A DPG method for steady viscous compressible flow

Jesse Chan*, Leszek Demkowicz**, Robert Moser**

Abstract

The Discontinuous Petrov-Galerkin (DPG) method is a class of novel higher order adaptive finite element methods derived from the minimization of the residual of the variational problem [1], and has been shown to deliver a method for convection-diffusion that is provably robust in the diffusion parameter [2, 3]. In this work, the DPG method is extrapolated to nonlinear systems, and applied to several problems in fluid dynamics whose solutions exhibit boundary layers or singularities in stresses. In particular, the effectiveness of DPG as a numerical method for compressible flow is assessed with the application of DPG to two model problems over a range of Mach numbers and laminar Reynolds numbers using automatic adaptivity for higher order elements, beginning with highly under-resolved coarse initial meshes.

1. Introduction

Standard numerical methods tend to perform poorly across the board for the class of PDEs known as singular perturbation problems; these problems are often characterized by a parameter that may be either very small or very large. An additional complication of singular perturbation problems is that very often, in the limiting case of the parameter blowing up or decreasing to zero, the PDE itself will change types (e.g. from elliptic to hyperbolic). A canonical example of a singularly perturbed problem is the convection-diffusion equation in domain $\Omega \subset \mathbb{R}^3$,

$$\nabla \cdot (\beta u) - \epsilon \Delta u = f.$$

The equation models the steady-state distribution of the scalar quantity u, representing the concentration of a quantity in a given medium, taking into account both convective and diffusive effects. Vector $\beta \in \mathbb{R}^3$ specifies the direction and magnitude of convection, while the singular perturbation parameter ϵ represents the diffusivity of the medium. In the limit of an inviscid medium as $\epsilon \to 0$, the equation changes types, from elliptic to hyperbolic, and from second order to first order.

The standard finite element method applied to the convection-dominated diffusion problem can perform very poorly for the case of small ϵ.[1] This poor performance can be observed numerically – as the singular perturbation parameter ϵ decreases, the finite element solution can diverge significantly from the best finite element approximation of the solution. For example, it is well known that, on a fixed coarse mesh and for small values of ϵ (or a large Peclet number, the ratio h/ϵ), the Galerkin

*Principal Corresponding Author, Institute for Computational Engineering and Sciences, The University of Texas at Austin

**Institute for Computational Engineering and Sciences, The University of Texas at Austin

Email addresses: jchan@ices.utexas.edu (Jesse Chan), leszek@ices.utexas.edu (Leszek Demkowicz), rmoser@ices.utexas.edu (Robert Moser)

[1] This is especially true in the presence of boundary layers in the solution [4].

approximation of the solution to the convection-diffusion equation with a boundary layer develops spurious oscillations everywhere in the domain, even where the best approximation error is small. These oscillations grow in magnitude as $\epsilon \to 0$, eventually polluting the entire solution.[2]

Traditionally, instability/loss of robustness in finite element methods has been dealt with using residual-based stabilization techniques. Given some variational form, the problem is modified by adding to the bilinear form the strong form of the residual, weighted by a test function and scaled by a stabilization constant τ. The most well-known example of this technique is the streamline-upwind Petrov-Galerkin (SUPG) method, which is a stabilized FE method for solving the convection-diffusion equation using piecewise linear continuous finite elements [5]. SUPG stabilization not only removes the spurious oscillations from the finite element solution of the convection-diffusion equation, but delivers the best finite element approximation in the H^1 norm in 1D.

From the perspective of the compressible Navier-Stokes equations, this loss of robustness is doubly problematic. Not only will any nonlinear solution suffer from similar unstable oscillations, but nonlinear solvers themselves may fail to yield a solution due to such instabilities. A nonlinear solution is almost always computed by solving a series of linear problems whose solutions will converge to the nonlinear solution under appropriate assumptions, and the presence of such oscillations in each linear problem can cause the solution convergence to slow significantly or even diverge. Artificial viscosity and shock capturing methods have been used to suppress such oscillations and regularize the problem. While these methods will usually yield smooth and qualitatively resolved solutions, these methods are often overly diffusive, yielding results which are poor approximations of the true solution [6], though modern artificial viscosity and shock capturing schemes have improved greatly in recent years [7, 8]. We have taken an alternative approach in this work, avoiding artificial diffusion and shock capturing for the moment.

Our aim is to develop a stable, higher order scheme for the steady compressible laminar Navier-Stokes equations in transonic/supersonic regimes that is automatically adaptive beginning with very coarse meshes. This requires that both the method and the refinement scheme to perform adequately on coarse meshes with high Peclet numbers – in other words, that the adaptive method is robust in the diffusion parameter. We construct such a method in this paper as follows: we begin by deriving the DPG method for linear problems, then constructing a DPG method for the scalar convection-diffusion equation that is robust for very small viscosities. Unlike common adaptive methods in computational fluid dynamics, which often refine based on physical features (such as high gradients in the solution), adaptivity under this method is driven by the minimization of a residual, which measures error accurately even for highly underresolved meshes. The DPG method for scalar convection-diffusion is then extrapolated to systems of nonlinear equations, and applied to the compressible Navier-Stokes equations. Numerical experiments are given, demonstrating the robustness of the method and the effectiveness of automatic adaptivity for two model problems in viscous compressible flow.

[2] For nonlinear shock problems, the solution often exhibits sharp gradients or discontinuities, around which the solution would develop spurious Gibbs-type oscillations. These are a result of underresolution of the solution, and are separate from the oscillations resulting from a lack of robustness.

2. DPG as a minimum-residual method

Our starting point is the minimization of some measure of error over a finite-dimensional space, given an abstract variational formulation

$$\left\{\begin{array}{l}\text{Given } l \in V^*, \text{ find } u \in U \text{ such that} \\ b(u,v) = l(v), \quad \forall v \in V,\end{array}\right. \tag{1}$$

where $b(\cdot,\cdot) : U \times V \to \mathbb{R}$ is a continuous bilinear form. Throughout the paper, we assume that the trial space U and test space V are real Hilbert spaces, and denote U^* and V^* as the respective topological dual spaces. Throughout the paper, we suppose the variational problem (1) to be well-posed in the inf-sup sense. We can then identify a unique operator $B : U \to V^*$ such that

$$\langle Bu, v\rangle_{V^*\times V} \coloneqq b(u,v), \quad u \in U, v \in V$$

with $\langle\cdot,\cdot\rangle_{V^*\times V}$ denoting the duality pairing between V^* and V, to obtain the operator form of the our variational problem

$$Bu = l \quad \text{in } V^*. \tag{2}$$

We are interested in minimizing the residual over the discrete approximating subspace $U_h \subset U$

$$u_h = \underset{u_h \in U_h}{\arg\min}\, J(u_h) \coloneqq \frac{1}{2}\|l - Bu_h\|_{V^*}^2 \coloneqq \frac{1}{2} \sup_{v \in V\setminus\{0\}} \frac{|l(v) - b(u_h,v)|^2}{\|v\|_V^2}.$$

For convenience in writing, we will abuse the notation $\sup_{v\in V}$ to denote $\sup_{v\in V\setminus\{0\}}$ for the remainder of the paper. If we define the problem-dependent *energy norm*

$$\|u\|_E \coloneqq \|Bu\|_{V^*},$$

then we can equate the minimization of $J(u_h)$ with the minimization of error in $\|\cdot\|_E$.

The first order optimality condition for minimization of the quadratic functional $J(u_h)$ requires the Gâteaux derivative to be zero in all directions $\delta u \in U_h$,

$$(l - Bu_h, B\delta u)_{V^*} = 0, \quad \forall \delta u \in U, \tag{3}$$

which is nothing more than the least-squares condition enforcing orthogonality of error with respect to the inner product on V.

The difficulty in working with the first-order optimality condition (3) is that the inner product $(\cdot,\cdot)_{V^*}$ cannot be evaluated explicitly. However, we have that

$$(l - Bu_h, B\delta u)_{V^*} = \left(R_V^{-1}(l - Bu_h), R_V^{-1} B\delta u\right)_V, \tag{4}$$

where $R_V : V \to V^*$ is the Riesz map mapping elements of a Hilbert space V to elements of the dual V^* defined by

$$\langle R_V v, \delta v\rangle_{V^*\times V} \coloneqq (v, \delta v)_V.$$

Furthermore, the Riesz operator is an isometry, such that $J(u_h) = \frac{1}{2}\|l - Bu_h\|_{V^*}^2 = \frac{1}{2}\left\|R_V^{-1}(l - Bu_h)\right\|_V^2$. Thus, satisfaction of (4) is exactly equivalent to satisfaction of the original optimality conditions (3).

2.1. Optimal test functions

We define, for a given trial function $\delta u \in U$, the corresponding *optimal test function* $v_{\delta u}$

$$v_{\delta u} := R_V^{-1} B \delta u \quad \text{in } V. \tag{5}$$

By definition of the Riesz inverse, the optimality condition (4) becomes

$$\langle B u_h - l, v_{\delta u} \rangle_V = 0, \quad \forall \delta u \in U$$

which is exactly the standard variational equation in (1) with $v_{\delta u}$ as the test functions. By defining the optimal test space $V_{\text{opt}} := \{v_{\delta u} \text{ s.t. } \delta u \in U\}$, the solution of the discrete variational problem (1) with test space $V_h = V_{\text{opt}}$ minimizes the residual in the dual norm $\|B u_h - l\|_{V'}$.

The inversion of the Riesz operator required to determine optimal test functions is done through the solution of the auxiliary variational problem

$$(v_{\delta u}, \delta v)_V = b(\delta u, \delta v), \quad \forall \delta v \in V. \tag{6}$$

In general, for conforming finite element methods, test functions are continuous over the entire domain, and hence solving variational problem (6) for each optimal test function is a global operation over the entire mesh, rendering the method impractical. The development of discontinous Galerkin (DG) methods allows us to avoid this by adopting basis functions which are discontinuous over elements. In particular, the use of discontinuous test functions δv and a *localizable* norm $\|v\|_{V(\Omega_h)}^2 = \sum_{K \in \Omega_h} \|v\|_{V(K)}^2$, (where $\|v\|_{V(K)}$ is a norm over the element K), reduces the problem of determining global optimal test functions in (6) to local problems that can be solved in an element-by-element fashion.

We note that solving (6) on each element exactly is still impossible since it amounts to inverting the infinite-dimensional Riesz map R_V. Instead, optimal test functions are approximated using the standard Bubnov-Galerkin method on an "enriched" subspace $\tilde{V} \subset V$ such that $\dim(\tilde{V}) > \dim(U_h)$ elementwise [9, 1]. We assume in this paper that the approximation error in optimal test functions is negligible, and refer to [10] for estimating the effects of approximation error on DPG.

Finally, the DPG method boasts several attractive properties:

- DPG provides a symmetric positive-definite stiffness matrix. Let $\{\phi_j\}_{j=1}^m$ be a basis for U_h, and $\{v_i\}_{i=1}^n$ a basis for V_h, such that $n > m$. Then, for $B_{ji} = b(\phi_j, v_i)$ and $l_i = \ell(v_i)$, DPG solves the "internally preconditioned" algebraic least squares system for degrees of freedom u

  $$\left(B^T R_V^{-1} B\right) u = \left(B^T R_V^{-1}\right) l,$$

  where, under a localizable norm and discontinuous test functions, R_V^{-1} is block diagonal.

- DPG delivers the best approximation error in the "energy norm" [11, 1, 12]

  $$\|u - u_h\|_E = \inf_{w \in U_h} \|u - w\|_E\,, \tag{7}$$

  where the energy norm $\|u\|_E$ is defined

  $$\|u\|_E := \|Bu\|_{V'} = \sup_{v \in V} \frac{b(u,v)}{\|v\|_V}. \tag{8}$$

- DPG inherits the natural property of minimum-residual methods that the energy error can be determined without knowing the exact solution – $\|u-u_h\|_{U,E} = \|Bu - Bu_h\|_{V'} = \left\|R_V^{-1}\left(l - Bu_h\right)\right\|_V$ shows that the energy error is simply the residual measured in a specific norm. The energy error for DPG can be determined by computing the *error representation function* $e := R_V^{-1}\left(l - Bu_h\right)$ through
$$(e, \delta v)_V = b(u - u_h, \delta v) = l\left(\delta v\right) - b(u_h, \delta v)$$
and measuring $\|e\|_V$. We can compute the squared norm over the domain $\|e\|^2_{V(\Omega_h)}$ as the sum of element contributions $\sum_{K\in\Omega_h} \|e\|^2_{V(K)}$, and we define $e_K^2 := \|e\|^2_{V(K)}$ as a local error indicator with which we can drive adaptive mesh refinement.

2.2. The ultra-weak variational formulation

We note that the above theory holds for any variational formulation that allows for discontinuous test functions. Though alternative discretizations have been successful [13], we focus primarily on the ultra-weak variational formulation.

The ultra-weak variational formulation begins with a first order system $Au = f$. After multiplying by a test function v, we integrate all equations by parts, such that over one element, we have
$$(Au, v)_{L^2(K)} = (u, A_h^* v)_{L^2(K)} + \langle \widehat{u}, v\rangle_{\partial K}\,.$$
$A_h^* v$ is the broken formal adjoint, which corresponds to the adjoint operator acting element-wise on v. In traditional DG methods, the numerical flux $\widehat{u}$ is constructed as some function of solution values on elements adjacent to ∂K. Under the ultra-weak formulation, these numerical fluxes are identified as additional unknowns, akin to hybridized discontinuous Galerkin (HDG) methods [14].[3]

We define the mesh Ω_h as the partitioning of Ω into non-overlapping elements $K_j, j = 1, \ldots, N^{\text{el}}$ through $\Omega_h = \cup_{j=1}^{N^{\text{el}}} K_j$. We define also the mesh "skeleton" by $\Gamma_h = \cup_{j=1}^{N^{\text{el}}} \partial K_j$; the set of all element faces/edges. The ultra weak formulation is given as the sum of the independent element contributions in the mesh Ω_h, which, under proper regularity assumptions, yields the bilinear form $b\left((u, \widehat{u}), v\right)$
$$b\left((u, \widehat{u}), v\right) = \sum_{K\in\Omega_h} (u, A_h^* v)_{L^2(K)} + \langle \widehat{u}, v\rangle_{\partial K} = (u, A_h^* v)_{L^2(\Omega)} + \langle \widehat{u}, [\![v]\!]\rangle_{\Gamma_h}\,,$$
where $[\![v]\!]$ is the jump of v across all edges that make up the skeleton Γ_h. We refer to [15, 16, 17] for a detailed discussion of the proper energy setting and spaces for each variable, as well as detailed analysis of well-posedness of the ultra-weak variational formulation for a large range of problems.

3. DPG for convection-diffusion

The convection-diffusion problem can be given as a first order system:
$$\begin{aligned} \nabla \cdot (\beta u - \sigma) &= f \\ \frac{1}{\epsilon}\sigma - \nabla u &= 0, \end{aligned}$$

[3] While the boundary term resulting from integration by parts of a gradient (the numerical trace), is typically identified as an additional unknown, the boundary term resulting from integration by parts of a divergence (the numerical flux) is not, and is approximated using a DG flux. The ultra-weak variational formulation identifies both as independent unknowns.

with inflow boundary conditions $u|_{\Gamma_{\text{in}}} = u_{\text{in}}$, and outflow boundary conditions $u|_{\Gamma_{\text{out}}} = u_{\text{out}}$. The inflow and outflow boundaries are defined

$$\begin{aligned} \Gamma_{\text{in}} &\coloneqq \{x \in \Gamma; \beta_n(x) \leq 0\}, \quad \text{(inflow)} \\ \Gamma_{\text{out}} &\coloneqq \{x \in \Gamma; \beta_n(x) > 0\}, \quad \text{(outflow)} \end{aligned}$$

On domain Ω, with mesh Ω_h and mesh skeleton Γ_h, the ultra-weak variational formulation for convection-diffusion is

$$b\left(\left(u, \sigma, \widehat{u}, \widehat{f}_n\right), (v, \tau)\right) = (u, \nabla \cdot \tau - \beta \cdot \nabla v)_{\Omega_h} + \left(\sigma, \epsilon^{-1}\tau + \nabla v\right)_{\Omega_h} - \langle [\![\tau \cdot n]\!], \widehat{u}\rangle_{\Gamma_h} + \left\langle \widehat{f}_n, [\![v]\!] \right\rangle_{\Gamma_h}.$$

with $v \in H^1$ and $\tau \in H(\text{div}, \Omega_h)$.

3.1. Inflow boundary condition and choice of test norm

Given the above properties of DPG, we can see that the choice of test norm will determine the behavior of DPG; however, the proper choice of test norm can be unclear. There exists a canonical graph norm for the ultra-weak variational formulation under which well-posedness can be proven for a large class of problems, and near-optimality is observed in numerical experiments [12, 16, 17]. However, for the specific problem of convection-diffusion, the resolution of optimal test functions resulting from the graph norm prove very difficult to approximate [18]. In [2], Demkowicz and Heuer introduced guidelines for the construction of alternative test norms based on energy estimates for the adjoint equation. We follow [3], where, in addition to constructing an alternative test norm, a new inflow boundary condition is introduced in order to regularize the adjoint equation.

