Improving Shock-capturing Robustness for Higher-order Finite Element Solvers

by

Carlee F. Wagner

B.S.E., University of Pennsylvania (2012)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2015

Signature redacted

Author ....
Department of Aeronautics and Astronautics
August 20, 2015

Signature redacted

Certified by ....
David Darmofal
Professor of Aeronautics and Astronautics
Thesis Supervisor

Signature redacted

Accepted by ....
Paulo C. Lozano
Associate.Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Improving Shock-capturing Robustness for Higher-order Finite Element Solvers

by

Carlee F. Wagner

Submitted to the Department of Aeronautics and Astronautics
on August 20, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Simulation of high speed flows where shock waves play a significant role is still an area of development in computational fluid dynamics. Numerical simulation of discontinuities such as shock waves often suffer from nonphysical oscillations which can pollute the solution accuracy. Grid adaptation, along with shock-capturing methods such as artificial viscosity, can be used to resolve the shock by targeting the key flow features for grid refinement. This is a powerful tool, but cannot proceed without first converging on an initially coarse, unrefined mesh. These coarse meshes suffer the most from nonphysical oscillations, and many algorithms abort the solve process when detecting nonphysical values.

In order to improve the robustness of grid adaptation on initially coarse meshes, this thesis presents methods to converge solutions in the presence of nonphysical oscillations. A high order discontinuous Galerkin (DG) framework is used to discretize Burgers' equation and the Euler equations. Dissipation-based globalization methods are investigated using both a pre-defined continuation schedule and a variable continuation schedule based on homotopy methods, and Burgers' equation is used as a test bed for comparing these continuation methods. For the Euler equations, a set of surrogate variables based on the primitive variables (density, velocity, and temperature) are developed to allow the convergence of solutions with nonphysical oscillations. The surrogate variables are applied to a flow with a strong shock feature, with and without continuation methods, to demonstrate their robustness in comparison to the primitive variables using physicality checks and pseudo-time continuation.

Thesis Supervisor: David L. Darmofal
Title: Professor of Aeronautics and Astronautics

Acknowledgments

First I would like to thank my advisor, Professor David Darmofal, for allowing me to be a part of the research team, the ACDL, and the greater CFD community, for getting me started virtually from square one in coding CFD, for always clarifying my vague statements during meetings, and most importantly, for teaching me to be thorough in my work. Although I feel like I did not pick his brain as much as I wanted to, I would also like to Thank Dr. Steven Allmaras for his long-distance advice at my meetings and over email, for the fictitious gas research, and of course for always diagnosing Jacobian bugs without even seeing the computer screen. Dr. Marshall Galbraith deserves acknowledgement for being a tremendous help to me throughout the thesis process, especially in the final throes. I would like to thank him for showing me how implement all sorts of code, how to use the cluster, for teaching me about pointers, lifting operators and ping tests, for dropping everything to help me debug, and for always reminding me that "if it were easy, you wouldn't be getting a Master's degree."

I'd like to thank the ProjectX team when I entered the group: Steve, Phil, and Jun, for helping me understand our group's research, teaching me the fundamentals–from using the command line, to quadrature, to .func files, for helping accustom me to MIT, and for being there to help with technical problems or just to blow off steam. My fellow first-year students on the team: Yixuan and Savithru, also deserve acknowledgments, for working together to help each other understand ProjectX, to run those turbulence cases, and to tackle those 920 projects.

Keeping in good spirits has been important to staying motivated at MIT. My ACDL friends have provided a great environment for learning, working and socializing, especially in the new lab space. Big thanks go out to my cube-mates (and on occasion, classmates): Giulia, Hugh, and Philippe, in part for helping with classwork and research, but mostly for making me look forward to going into lab, even when it got hard, if just to share some laughs in DogeCube. I'd also like to thank Patrick for encouraging ACDL socialization outside of lab, and bonding over our cars. In this vein, I would also like to thank my SidPac friends, especially Renee and Jeff, for providing me with social support when I needed it, as well as MIT's DanceTroupe for being so welcoming. I'd also like to recognize the department's space manager, Anthony Zolnik, for so fervently helping me when I lost my laptop.

My family deserves thanks for understanding, and not taking personally, my bouts of radio silence while hard at work in Boston, especially Cole. I thank them for their continual encouragement all along, and hope that they also do not take my impending cross-country move personally either.

Finally, I would like to acknowledge the financial support provided by the Boeing Company (technical monitor Dr. Mori Mani), and by the MIT AeroAstro Department through their Hardy fellowship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As technology has advanced over the last several decades, computation has grown into an integral tool in engineering design and analysis. Numerical simulation is particularly well suited to solving complex engineering problems that do not have closed-form analytical solutions, or problems that are hard to test experimentally. Computational fluid dynamics (CFD) is a popular tool in the aerospace industry to analyze flows and reduce the need for wind tunnel testing. Various flow problems of interest in the aerospace industry involve complex geometries and/or freestream flow at high speeds. The setup, instrumentation, and execution of such tests may be prohibitively expensive, time consuming, and prone to human error. This is especially true of high speed flows where shock waves play a significant role, such as the transonic, supersonic, and hypersonic regimes. CFD simulation can significantly reduce reliance on physical testing during the engineering design phase, which in turn can increase the speed of product development. However, despite CFD's advantages of cost, time, and quality relative to wind tunnel testing, there are improvements to be made before engineers can fully rely on computational results.

Numerical simulation of high speed flow is not without its challenges. The physical processes to be modeled are complex and vary over extreme ranges. Rigorous modeling of supersonic and hypersonic flows involves modeling many physical phenomena, including: transition prediction, unsteady chemical reactions due to gas ionization at high temperature, radiative heat transfer behind shocks, use of both continuum and rarefied gas models, and unsteady shock and boundary layer interactions [3].

Even when the problem is vastly simplified to not include these various complex physical processes, computational challenges still remain. The computational mesh used to discretize the domain of the problem is a critical factor in overall solution quality. It is even possible that a solution does not exist on an under-resolved grid [15]. Unstructured meshes are capable of being iteratively refined and adapted to resolve key flow features and thereby minimize solution error. Such an adaptive framework eliminates a significant amount of user intervention and makes the CFD process more autonomous. Unstructured adapted grids can save computational resources by allowing the mesh to focus cells in regions which are most important to controlling the accuracy, and have been shown to require fewer degrees of freedom (DOF) to achieve the same level of accuracy as structured meshes [34]. However, meshes which do not align with strong shocks, which is frequently true of unstructured meshes, often do not perform well, lacking in both accuracy and robustness [12, 28]. Numerical error due to grid misalignment around shocks causes error to propagate and pollute the downstream solution. The key to mitigating these errors is in achieving a smoothed shock representation in which many DOF are used to represent a shock. There are many approaches to achieving this end. The use of higher-order discretizations, along with artificial viscosity and grid adaptation, has been shown to mitigate these errors in unstructured meshes [3].

However, grid adaptation algorithms cannot proceed if the initial solve on a coarse mesh does not converge [40]. Nonphysical oscillations around discontinuities, such as shocks, are an additional source of error that can impede the solution process from converging by making the system poorly conditioned. Another factor that is ubiquitous among CFD problems of all flow regimes is that of having a poor initial guess of the solution. An initial condition that poorly matches the boundary conditions can contribute to the ill-conditioning of a nonlinear system. This problem can occur even in a flow without strong features that would be difficult to guess or otherwise easily incorporate into an initial condition. It is important then to develop robust nonlinear solution algorithms capable of converging on poorly conditioned problems and poorly resolved meshes. High accuracy of such converged solutions is not required; as long as the solution on the current mesh converges, grid adaptation can subsequently refine the mesh to reach a high-fidelity solution.

Developing robust nonlinear solution methods for convergence of shock-dominated problems is the focus of this thesis. For simplicity, all work is done in one dimension on uniform grids.

1.2 Background

1.2.1 High Order Methods

High order methods are used to achieve higher accuracy at less computational cost compared to lower order methods. This objective is realized by reducing the discretization error. Despite the increase in accuracy, higher order methods have not been extensively adopted in industry.

The current industry standard for aerospace CFD software uses finite volume discretization. The typical approach for increasing the order of finite volume methods amounts to extending the numerical stencil to include cell "neighbors" increasingly far away. Increasing the numerical stencil complicates the boundary discretization (because there are no neighbors to one side) and increases computational expense on the interior by coupling more cells. Increasing the stencil also makes problems more difficult to linearize, so approximate Jacobians are typically used for Newton's method. To this end, industry codes typically use only second order accurate finite volume. This scheme is generally fast and robust, but lacks a high degree of accuracy:

the error of a second order finite volume scheme converges at a rate of $\mathcal{O}(h^2)$, where h is a measure of the grid size.

Finite element methods (FEM) are easily extended to high order accuracy, that is, error rates that converge faster than $\mathcal{O}(h^2)$. In general, the FEM solution is represented by:

$$u(x) = \sum_i \phi_i(x)\hat{u}_i \tag{1.1}$$

where $\phi(x)$ represents the basis polynomial functions and $\hat{u}$ represents the amplitude of each basis function, which are the unknowns of the problem. In particular, the discontinuous Galerkin (DG) finite element method can compute higher order solutions by increasing the polynomial order, p, of the basis functions inside each element. For smooth problems, the error (measured in the L^2 norm) converges at a rate of $\mathcal{O}(h^{p+1})$ [46]. Increasing the polynomial order does not increase the elemental stencil, only the number of modes in each element. Also, the DG scheme only requires information from the face neighboring elements to compute the residual.

The DG method was introduced by Reed and Hill [45] in 1973 for scalar hyperbolic equations for neutron transport. The DG method was further applied to nonlinear hyperbolic problems by Chavent, Salzano, Cockburn, and Shu [16, 20, 19, 18, 21]. Bassi and Rebay demonstrated DG for use on the Euler and Navier Stokes equations, developing the BR1 [5] and BR2 [6] schemes for viscous discretization to stabilize elliptic problems.

1.2.2 Shock Capturing

Dealing with discontinuities has long been a challenge for high order discretizations. Nonphysical oscillations arise in the vicinity of discontinuities; this is known as Gibbs phenomenon. The magnitude of Gibbs oscillations are constant and bounded away from zero [29], regardless of mesh size. In some cases, these nonphysical oscillations can spread into smoother regions of the solution, adversely affecting the solution accuracy [27] and global convergence rate.

First order schemes, on the other hand, do not suffer from such oscillations around shocks. Such schemes have second order truncation error that serves as numerical dissipation, which smears out sharp features like discontinuities. The issue of shock capturing deals with how to dissipate these oscillations in the presence of discontinuities in order to still take advantage of the high accuracy that high order discretizations otherwise enjoy.

A broad range of shock capturing methods has been investigated in the literature. One approach to shock capturing is slope limiting, originated by van Leer [52], where the cell gradient is decreased according to the neighboring cells, such that the solution is monotonicity preserving. Cockburn and Shu [20] extended this method to DG by reducing the polynomial of the cell to piece-wise constant. The limiting is applied after the residual calculation, making implicit time-stepping difficult. Moreover, these types of methods in [20] do not guarantee positive values of density and pressure [53].

Another class of shock capturing methods are referred to as Essentially Non-Oscillatory (ENO) and Weighted Essentially Non-Oscillatory (WENO). These methods use a finite volume stencil to reconstruct the polynomial, at the cost an increased stencil [31, 49]. Applied to DG, the size of the stencil can be decreased by using Hermite polynomials (HWENO) [42, 38]. The downsides to these reconstruction methods are that they are still applied outside the residual evaluation and therefore inhibit implicit time-stepping, and that they require additional programming of the finite volume scheme [27].

A third class of shock capturing methods, which is popular in the DG community, is artificial viscosity. Persson and Peraire [41] developed a method for directly adding artificial viscosity to the governing equations in the region of shocks. The artificial viscosity is piece-wise constant and scales with the sub-cell resolution, $h/(p+1)$.

Scaling the viscosity with the sub-cell resolution also scales the shock width with the sub-cell resolution, meaning that the shock can be captured in only one element for high enough p, as opposed to the shock being smeared out over several elements. So as to not add viscosity in smooth regions, a shock sensor is used to detect high frequency oscillations and turn on the artificial viscosity only in the vicinity of shocks. While this method has been successful, it does require several tuning parameters, namely the magnitude of the artificial viscosity. Too little viscosity will not capture the shock and mitigate oscillations, while too much viscosity will smear the shock too much and hurt solution accuracy. Another drawback to this method is that the piece-wise constant viscosity is not smooth, which can itself cause oscillations in the solution gradient.

