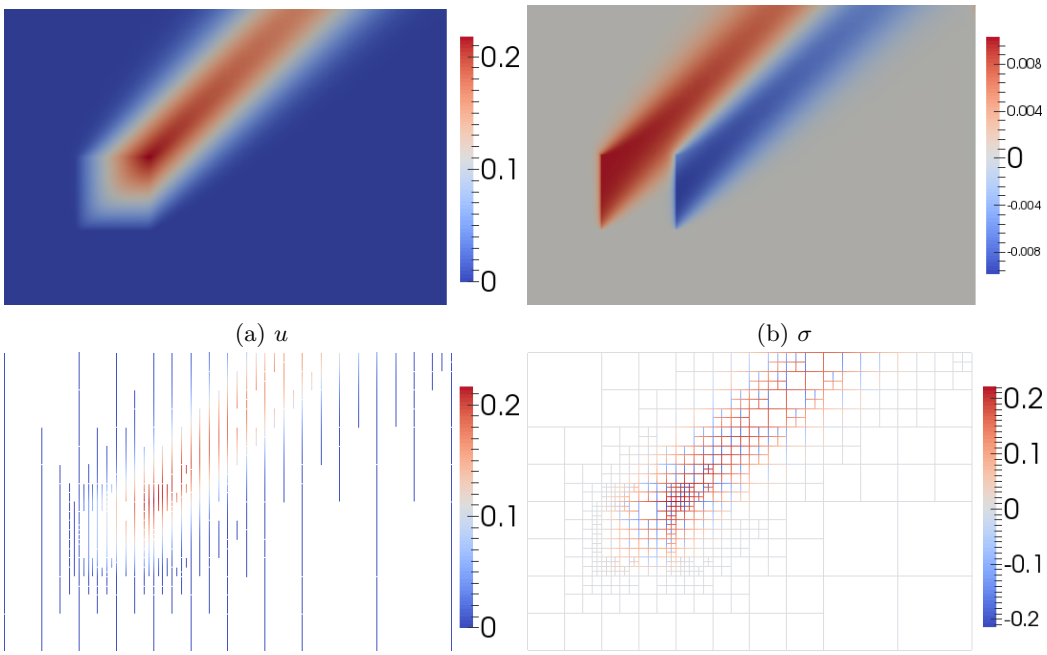
where $\boldsymbol{S}^*$ is the trace-less viscous strain rate tensor. The heat flux is given by Fourier's law:

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

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

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

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



(a) u (b) σ

(c) $\hat{u}$ (d) $\hat{t}$

Figure 4.4: Space-time convection-diffusion problem after 4 refinements

fluid. The total 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:

$$\frac{1}{\mu}\mathbb{D} - \left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) + \frac{2}{3}\nabla\cdot\boldsymbol{u}\boldsymbol{I} = 0 \tag{4.8a}$$

$$\frac{Pr}{C_p\mu}\boldsymbol{q} + \nabla T = 0 \tag{4.8b}$$

$$\nabla_{xt}\cdot\begin{pmatrix}\rho\boldsymbol{u}\\ \rho\end{pmatrix} = f_c \tag{4.8c}$$

$$\nabla_{xt}\cdot\begin{pmatrix}\rho\boldsymbol{u}\otimes\boldsymbol{u} + \rho RT\boldsymbol{I} - \mathbb{D}\\ \rho\boldsymbol{u}\end{pmatrix} = \boldsymbol{f}_m \tag{4.8d}$$

$$\nabla_{xt}\cdot\begin{pmatrix}\rho\boldsymbol{u}\left(C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \boldsymbol{u}\rho RT + \boldsymbol{q} - \boldsymbol{u}\cdot\mathbb{D}\\ \rho\left(C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right)\end{pmatrix} = f_e\,, \tag{4.8e}$$

where our solution variables are ρ, $\boldsymbol{u}$, T, $\mathbb{D}$, and $\boldsymbol{q}$.

4.3.1 Derivation of Space-Time DPG Formulation

We start with (4.8) and multiply by test functions $\mathbb{S}$ (symmetric tensor), $\boldsymbol{\tau}$, v_c, $\boldsymbol{v}_m$, v_e, then integrate by parts over each space-time element K:

$$\left(\frac{1}{\mu}\mathbb{D},\mathbb{S}\right) + (2\boldsymbol{u},\nabla\cdot\mathbb{S}) - \left(\frac{2}{3}\boldsymbol{u},\nabla\operatorname{tr}\mathbb{S}\right) - \left\langle\frac{4}{3}\hat{\boldsymbol{u}},\mathbb{S}\boldsymbol{n}_x\right\rangle = 0 \tag{4.9a}$$

$$\left(\frac{Pr}{C_p\mu}\boldsymbol{q},\boldsymbol{\tau}\right) - (T,\nabla\cdot\boldsymbol{\tau}) + \left\langle\hat{T},\tau_n\right\rangle = 0 \tag{4.9b}$$

$$-\left(\begin{pmatrix}\rho\boldsymbol{u}\\ \rho\end{pmatrix},\nabla_{xt}v_c\right) + \left\langle\hat{t}_c,v_c\right\rangle = (f_c,v_c) \tag{4.9c}$$