Typical boundary conditions for the convection-diffusion problem enforce a Dirichlet condition on both inflow and outflow. We adopt instead a“conserved flux” inflow boundary condition,

$$\beta_n u - \epsilon \frac{\partial u}{\partial n} \approx u_{\text{in}}.$$

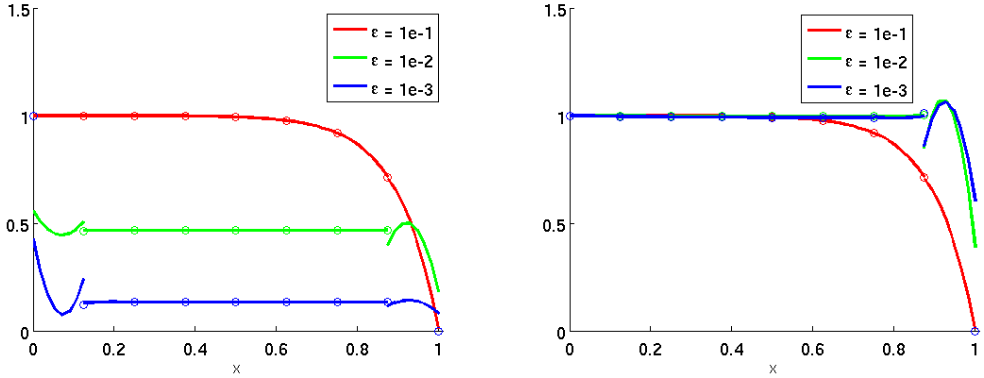
This corresponds to the imposition of a boundary condition on the conserved flux $\widehat{f}_n = \beta_n u - \sigma_n$. For small ϵ and most problems of interest, both $\frac{\partial u}{\partial n}$ and ϵ tend to be small, such that the above is a good approximation of $u|_{\Gamma_{\text{in}}} = u_{\text{in}}$.

Under this boundary condition, the adjoint problem is regularized such that its solutions are smoother, which in turn improves the stability of the DPG method [3]. We illustrate this improvement with a simple numerical experiment; Figure 1 demonstrates the behavior of the DPG method under both the standard Dirichlet and the new conserved flux boundary conditions. In both cases, the test norm used is the 1D Sobolev norm $\|(v,\tau)\|_V \coloneqq \|v\|_{H^1} + \|\tau\|_{H^1}$. The left figure shows the behavior of DPG under a Dirichlet boundary condition; degeneration of the solution is observed as ϵ decreases. In contrast, under the new conserved flux boundary condition, as ϵ decreases, the solution converges to the L^2 projection of the exact solution onto our trial space.

The test norm for convection-diffusion in multiple dimensions is defined elementwise on K as

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

DPG under $\|\cdot\|_V$ both delivers robust control of the L^2 error in the variables u and σ and minimizes approximation error in the solution of (6).



(a) Dirichlet boundary condition $u(0) = 1$

(b) "Conserved flux" boundary condition $u(0) - \epsilon u'(0) = 1$

Figure 1: Behavior of the 1D DPG method for convection diffusion under both standard Dirichlet and conserved flux boundary conditions.

3.2. Anisotropic refinement

Isotropic adaptive mesh refinement has shown itself to be an effective way to resolve isolated solution features with large gradients, such as point singularities [19, 20]. However, for the resolution of phenomena such as shocks or boundary layers, anisotropic mesh refinement can resolve solution features for a much lower cost per degree-of-freedom, due to the fact that boundary layers in n-dimensions are primarily phenomena supported over $n-1$ dimensions.

As a least squares method, DPG already includes a natural error indicator with which to drive adaptive mesh refinement. To introduce anisotropic refinements, we need to introduce an anisotropy indicator in order to detect in which direction solution features are aligned. In general, a test norm can be expressed as the sum of normed quantities, both scalar and vector valued. If we restrict ourselves to quadrilateral elements for the moment, a general anisotropy indicator for DPG can be constructed by evaluating the $L^2(\Omega)$ norms of the individual components of vector valued terms in the test norm.

Under the robust test norm derived in this chapter for the convection-dominated diffusion problem, we can define x and y error contributions over a single element

$$
\begin{aligned}
e_{x,K} &= \epsilon \left\| \frac{\partial v}{\partial x} \right\|^2_{L^2(K)} + \left\| \tau_x \right\|^2_{L^2(K)} \\
e_{y,K} &= \epsilon \left\| \frac{\partial v}{\partial y} \right\|^2_{L^2(K)} + \left\| \tau_y \right\|^2_{L^2(K)} .
\end{aligned}
$$

We define the anisotropy indicator as the ratio $r_K = \frac{e_{x,K}}{e_{y,K}}$, and implement a simple refinement scheme following [21]. Given some anisotropic threshold ϵ_r, if $r_K > \epsilon_r$, then we can conclude that the error in the x direction is large compared to the y direction, and we anisotropically refine the element along the x-axis. Likewise, if $r_K < \frac{1}{\epsilon_r}$, this implies that the opposite is true, and we refine the element anisotropically along the y-axis.

4. DPG for nonlinear problems

In this chapter, we extend DPG to the nonlinear setting and apply it to two problems in computational fluid dynamics. We take as our starting point a nonlinear variational formulation $b(u,v) = l(v)$, which is linear in v, but not in u. An appropriate linearization gives

$$b_u(\Delta u, v) = l(v) - b(u,v),$$

where $b_u(\Delta u, v)$ is the linearization of $b(u,v)$ with respect to u. Let $B(u)$ and $B_u \Delta u$ be the variational operators associated with $b(u,v)$ and $b_u(\Delta u, v)$, respectively. We define two additional measures:

$$\begin{aligned}
\|\Delta u\|_E &\coloneqq \|B_u \Delta u\|_{V'} = \left\|R_V^{-1} B_u \Delta u\right\|_V = \sup_{v\in V} \frac{b_u(\Delta u, v)}{\|v\|_V} \\
\|R(u)\|_E &\coloneqq \|B(u) - l\|_E = \|B(u) - l\|_{V'} = \left\|R_V^{-1} B(u) - l\right\|_V = \sup_{v\in V} \frac{b(u,v) - l(v)}{\|v\|_V}
\end{aligned}$$

These two quantities are measures of the linearized update Δu and the nonlinear residual in the appropriate norm in the dual space V'. The first will be used to measure convergence of a nonlinear solution scheme to a stable discrete solution, while the second will be used to assess the convergence of the discrete solution to the continuous solution.

4.1. Nonlinear solution strategies

The solution of a nonlinear problem is most commonly found using an iterative method, where a series of solutions to linear problems is expected to converge to the nonlinear solution. We use two main methods to iterate to a nonlinear solution.

- **(Damped) Newton iteration :** Given the linearized system $b_u(\Delta u, v) = b(u,v) - l(v)$, we begin with some initial guess $u \coloneqq u_0$ and solve for Δu_0. The process is then repeated with $u \coloneqq u_{i+1} \coloneqq u_i + \alpha_i \Delta u_i$, where $\alpha_i \in (0,1]$ is some damping parameter that may limit the size of the Newton step in order to optimize the rate of convergence or preserve physicality of the solution. The solution is considered converged when $\|\Delta u\|_E \leq tol$.

- **Pseudo-time stepping:** An alternative method for the solution of steady-state systems is to use a pseudo-timestepping method. The most common approach is to discretize the equations in time using a stable, implicit method — if $Lu = f$ is our nonlinear problem and L_u is the linearization of the nonlinear operator L with respect to u, then the pseudo-timestepping method solves at each discrete time t_i

  $$\frac{\partial u}{\partial t} + Lu = f \rightarrow \frac{u(t_i) - u(t_{i-1})}{\Delta t} + L_{u(t_i)} \Delta u(t_i) = (f - Lu(t_i)).$$

  The solution at the next timestep is then set $u(t_{i+1}) \coloneqq u(t_i) + \Delta u(t_i)$. This procedure is then repeated for the next timestep t_{i+1} until the transient residual decreases such that $\|u(t_i) - u(t_{i-1})\|_{L^2(\Omega)} = \|\Delta u(t_i)\|_{L^2(\Omega)} \leq tol.$[4]

[4] Strictly speaking, seeking the solution at each timestep involves the solution of a nonlinear problem, requiring a Newton-type iteration to solve for $u(t_i)$. For most applications, it is sufficient to approximate the nonlinear solution using a single Newton solve at each time step.

In practice, most compressible flow solvers opt for the pseudo-time step method over the direct Newton iteration due to the difficulty of convergence and sensitivity of the Newton iteration to initial guess[22, 23]. Though the convergence of the pseudo-time step is slower, the addition of the zero-order transient terms "regularizes" the problem and makes it less difficult to solve.[5]

A second class of nonlinear solvers are optimization methods. Since DPG allows for the formulation of a discrete nonlinear residual, it is possible to formulate the nonlinear DPG problem as a minimization problem and use an optimization method to solve the discrete nonlinear problem. This approach has been successfully implemented in solving compressible gas dynamics problems on uniform grids using a modified version of the ultra-weak variational formulation [24, 25]. An additional advantage of such an approach would be the more direct enforcement of physical constraints, which are treated in an ad-hoc manner in most compressible Navier-Stokes solvers.

4.2. DPG as a nonlinear minimum residual method

A recent theoretical development is the formulation of a DPG method that aims to minimize a nonlinear residual. Given two Hilbert spaces — a trial space U and test space V — our nonlinear variational formulation can be written as $b(u, v) = l(v)$, with the corresponding operator form of the formulation in V'

$$B(u) = l.$$

We can apply the steps used to derive the DPG method for linear problems to the nonlinear setting as well. Given a finite dimensional subspace $U_h \subset U$, we consider the discrete nonlinear residual

$$J(u_h) \coloneqq \frac{1}{2} \left\| R_V^{-1} \left(B(u_h) - l \right) \right\|_V^2 .$$

Our goal is to solve

$$u_h = \arg \min_{w_h \in U_h} J(w_h).$$

We note that, under convergence of any of the nonlinear iterations described above, the converged DPG solution does in fact satisfy the above minimization problem [24], and that the DPG energy error converges to the nonlinear residual. However, it is theoretically possible to accelerate convergence of a nonlinear iteration to the minimizer u_h, using the same minimum residual framework DPG is based upon.

Similarly to the linear case, we can take the Gateáux derivative to arrive at a necessary condition for u_h to minimize $J(u_h)$

$$\langle J'(u_h), \delta u_h \rangle = \left(R_V^{-1} \left(B(u_h) - l \right), R_V^{-1} B'(u_h; \delta u_h) \right)_V , \quad \delta u_h \in U_h.$$

As the above is a nonlinear equation, we seek its solution through linearization. Differentiating a second time in u, we arrive at

$$\begin{aligned} \langle J''(u_h), \Delta u_h \rangle = & \langle B'(u_h; \Delta u_h), B'(u_h; \delta u_h) \rangle_V \\ & + \langle \left(B(u_h) - l \right), B''(u_h; \delta u_h, \Delta u_h) \rangle_V \\ = & \left(R_V^{-1} B'(u_h; \Delta u_h), R_V^{-1} B'(u_h; \delta u_h) \right)_V \\ & + \left(R_V^{-1} \left(B(u_h) - l \right), R_V^{-1} B''(u_h; \delta u_h, \Delta u_h) \right)_V \end{aligned}$$

[5] The addition of a zero-order term "regularizes" an equation by adding to it a positive-definite L^2 projection operator. In the limit as $\Delta t \to 0$, the solution at t_i will simply return the L^2 projection of the solution at the previous timestep.

where $B''(u_h; \delta u_h, \Delta u_h)$ denotes the Hessian of $B(u_h)$, evaluated using both δu_h and Δu_h.

Examining the above formulation, we note that DPG as applied to the linearized problem produces the term $\left(R_V^{-1} B'(u_h; \Delta u_h), R_V^{-1} B'(u_h; \delta u_h)\right)_V$. However, in approaching the nonlinear problem through the minimization of the discrete residual, we gain a second-order term involving the Hessian

$$\left(R_V^{-1}\left(B(u_h) - l\right), R_V^{-1} B''(u_h; \delta u_h, \Delta u_h)\right)_V.$$

The evaluation of this term can be done in a computationally efficient manner — if we define the image of the nonlinear residual under the Riesz inverse

$$v_{R(u)} = R_V^{-1}\left(B(u_h) - l\right),$$

then we can compute this additional term through

$$\left(v_{R(u)}, R_V^{-1} B''(u_h; \delta u_h, \Delta u_h)\right)_V = \left\langle v_{R(u)}, B''(u_h; \delta u_h, \Delta u_h)\right\rangle_V$$

which can be computed in the same fashion as a Bubnov-Galerkin stiffness matrix. This addition, though not positive definite, is symmetric due to the nature of second order derivatives.

We note that we have not implemented this Hessian-based nonlinear solver in the numerical experiments to follow, and instead plan to do so in future work.

4.3. DPG as a Gauss-Newton approximation

The above Hessian-based nonlinear DPG method essentially replaces a simple linearization of with a higher-order expansion of the residual $J(u_h)$. However, the application of DPG to the linearized equations actually yields a Gauss-Newton method for minimizing $J(u_h)$.

Recall that the discrete DPG method applied to a linear problem solves

$$B^T {R_V}^{-1} B u - B^T {R_V}^{-1} l = 0$$

where B is the rectangular matrix of size $m \times n$ resulting from the variational form $b(u, v)$ (where $m = \dim(U_h)$ and $n = \dim(V_h)$), and R_V is the Gram matrix of dimension $n \times n$ resulting from choice of norm $\|\cdot\|_V$ and inner product on V. If we have the nonlinear equation $B(u_h) - l$ and linearize it, we have $B_u \triangle u_h = r(u_h)$, where $r(u_h) = l - B(u_h)$ is the residual. The solution of the linearized equation by DPG minimizes then $\|r(u_h) - B_u \triangle u_h\|_{V'}^2$, which we can recognize as linear least squares equations from which the Gauss-Newton method is derived [26].

Initial experiments with Hessian-based nonlinear DPG under Burgers' equation indicate that eigenvalues of the stiffness matrix can become negative if the Hessian is applied. While the Gauss-Newton iteration may not converge as rapidly as the Hessian-based version of DPG, it does guarantee positive eigenvalues of the stiffness matrix, such that Δu_h is always a descent direction.

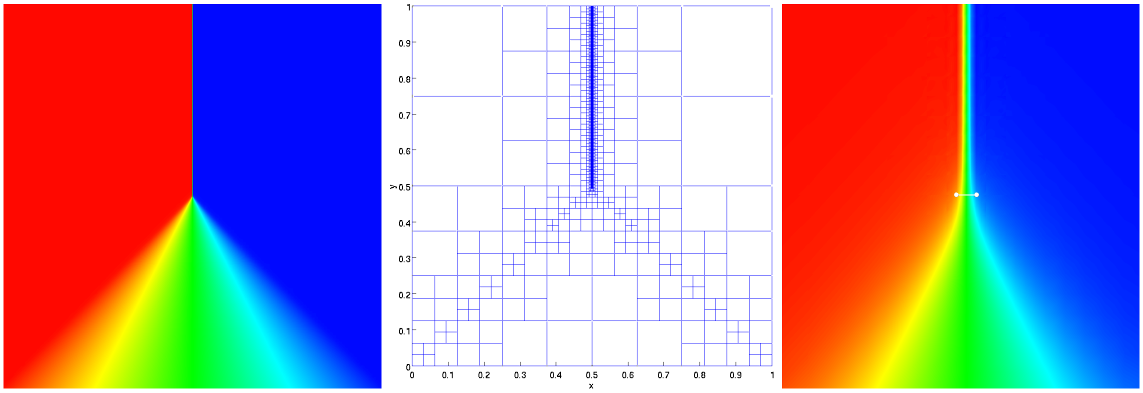
4.4. A viscous Burgers equation

We will illustrate the application of DPG to nonlinear problems using a viscous Burgers' equation on domain $\Omega = [0, 1]^2 \in \mathbb{R}^2$ [24]

$$\frac{\partial\left(u^2/2\right)}{\partial x} + \frac{\partial u}{\partial y} - \epsilon \Delta u = f.$$

If we remove the viscous term, the above problem reduces to the form of the 1D transient Burgers equation, whose solution we can determine via the method of characteristics. For boundary conditions

$$u(x, y) = 1 - 2x, \quad x = 0, y = 0,$$



(a) Solution u (b) Adaptive mesh (c) Zoomed solution at the shock

Figure 2: Solution and adaptive mesh after 9 refinements, and zoom view at point (.5,.5) with shock formation and $1e-3$ width line for reference.

the solution forms a shock discontinuity starting at $(x, y) = (.5, .5)$, which then propagates upward in the y-direction. The addition of the viscous term smears this discontinuity, leading to a solution with a smooth shock of width ϵ.