Barter [3] addressed this smoothness issue by using a separate PDE to determine the artificial viscosity. This PDE-based artificial viscosity was applied to hypersonic compressible Navier Stokes flows on unstructured grids, successfully mitigating non-physical oscillations when compared to non-smooth artificial viscosity models. The drawback to PDE-based artificial viscosity is that it introduces more unknowns to the system of equations.

Guermond [29] proposed another type of artificial viscosity that scales with entropy production. Entropy production is high near shocks, so the added viscosity is large there. Smooth areas do not exhibit much entropy production, making the added viscosity negligible there. The entropy viscosity method is applicable to any conversation equations with one or more entropy inequalities, and was shown to successfully smooth shocks for scalar conservation laws and the compressible Euler equations [57].

1.2.3 Preserving Physicality

In addition to degrading solution accuracy, Gibbs oscillations can cause values that are required to be greater than zero, e.g. density or pressure, to go negative [27]. Additionally, small negative values of density or pressure in regions of high speed (such as high Mach number flow around a corner) might approximate the solution within the truncation error of the scheme [37]. Internal energy (and equivalently temperature) is also prone to going negative since it is calculated as the difference between total and kinetic energy- a small difference between two large values in high speed areas. Whatever the cause, negative values of density or pressure can cause many CFD codes to detect nonphysical errors and abort the solution process entirely. For example, a negative pressure would cause the calculation of the speed of sound, $a = \sqrt{\gamma p/\rho}$, to either fail, or at the very least, to produce a value that is not physically meaningful. Einfeldt et. al [23] defined the set of physically feasible states for the Euler equations as those for which density and internal energy are positive; Linde and Roe [37] equivalently replaced the positive internal energy constraint with a positive pressure constraint in the definition of physically feasible states for the Euler equations.

Einfeldt et. al [23] deemed the term "positively conservative" to refer to schemes that always compute positive values of density and pressure. They noted that the Roe flux scheme, without any entropy fix, is not positively conservative. For finite volume Euler schemes, Linde and Roe [37] showed that schemes which are not positively conservative can fail when nearby nonphysical states, no matter how small the time step, and offered methods for determining whether a scheme is positively conservative or not.

When using a scheme that is not positively conservative, methods need to be employed to avoid nonphysical states. In an implicit scheme, one such method is to use a line search to limit the state update. The line search is to find the maximum solution update factor that keeps the change in density and pressure under a defined fraction, and moreover, that these changes result in physical quantities [24, 39].

Ceze and Fidkowski [15] investigated pseudo-unsteady algorithms that incorporate physicality constraints. These constraints penalize the residual as nonphysical states are approached, thus repelling nonfeasible states. Incorporating physicality constraints with various line search and pseudo-unsteady continuation algorithms showed consistent convergence and decreased sensitivity to nonphysical transients. These algorithms are, however, rather sensitive to certain tuning parameters.

1.3 Thesis Overview

This thesis describes work toward developing robust solution methods in the presence of oscillations for higher order DG discretizations. The methods presented are intended to be used with a pre-existing grid-adaptive framework in conjunction with local artificial viscosity methods. The contributions made are:

- Application of alternative nonlinear solution methods to solutions with shocks
- Development of surrogate variables to eliminate nonphysical transient solutions for the Euler and Navier Stokes equations

CHAPTER 2 introduces the the governing equations and flux functions used for the Burgers and Euler equations. This chapter then describes the DG discretization. CHAPTER 3 describes the solution technique and provide details for the following continuation methods: pseudo-transient continuation, p-sequencing, dissipation-based continuation, and homotopy. CHAPTER 4 applies these continuation methods to Burgers' equation and compares the results. CHAPTER 5 introduces surrogate variables for use with fluid flow equations, and applies the surrogate variables with the previous solution methods to the Euler equations. CHAPTER 6 makes concluding remarks and future work considerations.

Chapter 2

Governing Equations and Discretization

This chapter first describes the governing equations, then summarizes the discontinuous Galerkin (DG) method for general conservation laws.

2.1 Governing Equations

In this work, two primary governing equations are considered: Burgers' equation and the Euler equations. These equations can all be written in a general conservation form. For compactness, let the subscript x be the partial derivative with respect to x: $U_x \equiv \frac{\partial U}{\partial x}$. Let $\Omega \in \mathbb{R}^1$ be a bounded domain in a 1−dimensional space. The strong form of a time-dependent conservation law in the domain, Ω, can be expressed as:

$$\frac{\partial U}{\partial t} + \frac{F^i(U,x)}{\partial x} - \frac{F^v(U,U_x,x)}{\partial x} = S(U,x), \quad \forall x \in \Omega,\ t \in I \tag{2.1}$$

with initial condition:

$$U(x,0) = U_0(x), \quad \forall x \in \Omega$$

and boundary conditions:

$$\mathcal{B}(U,U_x,x;BC) = 0 \quad \forall x \in \partial\Omega,\ t \in I$$

where $U(x,t) : \mathbb{R}^m$ is the $m-$state solution vector, $F^i(U,x) : \mathbb{R}^m$ is the inviscid flux, $F^v(U,U_x,x) : \mathbb{R}^m$ is the viscous flux, $S(U,x) : \mathbb{R}^m$ is the source term, and $\mathcal{B}$ imposes the boundary condition.

2.1.1 Burgers' Equation

The conservative state used for Burgers' equation is $U = u$. The equation for the viscous Burgers' equation is given by:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) - \frac{\partial}{\partial x}\left(\mu(u,x)\frac{\partial u}{\partial x}\right) - \alpha u - g(x) = 0, \quad \forall x \in \Omega, \ t \in I \qquad (2.2)$$

where $\alpha u + g(x) = S(u,t)$ is the source term. αu is a reaction term with α being a scalar, and $g(x)$ is a forcing term. The reaction and forcing terms are included to support a steady state shock solution at a particular x-location [3]. Without these terms, a shock just based on the boundary conditions could set up anywhere in the domain.

The solution to this equation has a state rank $m = 1$ with u being the conservative state. In terms of the general conservative form given by EQUATION 2.1, the inviscid flux is given by $F^i = \frac{1}{2}u^2$, and the viscous flux is given by $F^v = \mu(u,x)u_x$. Throughout most of this work, μ is constant in time and space; however, it is possible to use a predefined function $\mu = \mu(x)$ that is constant in time but varies in space. When using artificial viscosity, $\mu = \mu(u)$.

2.1.2 Fluid Flow Equations

Quasi-1D Euler Equations

The conservative state vector used for the one-dimensional compressible Euler equations is $U = [\rho, \rho u, \rho E]^T$, with state rank $m = 3$, where ρ is the density, u is the x-directional velocity, and E is the specific total internal energy. The area can vary with x, making the system quasi-1D. The inviscid flux vector F^i is:

$$F^i = \begin{pmatrix} \rho u A \\ \left(\rho u^2 + p\right) A \\ \rho u H A \end{pmatrix} \tag{2.3}$$

where p is the static pressure and $H = E + p/\rho$ is the specific total enthalpy, and A is the area. The state equation, calculated in terms of the primitive variables, $Z = [\rho, u, T]^T$, is:

$$p = \rho R T. \tag{2.4}$$

Specific total internal energy can also be calculated from the primitive variables as

$$E = c_v T + \frac{1}{2} u^2. \tag{2.5}$$

The steady version of EQUATION 2.1 for the quasi-1D Euler equations is:

$$\frac{\partial}{\partial x} \begin{pmatrix} \rho u A \\ (\rho u^2 + p) A \\ \rho u H A \end{pmatrix} - \begin{pmatrix} 0 \\ p \frac{\partial A}{\partial x} \\ 0 \end{pmatrix} = 0. \tag{2.6}$$

Compressible Navier Stokes

For a Newtonian fluid, the sheer stress τ is given by:

$$\tau_{xx} = \mu \left(2 \frac{\partial u}{\partial x} \right) + \lambda \frac{\partial u}{\partial x} \tag{2.7}$$

where μ is the dynamic viscosity, and $\lambda = -2/3\mu$ is the bulk viscosity coefficient. For a quasi-1D problem, the inviscid flux vector, F^i, is given by EQUATION 2.3, and the viscous flux vector, F^v, written in terms of the primitive variables is:

$$F^v = \begin{pmatrix} 0 \\ \frac{4}{3}\mu\frac{\partial u}{\partial x} \\ \frac{4}{3}\mu\frac{\partial u}{\partial x}u + k\frac{\partial T}{\partial x} \end{pmatrix} \tag{2.8}$$

where T is the temperature, k is the thermal conductivity, and the dynamic viscosity, μ, is assumed to be constant throughout the domain. This assumption simplifies the linearization. With a specific heat at constant pressure $c_p = \gamma R/(\gamma - 1)$ and a specified Prandtl number, $Pr = 0.72$ for air, the thermal conductivity, k, is determined by:

$$k = c_p \frac{\mu}{Pr}. \tag{2.9}$$

Nondimensionalization

All of the variables for the fluid flow equations are nondimensionalized in order to make the Jacobian better conditioned, as well as for plotting purposes. The nondimensionalization scheme is summarized by TABLE 2.1.

Quantity	**Symbol**	**Normalization**	**FreestreamValue**
Density	ρ	ρ_∞	1
Pressure	p	p_∞	1
Velocity	u	$\sqrt{p_\infty/\rho_\infty}$	$\sqrt{\gamma}M_\infty$
Speed of sound	$a = \sqrt{\gamma p/\rho}$	$\sqrt{p_\infty/\rho_\infty}$	$\sqrt{\gamma}$
Specific energy	$E = \frac{p}{(\gamma-1)\rho} + u^2/2$	p_∞/ρ_∞	$1/(\gamma-1) + \gamma M_\infty^2/2$
Temperature	T	T_∞	1
Gas consant	R	R_∞	1
Viscosity	$\mu = \rho u L/Re$	$L_\infty\sqrt{p_\infty/\rho_\infty}$	$\gamma M_\infty/Re_\infty$

TABLE 2.1: Inviscid flow nondimensionalization

2.2 Discontinuous Galerkin Discretization

For the discontinuous Galerkin discretization, let $\mathcal{T}_h$ be a triangulation of the 1-dimensional domain Ω with non-overlapping elements, κ, of length h. Also define a function space $\Phi_{h,p}$ as:

$$\Phi_{h,p} \equiv \{\phi \in \left(L^2(\Omega)\right)^m : \phi|_\kappa \in \left(\mathcal{P}^p(\kappa)\right)^m, \forall \kappa \in \mathcal{T}_h\}, \tag{2.10}$$

where $\mathcal{P}^p(\kappa)$ represents the solution space of p-th degree polynomials on a physical element κ. Taking the product of EQUATION 2.1 with a test function $\phi_{h,p} \in \Phi_{h,p}$, and integrating by parts yields the weak formulation of the governing equation. Solving the weak formulation finds a solution $u_{h,p}(\cdot, t) \in \Phi_{h,p}$ such that:

$$\sum_{\kappa\in\mathcal{T}_h} \int_\kappa \phi_{h,p}^T \frac{\partial u_{h,p}}{\partial t} + \mathcal{R}_{h,p}(u_{h,p}, \phi_{h,p}) = 0 \quad \forall \phi_{h,p} \in \Phi_{h,p}. \tag{2.11}$$

where the weighted residual $\mathcal{R}_{h,p}$ is comprised of inviscid ($\mathcal{R}^i$), viscous ($\mathcal{R}^v$), and source ($\mathcal{R}^s$) discretization terms:

$$\mathcal{R}_{h,p}(w_{h,p}, \phi_{h,p}) = \mathcal{R}^i_{h,p}(w_{h,p}, \phi_{h,p}) + \mathcal{R}^v_{h,p}(w_{h,p}, \phi_{h,p}) + \mathcal{R}^s_{h,p}(w_{h,p}, \phi_{h,p}). \tag{2.12}$$

2.2.1 Inviscid Discretization

The DG discretization of the inviscid term is given by:

$$\begin{aligned}\mathcal{R}^i_{h,p}(w, \phi) = &- \sum_{\kappa\in\mathcal{T}_h} \int_\kappa \phi_x^T F^i(w) \\ &+ \sum_{f\in\Gamma_i} (\phi^+ - \phi^-)^T \mathcal{H}(w^+, w^-; \hat{n}^+) \\ &+ \sum_{f\in\Gamma_b} \phi^{+^T} \mathcal{H}^b(w^+, u^b(w^+; BC); \hat{n}^+)\end{aligned} \tag{2.13}$$

where $(\cdot)^+$ and $(\cdot)^-$ denote trace values taken from opposite sides of a face f, $\hat{n}^+$ is the normal vector pointing from the (+) side to the (-) side and equal to either (+1) or (-1) in 1 dimension, $\mathcal{H}$ and $\mathcal{H}^b$ are numerical flux functions on interior and boundary faces respectively, u^b is the boundary state constructed from the interior state and a specified boundary condition, and Γ_i and Γ_b are the interior and boundary faces, respectively. The inviscid boundary flux $\mathcal{H}^b$, is calculated by evaluating the flux at a boundary state, u_b, which is a function of both the interior state, w^+, and a user-specified boundary condition, BC.