$$-\left(\begin{pmatrix}\rho\boldsymbol{u}\otimes\boldsymbol{u} + \rho RT\boldsymbol{I} - \mathbb{D}\\ \rho\boldsymbol{u}\end{pmatrix},\nabla_{xt}\boldsymbol{v}_m\right) + \left\langle\hat{\boldsymbol{t}}_m,\boldsymbol{v}_m\right\rangle = (\boldsymbol{f}_m,\boldsymbol{v}_m) \tag{4.9d}$$

$$-\left(\begin{pmatrix}\rho\boldsymbol{u}\left(C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \boldsymbol{u}\rho RT + \boldsymbol{q} - \boldsymbol{u}\cdot\mathbb{D}\\ \rho\left(C_vT + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right)\end{pmatrix},\nabla_{xt}v_e\right) + \left\langle\hat{t}_e,v_e\right\rangle = (f_e,v_e)\,, \tag{4.9e}$$

where

$$\hat{\boldsymbol{u}} = \operatorname{tr}(\boldsymbol{u})$$

$$\hat{T} = \operatorname{tr}(T)$$

$$\hat{t}_c = \operatorname{tr}(\rho \boldsymbol{u}) \cdot \boldsymbol{n}_x + \operatorname{tr}(\rho)\, n_t$$

$$\hat{\boldsymbol{t}}_m = \operatorname{tr}(\rho \boldsymbol{u} \otimes \boldsymbol{u} + \rho R T \boldsymbol{I} - \mathbb{D}) \cdot \boldsymbol{n}_x + \operatorname{tr}(\rho \boldsymbol{u})\, n_t$$

$$\hat{t}_e = \operatorname{tr}\left(\rho \boldsymbol{u}\left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right) + \boldsymbol{u}\rho R T + \boldsymbol{q} - \boldsymbol{u}\cdot\mathbb{D}\right)\cdot \boldsymbol{n}_x + \operatorname{tr}\left(\rho\left(C_v T + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\right)\right) n_t\,.$$

Note that integrating $\mathbb{S}$ against the symmetric gradient only picks up the symmetric part. This is a much more complicated system of equations than we had for the space-time heat equation, but the situation has many similarities. Test function $\boldsymbol{\tau} \in \boldsymbol{H}(\operatorname{div}, K)$ where the divergence is taken only over spatial dimensions, $v_c, v_e \in H^1(K)$, and $\boldsymbol{v}_m \in \boldsymbol{H}^1(K)$. These are all familiar spaces from our work with the heat equation. Unfortunately, $\mathbb{S}$ has some weird requirements: each $d \times d$ components must be at least in $L^2(K)$, $\nabla \cdot \mathbb{S} \in \mathbf{L}^2(K)$, and $\nabla \operatorname{tr} \mathbb{S} \in \mathbf{L}^2(K)$. In practice, we will probably just seek each component in $H^1(K)$.

4.3.1.1 Linearization

We follow a standard residual-Jacobian linearization procedure coupled with a Gauss-Newton solve. Let $U = \left\{\rho, \boldsymbol{u}, T, \mathbb{D}, \boldsymbol{q}, \hat{\boldsymbol{u}}, \hat{e}, \hat{t}_c, \hat{\boldsymbol{t}}_m, \hat{t}_e\right\}$ be a group solution variable which we can decompose into two parts: $U := \tilde{U} + \Delta U$, where $\tilde{U} = \left\{\tilde{\rho}, \tilde{\boldsymbol{u}}, \tilde{T}, \tilde{\mathbb{D}}, \mathbf{0}, \mathbf{0}, 0, 0, \mathbf{0}, 0\right\}$ is the previous iteration approximation, and $\Delta U = \left\{\Delta\rho, \Delta\boldsymbol{u}, \Delta T, \Delta\mathbb{D}, \boldsymbol{q}, \hat{\boldsymbol{u}}, \hat{e}, \hat{t}_c, \hat{\boldsymbol{t}}_m, \hat{t}_e\right\}$ is the update. Note that $\tilde{U}$ only contains terms which participate in nonlinearities in (4.9) while ΔU contains the full linear terms and the updates to the nonlinear terms. Also, we drop the Δ and $\tilde{\cdot}$ notation for linear terms. Define residual $R(U)$ as the left hand side of (4.9) minus the right hand side. Approximating $R(U) = 0$ by $R(\tilde{U}) + R'(\tilde{U})\Delta U = 0$, where $R'(\tilde{U})$ is the Jacobian of R evaluated at $\tilde{U}$, we get a

linear system:

$$R'(\tilde{U})\Delta U = -R(\tilde{U})\,. \tag{4.10}$$

This is an instance of a Gauss-Newton nonlinear solve. We only need to define our Jacobian and residual for each component of (4.9). The Jacobian of our compressible Navier-Stokes system, $R'(\tilde{U})\Delta U$ is