We begin by writing the equation as a first order system. Defining $\beta(u) = (u/2, 1)$, the above Burgers equation can be written as

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

Analogously to the convection-diffusion problem, the DPG nonlinear variational formulation can then be given

$$b\left(\left(u, \sigma, \widehat{u}, \widehat{f}_n\right), (v, \tau)\right) = (u, \nabla \cdot \tau - \beta(u) \cdot \nabla v) + \left(\sigma, \frac{1}{\epsilon}\tau + \nabla v\right) + \langle \widehat{u}, \tau \cdot n \rangle + \left\langle \widehat{f}_n, v \right\rangle = (f, v)$$

Linearizing the above then gives us

$$\begin{aligned} b_u\left(\left(\Delta u, \sigma, \widehat{u}, \widehat{f}_n\right), (v, \tau)\right) &= \left(\Delta u, \nabla \cdot \tau - \begin{bmatrix} u \\ 1 \end{bmatrix} \cdot \nabla v\right) + \left(\sigma, \frac{1}{\epsilon}\tau + \nabla v\right) + \langle \widehat{u}, \tau \cdot n \rangle + \left\langle \widehat{f}_n, v \right\rangle \\ &= (u, \nabla \cdot \tau - \beta(u) \cdot \nabla v) \end{aligned}$$

Notice that the nonlinear term is only dependent on u, and thus there is no need to linearize in the variables σ, $\widehat{u}$, and $\widehat{f}_n$. Since the linearized Burgers' equation is of the form of a convection-diffusion problem with non-homogeneous load, we adopt the test norm described in Section 3.1 with convection vector $\beta = (u, 1)$.

Recall that we did not linearize in the flux variables $\widehat{u}$ and $\widehat{f}_n$, so we can directly apply the nonlinear boundary conditions to our variational formulation. Additionally, since Burgers' equation does not have any physical constraints, we can employ a direct Newton iteration to solve the nonlinear equation. The adaptivity algorithm is identical to the greedy algorithm described previously, except that the linear solve is replaced by a nonlinear solve. The results of an adaptive simulation are shown in Figure 2 for $\epsilon = 1e-4$.

For the solution of Burgers' equation, adaptivity began on a cubic 4×4 mesh. 9 iterations of adaptive mesh refinement were performed, and the resulting mesh and zoomed solution are also displayed in Figure 2, demonstrating a fully resolved smooth transition of the solution into a smeared shock.

5. The compressible Navier-Stokes equations

We briefly review the 2D compressible Navier-Stokes equations and formulate DPG for the nonlinear system.

- **Conservation equations**

$$\begin{aligned}
\nabla \cdot \begin{bmatrix} \rho u_1 \\ \rho u_2 \end{bmatrix} &= 0 \\
\nabla \cdot \begin{bmatrix} \rho u_1^2 + p \\ \rho u_1 u_2 \end{bmatrix} &= \nabla \cdot (\vec{\sigma_{i1}}) \\
\nabla \cdot \begin{bmatrix} \rho u_1 u_2 \\ \rho u_2^2 + p \end{bmatrix} &= \nabla \cdot (\vec{\sigma_{i2}}) \\
\nabla \cdot \begin{bmatrix} ((\rho e) + p) u_1 \\ ((\rho e) + p) u_2 \end{bmatrix} &= \nabla \cdot \left[\boldsymbol{\sigma} \boldsymbol{u} + \vec{q} \right],
\end{aligned}$$

  where $\boldsymbol{\sigma}$ is the stress tensor whose ijth term is σ_{ij}, and $\boldsymbol{u}$ is the vector $(u_1, u_2)^T$.

- **Newtonian fluid laws**

  We represent $\boldsymbol{\sigma}$ using the Newtonian fluid law

$$\sigma_{ij} = \mu(u_{i,j} + u_{j,i}) + \lambda u_{k,k} \delta_{ij}$$

  where μ is viscosity and λ is bulk viscosity. We can invert the stress tensor under isotropic and plane strain assumptions to get

$$\frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u}\right) = \frac{1}{2\mu} \sigma_{ij} - \frac{\lambda}{4\mu(\mu + \lambda)} \sigma_{kk} \delta_{ij}$$

  We also have

$$\frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u}\right) = \nabla \boldsymbol{u} - \boldsymbol{\omega}$$

  where $\boldsymbol{\omega}$ is the antisymmetric part of the infinitesimal strain tensor:

$$\boldsymbol{\omega} = \frac{1}{2}\left(\nabla \boldsymbol{u} - \nabla^T \boldsymbol{u}\right).$$

  Thus our final form is

$$\nabla \boldsymbol{u} - \boldsymbol{\omega} = \frac{1}{2\mu} \boldsymbol{\sigma} - \frac{\lambda}{4\mu(\mu + \lambda)} \operatorname{tr}(\boldsymbol{\sigma}) \boldsymbol{I}.$$

  Notice that $\boldsymbol{\omega}$ is implicitly defined to be the antisymmetric part of ∇u by taking the symmetric part of the above equation.

  We note that, though this is a standard approach in solid mechanics, it is nonstandard compared to the usual finite element and DG approaches to the viscous stresses. We adopt such an approach to better mirror our experiences with the convection-diffusion equation [2, 3].

- **Fourier's heat conduction law**

  We assume Fourier's law

  $$\vec{q} = \kappa \nabla T,$$

  We introduce here the Prandtl number here as well

  $$\mathrm{Pr} = \frac{\gamma c_v \mu}{\kappa}.$$

  In this case, we assume a constant Prandlt number, which implies that the heat conductivity κ is proportional to viscosity μ.

5.1. Nondimensionalization

To nondimensionalize our equations, we introduce nondimensional quantities for length, density, velocity, temperature, and viscosity.

$$\boldsymbol{x}^* = \frac{\boldsymbol{x}}{L}, \qquad \rho^* = \frac{\rho}{\rho_\infty}, \qquad u_1^* = \frac{u_1}{V_\infty}, \qquad u_2^* = \frac{u_2}{V_\infty}, \qquad T^* = \frac{T}{T_\infty}, \qquad \mu^* = \frac{\mu}{\mu_\infty}$$

Pressure, internal energy, and bulk viscosity are then nondimensionalized with respect to the above variables

$$p^* = \frac{p}{\rho_\infty V_\infty^2}, \qquad \iota^* = \frac{\iota}{V_\infty^2}, \qquad \lambda^* = \frac{\lambda}{\mu_\infty}$$

We introduce, for convenience, the Reynolds number

$$\mathrm{Re} = \frac{\rho_\infty \mathrm{V}_\infty \mathrm{L}}{\mu_\infty}$$

and the reference (free stream) Mach number

$$M_\infty = \frac{V_\infty}{\sqrt{\gamma(\gamma-1)c_v T_\infty}}$$

Note that

$$a = \sqrt{\frac{\gamma p_\infty}{\rho_\infty}} = \sqrt{\gamma p_\infty} = \sqrt{\gamma(\gamma-1)c_v T_\infty}$$

The equations take the same form as before after nondimensionalization, so long as we define new material constants

$$\tilde{\mu} = \frac{\mu^*}{\mathrm{Re}}, \qquad \tilde{\lambda} = \frac{\lambda^*}{\mathrm{Re}}, \qquad \tilde{c}_v = \frac{1}{\gamma(\gamma-1)M_\infty^2}, \qquad \tilde{\kappa} = \frac{\gamma \tilde{c}_v \tilde{\mu}}{\mathrm{Pr}}$$

From here on, we drop the * superscript and assume all variables refer to their nondimensionalized quantities.

To summarize, our system of equations in the classical variables is now

$$
\begin{aligned}
\nabla\cdot\left[\begin{array}{c}\rho u_1\\ \rho u_2\end{array}\right] &= 0\\
\nabla\cdot\left(\left[\begin{array}{c}\rho u_1^2+p\\ \rho u_1u_2\end{array}\right]-\boldsymbol{\sigma}_1\right) &= 0\\
\nabla\cdot\left(\left[\begin{array}{c}\rho u_1u_2\\ \rho u_2^2+p\end{array}\right]-\boldsymbol{\sigma}_2\right) &= 0\\
\nabla\cdot\left(\left[\begin{array}{c}((\rho e)+p)u_1\\ ((\rho e)+p)u_2\end{array}\right]-\boldsymbol{\sigma}\mathbf{u}-\vec{q}\right) &= 0\\
\frac{1}{2\mu}\boldsymbol{\sigma}-\frac{\lambda}{4\mu(\mu+\lambda)}\mathrm{tr}(\boldsymbol{\sigma})\boldsymbol{I} &= \nabla\mathbf{u}-\mathrm{Re}\,\boldsymbol{\omega}\\
\frac{1}{\kappa}\vec{q} &= \nabla T
\end{aligned}
$$

We strongly enforce symmetry of the stress tensor $\boldsymbol{\sigma}$ by setting $\sigma_{21}=\sigma_{12}$. Additionally, we have scaled the antisymmetric tensor $\boldsymbol{\omega}$ by the Reynolds number to ensure that $\boldsymbol{\omega}=O(1)$ for all ranges of Re.

5.2. Linearization

As the equations for viscous compressible flow are nonlinear and cannot be solved exactly, we linearize the equations and adopt an iterative procedure for approximating the nonlinear solution.[6] We outline the linearization of both the conservation and stress laws in this section.

5.2.1. Conservation laws

The Navier-Stokes conservation laws can be written as

$$
\begin{aligned}
\nabla\cdot\left[\begin{array}{c}\rho u_1\\ \rho u_2\end{array}\right] &= 0\\
\nabla\cdot\left(\left[\begin{array}{c}\rho u_1^2+p\\ \rho u_1u_2\end{array}\right]-\boldsymbol{\sigma}_1\right) &= 0\\
\nabla\cdot\left(\left[\begin{array}{c}\rho u_1u_2\\ \rho u_2^2+p\end{array}\right]-\boldsymbol{\sigma}_2\right) &= 0\\
\nabla\cdot\left(\left[\begin{array}{c}((\rho e)+p)u_1\\ ((\rho e)+p)u_2\end{array}\right]-\boldsymbol{\sigma}_1\cdot\boldsymbol{u}-\boldsymbol{\sigma}_2\cdot\boldsymbol{u}-\vec{q}\right) &= 0,
\end{aligned}
$$

where $\boldsymbol{\sigma}_1$ is the ith column or row of $\boldsymbol{\sigma}$, or more generally, if we group our Eulerian and stress variables into the vector variables $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$, respectively

$$
\nabla\cdot(F_i(\boldsymbol{U})-G_i(\boldsymbol{U},\boldsymbol{\Sigma})=0,\qquad i=1,\dots,4,
$$

[6] We note that it is possible to linearize the strong form of the equations and then derive a linearized weak form, instead of linearizing the weak form of the nonlinear equations, which is done here. The two formulations are equivalent; however, extraneous linearization of the fluxes is avoided using the latter approach.

where $F_i$and G_i are given as

$$F_1 = \begin{bmatrix} \rho u_1 \\ \rho u_2 \end{bmatrix}, \quad G_1 = 0$$

$$F_2 = \begin{bmatrix} \rho u_1^2 + p \\ \rho u_1 u_2 \end{bmatrix}, \quad G_2 = \boldsymbol{\sigma}_1$$

$$F_3 = \begin{bmatrix} \rho u_1 u_2 \\ \rho u_2^2 + p \end{bmatrix}, \quad G_3 = \boldsymbol{\sigma}_2$$

$$F_4 = \begin{bmatrix} ((\rho e) + p) u_1 \\ ((\rho e) + p) u_2 \end{bmatrix}, \quad G_4 = \boldsymbol{\sigma}_1 \cdot \boldsymbol{u} + \boldsymbol{\sigma}_2 \cdot \boldsymbol{u} + \vec{q}$$

The variational form restricted to a single element gives

$$\langle \widehat{F}_i \cdot n, v \rangle - \int_K (F(\boldsymbol{U}) - G_i(\boldsymbol{U}, \boldsymbol{\Sigma})) \cdot \nabla v_i = 0, \qquad i = 1, \dots, 4$$

and the variational form over the entire domain is given by summing up the element-wise contributions.

The presence of terms such as $\boldsymbol{\sigma}_i \cdot \boldsymbol{u}$ means that we will need to linearize in the stress variables $\sigma_i j$ in addition to our Eulerian quantities. Since fluxes and traces are linear functions of the unknowns, we do not need to linearize them. Instead, fluxes $\widehat{F}_{i,n}$ and traces $\widehat{u}, \widehat{v}, \widehat{T}$ will represent normal traces and traces of the accumulated nonlinear solution. The linearized variational formulation is thus

$$\begin{aligned} \langle \widehat{F}_i \cdot n, v \rangle - \int_K & (F_{i,\boldsymbol{U}}(\boldsymbol{U}) \cdot \Delta \boldsymbol{U} - G_{i,\boldsymbol{U}}(\boldsymbol{U}, \boldsymbol{\Sigma}) \cdot \Delta \boldsymbol{U} - G_{i,\boldsymbol{\Sigma}}(\boldsymbol{U}, \boldsymbol{\Sigma}) \cdot \Delta \boldsymbol{\Sigma}) \cdot \nabla v_i \\ = \int_K & (F_i(\boldsymbol{U}) - G_i(\boldsymbol{U})) \cdot \nabla v_i \\ i = 1, \dots, 4 & \end{aligned}$$

where $F^i_{j,\boldsymbol{U}}$, $G^i_{j,\boldsymbol{U}}$, and $G^i_{j,\boldsymbol{\Sigma}}$ are the Eulerian and two viscous Jacobians (linearized w.r.t. the Eulerian/viscous variables), respectively.

5.2.2. Viscous equations

We have two equations left to linearize - the constitutive laws defining our viscous stresses and heat flux terms.

$$\frac{1}{2\mu}\boldsymbol{\sigma} - \frac{\lambda}{4\mu(\mu+\lambda)} \mathrm{tr}(\boldsymbol{\sigma})\boldsymbol{I} + \mathrm{Re}\boldsymbol{\omega} = \nabla \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\frac{1}{\kappa} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \nabla T$$

We treat the first tensor equation as two vector equations by considering each column:

$$\frac{1}{2\mu} \begin{bmatrix} \sigma_{11} \\ \sigma_{12} \end{bmatrix} - \frac{\lambda}{4\mu(\mu+\lambda)} \begin{bmatrix} \sigma_{11} + \sigma_{22} \\ 0 \end{bmatrix} + \mathrm{Re} \begin{bmatrix} 0 \\ -\omega \end{bmatrix} - \nabla \mathrm{u}_1 = 0$$

$$\frac{1}{2\mu} \begin{bmatrix} \sigma_{12} \\ \sigma_{22} \end{bmatrix} - \frac{\lambda}{4\mu(\mu+\lambda)} \begin{bmatrix} 0 \\ \sigma_{11} + \sigma_{22} \end{bmatrix} + \mathrm{Re} \begin{bmatrix} \omega \\ 0 \end{bmatrix} - \nabla \mathrm{u}_2 = 0$$

Since all equations are linear in variables q_1, q_2, w for all combinations of variables, we do not need to linearize any equations in q_1, q_2, w. We do not linearize the viscosities μ and λ, but instead set them based on the power law and the solution at the previous timestep for simplicity.

5.3. Test norm

Recall that the test norm adopted for convection-diffusion is defined elementwise as

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

This test norm is extrapolated to the Navier-Stokes equations as follows: we denote the vector of H^1 test functions as $\boldsymbol{V} = \{v_1, v_2, v_3, v_4\}$, and similarly for $\boldsymbol{W} = \{\tau_1, \tau_2, \tau_3\}$. If $R_{\text{Euler}}(\boldsymbol{U},\boldsymbol{\Sigma})$ and $R_{\text{visc}}(\boldsymbol{U},\boldsymbol{\Sigma})$ are Eulerian and viscous nonlinear residuals, our formulation for the linearized Navier-Stokes equations can be written as

$$\begin{aligned}\nabla\cdot\left(A_{\text{Euler}}\boldsymbol{\delta}U - A_{\text{visc}}\boldsymbol{\delta}\Sigma\right) &= R_{\text{Euler}}(\boldsymbol{U},\boldsymbol{\Sigma})\\ E_{\text{visc}}\boldsymbol{\delta}\Sigma - \nabla\boldsymbol{\delta}U &= R_{\text{visc}}(\boldsymbol{U},\boldsymbol{\Sigma})\end{aligned}$$

with variational formulation

$$\begin{aligned}\left\langle \widehat{F}_n, \boldsymbol{V}\right\rangle_{\Gamma_h} + \left(\boldsymbol{\delta}U, \nabla\cdot\boldsymbol{W} - A_{\text{Euler}}^T\nabla\boldsymbol{V}\right) + \left\langle \widehat{\boldsymbol{U}}, \boldsymbol{W}\cdot n\right\rangle_{\Gamma_h} + \left(\boldsymbol{\delta}\Sigma, E_{\text{visc}}^T\boldsymbol{W} - A_{\text{visc}}^T\nabla\boldsymbol{V}\right) &=\\ \langle R_{\text{Euler}}(\boldsymbol{U},\boldsymbol{\Sigma}),\boldsymbol{V}\rangle + \langle R_{\text{visc}}(\boldsymbol{U},\boldsymbol{\Sigma}),\boldsymbol{W}\rangle&\end{aligned}$$

Identifying vector-valued terms in the Navier-Stokes formulation with equivalent scalar terms in the convection-diffusion equation allows us to extrapolate our test norm to systems of equations

$$\begin{aligned}\|\,(\boldsymbol{V},\boldsymbol{W})\,\|_{V,K}^2 =& \|\boldsymbol{V}\|^2 + \frac{1}{\text{Re}}\|A_{\text{visc}}^T\nabla\boldsymbol{V}\|^2 + \|A_{\text{Euler}}^T\nabla\boldsymbol{V}\|^2\\ &+ \|\nabla\cdot\boldsymbol{W} - A_{\text{Euler}}^T\nabla\boldsymbol{V}\|^2 + \min\left\{\text{Re},\frac{1}{|\text{K}|}\right\}\|E_{\text{visc}}^T\boldsymbol{W}\|^2.\end{aligned}$$

An advantage of this extrapolation approach is that the incompletely parabolic nature of the Navier-Stokes equation is taken into account; there is no diffusive term present in the mass conservation equation, and the test norm reflects that by requesting only limited regularity of v_1, the test function for the conservation equation.[7]

5.4. Boundary conditions

As a consequence of the ultra-weak variational formulation, our solution is linear in the flux and trace variables. Thus, the nonlinear boundary conditions can be applied directly to our fluxes $\widehat{f}_{i,n}, i = 1,\ldots,4$, and traces $\widehat{u}_1$, $\widehat{u}_2$, and $\widehat{T}$. Additionally, inflow boundary conditions are applied not directly to the trace variables $\widehat{u}_1$, $\widehat{u}_2$, and $\widehat{T}$, but to the fluxes $\widehat{f}_{i,n}$. Extrapolating from the convection-diffusion problem, this allows us to use a stronger test norm without experiencing adverse effects for smaller diffusion/higher Reynolds numbers, as described in Section 3.1 and [2, 3].