Local Lax-Friedrichs Flux

The upwinding flux function used for Burgers' equation is the Local Lax-Friedrichs flux [20]. The flux function is written for Burgers' equation as

$$\mathcal{H}_{LLF}(u^-, u^+) = \frac{1}{2}\left(\frac{1}{2}(u^-)^2 + \frac{1}{2}(u^+)^2\right) - \frac{1}{2}\max\left(|(u^-)|, |(u^+)|\right)(u^+ - u^-) \qquad (2.14)$$

Both the max$(\cdot,\cdot)$ function and the absolute value of u must be evaluated using a smooth function to ensure that the derivatives are continuous. The modified smooth max$(\cdot,\cdot)$ function, denoted with an asterisk (*), and its derivative are defined as:

$$\max{}^*(u_1, u_2) = \frac{u_1 e^{\alpha u_1} + u_2 e^{\alpha u_2}}{e^{\alpha u_1} + e^{\alpha u_2}} \qquad (2.15)$$

$$\frac{\partial \max{}^*(u_1, u_2)}{\partial u_i} = \frac{e^{\alpha u_i}\left(1 + \alpha(u_i - \max{}^*(u_1, u_2))\right)}{e^{\alpha u_1} + e^{\alpha u_2}} \quad , \; i = 1, 2 \qquad (2.16)$$

with $\alpha = 5$. Note that as $\alpha \to +\infty$, the function better approximates the maximum, but can fall susceptible to floating point errors as $e^{\alpha u_i}$ approaches infinity. Also note that as $\alpha \to -\infty$, the function approximates min$(\cdot,\cdot)$.

The modified smooth absolute value function, denoted with an asterisk (*) and its derivative are defined as:

$$|u|^* = \frac{u^2}{\text{sign}(u)u + \varepsilon} \tag{2.17}$$

$$\frac{\partial |u|^*}{\partial u} = \frac{2u - \text{sign}(u)u^*}{\text{sign}(u)u + \varepsilon} \tag{2.18}$$

with $\varepsilon = 10^{-8}$.

Roe flux

The upwinding flux function used for the fluid flow equations is Roe's approximate Riemann solver [48] with entropy fix. Using the notation from [26] and letting $\Delta U = (U^- - U^+)$ represent the jump across a cell interface, the flux function is written as

$$\mathcal{H}_{Roe}(Z^-, Z^+) = \frac{1}{2}\left(F^i(Z^-) + F^i(Z^+)\right) + |\lambda_3|(U(Z^-) - U(Z^+)) + \begin{pmatrix} C_1 \\ C_1\bar{u} + C_2\hat{n} \\ C_1\bar{H} + C_2\bar{u}\hat{n} \end{pmatrix} \tag{2.19}$$

where $\hat{n}^+$ is the normal vector pointing from the (+) side to the (-) side and equal to either (+1) or (-1) in 1 dimension, and an overbar, $\bar{(\cdot)}$, denotes the Roe average value. The variables C_1 and C_2 are

$$C_1 = \frac{G_1 s_1}{\bar{a}^2} + \frac{G_2 s_2}{\bar{a}} \ , \ C_2 = \frac{G_1 s_2}{\bar{a}} + G_2 s_1 \tag{2.20}$$

with

$$G_1 = (\gamma - 1)\left(\frac{1}{2}\bar{u}^2 \Delta\rho - \bar{u}\Delta(\rho u) + \Delta(\rho E)\right) \tag{2.21}$$

$$G_2 = -\bar{u}\hat{n}\Delta\rho + \Delta(\rho u)\hat{n}$$

and

$$s_1 = \frac{1}{2}\left(|\lambda_1| + |\lambda_2|\right) - |\lambda_3| \ , \ s_2 = \frac{1}{2}\left(|\lambda_1| - |\lambda_2|\right). \tag{2.22}$$

λ_i represents the characteristic velocities, which are the Roe averaged eigenvalues, defined as

$$\lambda_1 = \bar{u}\hat{n} + \bar{a} \ , \ \lambda_2 = \bar{u}\hat{n} - \bar{a} \ , \ \lambda_3 = \bar{u}\hat{n}. \tag{2.23}$$

Using an entropy fix for when these values are small, the absolute values are written

$$|\lambda_i| = \begin{cases} \frac{1}{2}\left(\varepsilon\bar{a} + \frac{\lambda_i^2}{\varepsilon\bar{a}}\right) & -\varepsilon\bar{a} < \lambda_i < \varepsilon\bar{a} \\ \sqrt{\lambda_i^2} & \text{otherwise} \end{cases} \tag{2.24}$$

with $\varepsilon = 0.01$. The Roe averaged velocity, enthalpy, and speed of sound are defined as

$$\bar{u} = \frac{\sqrt{\rho^-}\bar{u}^- + \sqrt{\rho^+}\bar{u}^+}{\sqrt{\rho^-} + \sqrt{\rho^+}} \tag{2.25}$$

$$\bar{H} = \frac{\sqrt{\rho^-}\bar{H}^- + \sqrt{\rho^+}\bar{H}^+}{\sqrt{\rho^-} + \sqrt{\rho^+}} \tag{2.26}$$

$$\bar{a}^2 = (\gamma - 1)\left(\bar{H} - \frac{1}{2}\bar{u}^2\right). \tag{2.27}$$

However, to ensure that $\bar{a}^2$ is always positive, it is rewritten and evaluated as a function of the primitive variables, which can be guaranteed to be positive in ρ and T (see SECTION 5.1):

$$\bar{a}^2 = (\gamma - 1)\frac{(\rho^- T^- + \rho^+ T^+)(c_v + R) + \sqrt{\rho^-\rho^+}\left((T^- + T^+)(c_v + R) + \frac{1}{2}\left(u^- - u^+\right)^2\right)}{\left(\sqrt{\rho^-} + \sqrt{\rho^+}\right)^2}. \tag{2.28}$$

2.2.2 Viscous Discretization

The second method of Bassi and Rebay (BR2 scheme) [7] is used to discretize the viscous terms. For compactness, the jump $[[\cdot]]$ and average $\{\cdot\}$ operators are used.

For a scalar s, the jump and averages on interior faces are defined as:

$$\{s\} = \frac{1}{2}(s^+ + s^-), \qquad [[s]] = s^+\hat{n}^+ + s^-\hat{n}^-$$

and on boundary faces as:

$$\{s\} = s^+, \qquad [[s]] = s^+\hat{n}^+.$$

The viscous discretization is:

$$\begin{aligned}\mathcal{R}^v_{h,p}(w,\phi) = &\sum_{\kappa\in\mathcal{T}}\int_\kappa \phi_x^T F^v(w, w_x + R([[w]])) \\ &- \sum_{f\in\Gamma_i} [[\phi]]^T\{F^v(w, w_x + \eta_f r_f([[w]]))\} \\ &- \sum_{f\in\Gamma_b} \phi^{+^T}\left(F^v\left(u^b, u^b_x + \eta_f r^b_f(w^+ - u^b)\right)\right)\hat{n}^+.\end{aligned} \tag{2.29}$$

where $u^b(w^+, BC)$ and $u^b_x(w^+_x; BC)$ are chosen to specify the boundary viscous flux, r_f and r^b_f are the lifting operators on an interior and boundary face respectively, and η_f is the stabilizing parameter. The viscosity and corresponding viscous flux on the boundary is calculated as function of the boundary state, u_b, rather than the interior, w^+. This choice proved to give to viscous flux a better stabilizing effect on the solution. The stabilization parameter is set to $\eta_f = 2$, because it must be greater than or equal to the number of faces of an element [22]. The lifting operators provide coupling between neighboring elements by penalizing jumps in the solution. For every face f, find $r_f \in \Phi_{h,p}$ such that for interior faces

$$\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \phi^T r_f([[w]]) = [[w]]^T\{\phi\} \quad \forall\phi\in\Phi_{h,p} \tag{2.30}$$

and for boundary faces

$$\sum_{\kappa \in \mathcal{T}_h} \int_\kappa \phi^T r_f^b(w) = w^T \phi^+ \hat{n}^+ \quad \forall \phi \in \Phi_{h,p} \tag{2.31}$$

with

$$R([[w]]) = \sum_{f \in \Gamma_i} r_f([[w]]). \tag{2.32}$$

2.2.3 Source Discretization

The source terms are not a function of the state gradient, so no lifting operators need to be used. The source term is discretized as:

$$\mathcal{R}^s_{h,p}(w, \phi) = \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \phi^T S(w). \tag{2.33}$$

Chapter 3

Nonlinear Solution Algorithms

This chapter first summarizes the Newton-Raphson solution algorithm for generic residuals, as well as the line search methods used. The chapter then describes the continuation algorithms explored.

3.1 Newton-Raphson Solver

After choosing basis functions in the approximation space $\Phi_{h,p}$, EQUATION 2.11 becomes a discrete root-finding problem. The steady discrete equation can be expressed as a system of algebraic equations, which allows for finding Q such that:

$$R_s(Q) = 0 \tag{3.1}$$

where $R_s(Q)$ is the spatial residual vector and Q is the solution vector (i.e. basis function weights). Given an initial approximation to the solution vector, Q^n, the approximate solution at the next iteration, Q^{n+1}, is found by approximately solving

$$R_s(Q^{n+1}) = 0 \tag{3.2}$$

Specifically, at each iteration, the Newton-Raphson method is used to solve EQUATION 3.2 such that:

$$Q^{n+1} - Q^n \approx \Delta Q \equiv - \left(\frac{\partial R_s}{\partial Q}\bigg|_{Q^n} \right)^{-1} R_s(Q^n) \tag{3.3}$$

The Newton iterations are continued until the steady spatial residual's L^2 norm $||R_s(Q^{n+1})||_2$ is less than some specified tolerance.

3.1.1 Line search

Each Newton step ΔQ is limited by a line search with the update fraction, η, to ensure a decrease of the L^2 norm of the steady residual. ΔQ is considered the direction of the step, while η is the magnitude of the step. Once the line search finds an acceptable update fraction,

$$Q^{n+1} = Q^n + \eta \Delta Q. \tag{3.4}$$

Residual norm line search

Two line search algorithms are implemented to decrease the residual norm. The first algorithm halves the step size until the L^2 norm of the intermediate residual is less than the L^2 norm of the current residual. This procedure is summarized in ALGORITHM 1. The stopping criterion, $\eta_{min} = 9.5 \times 10^{-7}$, corresponds to 20 halving iterations when starting from $\eta_0 = 1$. In general, $\eta_0 = 1$, unless a physicality line search has imposed a smaller η_0. Continuation methods are considered when the line search is unable to find a suitable update fraction $\eta \geq \eta_{min}$.