$$\begin{aligned}
&\left(\frac{1}{\mu}\Delta\mathbb{D},\mathbb{S}\right) + (2\Delta\boldsymbol{u},\nabla\cdot\mathbb{S}) - \left(\frac{2}{3}\Delta\boldsymbol{u},\nabla\,\mathrm{tr}\,\mathbb{S}\right) - \left\langle\frac{4}{3}\hat{\boldsymbol{u}},\mathbb{S}\boldsymbol{n}_x\right\rangle \\
&+ \left(\frac{Pr}{C_p\mu}\boldsymbol{q},\boldsymbol{\tau}\right) - (\Delta T,\nabla\cdot\boldsymbol{\tau}) + \left\langle\hat{T},\tau_n\right\rangle \\
&- \left(\begin{pmatrix}\Delta\rho\tilde{\boldsymbol{u}}+\tilde{\rho}\Delta\boldsymbol{u}\\ \Delta\rho\end{pmatrix},\nabla_{xt}v_c\right) + \left\langle\hat{t}_c,v_c\right\rangle \\
&- \left(\begin{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}+\left(\Delta\rho R\tilde{T}+\tilde{\rho}R\Delta T\right)\boldsymbol{I}-\Delta\mathbb{D}\\ \Delta\rho\tilde{\boldsymbol{u}}+\tilde{\rho}\Delta\boldsymbol{u}\end{pmatrix},\nabla_{xt}\boldsymbol{v}_m\right) + \left\langle\hat{\boldsymbol{t}}_m,\boldsymbol{v}_m\right\rangle \\
&- \left(\begin{pmatrix}\left[C_v\Delta\rho\tilde{T}\tilde{\boldsymbol{u}}+C_v\tilde{\rho}\Delta T\tilde{\boldsymbol{u}}+C_v\tilde{\rho}\tilde{T}\Delta\boldsymbol{u}+\frac{1}{2}\left(\Delta\rho\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}+\tilde{\rho}\Delta\boldsymbol{u}\cdot\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}+\tilde{\rho}\tilde{\boldsymbol{u}}\cdot\Delta\boldsymbol{u}\tilde{\boldsymbol{u}}+\tilde{\rho}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\Delta\boldsymbol{u}\right)\right.\\ \left.+R\left(\Delta\rho\tilde{T}\tilde{\boldsymbol{u}}+\tilde{\rho}\Delta T\tilde{\boldsymbol{u}}+\tilde{\rho}\tilde{T}\Delta\boldsymbol{u}\right)+\boldsymbol{q}-\Delta\boldsymbol{u}\cdot\tilde{\mathbb{D}}-\tilde{\boldsymbol{u}}\cdot\Delta\mathbb{D}\right]\\ 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)\end{pmatrix},\nabla_{xt}v_e\right) \\
&+ \left\langle\hat{t}_e,v_e\right\rangle\,.
\end{aligned} \tag{4.11}$$

The residual, $R(\tilde{U})$, is then

$$\begin{aligned}
&\left(\frac{1}{\mu}\tilde{\mathbb{D}},\mathbb{S}\right) + (2\tilde{\boldsymbol{u}},\nabla\cdot\mathbb{S}) - \left(\frac{2}{3}\tilde{\boldsymbol{u}},\nabla\,\mathrm{tr}\,\mathbb{S}\right) \\
&- \left(\tilde{T},\nabla\cdot\boldsymbol{\tau}\right) \\
&- \left(\begin{pmatrix}\tilde{\rho}\tilde{\boldsymbol{u}}\\ \tilde{\rho}\end{pmatrix},\nabla_{xt}v_c\right) - (f_c,v_c) \\
&- \left(\begin{pmatrix}\tilde{\rho}\tilde{\boldsymbol{u}}\otimes\tilde{\boldsymbol{u}}+\tilde{\rho}R\tilde{T}\boldsymbol{I}-\tilde{\mathbb{D}}\\ \tilde{\rho}\tilde{\boldsymbol{u}}\end{pmatrix},\nabla_{xt}\boldsymbol{v}_m\right) - (\boldsymbol{f}_m,\boldsymbol{v}_m) \\
&- \left(\begin{pmatrix}\tilde{\rho}\tilde{\boldsymbol{u}}\left(C_v\tilde{T}+\frac{1}{2}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\right)+\tilde{\boldsymbol{u}}\tilde{\rho}R\tilde{T}-\tilde{\boldsymbol{u}}\cdot\tilde{\mathbb{D}}\\ \tilde{\rho}\left(C_v\tilde{T}+\frac{1}{2}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\right)\end{pmatrix},\nabla_{xt}v_e\right) - (f_e,v_e)\,.
\end{aligned} \tag{4.12}$$

4.3.1.2 Test Norm

The most obvious first choice for test norm in the local solve is the graph norm, which comes from the problem adjoint. We start by grouping terms in (4.11) by trial variable to get