6. Nonlinear solver

For our nonlinear solver, we use a pseudo-time stepping approach to iterate to a steady state solution, along with a greedy refinement scheme to eliminate spatial discretization error. We cover briefly the details of the pseudo-timestepping method in this section.

[7] The situation is analogous to using the full $H^1(\Omega_h)$ norm for the pure convection equation — the optimal test norm $\|v\|_V = \|\beta\cdot\nabla v\| + \|v\|$ implies only streamline regularity, whereas taking $\|v\|_V = \|\nabla v\| + \|v\|$ implies stronger regularity on the test space V than the graph norm. Consequently, convergence is suboptimal for DPG applied to the convection problem under the $H^1(\Omega_h)$ test norm.

6.1. Pseudo-timestepping

The solution of the compressible Navier-Stokes equations can be quite challenging; as mentioned previously, the direct application of a Newton algorithm often will not converge, especially for high Reynolds numbers. Typically, a pseudo-timestepping algorithm is used in lieu of a full nonlinear Newton algorithm. The pseudo-timestep proceeds as follows: given the transient terms present in the conservation laws of compressible flow

$$\begin{aligned}\frac{\partial\rho}{\partial t} &+ \dots\\ \frac{\partial\left(\rho u\right)}{\partial t} &+ \dots\\ \frac{\partial\left(\rho v\right)}{\partial t} &+ \dots\\ \frac{\partial\left(\rho e\right)}{\partial t} &+ \dots,\end{aligned}$$

we discretize each time derivative using an implicit timestepping method. Though second order time discretizations have been shown to be effective [27, 28], we choose a first order backwards Euler discretization for simplicity. Due to the fact that we've chosen to solve the Navier-Stokes equations under primitive variables ρ, u, v, and T, the time terms are nonlinear as well, and must be linearized. After time discretization and linearization, we are left with a coupled system of reaction terms in the problem for our solution update at every timestep

$$\begin{gathered}A_{\text{time}}\boldsymbol{\delta}U_i + \nabla\cdot\left(A_{\text{Euler}}\boldsymbol{\delta}U_i - A_{\text{visc}}\boldsymbol{\delta}\Sigma_i\right) = R_{\text{Euler}}(\boldsymbol{U}_i, \boldsymbol{\Sigma}_i) + R_{\text{time}}(\boldsymbol{U}_{i-1}, \boldsymbol{U}_i)\\ E_{\text{visc}}\boldsymbol{\Sigma}_i - \nabla\boldsymbol{U}_i = R_{\text{visc}}(\boldsymbol{U}_i, \boldsymbol{\Sigma}_i),\end{gathered}$$

where $R_{\text{time}}(\boldsymbol{U}_{i-1}, \boldsymbol{U}_i)$ is the transient residual. Including the transient terms in our test norm as well, our final test norm is

$$\begin{aligned}\left\|\left(\boldsymbol{V},\boldsymbol{W}\right)\right\|_{V,K}^2 = \left\|A_{\text{time}}^T\boldsymbol{V}\right\| + \|\boldsymbol{V}\|^2 + \frac{1}{\text{Re}}\|A_{\text{visc}}^T\nabla\boldsymbol{V}\|^2 + \|A_{\text{Euler}}^T\nabla\boldsymbol{V}\|^2\\ + \|\nabla\cdot\boldsymbol{W} - A_{\text{Euler}}^T\nabla\boldsymbol{V}\|^2 + \min\left\{\text{Re}, \frac{1}{|\text{K}|}\right\}\|E_{\text{visc}}^T\boldsymbol{W}\|^2.\end{aligned}$$

Finally, convergence of the pseudo-timestepping method is determined by measuring the transient residual in the energy norm; in other words, we measure $\|e_{\text{time}}\|_V$, where

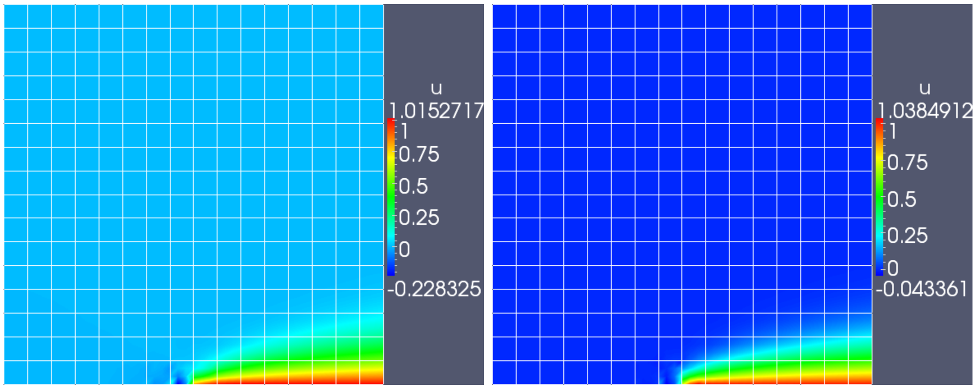
$$\left(\left(\boldsymbol{V},\boldsymbol{W}\right),\left(\boldsymbol{d}V,\boldsymbol{d}W\right)\right)_V = \left(R_{\text{time}}(\boldsymbol{U}_{i-1},\boldsymbol{U}_i),\boldsymbol{V}\right)_{L^2(\Omega)}, \quad \forall\left(\boldsymbol{d}V,\boldsymbol{d}W\right)\in V.$$

The energy norm thus provides a consistent measure in which convergence of the nonlinear iteration at each timestep, convergence of the pseudo-timestepping algorithm to steady state, and nonlinear residual can all be assessed.

6.1.1. Dependence of solution on dt

A surprising feature of pseudo-timestepping schemes for DPG is that, under the problem-dependent minimum-residual nature of the method, convergence to steady state can yield qualitatively slightly different solutions under different size timesteps. We illustrate this using a "plate" example for convection-diffusion. We consider the transient form of the conservation equation for convection-diffusion

$$\frac{\partial u}{\partial t} + \nabla\cdot\left(\beta u - \sigma\right) = f,$$



(a) $dt = .1$

(b) $dt = .01$

Figure 3: Comparison of pseudo-timestepping to steady state for a convection-diffusion problem under two different sizes of timestep.

where the stress law remains unchanged. Applying a backwards Euler time discretization, at each timestep, we will solve for the current solution u_i (as well as the current stress σ_i) given the previous timestep solution u_{i-1} under

$$\frac{u_i}{dt} + \nabla \cdot (\beta u_i - \sigma_i) = f + \frac{u_{i-1}}{dt}.$$

The conserved flux $\beta_n u - \sigma_n$ is set to freestream values on the inflow, and 0 on the top boundary $y = 1$ and half of the bottom boundary $x < .5, y = 0$. We set $u = 1$ for the boundary $x \in (.5, 1), y = 0$. For this example, $\beta = (1, 0)$ and $\epsilon = 10^{-3}$. This can be understood as the consequence of solving a "moving target" optimization problem – our variational formulation and test norm for this problem are

$$\left(u_i, \frac{1}{dt} v\right)_{L^2(\Omega)} + b\left(u_i, v\right) = l(v) + \left(u_{i-1}, \frac{1}{dt} v\right)_{L^2(\Omega)}$$
$$\left\|(\tau, v)\right\|_{V,dt}^2 = \frac{1}{dt}\left\|v\right\|_{L^2(\Omega)}^2 + \left\|v\right\|_V^2,$$

where $\left\|(\tau, v)\right\|_V$ is the test norm introduced in Section 3.1, and $b(u, v)$ and $l(v)$ are the bilinear form and load for the steady state form of the convection diffusion equation. DPG solutions minimize the functional

$$J(u_i) = \sup_{v \in V} \frac{\left(u_i - u_{i-1}, \frac{1}{dt} v\right)_{L^2(\Omega)} + b(u_i, v) - l(v)}{\left(\frac{1}{dt}\left\|v\right\|_{L^2(\Omega)}^2 + \left\|v\right\|_V^2\right)^{\frac{1}{2}}}$$

over a given mesh. As $u_{i-1} \to u_i$, which we expect to happen as the pseudo-timestepping algorithm converges to steady state, the minimized functional becomes

$$J(u_i) = \sup_{v \in V} \frac{b(u_i, v) - l(v)}{\left(\frac{1}{dt}\left\|v\right\|_{L^2(\Omega)}^2 + \left\|v\right\|_V^2\right)^{\frac{1}{2}}}.$$

While the transient portion of the residual disappears, a factor of $\frac{1}{dt}$ is still present in the test norm.[8] Thus, we can expect that the nature of the steady state solution achieved through convergence of the pseudo-time algorithm can depend on the timestep dt. We observe the same phenomena for analogous problems in compressible flow as well.

6.1.2. Adaptive time thresholding

Adaptive timestepping (also known as pseudo-transient continuation) has been implemented successfully for problems in compressible flow [28]. Typical adaptive time-stepping schemes modify the time-step based on some notion of the transient residual R_i at timestep i, such that

$$dt_{i+k} = \left(\frac{R_{i+k}}{R_i}\right)^r dt_i,$$

where $r > 1$ dictates the rate of change of the timestep based on residual reduction, and k indicates an integer interval at which to modify the timestep. However, the minimum-residual nature of the DPG method and the "moving target" problem make the effectiveness of adaptive time-stepping schemes questionable.

For our current experiments, we implement instead an adaptive time thresholding, where we adaptive decrease our convergence criterion based on the spatial energy error. Recall that, under convergence of the pseudo-timestepping algorithm, the DPG energy error converges to the measure of the nonlinear residual in the dual norm. We set convergence criterion for the pseudo-timestepping algorithm to be such that

$$\|e_{\text{time}}\|_V < \max\{\epsilon_t, \epsilon_{t,k}\},$$

where $\epsilon_{t,k}$ is the tolerance at the kth refinement iteration, and $\epsilon_t < \epsilon_{t,k}$ is an absolute tolerance. We initialize $\epsilon_{t,k}$ to ϵ_t, then based on the energy error $\|u - u_h\|_E$, we set

$$\epsilon_{t,k} = \alpha_t \|u - u_h\|_E \,.$$

Since the linearized error at a single timestep is composed of a sum of the linearization error, transient residual, and nonlinear residual, if the transient residual and linearization error are small, we expect that the nonlinear residual at that point will be sufficient to be an effective error indicator with which to drive adaptive mesh refinement. In the following numerical experiments, α_t is set to .005.

The aim of this adaptive thresholding is to relax the convergence criteria for solution of the nonlinear system at each refinement step such that the same refinement pattern is achieved with or without the use of adaptive time thresholding. Numerical experiments seem to indicate that the same refinement pattern is produced with or without the implementation of this simple adaptive thresholding scheme, though wall-clock convergence times under adaptive thresholding are faster. We hope to investigate both DPG-specific adaptive timestepping schemes and more advanced methods of balancing convergence criterion in the future.

[8] While the solution under smaller dt appears to give visually higher quality results, we stress that simply adding the term $\frac{1}{dt}\|v\|_{L^2(\Omega)}$ to the test norm under the steady state version of the convection-diffusion equations does not achieve the same effect. We are able to add this term without negative consequence due to the inclusion of the $\frac{u_i}{dt}$ term present in the variational formulation (see [2, 3] for mathematical details). Numerical experiments indicate that including $\alpha\|v\|_{L^2(\Omega)}$ in the test norm, where $\alpha > \frac{1}{dt}$, converges much more slowly to a fine-mesh reference solution than if $\alpha = \frac{1}{dt}$.

6.2. Linear solver

A clear choice for a linear solver under the ultra-weak variational formulation is static condensation, or the Schur-complement method. Given a block matrix structure of a stiffness matrix K, we can view the DPG system as

$$Ku = \left[\begin{array}{cc} A & B \\ B^T & D \end{array}\right] \left[\begin{array}{c} u_{\text{flux}} \\ u_{\text{field}} \end{array}\right] = \left[\begin{array}{c} f \\ g \end{array}\right] = l$$

where D has a block-diagonal structure, and A and D are both square matrices with $\dim A < \dim D$. This is due to the fact that, for the ultra-weak variational formulation (and for all HDG methods), the interior field degrees of freedom can be condensed out to yield a problem posed solely in terms of the coupled flux and trace degrees of freedom. The system can be reduced to yield the condensed system

$$\left(A - BD^{-1}B^T\right) u_{\text{flux}} = f - BD^{-1}g$$

where D^{-1} can be inverted block-wise. Once the globally coupled flux and trace degrees of freedom are solved for, the field degrees of freedom can be reconstructed locally. An additional advantage of the above approach is that the Schur complement maintains the same sparsity pattern implied by the connectivity of the globally coupled flux and trace degrees of freedom. Since the condensation process can be done locally, we can save memory by avoiding constructing the full stiffness matrix.

It has been shown that, unlike standard least-squares methods, DPG generates for the Poisson matrix a stiffness matrix with condition number $O(h^{-2})$ [10]. It is well known that, under standard finite element methods, if the condition number of the global stiffness matrix K is $O(h^{-2})$, the condition number of the Schur complement is $O(h^{-1})$. Additionally, through either diagonal preconditioning or matrix equilibration, the condition number of the Schur complement can often be made significantly smaller than $O(h^{-1})$, and the positive-definiteness of the resulting system allows the use of iterative solvers in solving the condensed system. Initial experiments indicate that, at least for quasi-uniform and low-order meshes, both algebraic multigrid and preconditioned conjugate gradients are able to solve the condensed system fairly rapidly. We hope to experiment further with solvers for the condensed system, and to develop multigrid methods and preconditioners for adaptive and higher order meshes under DPG.

7. Test problems

We applied the DPG method to two test problems in compressible flow – flow over a flate plate, and flow over a compression ramp. While the physics of both problems are fairly simple, they nonetheless display several features (shocks, singularities, boundary layers) that are computationally difficult to resolve without adaptivity. Furthermore, the problems themselves are not usually solved without the aid of artificial or numerical diffusion and/or shock-capturing terms, which we eschew in our application of DPG to these model problems.

The numerical parameters used are as follows for:

- DPG parameters: $p = 2$ and $\Delta p = 2$ uniformly across the mesh.
- Adaptivity parameters: Energy threshold α for refinements is $\alpha = .4$ for the Carter flat plate example and $\alpha = .5$ for the Holden ramp example.
- Time-stepping parameters: Initial timestep $\Delta t = .1$, and initial tolerance for transient residual $\epsilon_t = 1e - 7$.

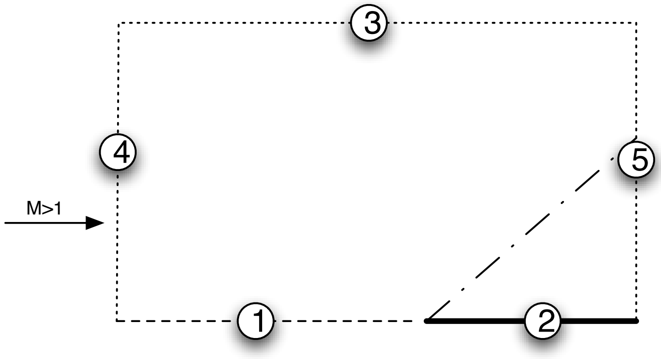


Figure 4: Carter flat plate problem.

Numerical experiments were run on a small cluster with 16 CPUs and 8GB memory, as well as on the Lonestar machine at the Texas Advanced Computing Center (TACC), using a varying number of cores, from 48 to 120 depending on problem geometry and parameters. Experiments were done using the Camellia library [29].

7.1. Numerical experiments: Carter flat plate

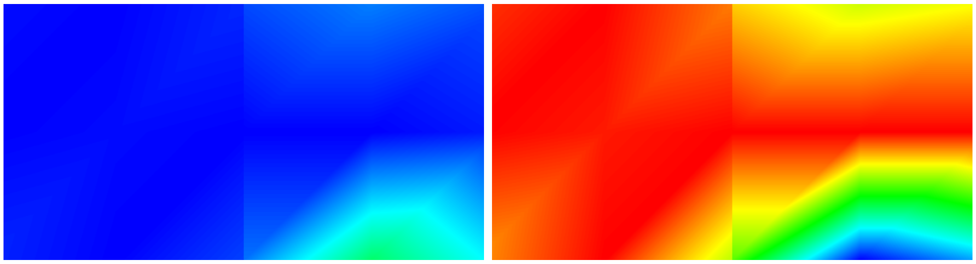
The first problem of interest is the Carter flat plate problem [30]. An infinitesmally thin flat plate disrupts a free stream flow, causing a shock to form at the tip of the plate, and a boundary layer forms and widens along the length of the plate.