Algorithm: *Halving line search algorithm*

```
$\eta = \eta_0$,  $Q^{n*} = Q^n + \eta\Delta Q$ ;
while $||R_s(Q^{n*})||^2 > ||R_s(Q^n)||^2$ & $\eta \geq \eta_{min}$ do
    $\eta \leftarrow \frac{\eta}{2}$,   $Q^{n*} = Q^n + \eta\Delta Q$;
end
if $\eta \geq \eta_{min}$ then
    $Q^{n+1} = Q^n + \eta\Delta Q$
else
    abort line search
end
```

Algorithm 1: Halving line search

Let an asterisk (*) denote the intermediate solution at a given line search iteration. $R_s^* = R_s(Q^{n*})$ and $Q^{n*} = Q^n + \eta\Delta Q$. The derivative of the square of the L^2 norm of R_s^* with respect to the update fraction is:

$$\begin{aligned}\frac{d||R_s^*||^2}{d\eta} &= \frac{d\left(R_s^{*T} R_s^*\right)}{d\eta} \\ &= 2R_s^{*T}\frac{dR_s^*}{d\eta} \\ &= 2R_s^{*T}\frac{\partial R_s^*}{\partial Q^*}\frac{dQ^*}{d\eta} \\ &= 2R_s^{*T}\frac{\partial R_s^*}{\partial Q^*}\Delta Q\end{aligned} \tag{3.5}$$

As $\eta \to 0, Q^{n*} \to Q^n$, thus $\frac{\partial R_s^*(Q^{n*})}{\partial Q^*}\Delta Q \to -R_s(Q^n)$ according to EQUATION 3.3, and $\frac{d||R_s^*||^2}{d\eta} \to -2R_s^T R_s \leq 0$. Then for small enough η, the slope of the square L^2 norm of the residual is negative, so the residual must decrease. The halving line search is then guaranteed to reduce the square of the L^2 norm of the residual for small enough η.

In order to reduce the number of Newton iterations needed, it can be advantageous to find an optimal update fraction. An optimal update fraction is one that decreases the residual as much as possible given the current step direction ΔQ. To do this, Brent's method [9] is used as the root-finding algorithm which solves $f(\eta) = 0$, where $f(\eta)$ for our problem is

$$f(\eta) = \frac{d||R_s^*||^2}{d\eta}. \tag{3.6}$$

While Brent's method is known for its rather complicated logic, it is also known for its efficiency [56, 50]. The algorithm selects the fastest root-finding method of the three choices: inverse quadratic interpolation, secant interpolation, and bisection. Inverse quadratic interpolation and secant interpolation both converge superlinearly, provided the function to be minimized is C^2 smooth near its minimum [10]. If neither inverse quadratic or secant interpolation are feasible, Brent's method relies on the more robust bisection method. ALGORITHM 2 summarizes how the Brent algorithm solves $f(\eta) = 0$ given the functional $f(\eta)$ on the interval $\eta \in [a, b]$. In practice, the interval $\eta \in [0, 1]$ is used, in which $\eta = 0$ corresponds to no update at all, and $\eta = 1$ corresponds to the full Newton update.

If the tolerances for the Brent algorithm are set too tight, or if the system is ill-conditioned or highly nonlinear- as are many of the cases investigated in this work- Brent's method may not converge before hitting a user-defined maximum number of iterations. If this occurs, it is possible that the output of the Brent algorithm still does not reduce the residual. It is also possible, although uncommon, that the Brent algorithm converges on a local maximum instead of a local minimum. For this reason, whenever the Brent algorithm is used, the halving line search is also used in sequence as a fail-safe, as summarized by ALGORITHM 3.

Finding the root of EQUATION 3.6 using Brent's algorithm requires one Jacobian evaluation per functional evaluation, which can be noticeably computationally expensive for larger, nonlinear systems of equations such as Euler or Navier Stokes. For smaller problems where the Jacobian is not so large, like Burgers' equation, the benefit of minimizing $R_s(Q^{n*})$ every time the line search is called tends to outweigh the cost of the additional associated Jacobian evaluations. A cheaper alternative to finding the root of EQUATION 3.6 is to instead find the root of the energy functional, $f = R_s^{*T}\Delta Q$ [25]. While cheaper, use of this functional proved to be less reliable than the L^2 functional in practice.

Physicality line search

A physicality line search is used prior to using the halving line search when solving the Euler equations using the physical primitive variables, $Q = Z$. The Brent line search is not used due to its expensive Jacobian evaluations. The physicality line search sets an upper limit on the update fraction to ensure that density and temperature do not go below some small critical value, ε. This upper limit sets η_0, which is then used by ALGORITHM 1. Since these variables are assumed to be greater than zero,

```
Algorithm: Brent's method line search

Calculate $f(a), f(b)$;
if $f(a)f(b) > 0$ then
    the root is not bracketed, exit function
end
if $|f(a)| < |f(b)|$ then
    swap $a, b$
end
$c = a$;
flag = 1;
while $f(b) = 0$ or $(|b - a| \leq tol$ & $iter \leq maxiter)$ do
    if $f(a) \neq f(c)$ & $f(b) \neq f(c)$ then
        Inverse quadratic interpolation;
        $s = \frac{af(a)f(c)}{(f(a) - f(b))(f(a) - f(c))} + \frac{bf(a)f(c)}{(f(b) - f(a))(f(b) - f(c))} + \frac{cf(a)f(b)}{(f(c) - f(a))(f(c) - f(b))}$
    else
        Secant interpolation;
        $s = b - f(b)\frac{b - a}{f(b) - f(a)}$
    end
    if $s$ is not $\in \left[\frac{3a+b}{4}, \; b\right]$ or
    $flag = 1$ & $|a - b| \geq |b - c|/2$ or
    $flag = 0$ & $|a - b| \geq |c - d|/2$ or
    $flag = 1$ & $|b - c| < tol$ or
    $flag = 0$ & $|c - d| < tol$ or then
        Bisection;
        $s = \frac{a + b}{2}$;
        flag = 1
    else
        flag = 0
    end
    Calculate $f(s)$;
    $d = c$;
    $c = b$;
    if $f(a)f(s) < 0$ then
        $b = s$
    else
        $a = s$
    end
    if $|f(a)| < |f(b)|$ then
        swap $a, b$
    end
    iter = iter+1
end
Output $b$
```

Algorithm 2: Brent's method line search

Algorithm: *Brent's method line search with halving fail-safe*

```
$\eta = 1$;
if $||R_s(Q^n + \eta \Delta Q)||^2 > ||R_s(Q^n)||^2$ then
    Brent's algorithm on the interval $\eta \in [0, 1]$;
    $\eta = \eta_{Brent}$;
    if $||R_s(Q^n + \eta \Delta Q)||^2 > ||R_s(Q^n)||^2$ then
        Halving algorithm;
        $\eta = \eta_{Halving}$
    end
    if $||R_s(Q^n + \eta \Delta Q)||^2 > ||R_s(Q^n)||^2$ then
        Cannot reduce residual; end run
    end
end
```

Algorithm 3: Brent's method line search with halving fail-safe

and the update fraction is never allowed to be negative, only negative ΔQ values are considered.

$$\eta_{\rho,i} = \begin{cases} \dfrac{\varepsilon_\rho - \rho_i}{\Delta \rho_i} & \Delta \rho_i < 0 \\ 1 & \Delta \rho_i \geq 0 \end{cases} , \quad \eta_{T,i} = \begin{cases} \dfrac{\varepsilon_T - T_i}{\Delta T_i} & \Delta T_i < 0 \\ 1 & \Delta T_i \geq 0 \end{cases} \tag{3.7}$$

$$\eta_0 = \min_i \left(\eta_{\rho,i}, \eta_{T,i}, 1 \right) \tag{3.8}$$

For consistency with the critical values used by the surrogate primitive variables introduced in CHAPTER 5, $\varepsilon_\rho = 0.01$ and $\varepsilon_T = 0.001$ are used.

3.2 Continuation Methods

Newton's method often fails for highly nonlinear problems with poor initial guesses. In order to globalize Newton's method, continuation methods are used to obtain better initial guesses. For well-behaved problems, globally convergent continuation algorithms are slower to converge than Newton's method by itself. However, continuation methods can help poorly-behaved problems converge when they may have otherwise failed. The line search is always used in conjunction with the continuation methods unless otherwise noted. This section describes the continuation techniques explored in conjunction with Newton's method: pseudo-transient continuation, p-sequencing, and diffusion-based continuation.

3.2.1 Pseudo-Transient Continuation

Pseudo-transient continuation, PTC, is a popular continuation method that imitates physical time-marching. Decreasing the time step makes the problem more similar to a real physical, unsteady system, which means the transient solution path is less likely to go through a nonphysical state. Since the user is only interested in a steady state solution, time accuracy is not necessary. If the system becomes poorly conditioned and must employ a line search to decrease the residual, smaller time steps can be taken to better imitate a real system. When the system is well conditioned, larger time steps can be taken to drive the solution to the steady state. The pseudo-transient version of the problem is solved using implicit backward Euler time marching. The unsteady version of EQUATION 3.2 becomes:

$$R_t(Q^{n+1}) \equiv M^t(U(Q^{n+1}) - U(Q^n)) + R_s(Q^{n+1}) = 0 \tag{3.9}$$

where R_t is the pseudo-unsteady residual and M^t is the mass matrix weighted by a local elemental time step Δt_κ. This time step is calculated as a function of the global CFL number defined as:

$$\mathrm{CFL} = \frac{\Delta \mathrm{t}_\kappa \lambda_{\mathrm{max}}}{\mathrm{h}_\kappa} \tag{3.10}$$

where h_κ is the element size and λ_{max} is the maximum characteristic speed. For Burgers' equation, λ_{max} refers to the maximum characteristic speed in the entire domain, and $\lambda = |u|^*$; for the Euler equations, λ_{max} refers to the maximum characteristic speed within the element κ, and $\lambda = |u|^* + a$. The Newton-Raphson method now solves the pseudo-unsteady version of EQUATION 3.3:

$$Q^{n+1} - Q^n \approx \Delta Q \equiv - \left(M^t + \left.\frac{\partial R_s}{\partial Q}\right|_{Q^n} \right)^{-1} R_s(Q^n) \tag{3.11}$$

A number of CFL evolution strategies exist to increase the robustness of the PTC method, since pseudo-time is less needed as the solution approaches the steady state solution. When using PTC, unless otherwise noted, exponential progression with under-relaxation is used to evolve the CFL. If the residual is successfully reduced without the help of a line search, the CFL is increased, meaning that a larger time step is taken towards steady state on the next Newton iteration. On the other hand, if the Newton iteration requires a line search but the line search fails to find an update fraction that reduces the residual, the CFL is decreased and the Newton iteration is retried with a correspondingly smaller time step. This method is summarized by ALGORITHM 4. Other potential CFL evolution algorithms are switched evolution relaxation, and the residual difference method. In both of these methods, the CFL is evolved based on the relative change in the residual [14].

While PTC methods are widely used in both industry and academia, it remains unclear how to implement PTC when using unstructured space-time grids. This is one reason for exploring alternative continuation strategies.

```
Algorithm: Exponential Progression with Under-relaxation
$\beta > 1, \quad \kappa < 1$;
if $\eta = 1$ then
    CFL $\leftarrow \min(\beta \mathrm{CFL}, \mathrm{CFL}_{\mathrm{max}})$
else
    if $\eta \leq \eta_{min}$ then
        $\eta = 0$;
        CFL $\leftarrow \max(\kappa \mathrm{CFL}, \mathrm{CFL}_{\mathrm{min}})$
    end
end
```

Algorithm 4: Exponential Progression with Under-relaxation

3.2.2 P-Sequencing

P-sequencing is another common continuation method for higher-order solutions that can be used in addition to other methods. Lower-order solutions are used as the initial condition for higher-order solutions. Higher-order problems with strong shocks tend to have difficulty converging without p-sequencing [4]. Since P0 (that is, zero-order polynomial or finite volume) solutions converge very robustly, p-sequencing is an effective tool for generating better initial guesses for the Newton-Raphson method with higher-order polynomials. The system is solved in discrete sub-problems of increasing polynomial order. A solution is first obtained for a low polynomial order, either P0 (for Burgers' equation) or P1 (for the Euler equations). The BR2 viscous discretization is only consistent for P0 in 1D; so while starting from P0 is fine for 1D or quasi-1D, extensions to higher dimensions may need to start with P1. Then, that solution is projected onto the next-higher polynomial space to serve as the initial condition for the sub-problem of the next-higher polynomial order, until the desired polynomial order is reached.