$$
\begin{aligned}
&\left(\Delta\mathbb{D}, \frac{1}{\mu}\mathbb{S} + \nabla \boldsymbol{v}_m + \nabla v_e \otimes \tilde{\boldsymbol{u}}\right) \\
&+ \left(\boldsymbol{q}, \frac{Pr}{C_p\mu}\boldsymbol{\tau} - \nabla v_e\right) \\
&+ \left(\Delta\rho, -\tilde{\boldsymbol{u}}\cdot\nabla v_c - \frac{\partial v_c}{\partial t} - \tilde{\boldsymbol{u}}\otimes\tilde{\boldsymbol{u}} : \nabla\boldsymbol{v}_m - R\tilde{T}\nabla\cdot\boldsymbol{v}_m - \tilde{\boldsymbol{u}}\cdot\frac{\partial \boldsymbol{v}_m}{\partial t}\right. \\
&\quad \left. - C_v\tilde{T}\tilde{\boldsymbol{u}}\cdot\nabla v_e - \frac{1}{2}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}\cdot\nabla v_e - R\tilde{T}\tilde{\boldsymbol{u}}\nabla v_e - C_v\tilde{T}\frac{\partial v_e}{\partial t} - \frac{1}{2}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t}\right) \\
&+ \left(\Delta\boldsymbol{u}, 2\nabla\cdot\mathbb{S} - \frac{2}{3}\nabla\operatorname{tr}\mathbb{S} - \tilde{\rho}\nabla v_c - \tilde{\rho}\tilde{\boldsymbol{u}}\cdot\nabla\boldsymbol{v}_m - \tilde{\rho}\nabla\boldsymbol{v}_m\cdot\tilde{\boldsymbol{u}} - \tilde{\rho}\frac{\partial\boldsymbol{v}_m}{\partial t} - C_v\tilde{\rho}\tilde{T}\nabla v_e\right. \\
&\quad \left. - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\cdot\tilde{\boldsymbol{u}}\nabla v_e - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\cdot\nabla v_e\tilde{\boldsymbol{u}} - \frac{1}{2}\tilde{\rho}\nabla v_e\cdot\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} - R\tilde{\rho}\tilde{T}_\nabla v_e + \tilde{\mathbb{D}}\cdot\nabla v_e - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t}\right) \\
&+ \left(\Delta T, -\nabla\cdot\boldsymbol{\tau} - R\tilde{\rho}\nabla\cdot\boldsymbol{v}_m - C_v\tilde{\rho}\tilde{\boldsymbol{u}}\nabla v_e - R\tilde{\rho}\tilde{\boldsymbol{u}}\nabla v_e - C_v\tilde{\rho}\frac{\partial v_e}{\partial t}\right) \\
&+ \left(\hat{\boldsymbol{u}}, -\frac{4}{3}\mathbb{S}\boldsymbol{n}_x\right) \\
&+ \left(\hat{T}, \tau_n\right) \\
&+ \left(\hat{t}_c, v_c\right) \\
&+ \left(\hat{\boldsymbol{t}}_m, \boldsymbol{v}_m\right) \\
&+ \left(\hat{t}_e, v_e\right) .
\end{aligned}
\tag{4.13}
$$

Then the graph norm would be defined by

$$
\begin{aligned}
&\left\| \frac{1}{\mu}\mathbb{S} + \nabla \boldsymbol{v}_m + \nabla v_e \otimes \tilde{\boldsymbol{u}} \right\|^2 \\
&+ \left\| \frac{Pr}{C_p \mu}\boldsymbol{\tau} - \nabla v_e \right\|^2 \\
&+ \Bigg\| -\tilde{\boldsymbol{u}} \cdot \nabla v_c - \frac{\partial v_c}{\partial t} - \tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} : \nabla \boldsymbol{v}_m - R\tilde{T}\nabla \cdot \boldsymbol{v}_m - \tilde{\boldsymbol{u}} \cdot \frac{\partial \boldsymbol{v}_m}{\partial t} \\
&\quad - C_v \tilde{T}\tilde{\boldsymbol{u}} \cdot \nabla v_e - \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} \cdot \nabla v_e - R\tilde{T}\tilde{\boldsymbol{u}}\nabla v_e - C_v \tilde{T}\frac{\partial v_e}{\partial t} - \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} \Bigg\|^2 \\
&+ \Bigg\| 2\nabla \cdot \mathbb{S} - \frac{2}{3}\nabla \operatorname{tr}\mathbb{S} - \tilde{\rho}\nabla v_c - \tilde{\rho}\tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{v}_m - \tilde{\rho}\nabla \boldsymbol{v}_m \cdot \tilde{\boldsymbol{u}} - \tilde{\rho}\frac{\partial \boldsymbol{v}_m}{\partial t} - C_v \tilde{\rho}\tilde{T}\nabla v_e \\
&\quad - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\nabla v_e - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \nabla v_e \tilde{\boldsymbol{u}} - \frac{1}{2}\tilde{\rho}\nabla v_e \cdot \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} - R\tilde{\rho}\tilde{T}\nabla v_e + \tilde{\mathbb{D}} \cdot \nabla v_e - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} \Bigg\|^2 \\
&+ \left\| -\nabla \cdot \boldsymbol{\tau} - R\tilde{\rho}\nabla \cdot \boldsymbol{v}_m - C_v \tilde{\rho}\tilde{\boldsymbol{u}}\nabla v_e - R\tilde{\rho}\tilde{\boldsymbol{u}}\nabla v_e - C_v \tilde{\rho}\frac{\partial v_e}{\partial t} \right\| \\
&+ \alpha_c \left\| v_c \right\|^2 + \alpha_m \left\| \boldsymbol{v}_m \right\|^2 + \alpha_e \left\| v_e \right\|^2 + \alpha_s \left\| \mathbb{S} \right\| + \alpha_f \left\| \boldsymbol{\tau} \right\| ,
\end{aligned}
\tag{4.14}
$$

where α_c, α_m, α_e, α_s, and α_f are scaling constants, usually one.