1. **Symmetry boundary conditions:** $u_n = q_n = \frac{\partial u_s}{\partial n} = 0$. Here, this implies $u_2 = q_2 = \sigma_{12} = 0$. We impose the stress condition by noting that, for the flat plate geometry, if $u_2 = 0$, then at the top and bottom, with $n = (0, 1)$, $\widehat{f}_{2,n} = \sigma_{12}$, and $\widehat{f}_{4,n} = q_2$ if σ_{12} and $u_2 = 0$. They are applied here to the bottom free-stream boundary.
2. **Flat plate boundary conditions:** $u_1 = u_2 = 0$, and $T = T_w = \left[1 + (\gamma - 1)M_\infty^2/2\right] T_\infty = 2.8T_\infty$ (for Mach 3 flow). We impose these strongly on the trace variables $\widehat{u}_1, \widehat{u}_2, \widehat{T}$.
3. **Symmetry boundary conditions** are applied also to the top free-stream boundary.
4. **Inflow boundary conditions:** free stream conditions are applied here to all four fluxes $\widehat{f}_{i,n}$.
5. **Outflow boundary conditions:** the exact boundary conditions to enforce here are not universally agreed on. Many enforce $\frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} = 0$ and $\frac{\partial T}{\partial n} = 0$, while others enforce an outflow boundary condition only in regions where the flow is subsonic [31]. We adopt a "no boundary condition" outflow condition, mimicing what is done in [32].

We initialize our solution to

$$\rho = 1, \qquad u_1 = 1, \qquad u_2 = 0, \qquad T = 1$$

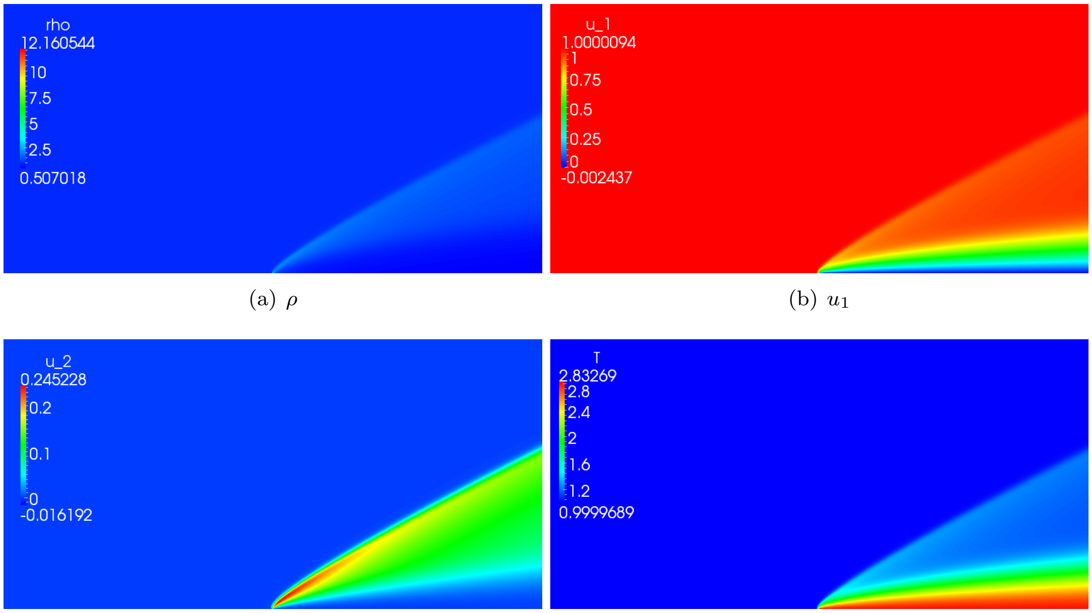
which, we also take as the freestream values for the above variables, and is consistent with what was done by Demkowicz, Oden, and Rachowicz in [27]. Stresses are set uniformly to zero. We take the computational domain to be $\Omega = [0, 2] \times [0, 1]$. Under Dirichlet wall boundary conditions for all 3 traces u_1, u_2, and T, the solution develops a singularity in the density ρ at the plate beginning, and both T and u_1 form a boundary layer along the leading edge of the plate.



(a) T

(b) u_1

Figure 5: Converged solution on 2 cells.



(a) ρ

(b) u_1

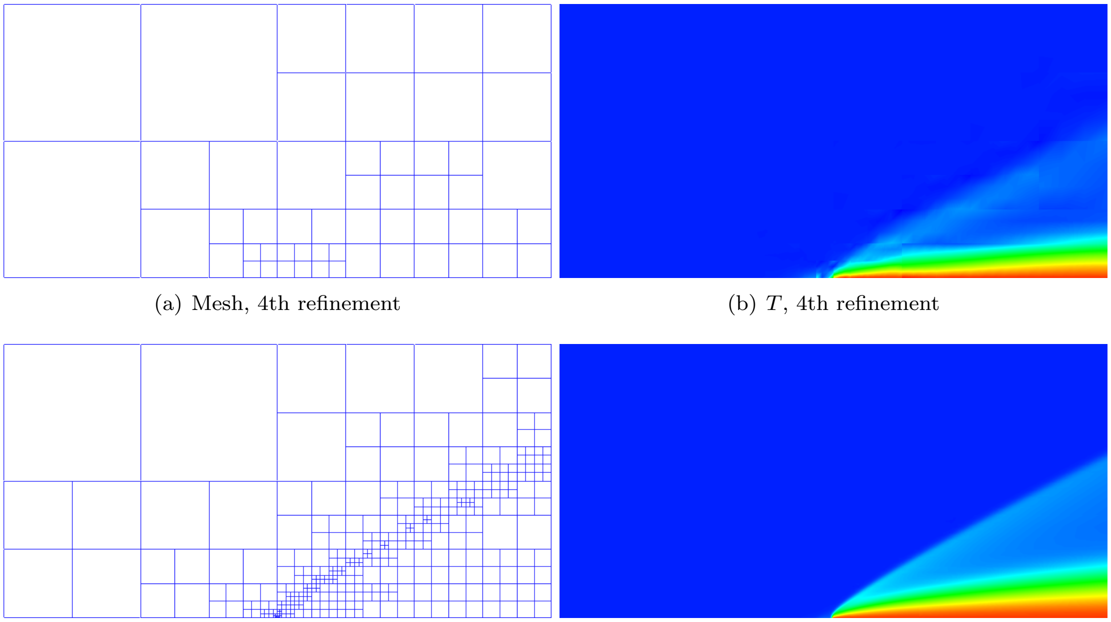
(c) u_2

(d) T

Figure 6: Solutions after twelve refinements for $p = 2$ and $\mathrm{Re} = 1000$, starting from a mesh of 2 elements.

We perform 10 steps of adaptive mesh refinement, beginning with a mesh of only two square elements. The solution on this coarsest mesh is given in Figure 5, and the final solutions after 10 refinement steps are given in Figure 6. For this specific example, we use only isotropic refinements.

Typical coarse meshes for adaptive CFD computations aim to resolve, at least to some degree, the features of the solution; often, physical features such as high gradients are used to drive refinement. For feature-based adaptivity to be effective, coarse mesh solutions must be of sufficiently high quality to resolve basic solution features. Often, artificial diffusion and shock capturing must also be applied in order to produce visually clean solutions on underresolved meshes. In contrast to this, the residual-based approach of DPG is able to place refinements accurately and efficiently despite the underresolution of solution features.



(a) Mesh, 4th refinement

(b) T, 4th refinement

(c) Mesh, 8th refinement

(d) T, 8th refinement

Figure 7: Snapshots of adaptive meshes and solutions for two different steps of adaptivity $\mathrm{Re} = 1000$.

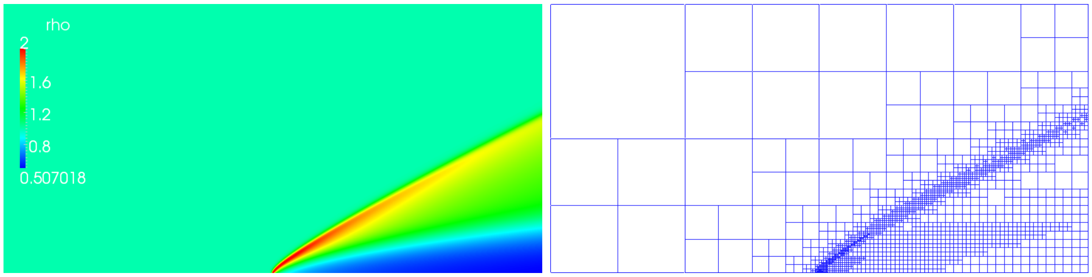
Figure 7 shows snapshots of the third and sixth steps of adaptive mesh refinement. The main contribution to energy error is at the plate tip – due to the change in boundary condition across the point $(.5, 0)$, the viscous stresses are singular at this point (this is analogous to the convection-diffusion plate example given earlier). Underresolution of the stresses at this point results in some pollution effects slightly upstream of the beginning of the plate; however, once $h \approx \mathrm{Re}^{-1}$ near the plate edge, these pollution effects disappear. We observe numerically that decreasing dt also limits how far upstream of the plate edge this pollution effect travels.

We observe numerically that ρ also behaves singularly at the plate tip – due to the presence of this strong singularity, the coloring of the wide range of values for ρ in Figure 6 causes the solution to appears largely uniform, save for a flare up at the tip of the plate. To better visualize density, ρ is rescaled such that the features of the solution away from the singularity, as well as the final mesh after 10 refinement steps, are visible in Figure 13.

We can also zoom in on the plate tip to view the solution quality at the singular point. Figure 9 demonstrates that the solution remains smooth and well-resolved at the plate tip, despite the presence of a singularity in the viscous stresses.

We increased the Reynolds number to 10,000 to assess the behavior of DPG for higher Reynolds numbers. However, we found it necessary to modify the method in several ways in order to achieve satisfactory results.

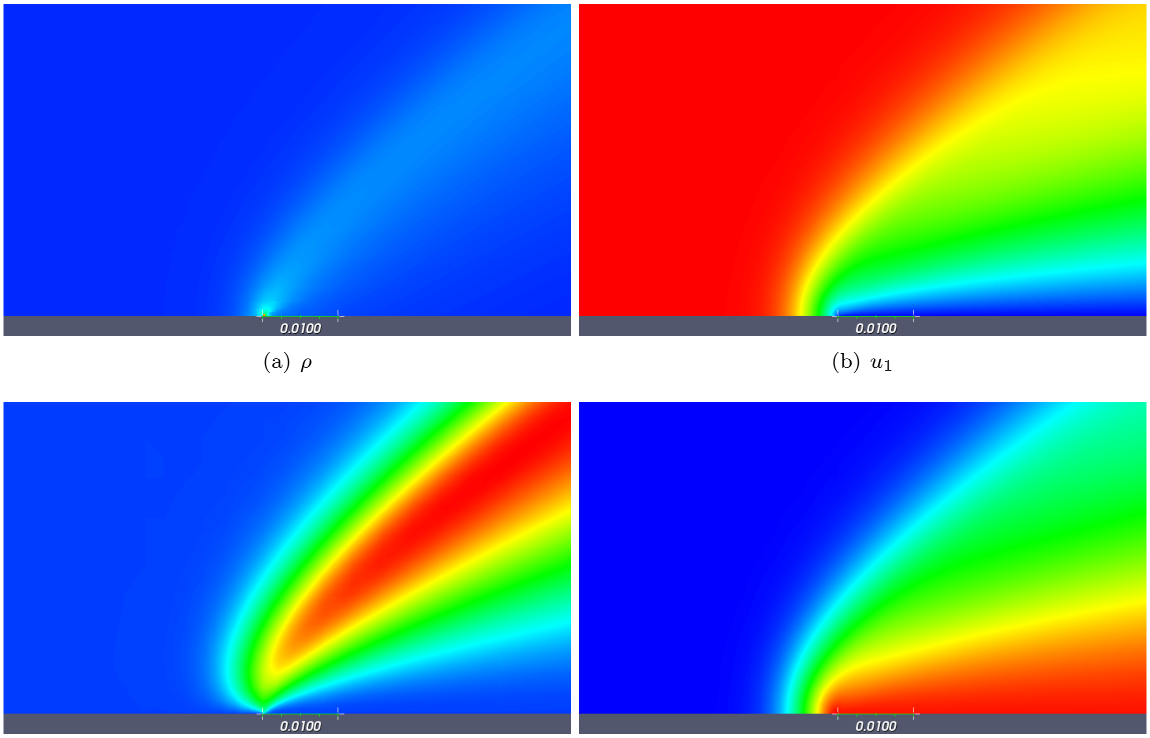
First, we implemented a line search algorithm to enforce positivity of both temperature and density, which are physically defined to be positive quantities. Given updates $\triangle\rho$ and $\triangle T$, we



(a) ρ

(b) Mesh

Figure 8: Rescaled solution for ρ in the range $[\rho_{\min}, 2]$ and adaptive mesh after 12 refinement steps.



(a) ρ

(b) u_1

(c) u_2

(d) T

Figure 9: Zoom of solutions at the beginning of the plate for $p = 2$ and $\mathrm{Re} = 1000$.

update our previous solution by setting

$$\begin{aligned} \rho &\coloneqq \rho + \alpha_{\text{line}} \triangle \rho \\ T &\coloneqq T + \alpha_{\text{line}} \triangle T, \end{aligned}$$

where α_{line} is chosen, for some $\delta > 0$, such that

$$\begin{aligned} \rho + \alpha_{\text{line}} \triangle \rho - \delta &= 0 \\ T + \alpha_{\text{line}} \triangle T - \delta &= 0. \end{aligned}$$

Since the addition of a line search can slow the convergence of a nonlinear algorithm, we incorporate also a Newton iteration at each timestep to effectively solve the nonlinear system at each timestep. We consider $\|\triangle U\|_E < \epsilon_{\text{Newton}}$ to be our condition for convergence of the Newton iteration, though we also limit the number of allowed Newton steps for computational efficiency. The full solver algorithm is given in Algorithm 2. For higher Reynolds numbers and highly refined meshes, the solution update exhibits large oscillations, such that the solution at a timestep can become negative, under which the pseudo-timestepping algorithm can stall or even diverge.[9] However, we have observed that requiring a strictly positive solution appears to be too restrictive a constraint for some meshes; the use of line search does not appear to be necessary for convergence of the pseudo-timestepping algorithm on coarser meshes (until the 10th refinement iteration, or $h < .001$).

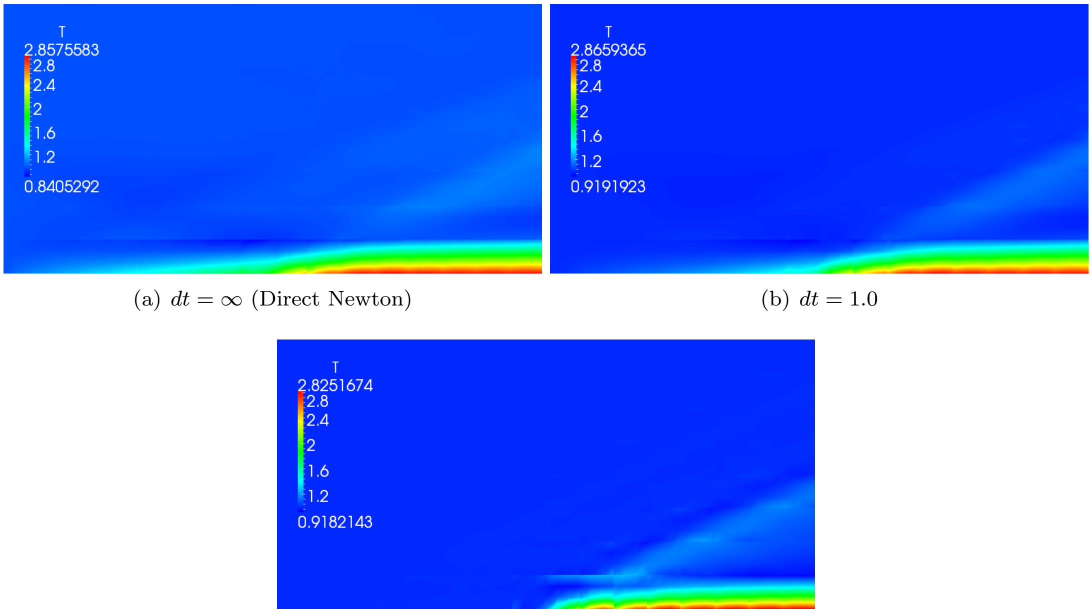
Algorithm 2 Pseudo-timetepping adaptive algorithm with line search.

```
for number of refinement steps do
  while $R_{\text{time}} > \epsilon_t$ do
    k = 0
    while $\|\triangle U\|_E > \epsilon_{\text{Newton}}$ and $k <$ maximum Newton steps do
      Solve for $\triangle U$, determine $\alpha_{\text{line}}$.
      $U \coloneqq U + \alpha_{\text{line}} \triangle U$.
      k = k+1
    end while
    Increment timestep.
  end while
  Compute energy error and refine based on a greedy refinement algorithm.
  Set $\epsilon_t = \alpha_t \|u - u_h\|_E$.
end for
```

Finally, we implement an "effective" CFL number; though implicit time integration schemes are unconditionally stable (compared to explicit schemes), a CFL condition relating the size of the time increment to the (minimum) mesh size is often still used in practice to improve convergence speed

[9] We note that these large oscillations mimic experiences in 1D as well, where the linearized solution was shown to exhibit sharp gradients near shocks that did not disappear, even with additional mesh refinement, indicating that the presence of such overshoots and undershoots is a consequence of the linearization, as opposed to the stability of the discretization [33]. This is discussed in more detail in Section 7.3.