3.2.3 Dissipation-Based Continuation

In order to capture shocks and prevent oscillations, it is common to add artificial viscosity or some type of numerical dissipation to the problem [41, 3]. To this end, Hicken and Zingg [33, 32] showed that dissipation-based parameter continuation (DBC) is capable of outperforming pseudo-transient continuation for the Euler equations solved with an inexact Newton method. Similar to p-sequencing, DBC uses the solution to a higher-dissipation problem as the initial condition for a lower-dissipation problem. EQUATION 2.12 becomes:

$$\mathcal{R} = \mathcal{R}^i + \mathcal{R}^v + \mathcal{R}^s + \lambda \mathcal{R}^d \tag{3.12}$$

where $\mathcal{R}^d$ is the residual for the added dissipation terms, and $\lambda \in [0, \lambda_{max}]$ is the continuation parameter which scales the magnitude of the added dissipation. The value of λ begins that $\lambda_0 = \lambda_{max}$ and is reduced towards zero after every sub-problem solve until $\lambda \rightarrow 0$ and EQUATION 2.12 is recovered.

The initial value and evolution of λ affect the robustness of the DBC algorithm. For both Burgers' equation and the Euler equations, the initial value of λ_0 is set to 1. Initially a tuning parameter, this value of λ_0 was chosen for Burgers' equation because it created enough dissipation to consistently converge the various initial conditions tested. For the Euler equations, $\lambda_0 = 1$ sets the artificial viscosity magnitude equal to the viscosity based on the user-input Reynolds number with global length scale. λ is reduced by a factor of 10 after each sub-problem, until a minimum value, $\lambda_{min} = 10^{-6}$. is hit where the additional dissipation becomes negligible, and the original problem can be solved with $\lambda = 0$. This evolution is summarized by ALGORITHM 5. Granted, this continuation schedule is somewhat arbitrary, and may not be as robust for the Euler equations as it is for Burgers' equation. This issue is addressed in SECTION 3.2.4.

Algorithm: *DBC Continuation Parameter Evolution*

```
$\kappa = 10,\ \ \lambda_{min} = 10^{-6}$;
if $\lambda > \kappa\lambda_{min}$ then
    $\lambda \leftarrow \lambda/\kappa$
else
    $\lambda \leftarrow 0$
end
```

Algorithm 5: DBC Continuation Parameter Evolution

3.2.4 Homotopy Continuation

The issue that remains with DBC is how to best evolve the continuation parameter. Instead of solving sub-problems at discrete values of λ, one can include λ as a variable rather than parameter, and solve a modified problem with a method known as homotopy continuation [1].

Homotopy is an established continuation method in numerical algebraic geometry and bifurcation analysis [35], and has been applied in chemical engineering analysis [51, 17, 43], and to some extent in circuitry analysis [36] and computer science [44]. Homotopy continuation methods had not seen many applications in CFD until recent years, and was initially investigated for individual flow problems using parameters specific to the problem [13, 47]. More recently, dissipation-based homotopy has been implemented as a general globalization method with Finite Difference and DG schemes for the

Euler equations [30, 55], as well as with Finite Volume schemes for the Euler, Navier Stokes, and Reynolds-Averaged Navier Stokes (RANS) equations [11]. In several of these cases, homotopy-based continuation methods have outperformed PTC methods in solver robustness and efficiency [11, 32].

Homotopy Overview

Let $F(Q) = 0$ represent a generic system of equations with $F(Q) : \mathbb{R}^m \rightarrow \mathbb{R}^m$. In this case, $F(Q) = 0$ represents Equation 2.11. $F(Q)$ might be highly nonlinear and hard to solve. Let $G(Q) = 0$ be a different system of equations with $G(Q) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ that has a known solution and is easy to solve regardless of the initial condition. Let Q_0 represent the initial condition. The homotopy continuation method first solves $G(Q_0) = 0$, then gradually morphs the system of equations $G(Q)$ into $F(Q)$ using $\lambda \in [0, 1]$.

$$H(Q, \lambda) = \lambda F(Q) + (1 - \lambda)G(Q) = 0 \tag{3.13}$$

For $\lambda = 0$, $H(Q, \lambda) = G(Q)$, which is the easy-to-solve problem. For $\lambda = 1$, $H(Q, \lambda) = F(Q)$, which is the hard-to-solve problem. If $G(Q)$ is linear, the Newton-Raphson solver will converge when $\lambda = 0$, regardless of the initial condition. This gets around the issue of having initial conditions that poorly match that boundary conditions. $G(Q_0) = 0$ must have a unique solution, and it must be twice differentiable [17].

For Burger's equation, $G(Q) = 0$ represents $\mathcal{R}^d = 0$. For the Euler and Navier Stokes equations, $G(Q) = 0$ represents $(\mathcal{R}^i + \mathcal{R}^v + \mathcal{R}^s) + \mathcal{R}^d = 0$, where the terms in parentheses represent the steady residual of the actual system. This choice of $G(Q)$ recovers the full steady residual in the homotopy equation, but with added dissipation that is gradually reduced, i.e.

$$\begin{aligned} H(Q,\lambda)_{Euler,NS} &= \lambda\left(\mathcal{R}^i+\mathcal{R}^v+\mathcal{R}^s\right)+(1-\lambda)\left(\left(\mathcal{R}^i+\mathcal{R}^v+\mathcal{R}^s\right)+\mathcal{R}^d\right) \\ &= \left(\mathcal{R}^i+\mathcal{R}^v+\mathcal{R}^s\right)+(1-\lambda)\mathcal{R}^d \end{aligned} \tag{3.14}$$

Because the $(\mathcal{R}^i+\mathcal{R}^v+\mathcal{R}^s)$ term is not scaled by λ, this continuation method is very similar to that of DBC. Including the steady residual also makes $G(Q)$ more nonlinear, meaning that convergence of $G(Q_0)=0$ is not guaranteed. Despite these drawbacks, this choice of $G(Q)$ is used because it ensures consistent boundary conditions for $H(Q,\lambda)$.

The curve $Q(\lambda)$ that satisfies EQUATION 3.13 and connects $\lambda=0$ to $\lambda=1$ is called the homotopy path. Provided there is a unique homotopy path connecting $H(Q_0,0)$ and $H(Q,1)$ for a given initial condition, Q_0, the method is probability-one convergent [54]. A homotopy path exists and is unique if and only if the Jacobian of $H(Q,\lambda)$ is full rank for all $\lambda\in[0,1]$. The Jacobian is size $n\times(n+1)$, and is thus required to have rank n. This means that $F(Q)$ can become rank-1 deficient without affecting the homotopy path [51]. This gives the homotopy method one advantage not only over PTC methods, but also over the very similar DBC method proposed in SECTION 3.2.3, wherein the Jacobian for both of these methods is always $n\times n$ and will fail when $F(Q)$ nears singularity.

A predictor-corrector method is used to numerically follow the homotopy path along λ. Once on the homotopy path, the predictor takes a step in the direction tangent to the path. Then, the corrector refines this rough guess to hone in on the actual path. This sequence of steps is represented in FIGURE 3-1.

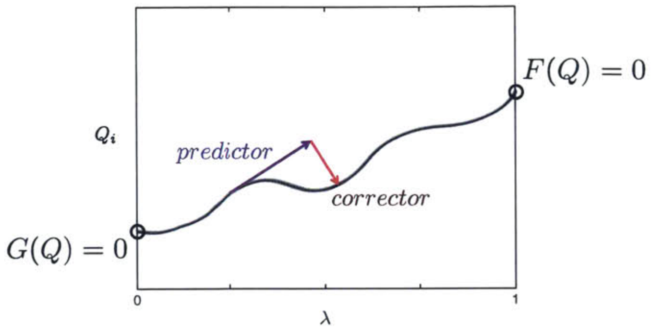


FIGURE 3-1: Example of a homotopy path

Euler Predictor

Let W be the augmented solution vector such that:

$$W \equiv \begin{pmatrix} Q \\ \lambda \end{pmatrix}. \tag{3.15}$$

A point on or near the homotopy path after k steps is

$$W_k = W(s_k) \tag{3.16}$$

where s_k is thought of as the arc length from the starting point for $k \geq 0$, such that

$$\Delta s_k = s_{k+1} - s_k \approx ||W_{k+1} - W_k||_2. \tag{3.17}$$

Note that the actual value of s_k is never needed, only the step size, δs_k. The step taken for each Euler predictor is equal to the step size times the unit tangent vector. The unit tangent vector, $\frac{dW}{ds}(s_k)$ satisfies:

$$\begin{pmatrix} \dfrac{\partial H}{\partial W}(W_k) \\ \dfrac{dW^T}{ds}(s_k) \end{pmatrix} \frac{dW}{ds}(s_k) = N \tag{3.18}$$

where N is a unit vector with the same dimensions as w, equal to all zeros and one in the last entry: $N = [0 \;\; 0 \;\; \cdots \;\; 1]^T$. In order to keep the path going in a consistent direction, and not turning back on itself, it is required that:

$$\frac{dW^T}{ds}(s_{k-1})\frac{dW}{ds}(s_k) > 0 \tag{3.19}$$

for $k \geq 1$. In order to make the system linear to calculate the tangent vector, the bottom row of EQUATION 3.18 is replaced by a known vector, the effect of which only changes the norm of the tangent vector [17]. For $k = 0$, the tangent vector, v_k, is calculated by:

$$\begin{pmatrix} \frac{\partial H}{\partial W}(W_k) \\ N^T \end{pmatrix} v_k = N. \tag{3.20}$$

For $k > 0$, the previous tangent vector is used, and the new tangent vector is calculated by:

$$\begin{pmatrix} \frac{\partial H}{\partial W}(W_k) \\ \frac{dW^T}{ds}(s_{k-1}) \end{pmatrix} v_k = N. \tag{3.21}$$

The unit tangent vector is then calculated by:

$$\frac{dW}{ds}(s_k) = \frac{\sigma_k v_k}{||v_k||_2} \tag{3.22}$$

where $\sigma_k = \pm 1$ in order to satisfy EQUATION 3.19.

The Euler predictor now approximates the next point along the homotopy path by:

$$W^0_{k+1} = W_k + \frac{dW}{ds}(s_k)\delta s_k. \tag{3.23}$$

While algorithms to strategically control the step size δs_k exist, these often require the calculation of the determinant of the Jacobian matrix [17], which may be computationally expensive. Such step size control is similar to the CFL algorithms of the PTC methods. This work uses a constant step size and leaves it as a tuning parameter, only requiring that $\delta s_k > 0$. For Burgers' equation, $\delta s_k = 0.1$ is used. For fluid flow equations, a larger $\delta s_k = 1$ is used for the sake of speeding up the computation time by reducing the number of predictor steps. The user may choose to decrease the step size for difficult problems in either case. The predictor steps are continued until $\lambda \geq 1$ is predicted, at which point, λ is set to a constant of 1 and the non-homotopy version of the problem is solved as in EQUATION 3.3.

Newton Corrector

Using W^0_{k+1} as an initial guess to the next point on the homotopy path, the Newton-Raphson method is used to hone in on the true path. Since the approximate point is only a small step away from the last known point on the homotopy path, it is expected that the approximate point is within the basin of attraction for sufficiently small δs_k, allowing the Newton solver to quickly converge. Since this work uses a constant δs_k, this step size might be too big on occasion, so a line search is still employed if necessary. Let W^n_{k+1} represent the point after n Newton corrector iterations. Newton's method is used to solve:

$$\begin{pmatrix} H(W^n_{k+1}) \\ \frac{dW^T}{ds}(s_k) \cdot \left(W^n_{k+1} - W^0_{k+1}\right) \end{pmatrix} = 0. \tag{3.24}$$

which is an augmented, homotopy-specific version of EQUATION 3.2. The bottom (augmented) equation in EQUATION 3.24 constrains the corrector path to be orthogonal to the tangent predictor path, with the expectation that this corrector path will be transversal to the homotopy path. It also allows the matrix in EQUATION 3.21 to be reused for the Jacobian computation. The Newton-Raphson solver now uses the augmented Jacobian [2],

$$\hat{A} = \begin{pmatrix} \left.\frac{\partial H}{\partial W}\right|_{W^n_{k+1}} \\ \frac{dW^T}{ds}(s_k) \end{pmatrix} \tag{3.25}$$

with

$$\frac{\partial H}{\partial W} = \begin{pmatrix} \frac{\partial H}{\partial Q} & \frac{\partial H}{\partial \lambda} \end{pmatrix} \tag{3.26}$$

$$\frac{\partial H}{\partial Q} = \lambda \frac{\partial F}{\partial Q}(Q) + (1-\lambda)\frac{\partial G}{\partial Q}(Q) \tag{3.27}$$

$$\frac{\partial H}{\partial \lambda} = F(Q) - G(Q) \tag{3.28}$$

and augmented residual vector,

$$\hat{R} = \begin{pmatrix} H(W_{k+1}^{n}) \\ 0 \end{pmatrix} \tag{3.29}$$

to solve

$$W_{k+1}^{n+1} - W_{k+1}^{n} \approx \Delta W_{k+1} \equiv -\hat{A}^{-1}\hat{R}(W_{k+1}^{n}). \tag{3.30}$$

The Newton corrector iterations are continued until $\max(\Delta W_{k+1})$ is less than some specified tolerance. For $\lambda = 0$ and $\lambda = 1$, the non-augmented version of the system is used to ensure that λ does not change throughout the newton iterations.