Unfortunately, the graph norm is known to not be robust for steady convection-diffusion or Navier-Stokes, and we saw that non-robustness manifest when we tried to use this norm for transient simulations as well. For steady state DPG, we developed a robust test norm for convection-diffusion and drew analogies to create a robust norm for Navier-Stokes. A similar analysis for transient convection-diffusion has not been done (this is part of the proposed work), so we are on shakier footing developing a robust norm for transient Navier-Stokes. Nevertheless, we can make some guesses about how to modify the test norm in order to obtain some preliminary results. The graph norm has proven to be sufficient for simulations of pure convection. So an obvious first guess might be to take the graph norm on the convective quantities and decouple the viscous terms. Indeed, this selection proved to be more robust for the test problems considered in the next section. This

modified graph norm is then:

$$
\begin{aligned}
&\left\|\nabla \boldsymbol{v}_m + \nabla v_e \otimes \tilde{\boldsymbol{u}}\right\|^2 \\
&+ \left\|-\nabla v_e\right\|^2 \\
&+ \left\| -\tilde{\boldsymbol{u}} \cdot \nabla v_c - \frac{\partial v_c}{\partial t} - \tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}} : \nabla \boldsymbol{v}_m - R\tilde{T}\nabla \cdot \boldsymbol{v}_m - \tilde{\boldsymbol{u}} \cdot \frac{\partial \boldsymbol{v}_m}{\partial t} \right. \\
&\quad \left. -C_v\tilde{T}\tilde{\boldsymbol{u}} \cdot \nabla v_e - \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} \cdot \nabla v_e - R\tilde{T}\tilde{\boldsymbol{u}}\nabla v_e - C_v\tilde{T}\frac{\partial v_e}{\partial t} - \frac{1}{2}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} \right\|^2 \\
&+ \left\| -\tilde{\rho}\nabla v_c - \tilde{\rho}\tilde{\boldsymbol{u}} \cdot \nabla \boldsymbol{v}_m - \tilde{\rho}\nabla \boldsymbol{v}_m \cdot \tilde{\boldsymbol{u}} - \tilde{\rho}\frac{\partial \boldsymbol{v}_m}{\partial t} - C_v\tilde{\rho}\tilde{T}\nabla v_e \right. \\
&\quad \left. -\frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{u}}\nabla v_e - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}} \cdot \nabla v_e\tilde{\boldsymbol{u}} - \frac{1}{2}\tilde{\rho}\nabla v_e \cdot \tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}} - R\tilde{\rho}\tilde{T}\nabla v_e + \tilde{\mathbb{D}} \cdot \nabla v_e - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} - \frac{1}{2}\tilde{\rho}\tilde{\boldsymbol{u}}\frac{\partial v_e}{\partial t} \right\|^2 \\
&+ \left\| -R\tilde{\rho}\nabla \cdot \boldsymbol{v}_m - C_v\tilde{\rho}\tilde{\boldsymbol{u}}\nabla v_e - R\tilde{\rho}\tilde{\boldsymbol{u}}\nabla v_e - C_v\tilde{\rho}\frac{\partial v_e}{\partial t} \right\| \\
&+ \frac{1}{\mu}\left\|\mathbb{S}\right\| + 2\left\|\nabla \cdot \mathbb{S} - \frac{1}{3}\nabla \operatorname{tr}\mathbb{S}\right\| + \frac{Pr}{c_p\mu}\boldsymbol{\tau}\left\|\boldsymbol{\tau}\right\| + \left\|\nabla \cdot \boldsymbol{\tau}\right\| \\
&+ \left\|v_c\right\|^2 + \left\|\boldsymbol{v}_m\right\|^2 + \left\|v_e\right\|^2 \ .
\end{aligned}
\tag{4.15}
$$

From a number of numerical tests, it appears that this norm is not completely robust, but it does seem to perform somewhat better than the standard graph norm.

4.3.2 Numerical Results

We consider two 1D test problems as verification. The Sod shock tube and Noh implosion both 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 numerical methods introduce a certain amount of artificial viscosity in order to select a unique solution. Most schemes also 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: $\mu = 10^{-5}$ for Sod and $\mu = 10^{-3}$ for Noh. Essentially we are just simulating low viscosity Navier-Stokes as a

stand-in for the unsolvable pure Euler. We mentioned previously that the test norm we are using is not entirely robust, and in fact these viscosity values were on the lower end of what we could simulate with this preliminary norm. Following the same polynomial representation as we did in the section on local conservation (Section 2.3), the field variables were represented with quadratics.

4.3.2.1 Sod Shock Tube

The Sod shock tube problem was developed by Gary Sod in 1978[44], 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 $\gamma = 1.4$. The $t = 0$ 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 $\hat{t}_c = \hat{t}_m = \hat{t}_e = 0$ on the left and right boundaries, 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 one continuous time slab starting with only 4 space-time elements.

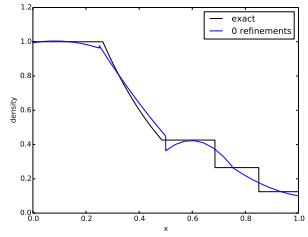
The results are plotted in Figure 4.5 for three different refinement levels: the initial coarse mesh, 7 adaptive refinements, and 14 refinements. The coarsest mesh is obviously not sufficient to resolve the features of the flow, but it is at least somewhat representative of the exact solution. We

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.

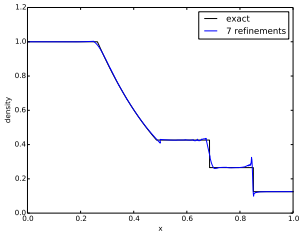
4.3.2.2 Noh Implosion

The Noh implosion problem[36] 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. We run this problem to a final time of $t = 1.0$. The longer time nature of this problem recommended the use of multiple time slabs rather than a single solve like the previous problem. We run with four time slabs of thickness 0.25 each with 4 initial space-time elements. We run the first slab to 8 adaptive refinements and set the initial conditions on the next slab to the refined solution on the previous slab.