We note also that theory developed in [3] for the convection-diffusion problem assumes a smooth convection field. However, under linearization of the Burgers' and Navier-Stokes equations, the solution around which we linearize dictates the convection field, and can display large gradients. While we have not observed issues in the Burgers' equation related to this, we hope to revisit the analysis done in [3] and generalize it for convection fields with large gradients.



(a) $dt = \infty$ (Direct Newton)

(b) $dt = 1.0$

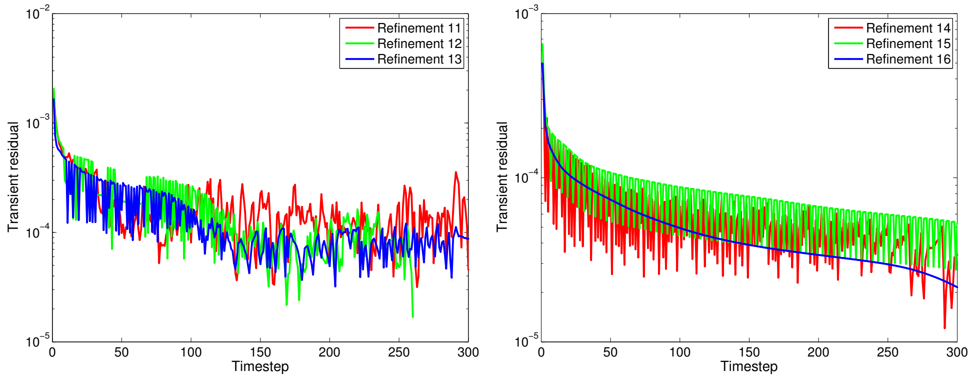
(c) $dt = .1$

Figure 10: Steady state solutions for Re $= 10,000$ under three different timesteps on a 16×8 uniform mesh.

and stability of the numerical scheme [32]. Our CFL number is chosen to be 64. We note that this CFL number is implemented for non-standard reasons. In fact, DPG is able to solve the steady-state system directly without the use of pseudo-timestepping (direct Newton iteration). However, as noted previously, the size of the timestep under which pseudo-timestepping converges greatly affects the qualitative nature of the solution. Figure 10 demonstrates the difference between convergence at large and small timesteps – for dt large relative to the mesh size, the solution experiences upstream "pollution" effects. Due to the upstream "pollution" present in the solution for $dt \geq 1$, the adaptive mesh refinement algorithm tends to add extraneous refinements on elements adjacent to the boundary $y = 0$, $x \in (0, 1)$. Decreasing the timestep alleviates this issue somewhat; however, an overly small timestep requires a large number of iterations to converge. The implementation of an effective CFL number aims to balance the size of the timestep with the mesh size.[10]

We note that, even with all of the above modifications, the pseudo-timetepping adaptive algorithm with line search would sometimes failed to converge below ϵ_t for Re $\geq 10,000$. Figure 11 shows a plot of the transient residual after the 14th refinement step; while the residual initially decreases, it stalls at about $R_{\text{time}} \approx 10^{-4}$. An examination of the difference in the solutions between the final

[10] An additional reason for the implementation of an effective CFL number is the conditioning of the local problem, which was discussed for the convection-dominated diffusion problem in [21]. The main problem concerning conditioning of local problems is the way that different test terms behave as a function of local element size. We illustrate this using the element Sobolev norm $\|v\|_{H^1(K)} = \|v\|_{L^2(\Omega)} + \|\nabla v\|_{L^2(\Omega)} - \|v\|_{L^2(\Omega)} = O(h^2)$ (where h is the element size), while $\|\nabla v\|_{L^2(\Omega)} = O(1)$. Thus, as $h \to 0$, the Sobolev norm over a single element approaches the Sobolev seminorm and loses positive definiteness, resulting in a highly ill-conditioned system to solve. The addition of a first-order pseudo-timestepping term allows us to increase the relative magnitude of $\|v\|_{L^2(\Omega)}$ with respect to $\|\nabla v\|_{L^2(\Omega)}$ and avoid conditioning issues while maintaining robustness of the method.



(a) Non-convergence

(b) Recovery of convergence

Figure 11: Stalling and recovery of the pseudo-timestep iteration for Re = 10000.

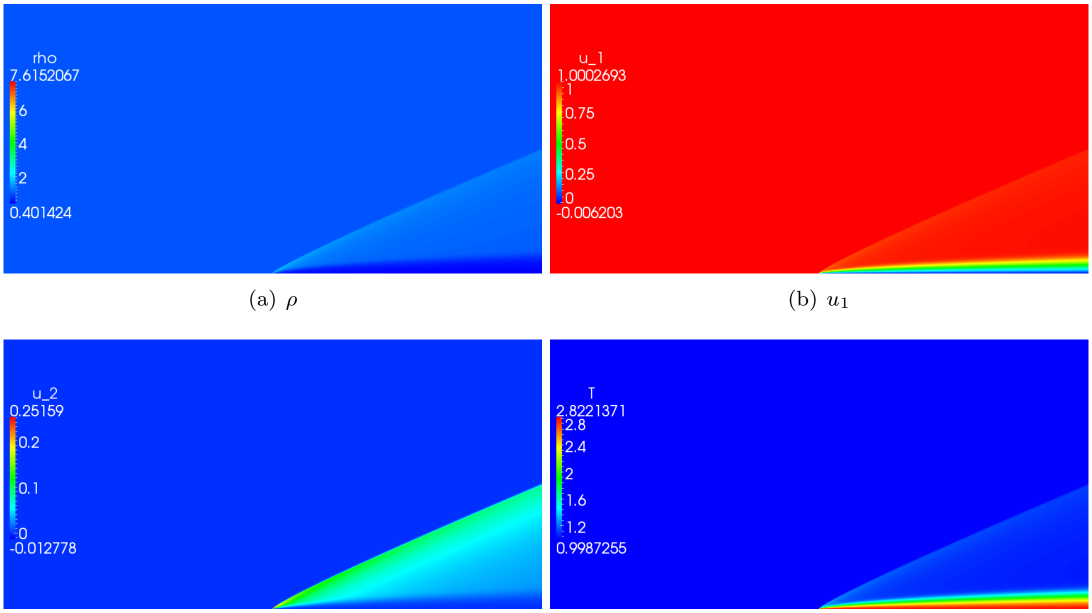
and 125th timesteps shows that the nonconvergence of the transient residual is due to oscillations in the solution (primarily in ρ) slightly upstream to the plate edge. Visually, the solution converges everywhere else, save for this area. Such behavior is also observed in [28], where, at the change in boundary conditions between the free stream and flat plate, an oscillation was observed in the solution which prevented convergence of his pseudo-time algorithm. His oscillations are of smaller magnitude ($O\left(10^{-6}\right)$ as opposed to $O\left(10^{-4}\right)$), which may be the result of several differences between the methods presented.[11] However, we note that, under additional refinements and increased resolution of the solution, the transient residual once again decreases in a smooth monotonic fashion, as shown in Figure 11.

For additional computational efficiency, we also implemented an anisotropic refinement scheme. In 2D, the boundary layer is a primarily 1D phenomena, which we expect to be far better resolved by anisotropic refinement than by isotropic refinement. We experimented first using the scheme described in Section 3.2 as an anisotropy indicator; however, the anisotropic scheme appeared to be too conservative near the boundary layer, placing primarily isotropic refinements. We modified our scheme in two ways – first, we incorporated spatially variable thresholding. Typically, anisotropic refinement in the x direction is chosen if $e_{x,K} > \epsilon_r e_{y,K}$ (and vice versa for anisotropic refinement in the y-direction), where $e_{x_i,K}$ is the error in the x_i direction. We set $\epsilon_r = \epsilon_{r,K}$; in other words, we allow our anisotropic threshold to vary element-by-element, and decrease it from $\epsilon_r = 10$ to $\epsilon_{r,K} = 2.5$ for elements adjacent to the wall upon which the boundary layer forms. Additionally, since the boundary layer typically displays rapid variation in the y-direction (the direction orthogonal to the wall), we relax the condition under which a vertically cut anisotropic refinement occurs to

$$2e_{y,K} > \epsilon_{r,K} e_{x,K}.$$

Despite the artificial modification of the anisotropic refinement scheme, the resulting meshes still resolve boundary layer solutions more efficiently than isotropic refinements. We hope to investigate

[11] Apart from the use of a standard Galerkin (continuous as opposed to discontinuous Galerkin) formulation, Kirk's approach differed from ours in the use of linear elements and the addition of artificial diffusion shock-capturing terms, both of which could explain the lower point at which error stagnates. Reasons for this loss of convergence in the pre-asymptotic range should be explored further.



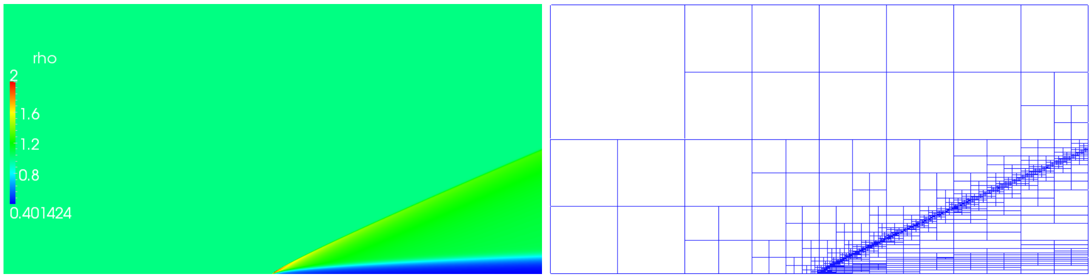
(a) ρ

(b) u_1

(c) u_2

(d) T

Figure 12: Solutions after 18 refinements for $p = 2$ and $\mathrm{Re} = 10000$.



(a) ρ

(b) Mesh

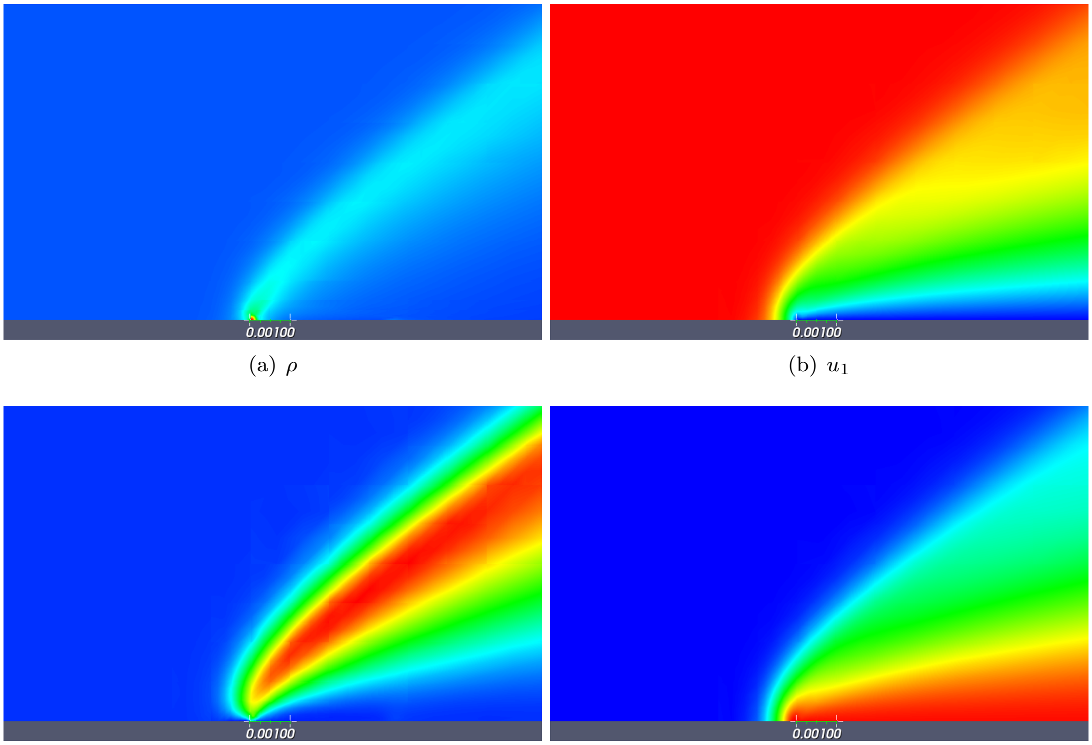
Figure 13: Rescaled solution for ρ in the range $[\rho_{\min}, 2]$ and adapted mesh.

reasons for the ineffectiveness of the pure anisotropic scheme for the compressible Navier-Stokes equations in future research.

We note that the resolution of the solution near the plate edge in Figure 14 for Reynolds number 10000 is qualitatively rougher than that for Reynolds 1000 in Figure 9; this is due to the greedy refinement algorithm emphasizing refinements most strongly at the singular point and not in shock resolution, as well as the fact that at a higher Reynolds number, the shock width is thinner and more difficult to resolve. Further refinement steps improve resolution of the shock features.

7.2. Holden ramp problem

Our second problem is a modified version of the Holden ramp problem, which models supersonic/hypersonic flow over a compression corner (the geometry of which is given in Figure 15).



(a) ρ

(b) u_1

(c) u_2

(d) T

Figure 14: Zoom of solutions at the beginning of the plate for $p = 2$ and Re $= 10000$.

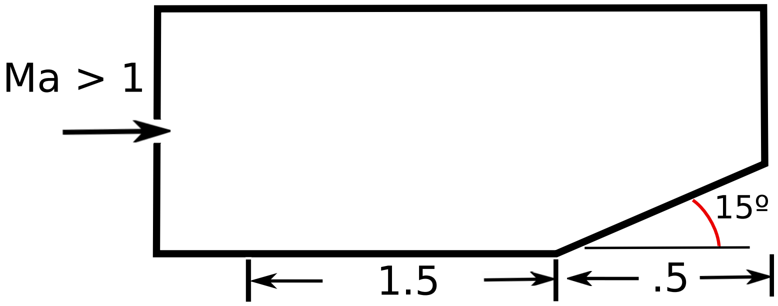
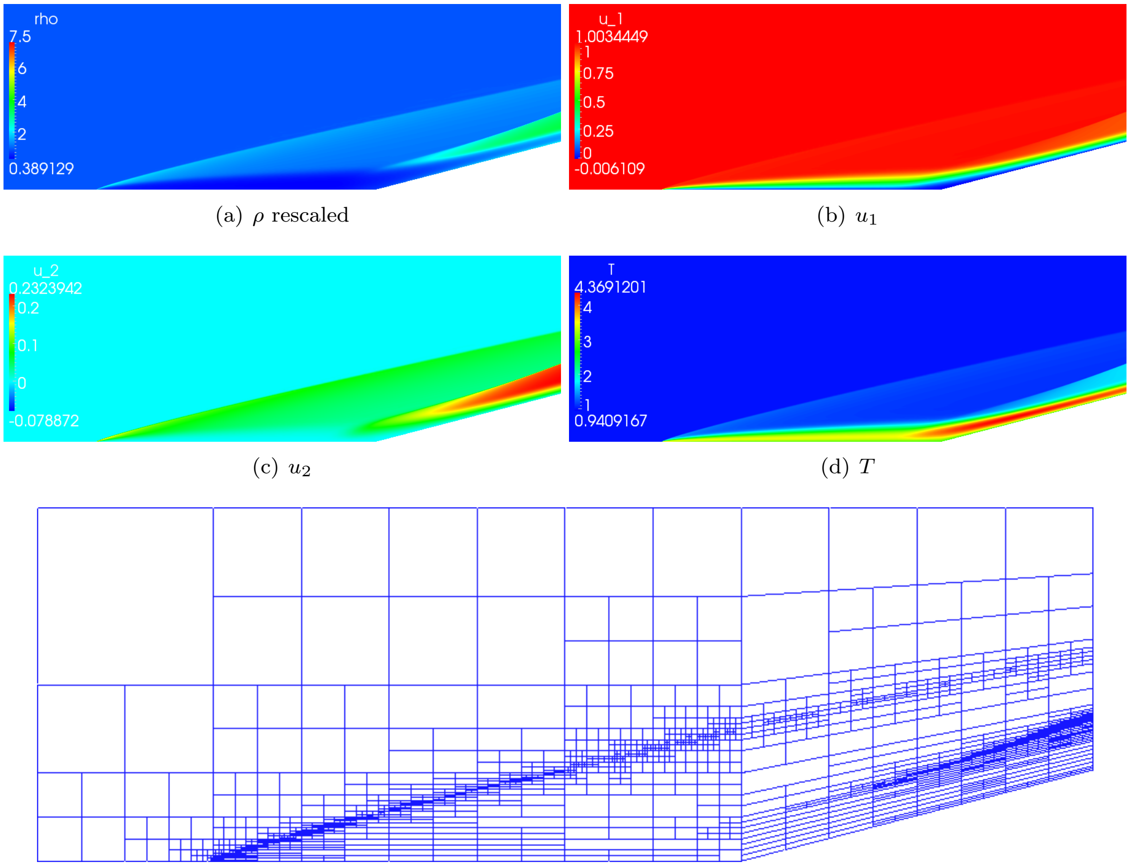


Figure 15: A modification of the Holden ramp/compression corner problem.