There is one caveat in that the bottom equation of Equation 3.25 does not exactly correspond to the bottom row of Equation 3.24, but rather to:

$$\frac{dW^T}{ds}(s_k) \cdot \left(W_{k+1}^{n+1} - W_{k+1}^{n}\right). \tag{3.31}$$

But since

$$W_{k+1}^{n} - W_{k+1}^{0} = \sum_{n} \Delta W_{k+1}^{n}, \tag{3.32}$$

then

$$\frac{dW^T}{ds}\left(W_{k+1}^{n} - W_{k+1}^{0}\right) = \sum_{n} \frac{dW^T}{ds} \Delta W_{k+1}^{n} = 0. \tag{3.33}$$

This is consistent with requiring an orthogonal corrector direction with an exact solve. However, this may not be the best choice when using an inexact linear solver, since an inexact update may be calculated and cause the direction of the corrector step to drift from the orthogonal direction.

Chapter 4

Burgers' Equation

This chapter demonstrates the performance of the various nonlinear solver techniques as applied to Burgers' equation.

The solution methods are first tested by solving Burger's equation because it is a simple nonlinear PDE with no physicality constraints. Since there are no physicality constraints on the solution, a broad range of initial conditions may be tested, and the solution process has more freedom when going through transients than when solving the Euler or Navier Stokes equations. Furthermore, Burgers' equation is scalar, meaning the computation expense is relatively small compared to the Euler equations, which facilitates extensive testing.

4.1 Viscous Burgers' Equation

4.1.1 Test Case

The equation for the viscous Burgers' equation is given by EQUATION 2.2. The specific case under consideration has $g(x) = \alpha \tanh(\frac{x}{2\mu})$ and $\alpha = -0.2$, such that the steady state solution, shown in FIGURE 4-1, is equal to

$$u = -\tanh\left(\frac{x}{2\mu}\right) \tag{4.1}$$

with $\mu = 0.005$ unless otherwise noted. The Dirichlet boundary conditions are set to be consistent with EQUATION 4.1 at the edges of the domain at $x = -1$ and $x = 1$.

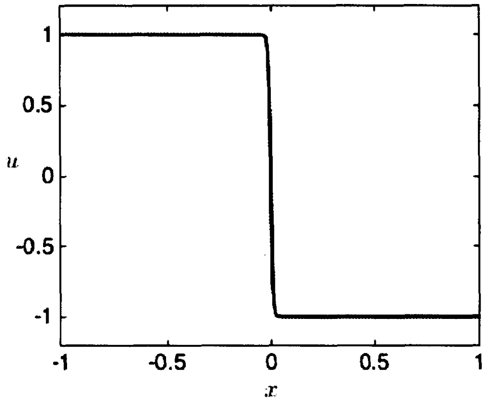
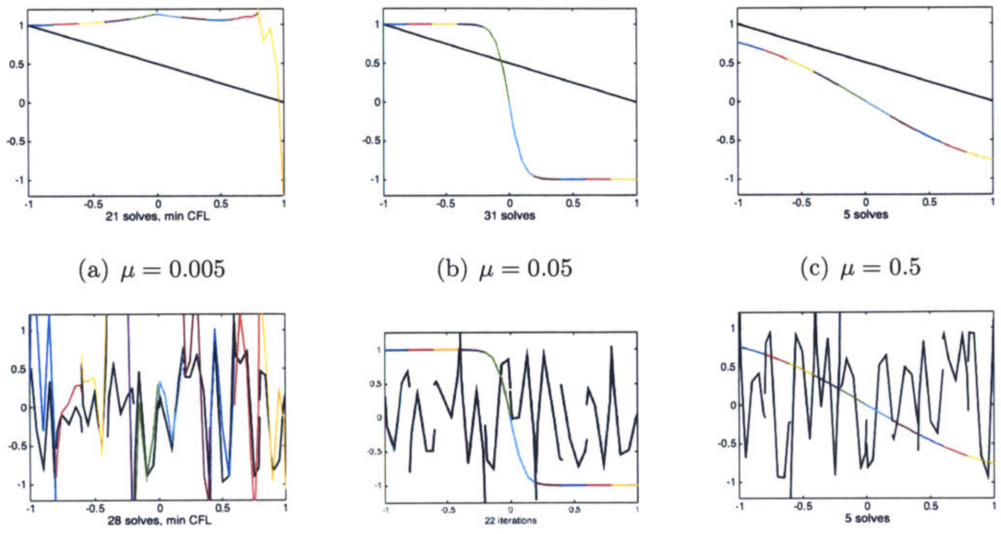


FIGURE 4-1: Exact solution for viscous Burgers' example

This manufactured solution is chosen such that the width of the shock can be scaled with viscosity. FIGURE 4-2 shows the effect of increasing viscosity in the governing equation, solving EQUATION 2.2 using pseudo-transient continuation, with one relatively simple initial condition (top row), and one randomized initial condition (bottom row), plotted in black. The X labels report number of linear solves performed, and failure mode if applicable. Of the three viscosity levels shown, $\mu = 0.005, 0.05, 0.5$, the least viscous cases with $\mu = 0.005$ hit the minimum CFL and do not converge, even for the more benign initial condition. FIGURE 4-2 demonstrates that solutions with more viscosity are more likely to converge, and in fewer iterations, even for poor initial conditions. This is the motivation behind diffusion-based continuation methods of DBC and homotopy.



(a) $\mu = 0.005$

(b) $\mu = 0.05$

(c) $\mu = 0.5$

(d) $\mu = 0.005$

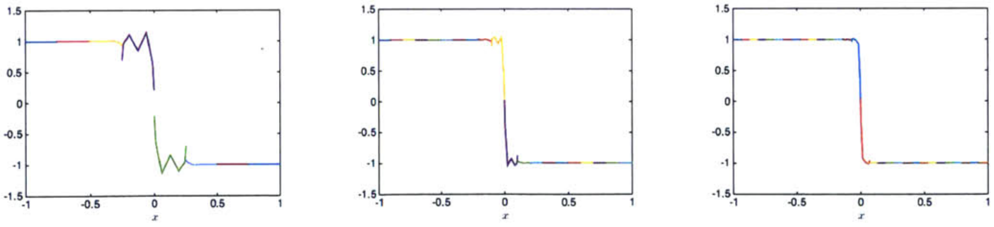
(e) $\mu = 0.05$

(f) $\mu = 0.5$

FIGURE 4-2: Effects of increasing viscosity. Black: initial condition; Multi: final output

4.1.2 Discrete Solution

On a coarse mesh, a thin shock (of finite width) might still appear as a discontinuity, and Gibbs oscillations are present. With enough DOF, the true shock can be resolved, and Gibbs oscillations go away. FIGURE 4-3 demonstrates this effect of increasing resolution on the presence of Gibbs oscillations around a finite width shock.



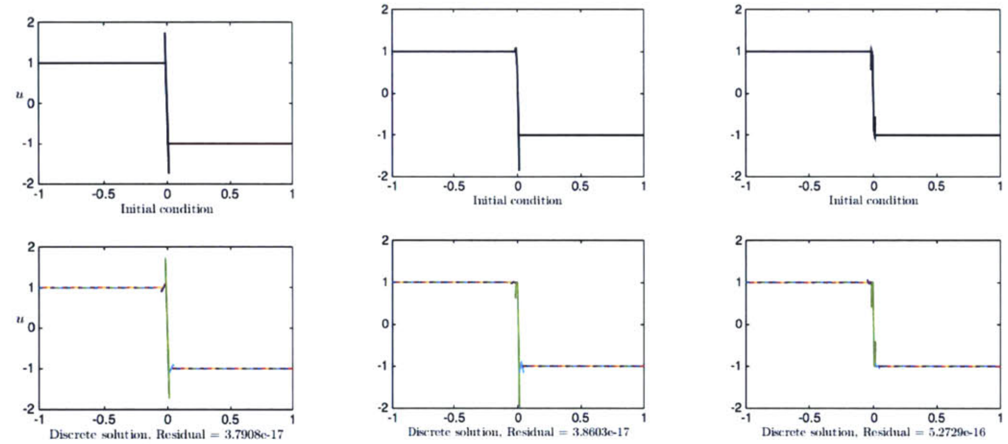
(a) P4 8 elements

(b) P4 20 elements

(c) P4 30 elements

FIGURE 4-3: Example of diminishing oscillations with refinement of finite width shock

Before attempting to solve this problem using various continuation methods, we first make sure that a discrete solution actually exists. This is checked by using the L^2 projection of the exact solution as the initial condition, and seeing that the residual is driven to machine zero (2.2204×10^{-16}). For grid of 65 elements, FIGURE 4-4 shows the discrete solutions for various polynomial orders. The X labels indicate the final residual.



(a) P1 65 elements

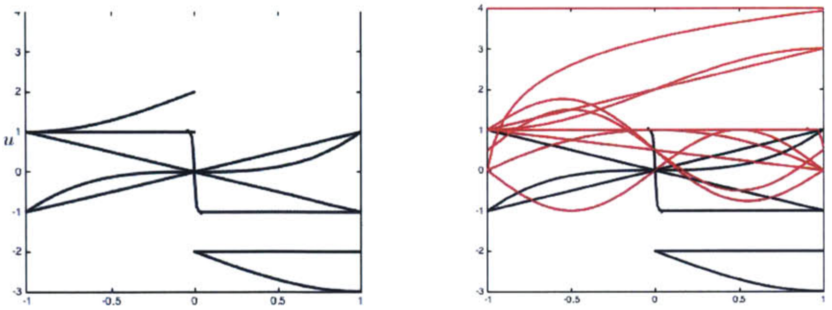
(b) P2 65 elements

(c) P3 65 elements

FIGURE 4-4: Discrete solutions for the viscous Burgers' case

4.1.3 Comparison of Solver Techniques

FIGURE 4-5 shows all of the initial conditions used to test the different solution techniques. The initial conditions that converged without the need for any continuation methods (labeled "Good") are plotted in black, while the remaining initial conditions that required some continuation method to converge are plotted in red.