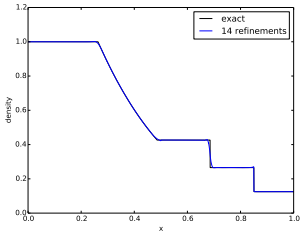
Each of the slabs are put together into one long time solution in Figure 4.6. Again we plot the solution on the initial mesh, a halfway resolved mesh, and a final mesh after 8 refinement steps. We get some very odd behavior around the shock on the middle mesh, but this goes away by the final mesh. We see the same behavior with overshoots and undershoots that we saw with the Sod problem.



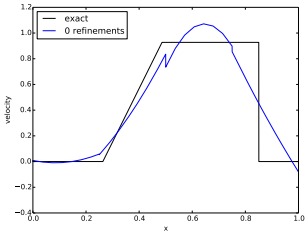
(a) Density on initial mesh



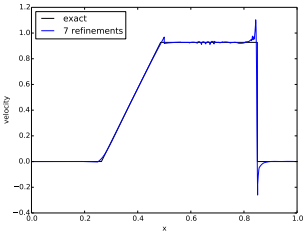
(b) Density after 7 refinements



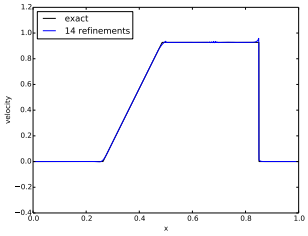
(c) Density after 14 refinements



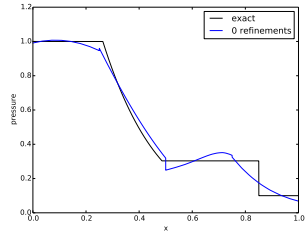
(d) Velocity on initial mesh



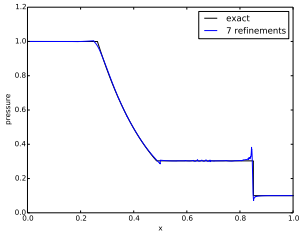
(e) Velocity after 7 refinements



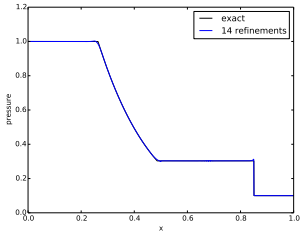
(f) Velocity after 14 refinements



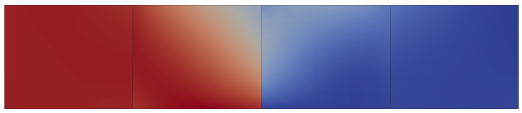
(g) Pressure on initial mesh



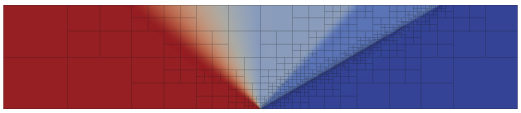
(h) Pressure after 7 refinements



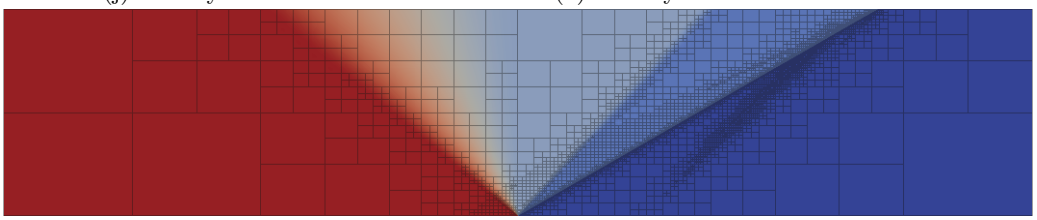
(i) Pressure after 14 refinements



(j) Density with initial mesh

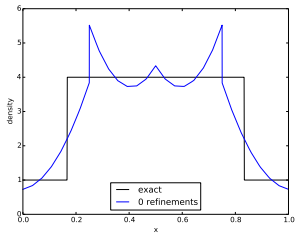


(k) Density with mesh after 7 refinements

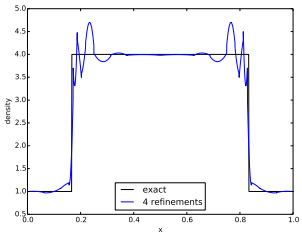


(l) Density with mesh after 14 refinements

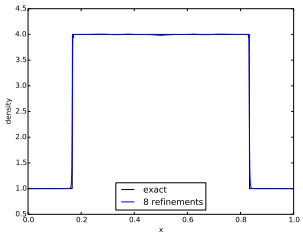
Figure 4.5: Sod problem with final time $t = 0.2$