Similarly to the Carter flat plate problem, a flat plate disrupts the flow and forms a weak shock at the plate tip due to viscous effects and no-slip boundary conditions. The boundary layer grows down the plate edge, deflecting upwards due to the presence of the compression corner. A stronger shock forms slightly upstream of the compression corner in order to deflect the incoming supersonic flow, and is a common test problem for adaptive finite element methods for compressible flow [34, 7, 28].

The original plate length is given to be .442, while the ramp length is given to be .269. We have modified the problem slightly in order to exactly represent the boundary conditions on a coarse mesh



(a) ρ rescaled

(b) u_1

(c) u_2

(d) T

(e) Mesh

Figure 16: Solutions and adaptive mesh for Ma = 6 and Re = 10000.

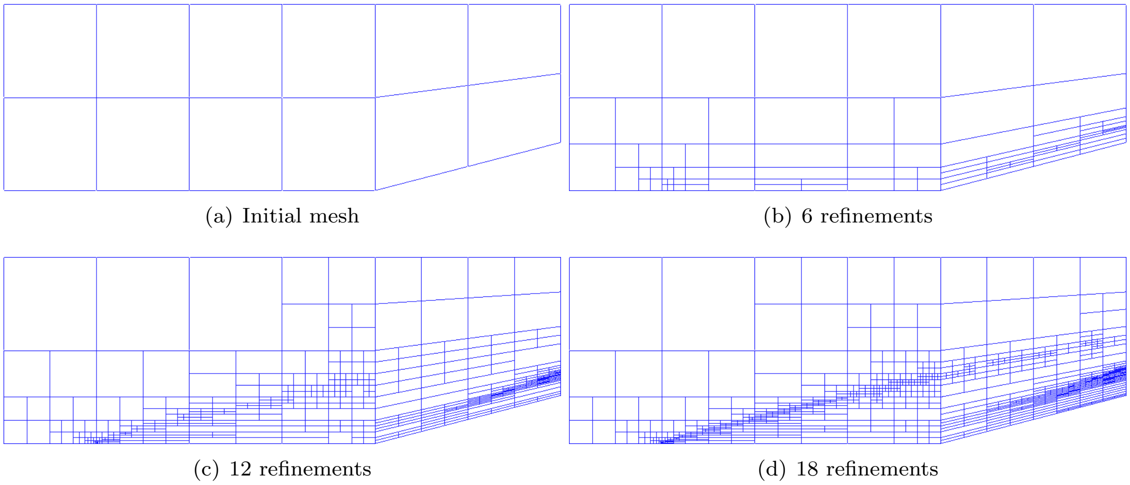
while keeping the ratio of plate length to ramp length roughly the same. Similarly to the Carter flat plate, we start out with a very coarse 2×3 mesh of 6 elements. We initialize our solution to the freestream values

$$\rho_\infty = 1, \qquad u_{1,\infty} = 1, \qquad u_{2,\infty} = 0, \qquad T_\infty = 1$$

and again set stresses uniformly to zero.

We first solve under Mach 6 flow.[12] and Reynolds number of 10,000, or a Reynolds number of 33,936 if measured with respect to the original plate length of .442. The wall temperature is set to $T_w = 2.8T_\infty$, identically to the flat plate problem. 24 mesh refinements were performed, resulting in a final mesh of 1858 elements. Figure 16 shows both solution values and final adaptive mesh. Due to a large maximum value of $\rho_{\max} = 14.1538$ at the plate tip, the resulting solution for ρ is scaled to better show qualitative features of the flow. The presence of the second shock deflecting

[12] The increase in Mach number is to change the angle of the shock; under the current setup, Mach 3 flow produced a shock which reflected off the top boundary $y = 1$ We note that the effect of Mach number under our nondimensionalization of choice is to decrease the thermal diffusivity constant κ relative to the viscosities μ and λ.



(a) Initial mesh

(b) 6 refinements

(c) 12 refinements

(d) 18 refinements

Figure 17: Sequence of adaptive meshes for Ma = 6 and Re = 10000.

the incoming supersonic flow at the ramp is clearly seen in both the solutions and the adaptive mesh refinements.

We can examine the sequence of adaptive meshes generated by the DPG method. Figure 17 shows the initial mesh, as well as the 6th, 12th, and 18th subsequent refinements generated automatically by the DPG method. Unlike the previous sequence of meshes generated by the flat plate example, refinements tend to be placed most heavily near the shock at the outflow ramp. By the 18th refinement, the adaptive mesh looks qualitatively very similar to the final mesh of 24 refinements, and further refinements focus on the resolution of the shocks originating at the plate edge and at the ramp outflow.

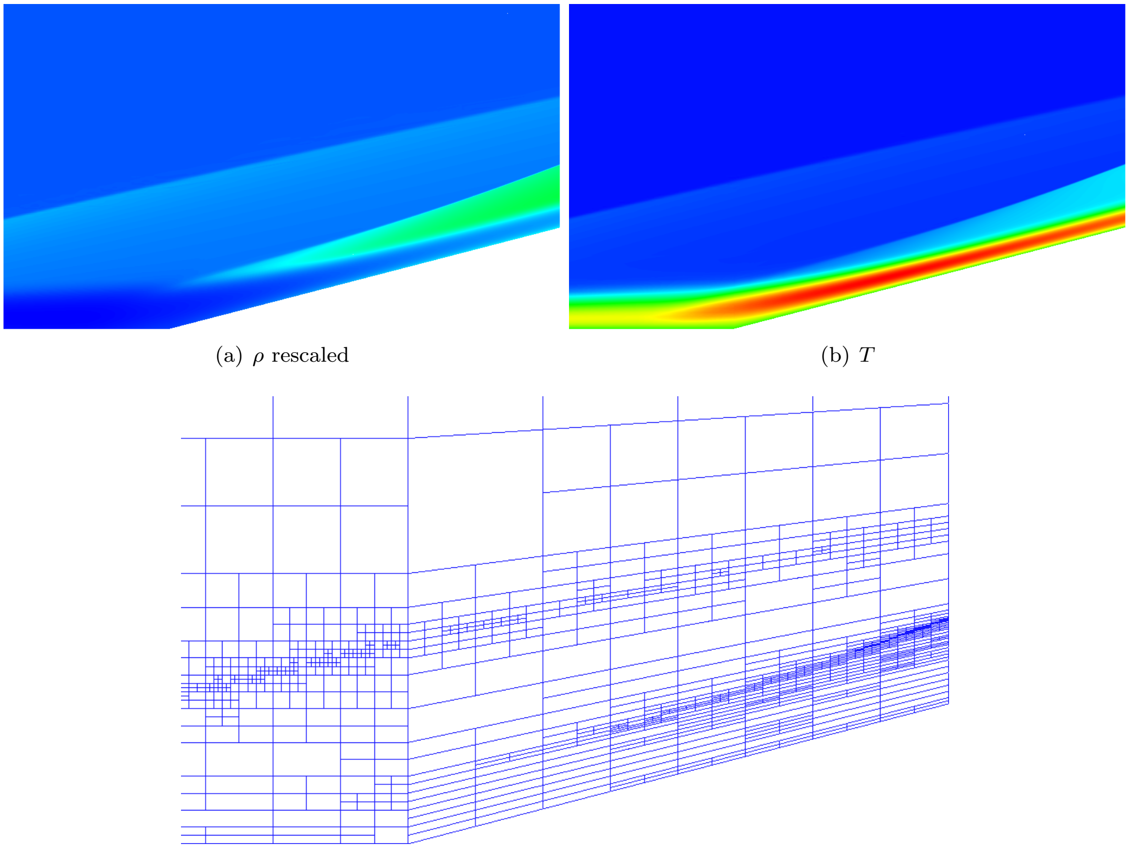
Figure 18 shows a zoom of the second stronger shock that develops near the ramp outflow. Refinements are placed very heavily near the shock, which we expect due to the fact that a shock forms a stronger gradient than a boundary layer. We note that our residual-driven adaptivity scheme places mesh refinements in a very precise manner; refinements on ramp boundary are placed slightly more heavily upstream than downstream. We believe this is due to the fact that solution gradients are slightly higher in ρ at the upstream section of the ramp.

The second set of conditions under which we solve are under Mach 11.68 flow and Reynolds number of 16,4 42.4, or 55,800 if measured with respect to original plate length, with the wall temperature set to $T_w = 4.6T_\infty$. 24 automatic mesh refinements are performed under an energy threshold of .5, resulting in a final mesh of 2385 elements.

Figure 19 shows the resulting solution and final adaptive mesh. The increased Mach number changes again the angle of the weak shock resulting from the change in boundary conditions at the plate edge. Compared to the previous case of Mach 6 flow, where only 18 refinements steps were performed, 24 refinements were performed for the Mach 11.68 case, leading to a more highly resolved solution, especially near the plate tip. Due to a large maximum value of $\rho_{\text{max}} = 50.1805$ at the plate tip, the resulting solution for ρ is scaled to better show qualitative features of the flow.

Compared to the Mach 6 case, where 18 refinements were performed, the resolution of the mesh near the shock at the ramp outflow for Mach 11.68 flow and 18 refinements is qualitatively similar. Further refinement steps increase resolution of the mesh near the shock, as shown in Figure 20.

Finally, we plot the normal heat flux over the plate and ramp for both Holden problems in



(a) ρ rescaled

(b) T

(c) Mesh

Figure 18: Zoom of solutions and adaptive mesh around shock.

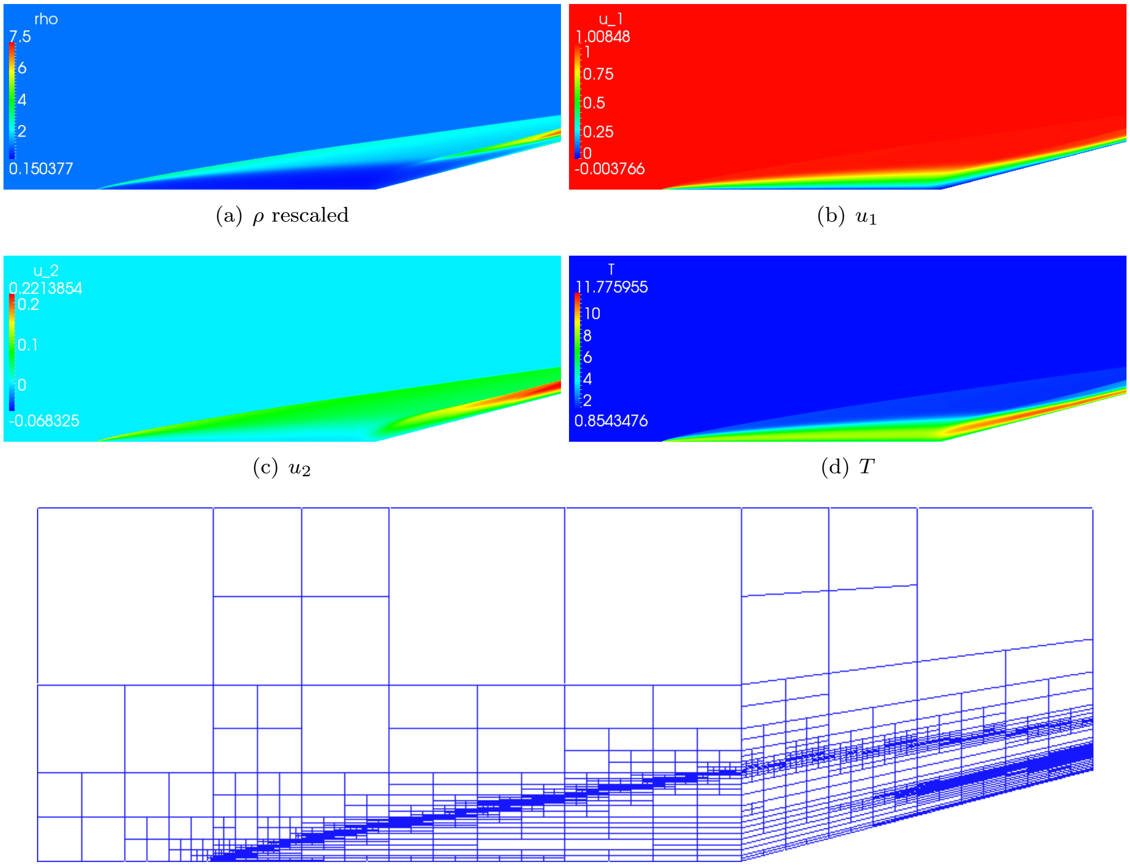
Figure 21. The normal heat flux over the flat plate is indicated by the blue line, while the dotted brown line indicates the heat flux over the ramp, and both are plotted against the x-coordinate along the plate/ramp boundary. The normal heat flux is given by the normal trace of the conserved quantity in the energy equation

$$\widehat{f}_{4,n} = (\rho e + p)u_n - \boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{u} + q_n.$$

Recognizing that $u_1 = u_2 = 0$ reduces the above to $\widehat{f}_{4,n} = q_n$.

The heat flux develops a strong singularity at the point $(-.5, 0)$, where the boundary condition changes from a Neumann/stress boundary condition to a Dirichlet/no-slip boundary condition.[13] It can be shown that the Laplace equation develops a singularity in stress at under any such change in boundary conditions, and the same phenomena is observed for a similar setup under convection-diffusion [35].

[13] Figure 21 cuts off this singularity in order to show the qualitative behavior of q_n over the remainder of the boundary. The maximum visualized values in the singular portion of q_n are .35 for Mach 6 flow and 1.268 for Mach 11.68 flow.



(a) ρ rescaled

(b) u_1

(c) u_2

(d) T

(e) Mesh

Figure 19: Solutions and adaptive mesh for $\mathrm{Ma} = 11.68$ and $\mathrm{Re} = 16442.4$.

7.3. Higher Reynolds numbers

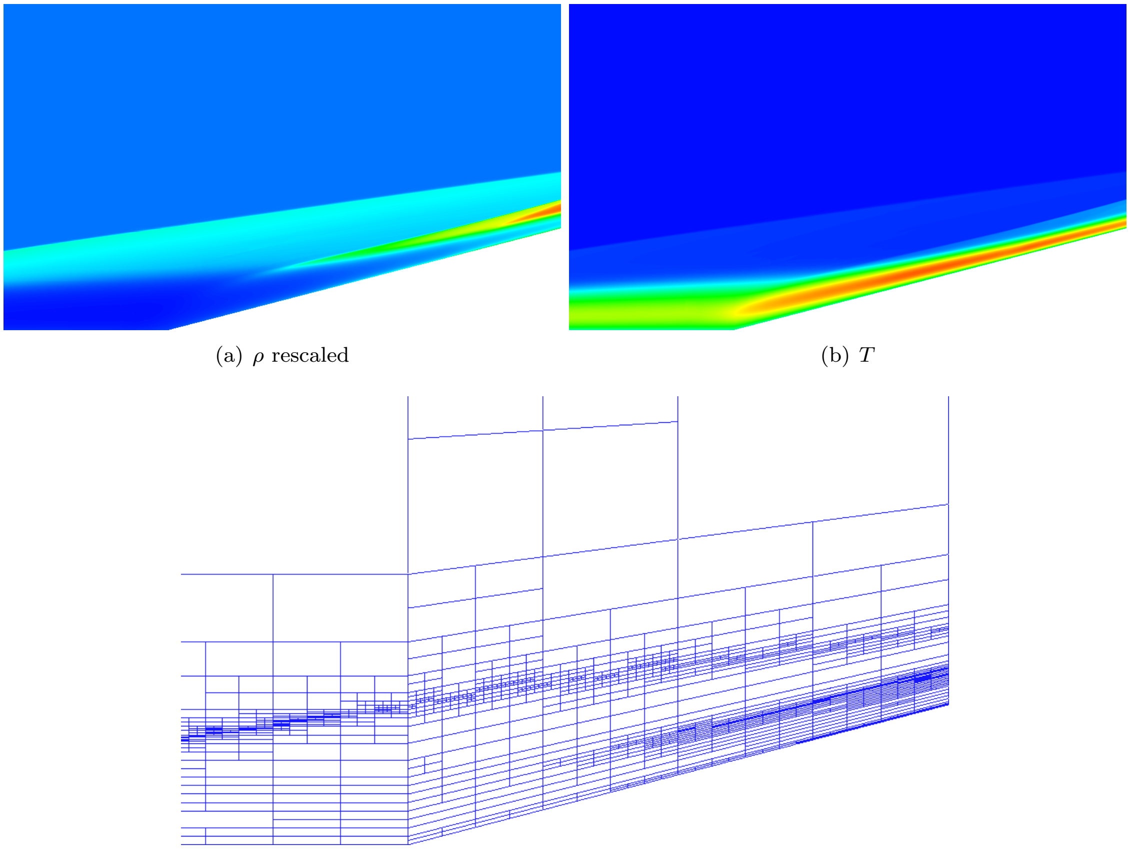
For higher Reynolds number, the pseudo-timestepping, we have encountered difficulties which have made it difficult to converge to a steady-state solution, even under the addition of an effective CFL number and line search to enforce positivity of density ρ and temperature T. However, we believe these difficulties to be related to the nature of the equations, rather than the robustness of the method. We illustrate this with a simple example.