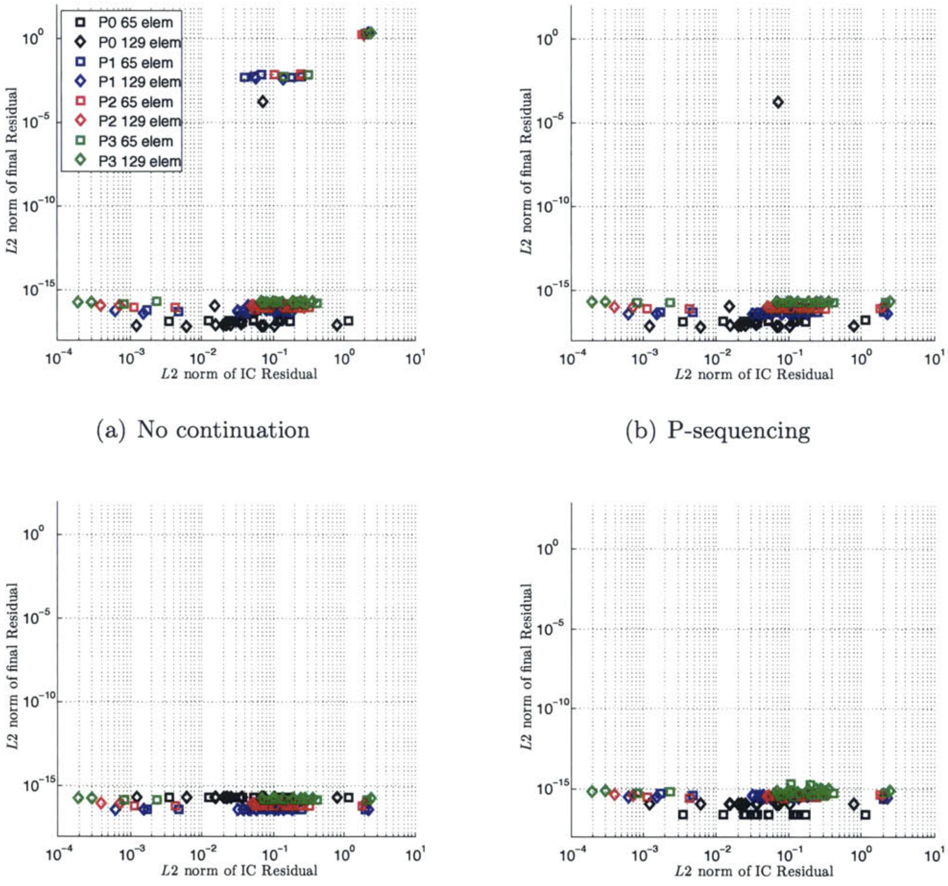
(a) "Good" ICs

(b) All ICs

FIGURE 4-5: Initial conditions used for testing with Burgers' equation

The problem is solved on 65- and 129-element uniform grids for various polynomial orders, using (a) no continuation method, (b) p-sequencing, (c) dissipation-based continuation, and (d) homotopy. FIGURE 4-6 shows the L^2 norm of the final residual plotted against the L^2 norm of the initial residual. The residuals for cases which "blew up"– i.e. whose residual increased beyond 10^4, are not plotted. This happens when the solver starts from a previous sub-problem that had not converged, and is more common to p-sequencing than the other methods. In each case, the number of linear solves is limited to 1000 per sub-problem, and the convergence tolerance is set to 10^{-14}.

We can clearly see in FIGURE 4-6(a) that there are several cases which do not converge. It is seen in FIGURE 4-6(b) that p-sequencing is a reliable method, and that while one P0 case does not converge within the maximum number of iterations, the subsequent solves with higher P do converge. FIGURE 4-6(c) and FIGURE 4-6(d) show consistent convergence even for P0 for DBC and homotopy.



(a) No continuation

(b) P-sequencing

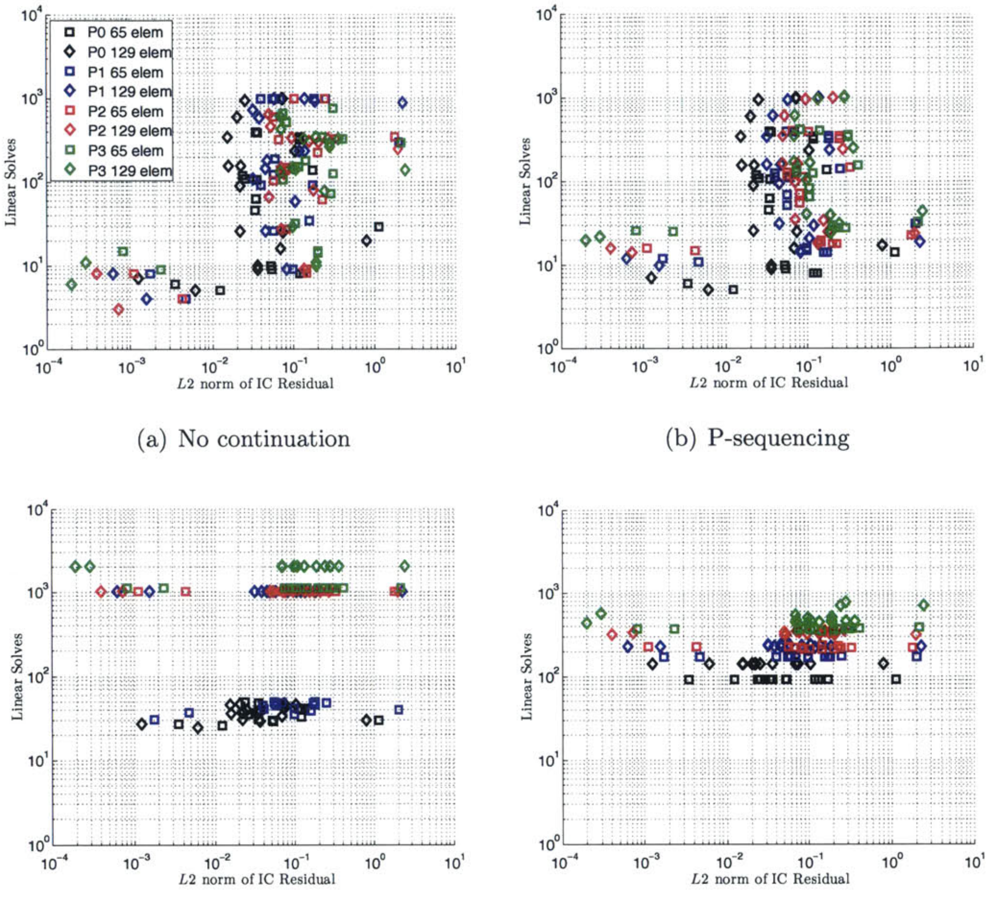
(c) DBC

(d) Homotopy

FIGURE 4-6: Final residuals using various solver techniques for viscous Burgers'

While FIGURE 4-6 indicates that p-sequencing is sufficient to converge high order solutions, we also want to consider the computational expense. The computational expense is determined by comparing the number of linear solves performed by each case, since the inversion of the Jacobian matrix is generally the most expensive part of the solution process. This comparison is shown in FIGURE 4-7. The number of solves include all sub-problems, for example, a given p-sequenced P2 solution totals the solves taken for P0, P1, and P2 for that case.

P-sequencing (b) shows slightly more linear solves than using no continuation method, as expected. For slightly more solves, nearly all the cases converge. DBC (c) requires fewer linear solves for the lower-DOF cases of P0 and most of P1 than the homotopy method and most IC cases with and without p-sequencing. The bands of high numbers of solves for DBC indicates that the maximum number of iterations were hit for one or more of the sub-problems. However, FIGURE 4-6 shows that all of these cases did converge. It is likely that relaxing the tolerance for the sub-problems would alleviate the number of solves required to converge using DBC. The homotopy method shows less of a spread in number of linear solves, likely due to the fact that aside from the initial condition, the same exact sequence of subproblems are solved since $G(Q_0)$ is linear for Burgers' equation and always results in the same solution for $\lambda = 0$. Increasing the predictor step size, δs, would likely reduce the required number of solves. Considering the number of linear solves taken to achieve the same level of convergence as p-sequencing, the homotopy method seems to be overkill. Even though DBC outperforms the other methods for the lower-DOF cases, the same issue remains with the method in that its continuation schedule is rather arbitrary. It is still unclear whether DBC would perform this well on other problems.



(a) No continuation

(b) P-sequencing

(c) DBC

(d) Homotopy

FIGURE 4-7: Number of linear solves for viscous Burgers'

4.2 Inviscid Burgers' Equation

The same solution methods are now investigated for an inviscid problem with a shock, where oscillations will be present regardless of the number of DOF.

4.2.1 Test Case

Following the test case in [3] the general governing equation is the same as EQUATION 2.2, with $\mu_0 = 0$, $\alpha = -0.1$ and forcing term $g(x)$ such that the inviscid exact solution has a shock at $x = 0$:

$$u(x) = \begin{cases} 2 + \sin\left(\frac{\pi x}{2}\right) & x < 0 \\ -2 - \sin\left(\frac{\pi x}{2}\right) & x > 0 \end{cases} \tag{4.2}$$

The Dirichlet boundary conditions are set to be consistent with EQUATION 4.2 at the edges of the domain at $x = -1$ and $x = 1$. FIGURE 4-8 shows the exact analytic solution.

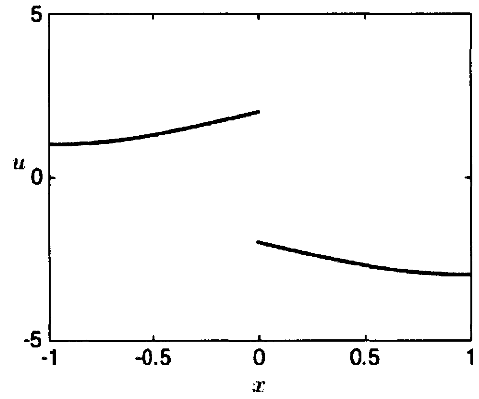
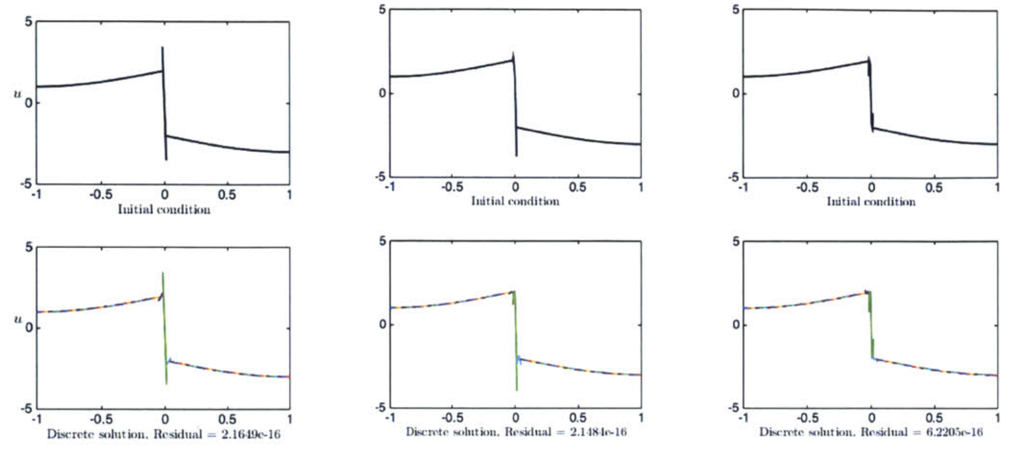


FIGURE 4-8: Exact solution for inviscid Burgers' example

4.2.2 Discrete Solution

We check that discrete solutions exist for this problem. For a grid of 65 elements, FIGURE 4-9 shows the discrete solutions for various polynomial orders. The X labels indicate the final residual.