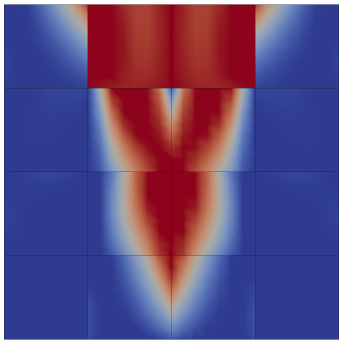
(a) Density on initial mesh



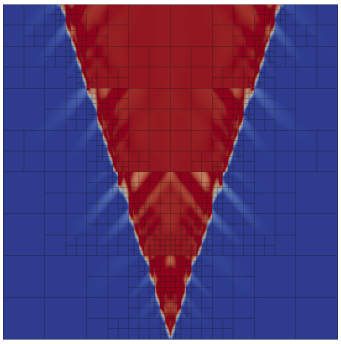
(b) Density after 4 refinements



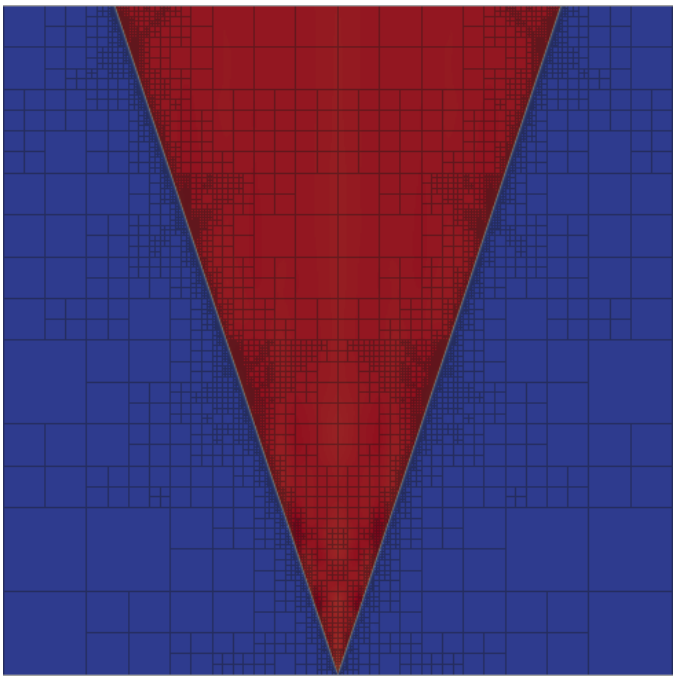
(c) Density after 8 refinements



(d) Density with initial mesh



(e) Density with mesh after 4 refinements



(f) Density with mesh after 8 refinements

Figure 4.6: Noh problem with final time $t = 1.0$

Chapter 5

Proposed work

The completed work in 1D space-time has been very promising, but there is still much to do. Obviously 1D computations are not going to be sufficient, but we are in active development extending our code to higher dimensional problems. The results for Sod and Noh suggest that the current method is not entirely robust in terms of shrinking viscosity, so we need to perform a robustness analysis and develop new test norms for transient convection-diffusion. We will then extend these ideas to the more complicated case of Navier-Stokes. Along the way I will be verifying our code on a number of standard benchmark problems for compressible and incompressible Navier-Stokes. A summary of my contributions divided into areas A, B, and C of the CSEM program follow.

Area A: Applicable mathematics. DPG is a method built on rigorous mathematical theory. While rooted in the standard theory of finite elements, DPG is novel enough that many of the old analysis tools do not directly apply. As a consequence, much of the early DPG literature has been laden with mathematical proofs of stability, convergence, and robustness. While developing the theory for a locally conservative DPG formulation we performed a stability and robustness analysis. Another robustness analysis will be necessary as we explore space-time convection-diffusion since the equation parabolic in space-time. Past DPG robustness analysis focused on purely elliptic problems.

One significant remaining obstacle to obtaining robust solutions for compressible Navier-Stokes is guaranteeing positivity of the density and energy in the presence of unresolved shocks. Positivity is an auxiliary stability condition that is not guaranteed by traditional stabilization techniques (including DPG), so it we are going to need to augment an additional technology to the method in order to guarantee physically realizable densities and energies. Unfortunately, positivity preservation is itself an incredibly difficult problem. We have several ideas we want to try in the context of DPG, but solving the issue may not lie within the scope of this dissertation.

Area B: Numerical analysis and scientific computation. I will support development of the DPG software framework *Camellia*[43] including running verification on several test problems.

This will include the development and verification of spatially 2D space-time simulations. I will also develop a means to perform time stepping via space-time slabs which should reduce the overall run time and memory requirements for longer simulations. I've already contributed several auxiliary features to *Camellia* including mesh readers and solution export, but I also plan on looking into adding HDF5 support along with parallel solution output.

A space-time implementation adds additional dimensionality to the problem under consideration increasing the problem size and memory requirements. This makes parallel simulation an increasingly important aspect of this work. The 16 core *Nozomi* cluster is sufficient for early experimentation, but as we ramp up to larger problems, I plan to run on allocations at TACC and on Mira at Argonne National Laboratory.

The major bottleneck to parallel simulations at the moment is that *Camellia* still relies on a serial direct solver for the global solve. Time permitting, I would like to investigate the implementation of an effective iterative solver for DPG.

Area C: Mathematical modeling and applications. As we are exploring a method that is still very much in development, I will be running a number of standard test problems. I've already explored the conservation properties of DPG through a couple of simulations of Stokes flow around a cylinder and over a backward facing step.