Let us consider the 1D steady state viscous Burgers' equation on $[-\infty, \infty]$

$$u\frac{\partial u}{\partial x} - \epsilon\frac{\partial^2 u}{\partial x^2} = 0.$$

The exact solution to this equation under boundary conditions

$$\begin{aligned} u(-\infty) &= 1 \\ u(\infty) &= -1 \end{aligned}$$

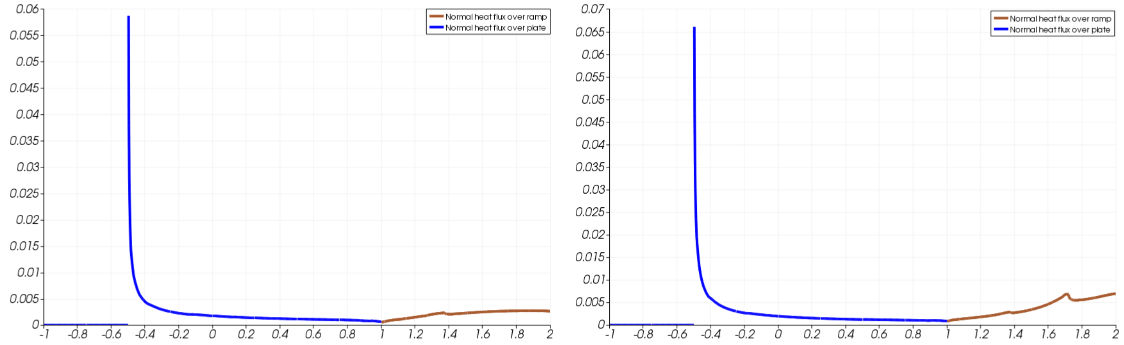


(a) ρ rescaled

(b) T

(c) Mesh

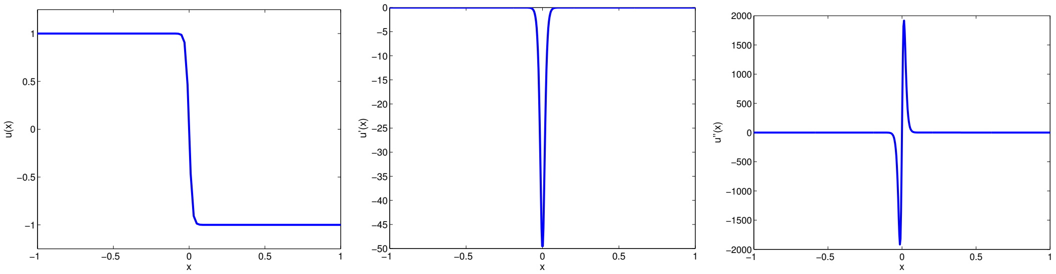
Figure 20: Zoom of solutions and adaptive mesh around shock.



(a) Mach 6, Reynolds 10,000

(b) Mach 11.68, Reynolds 16,442.4

Figure 21: Normal heat flux q_n over plate and ramp for the Holden problem. The blue line indicates q_n over the flat plate, while the brown line indicates q_n over the ramp.



(a) $u(x)$

(b) $u'(x)$

(c) $u''(x)$

Figure 22: $u(x)$ and its derivative and second derivative for $\epsilon = .01$.

can be easily derived (see [7] for a simple derivation), and the solution and its first two derivatives are

$$u(x) = \frac{1 - e^{\frac{x}{\epsilon}}}{1 + e^{\frac{x}{\epsilon}}}, \qquad u'(x) = \frac{-2e^{\frac{x}{\epsilon}}}{\left(1 + e^{\frac{x}{\epsilon}}\right)^2 \epsilon}, \qquad u''(x) = \frac{-2e^{\frac{x}{\epsilon}}\left(e^{\frac{x}{\epsilon}} - 1\right)}{\left(1 + e^{\frac{x}{\epsilon}}\right)^3 \epsilon^2}$$

Figure 22 shows each of these functions for $\epsilon = .01$. From the form of $u'(x)$ and $u''(x)$, we know these oscillations will grow rapidly as ϵ decreases. Consider now the linearized Burgers' equation

$$\frac{\partial u \Delta u}{\partial x} - \epsilon \frac{\partial^2 \Delta u}{\partial x^2} = -r(x)$$

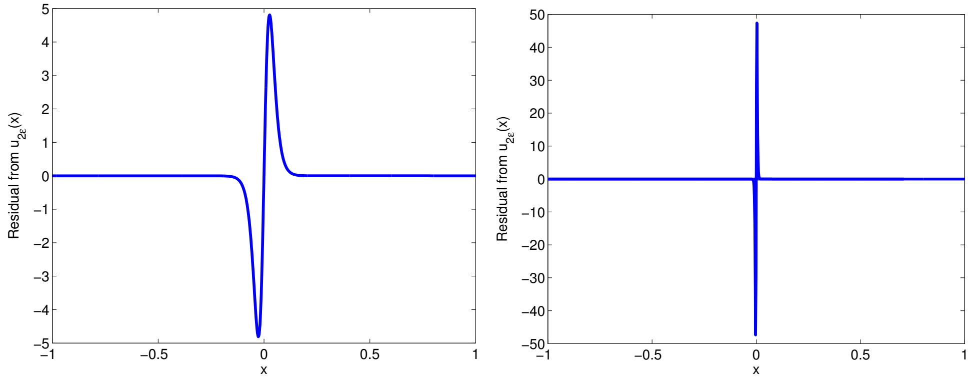
where $r(x) = u\frac{\partial u}{\partial x} - \epsilon \frac{\partial^2 u}{\partial x^2}$ is the strong form of the nonlinear residual. Recall that for a pure Newton iteration, $u(x)$ is assumed to be known, and the linearized problem is driven by the residual. The solution $u(x)$ is updated $u \coloneqq u + \Delta u$, and is repeated until Δu and $r(x)$ are both approximately zero.

Let $u_\epsilon(x)$ be the exact solution for a particular viscosity ϵ, and consider the setting of $u(x) = u_{\alpha\epsilon}(x)$, with $\alpha > 1$. In other words, the initial guess for the Newton iteration is taken to be the exact solution for the viscous Burgers' equation under a larger viscosity (a less sharp shock), a method known as continuation (specifically, continuation in viscosity ϵ).[14]

We plot $r(x)$ for $u(x) = u_{2\epsilon}(x)$ in Figure 23. While the form of the exact linearized solution Δu is unknown, we see that the forcing term in the above equation develops oscillations that grow in magnitude and decrease in support as ϵ decreases. We have observed similar behaviors for discretized solutions to the Burgers' and Navier Stokes equations – the discrete linearized solution will develop gradients as well, though their magnitude will be limited by the resolution of the mesh. However, additional refinements near shocks will introduce additional oscillations, which are subsequently damped by additional Newton iterations.

In other words, not all oscillations are related to the method of discretization – the exact linearized solution itself contains large oscillations, which are picked up more and more strongly in a discrete scheme as the mesh resolution approaches the viscous length scale ϵ (see [36, 37] for additional numerical examples). We note that these large oscillations are not an issue for the viscous Burgers' equation, as there is no positivity constraint on u. However, as noted previously, the convergence

[14] We note that continuation in Reynolds number was attempted for the Navier-Stokes equations; however, the presence of large oscillations in the Newton update on highly adapted meshes caused the line search to return a near-zero step length, which stalled the nonlinear iteration prior to convergence to a steady state solution.



(a) $\epsilon = .01$

(b) $\epsilon = .001$

Figure 23: Residual for Burgers' equation with viscosity ϵ under the exact solution for 2ϵ.

of the nonlinear iteration for the compressible Navier-Stokes equations can stall or even diverge for sufficiently negative values of ρ and T, which happens easily in the presence of large oscillations. When a line search is implemented to enforce a strictly positive solution, the nonlinear iteration can stall, and numerical experiments have generated cases in which the line search length goes to zero, returning an under-converged solution.

There are several approaches to controlling the magnitude of oscillations in the linearized solution – a simple option is decreasing the size of the timestep; however, doing so can cause the number of iterations required for convergence to greatly increase. The application of artificial diffusion near sharp gradients is another way in which to control such oscillations; however, this results in a modified set of equations, and, though solutions with contemporary artificial viscosity methods can produce solutions with expected qualitative features, there is a wide range of choices for artificial diffusion, and it was not clear in the scope of this paper which artificial viscosity method would be quantifiably superior, or most suitable for use with the DPG method.

8. Conclusions and future work

The goal of this work has been to explore the behavior of the Discontinuous Petrov-Galerkin (DPG) method as a method for the discretization and solution of convection-dominated diffusion problems, and to produce both theory and numerical results for model problems in this area. We begin by introducing the DPG method for linear problems; the concept of problem-dependent optimal test functions is derived through equivalence with the minimization of a specific residual, and discontinuous test functions are introduced in order to localize the determination of such optimal test functions to a single element.

The DPG framework is then extended to nonlinear problems, and equivalence is shown between the DPG method and a Gauss-Newton minimization scheme for the nonlinear residual. An anisotropic refinement scheme is implemented to more effectively capture lower-dimensional behavior of solutions of convection-diffusion problems, such as boundary layers. We extrapolate the DPG method to a nonlinear viscous Burgers' equation, and conclude by applying the DPG method to systems of equations and to the solution of two model problems in viscous compressible flow.

In particular, we demonstrate for both the flat plate and compression ramp problem in supersonic/hypersonic in compressible flow that the DPG method is able to begin from a highly underresolved meshes (two elements for the Carter plate problem, and 12 elements for the Holden ramp problem), and resolve physical features of the solution through automatic adaptivity, without the aid of artificial diffusivity or shock capturing terms. We believe this indicates both the robustness of the method on coarse grids and the effectiveness of the DPG error indicator for adaptive refinement.

As is the case with any research, much work remains to be done. We have chosen the classical variables in which to cast the compressible Navier-Stokes equations; however, investigation of alternative sets of variables may have merit, as different choices of variables yield differing linearizations with their own advantages (for example, all derivatives in time are linear with respect to the momentum variables, and the entropy variables of Hughes both symmetrize the Navier-Stokes equations and yield solutions obeying second law of thermodynamics for standard H^1 formulations [38]).

We also hope to investigate artificial viscosity methods as regularization for problems in viscous compressible flow. We present an analysis of the 1D Burgers' equation demonstrating that the exact solutions under Newton linearization contain large oscillations. While these oscillations are not the result of the stability of the spatial discretization, their presence can cause density and temperature to become non-physically negative, which can stall the convergence of the nonlinear solver. Our goal in incorporating artificial viscosity would not be to provide additional stabilization mechanisms for the discrete spatial discretization, but to provide regularization with which to suppress the presence of large oscillations in the linearized solution.

Finally, though the method is inf-sup stable for arbitrary meshes, our experiments have focused on meshes of uniform p. We hope to implement a true hp-adaptive DPG method in the future.

References

[1] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. II. Optimal test functions. *Num. Meth. for Partial Diff. Eq*, 27:70–105, 2011. see also ICES Report 9/16.

[2] L. Demkowicz and N. Heuer. Robust DPG method for convection-dominated diffusion problems. *SIAM J. Num. Anal*, 2013. accepted.

[3] J. Chan, N. Heuer, T. Bui Thanh, and L. Demkowicz. A robust DPG method for convection-dominated diffusion problems II: adjoint boundary conditions and mesh-dependent test norms. *Computers and Mathematics with Applications*, 2013. Accepted.

[4] H. Roos, M. Stynes, and L. Tobiska. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*. Springer series in computational mathematics. Springer, 2008.

[5] A. Brooks and T. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engr*, 32:199–259, 1982.

[6] P. M. Gresho and R. L. Lee. Don't suppress the wiggles - they're telling you something! *Computers and Fluids*, 9(2):223 – 253, 1981.

[7] G. Barter. *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order Discontinuous Galerkin Finite Element Method*. PhD thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, 2008.

[8] J.L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248 – 4267, 2011. Special issue High Order Methods for CFD Problems.

[9] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. part I: The transport equation. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1558 – 1572, 2010. see also ICES Report 2009-12.

[10] J. Gopalakrishnan and W. Qiu. An analysis of the practical DPG method. *Math. Comp.*, 2012. accepted.

[11] T. Bui-Thanh, Leszek Demkowicz, and Omar Ghattas. Constructively well-posed approximation method with unity inf-sup and continuity constants for partial differential equations. *Mathematics of Computation*, 2011. Accepted.

[12] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, and V.M. Calo. A class of discontinuous Petrov-Galerkin methods. Part IV: The optimal test norm and time-harmonic wave propagation in 1D. *Journal of Computational Physics*, 230(7):2406 – 2432, 2011.

[13] L. Demkowicz and J. Gopalakrishnan. A primal DPG method without a first order reformulation. *Computers and Mathematics with Applications*, 2013. To appear.

[14] 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.

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

[16] T. Bui-Thanh, L. Demkowicz, and O. Ghattas. A unified discontinuous Petrov-Galerkin Method and its analysis for Friedrichs' systems. Technical Report 34, ICES, 2011. Accepted to SIAM J. Num. Anal.

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

[18] J. Chan, J. Gopalakrishnan, and L. Demkowicz. Global properties of DPG test spaces for convection-diffusion problems. Technical Report 13-05, ICES, February 2013.

[19] I. Babuska and B.Q. Guo. Approximation properties of the h-p version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133(3-4):319 – 346, 1996.

[20] W. Rachowicz, J. T. Oden, and L. Demkowicz. Toward a universal h-p adaptive finite element strategy part 3. Design of h-p meshes. *Computer Methods in Applied Mechanics and Engineering*, 77(1-2):181 – 212, 1989.

[21] L. Demkowicz, J. Gopalakrishnan, and A. Niemi. A class of discontinuous Petrov-Galerkin methods. Part III: Adaptivity. *Appl. Numer. Math.*, 62(4):396–427, April 2012. see also ICES Report 2010/1.

[22] B. Kirk, J. Peterson, R. Stogner, and G. Carey. `libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4):237–254, 2006. http://dx.doi.org/10.1007/s00366-006-0049-3.

[23] M. D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows: A Guide to Theory, Practice, and Algorithms.* Computer Science and Scientific Computing. Elsevier Science, 1989.

[24] D. Moro-Ludeña, J. Peraire, and N. Nguyen. A Hybridized Discontinuous Petrov-Galerkin scheme for compressible flows. Master's thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, Boston, USA, 2011.

[25] O. Ghattas T. Bui-Thanh. A PDE-constrained optimization approach to the Discontinuous Petrov-Galerkin method with a trust region inexact Newton-CG solver. Technical Report 13-16, ICES, June 2013.

[26] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, New York, 2nd edition, 2006.

[27] L. Demkowicz, J.T. Oden, and W. Rachowicz. A new finite element method for solving compressible Navier-Stokes equations based on an operator splitting method and h-p adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 84(3):275 – 326, 1990.

[28] B. Kirk. *Adaptive Finite Element Simulation of Flow and Transport Applications on Parallel Computers.* PhD thesis in Computational and Applied Mathematics, The University of Texas at Austin, 2009.

[29] N. Roberts, D. Ridzal, P. Bochev, and L. Demkowicz. A Toolbox for a Class of Discontinuous Petrov-Galerkin Methods Using Trilinos. Technical Report SAND2011-6678, Sandia National Laboratories, 2011.

[30] J. E. Carter. Numerical solutions of the supersonic, laminar flow over a two-dimensional compression corner. Technical Report TR R-385, NASA, July 1972.

[31] L. Demkowicz, J. Oden, and W. Rachowicz. A new finite element method for solving compressible Navier-Stokes equations based on an operator splitting method and hp-adaptivity. *Comput. Methods Appl. Mech. Eng.*, 84(3):275–326, December 1990.

[32] F. Shakib, T. J.R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. the compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89(1 - 3):141 – 219, 1991. Second World Congress on Computational Mechanics.

[33] J. Chan, L. Demkowicz, N. Roberts, and R. Moser. A new Discontinuous Petrov-Galerkin method with optimal test functions. part v: Solution of 1d Burgers and Navier-Stokes equations. Technical Report 10-25, ICES, June 2010.

[34] W. Rachowicz. An anisotropic h-adaptive finite element method for compressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 146(3-4):231 – 252, 1997.

[35] T. Ellis, J. Chan, N. Roberts, and L. Demkowicz. Element conservation properties in the DPG method. Technical report, ICES, July 2013. In preparation.

[36] J. Chan, L. Demkowicz, and M. Shashkov. Space-time DPG for shock problems. Technical Report LA-UR 11-05511, LANL, September 2011.

[37] J. Chan, L. Demkowicz, R. Moser, and N. Roberts. A New Discontinuous Petrov-Galerkin Method with Optimal Test Functions. Part V: Solution of 1D Burgers and Navier-Stokes Equations. Technical Report 10-25, ICES, June 2010.

[38] T.J.R. Hughes, L.P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2):223 – 234, 1986.