The steady state DPG formulation failed to converge on the Carter plate problem for Reynolds numbers of 10^7. One hypothesis is that this was due to transient effects on the unresolved mesh which pushed the time step requirements very low. The hope is that by using a space-time formulation with temporal adaptivity, we would be able to resolve these temporal oscillations and they would die away to the correct steady state solution. Another hypothesis is that the non-convergence was caused by Gibbs phenomenon near the unresolved leading edge driving the density negative. Our line search algorithm scales the Newton updates to prevent negative densities, but sometimes the lines search factor can be driven so small as to effectively stall convergence. Therefore my first priority will be to investigate which factor is stalling convergence and attempt to solve it either via adaptive space-time refinements or some sort of positivity preserving technology yet to be decided upon. If this proves to be an effective way of handling transient shock problems, I would also like to simulate the Sedov explosion problem and the Noh implosion problem. Time permitting, I would also like to run the triple point shock interaction problem and a Rayleigh-Taylor instability problem. These will be ideal for demonstrating the local adaptivity of DPG.

That said, I believe that the most promising application for space-time DPG will be the incompressible Navier-Stokes equations. Here we don't run into any of the same issues with Gibbs phenomenon and we can use vanilla space-time DPG. The Taylor-Green vortex problem is an obvious choice since it has a exact solution that we can compare to. There are several vortex shedding problems that would be of interest including flow over a triangle, square, and cylinder.

Bibliography

[1] S.K. Aliabadi and T.E. Tezduyar. Space-time finite element computation of compressible flows involving moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 107(12):209 – 223, 1993.

[2] J.H. Argyris and D.W. Scharpf. Finite elements in time and space. *Nuclear Engineering and Design*, 10(4):456 – 464, 1969.

[3] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, May 2001.

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

[5] Hester Bijl, Mark H. Carpenter, Veer N. Vatsa, and Christopher A. Kennedy. Implicit time integration schemes for the unsteady compressible navierstokes equations: Laminar flow. *Journal of Computational Physics*, 179(1):313–329, June 2002.

[6] 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.

[7] J. Bramwell, L. Demkowicz, J. Gopalakrishnan, and W. Qiu. A locking-free *hp* DPG method for linear elasticity with symmetric stresses. *Num. Math.*, 2012.

[8] 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.

[9] F. Brezzi, B. Cockburn, L.D. Marini, and E. Sli. Stabilization mechanisms in discontinuous Galerkin finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 195(2528):3293 – 3310, 2006. Discontinuous Galerkin Methods.

[10] 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.

[11] J. Chan, N. Heuer, T. Bui-Thanh, and L. Demkowicz. Robust DPG method for convection-dominated diffusion problems II: A natural inflow condition. Technical Report 21, ICES, 2012.

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

[13] 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.

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

[15] 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.

[16] B. Cockburn and C. Shu. The RungeKutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141(2):199 – 224, 1998.

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

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

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

[20] L. Demkowicz and J. Gopalakrishnan. Analysis of the DPG method for the Poisson problem. *SIAM J. Num. Anal.*, 49(5):1788–1809, 2011.

[21] L. Demkowicz and J. Gopalakrishnan. An overview of the DPG method. Technical report, ICES, 2013.

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

[23] L. Demkowicz, J. Gopalakrishnan, I. Muga, and J. Zitelli. Wavenumber explicit analysis for a DPG method for the multidimensional Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 213-216:126–138, 2012.

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

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

[26] T. E. Ellis, L. F. Demkowicz, and J. L. Chan. Locally conservative discontinuous Petrov-Galerkin finite elements for fluid problems. Technical report, ICES, December 2013.

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

[28] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, {III}. *Journal of Computational Physics*, 131(1):3 – 47, 1997.

[29] Thomas J.R. Hughes, Gonzalo R. Feijo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method – a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(12):3 – 24, 1998. Advances in Stabilized Methods in Computational Mechanics.

[30] Thomas J.R. Hughes and Gregory M. Hulbert. Space-time finite element methods for elastodynamics: Formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66(3):339 – 363, 1988.

[31] Thomas J.R. Hughes and James R. Stewart. A space-time formulation for multiscale phenomena. *Journal of Computational and Applied Mathematics*, 74(12):217 – 229, 1996.

[32] C.M. Klaij, J.J.W. van der Vegt, and H. van der Ven. Spacetime discontinuous Galerkin method for the compressible navierstokes equations. *Journal of Computational Physics*, 217(2):589 – 611, 2006.

[33] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134(12):71 – 90, 1996.

[34] X. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.

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

[36] W.F Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics*, 72(1):78 – 120, 1987.

[37] J. Tinsley Oden. A general theory of finite elements. ii. applications. *International Journal for Numerical Methods in Engineering*, 1(3):247–259, 1969.

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

[39] 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.

[40] 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.

[41] Sander Rhebergen and Bernardo Cockburn. A spacetime hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. *Journal of Computational Physics*, 231(11):4185 – 4204, 2012.

[42] N. V. Roberts, T. Bui-Thanh, and L. F. Demkowicz. The DPG method for the Stokes problem. Technical Report 12-22, ICES, 2012.

[43] N. V. Roberts, D. Ridzal, P. B. Bochev, and L. F. Demkowicz. A toolbox for a class of discontinuous Petrov-Galerkin methods using Trilinos. Technical Report SAND2011-6678, Sandia National Laboratories.

[44] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1 – 31, 1978.

[45] 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. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339 – 351, 1992.

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

[47] J.J.W. van der Vegt and H. van der Ven. Spacetime discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. general formulation. *Journal of Computational Physics*, 182(2):546 – 585, 2002.

[48] H. Zhu, H. Shu, and M. Ding. Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method. *Computers & Mathematics with Applications*, 60(3):840–848, August 2010.