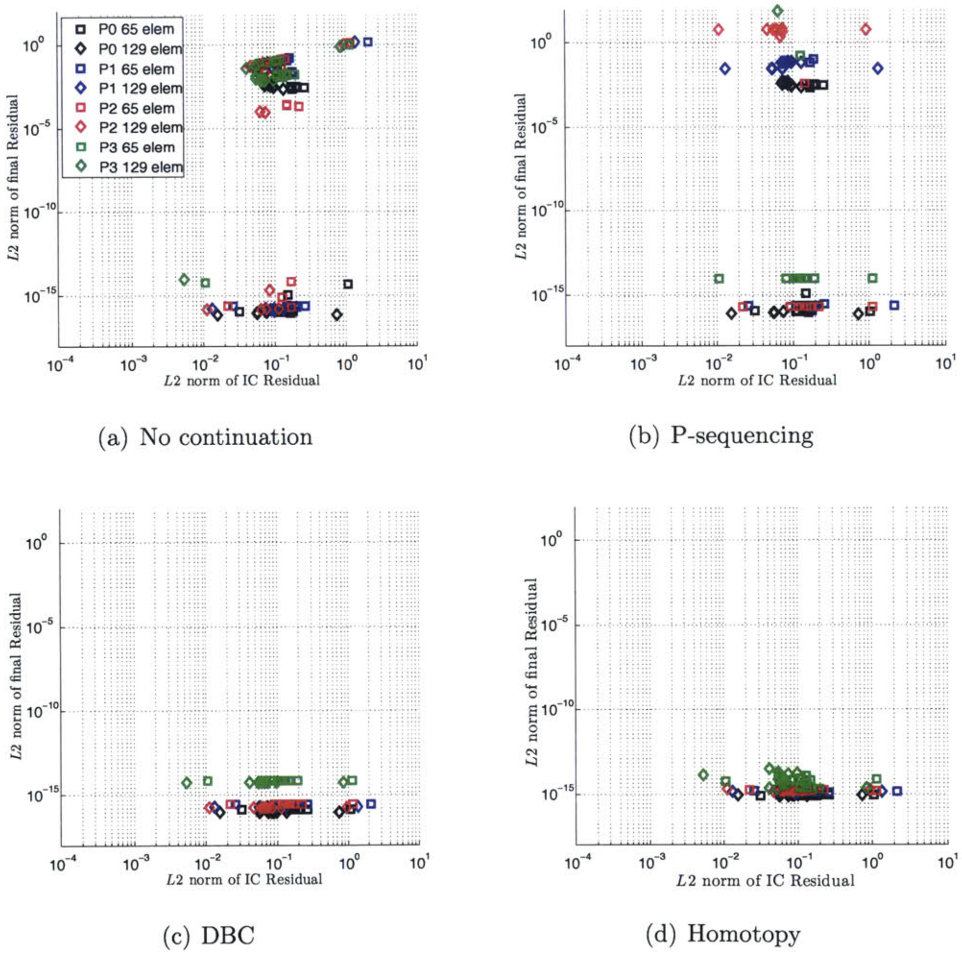
(a) P1 65 elements (b) P2 65 elements (c) P3 65 elements

FIGURE 4-9: Discrete solutions. Top row: initial condition, Bottom row: final solution

4.2.3 Comparison of Solver Techniques

The problem is solved on 65- and 129-element uniform grids for various polynomial orders, using (a) no continuation method, (b) p-sequencing, (c) dissipation-based continuation, and (d) homotopy. FIGURE 4-10 shows the L^2 norm of the final residual plotted against the L^2 norm of the initial residual. The residuals for cases which "blew up"– i.e. whose residual increased beyond 10^4, are not plotted. This happens when the solver starts from a previous solution that had not converged, and is more common to p-sequencing than to the other methods. In each case except for homotopy, the number of linear solves is limited to 1000 per sub-problem; for homotopy, the number of linear solves is limited to 100 per Newton corrector sub-problem, and number of predictor steps is limited to 300. The convergence tolerance is set to 10^{-14} for all cases.

FIGURE 4-10(a) shows several cases which do not converge, more so than with this viscous case. This is due to the Gibbs oscillations around the shock, a true discontinuity which no amount of DOF can resolve. FIGURE 4-10(b) shows that p-sequencing is unable to converge many more cases and is not much more reliable overall, sometimes performing worse than the cases with no continuation method. This is attributed to the Gibbs oscillations. It appears in FIGURE 4-10(c) that while all of the DBC cases converged, the P3 results lay very close to the set tolerance when compared with the other P cases. It is unclear whether those cases would converge closer to 10^{-15} like the rest of the results given a tighter tolerance. FIGURE 4-10(d) shows consistent convergence for all the homotopy cases in a similar residual range to DBC.



(a) No continuation

(b) P-sequencing

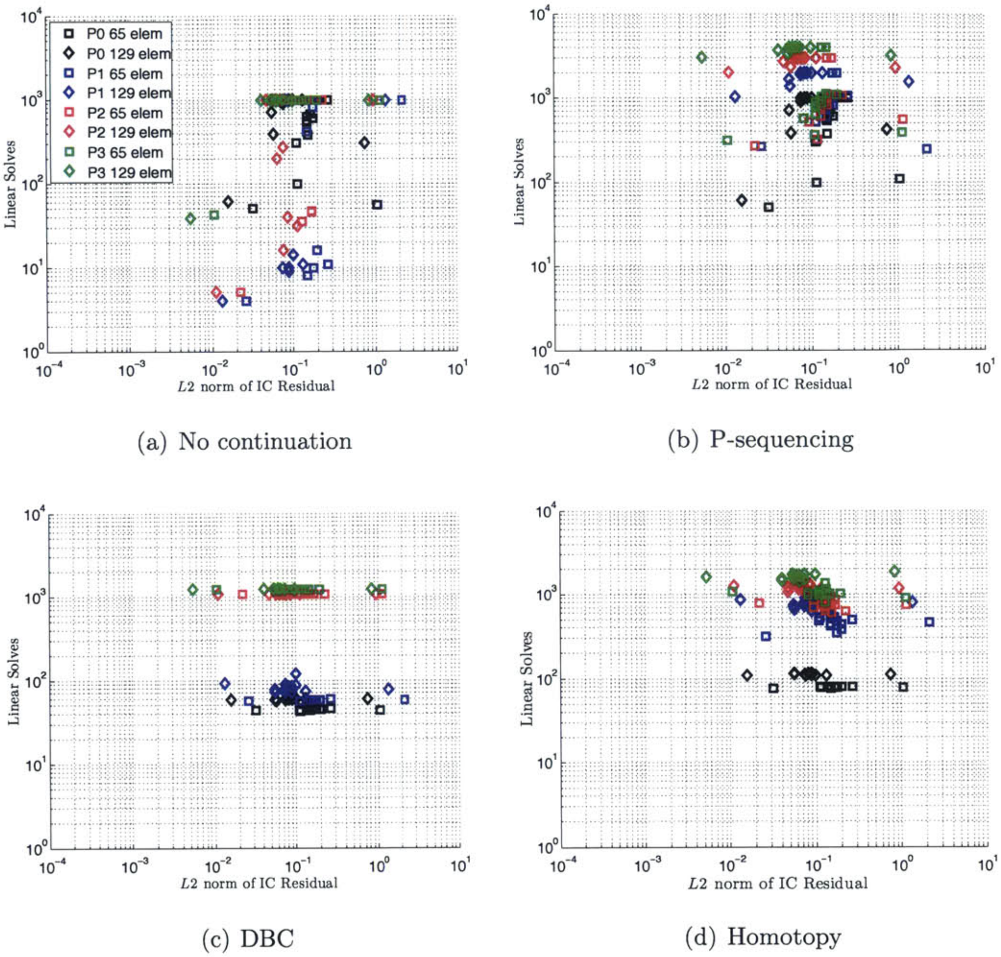
(c) DBC

(d) Homotopy

FIGURE 4-10: Final residuals using various solver techniques for inviscid Burgers'

We also consider the computational expense, which is compared in FIGURE 4-11. As expected, the p-sequencing method is more expensive than almost all of the cases without continuation. The DBC and homotopy performances are similar to those for the viscous case. DBC performs well for P0 and P1, but again seems to hit maximum iteration limits for higher P. However, all of these cases still converged, meaning that the brute force method of decreasing viscosity even on an unconverged sub-problem can still work. It is likely that those unconverged sub-problems were close to the tolerance such that the unconverged intermediate solutions were within the basin of attraction for the subsequent sub-problems. FIGURE 4-11(d) shows that homotopy performs similarly to DBC for P0, is more expensive than DBC for P1, but outperforms DBC for most P2 and several P3 problems.

For all four methods overall, DBC performed best for P0 and P2, while homotopy performed best for most higher P cases for the inviscid test case. This is in contrast to the viscous test case, where p-sequencing showed a wide range of linear solves, making it unclear exactly when one method would perform better over another, since almost all of the continuation methods converged robustly.



(a) No continuation

(b) P-sequencing

(c) DBC

(d) Homotopy

FIGURE 4-11: Number of linear solves for inviscid Burgers'

Chapter 5

Euler Equations

For Burgers' equation, any value of the state, u, is allowed; however, the same cannot be said for the Euler and Navier Stokes equations. For example, a negative pressure would cause the calculation of the speed of sound, $a = \sqrt{\gamma p/\rho}$, to fail. For the Euler equations using the conservative vartiables, Einfeldt et. al. [23] defined the set of physically admissible states, G (not to be confused with $G(Q)$ in CHAPTER 3), as those which contain positive density and internal energy:

$$G = \left\{ \; U \mid \rho > 0 \text{ and } 2\rho\left(\rho E\right) - \left(\rho u\right)^2 \; \right\} > 0. \tag{5.1}$$

Linde and Roe [37] defined the same set of admissible states as EQUATION 5.1, but referred to requiring positive pressure rather than internal energy. Indeed, requiring positive pressure is equivalent to requiring positive internal energy. The second constraint is rewritten as

$$\rho\left(E - u^2/2\right) > 0. \tag{5.2}$$

Since $(\gamma - 1) = 0.4 > 0$ for air, EQUATION 5.2 also corresponds to positive pressure when written in terms of the conservative variables:

$$p = (\gamma - 1)\,\rho\left(E - u^2/2\right) > 0. \tag{5.3}$$

Substituting $e = E - u^2/2$ and $c_v T = e$ for an ideal gas,

$$p = (\gamma - 1)\, \rho c_v T = \rho R T > 0. \tag{5.4}$$

We see that requiring density and temperature to be positive also corresponds to requiring positive pressure and energy, as well as speed of sound, written above. Linde and Roe also note that limiting the primitive variables has yielded good results in many other numerical experiments [37].

These physicality constraints cause robustness issues for fluid simulations that might pass through nonphysical states during the transient solves. A nonphysical state might be computed during a Newton iteration; even if a subsequent line search would reduce the update size and prevent the Newton update from making the state go nonphysical (nonphysical or close-to-nonphysical states are often associated with large residual values), some codes will immediately halt once detecting the nonphysical state and end the solve. A nonphysical state might also be computed around a shock where Gibbs oscillations are present due to the discontinuity. Gibbs oscillations can also appear on smooth features that are under-resolved, such as shocks of finite thickness (i.e. with viscosity) on coarse meshes.

Since this is a relatively ubiquitous problem in CFD, there are many approaches to avoiding nonphysical states. As described in Section 1.2, some of these approaches take the form of shock capturing or positivity preserving methods, including line searches, (constrained) pseudo-unsteady algorithms, and artificial viscosity. This chapter proposes an alternate strategy wherein density and temperature surrogate solution variables are allowed to go negative, while still preserving positivity of the actual density and temperature.

5.1 Surrogate Variables

When solving the Euler and Navier Stokes equations, we would like to emulate the property of Burgers' equation of having no nonphysical state. Specifically, we would like to allow density and temperature to go negative during the transient solves and/or on coarse meshes, knowing that the steady state and/or grid-refined solution (possible with local artificial viscosity as in [41] or [3]) will indeed be physical.

We consider the use of surrogate primitive variables, $\tilde{Z} = \left[\tilde{\rho}, \tilde{u}, \tilde{T}\right]^T$, which are allowed take on any real value. The conversion back to the actual primitive variables, $Z = [\rho, u, T]^T$, is chosen to ensure that $\rho(\tilde{\rho})$ and $T(\tilde{T})$ are always positive.
The specific surrogate primitive variables we use are:

$$\rho(\tilde{\rho}) = \begin{cases} \tilde{\rho} & \tilde{\rho} \geq \tilde{\rho}_c \\ \dfrac{\tilde{\rho}_c}{3 - 3\tilde{\rho}/\tilde{\rho}_c + (\tilde{\rho}/\tilde{\rho}_c)^2} & \tilde{\rho} < \tilde{\rho}_c \end{cases} \tag{5.5a}$$

$$u(\tilde{u}) = u \tag{5.5b}$$

$$T(\tilde{T}) = \begin{cases} \tilde{T} & \tilde{T} \geq \tilde{T}_c \\ \dfrac{\tilde{T}_c}{3 - 3\tilde{T}/\tilde{T}_c + \left(\tilde{T}/\tilde{T}_c\right)^2} & \tilde{T} < \tilde{T}_c \end{cases} \tag{5.5c}$$

A plot of this surrogate model is shown in FIGURE 5-1. The values and slopes of the surrogates match at the critical values, $\tilde{\rho}_c$ and $\tilde{T}_c$. The choices of these critical values are tuning parameters; for the test case presented, $\tilde{\rho}_c = 0.01$ and $\tilde{T}_c = 0.001$ were found to work well and are used for all results shown. $\tilde{T}_c$ is smaller than $\tilde{\rho}_c$ because $\tilde{T}$ tended to take on much larger-magnitude negative values than $\tilde{\rho}$, and decreasing the critical value helped to lessen its drastic negative spikes during the transient solves.

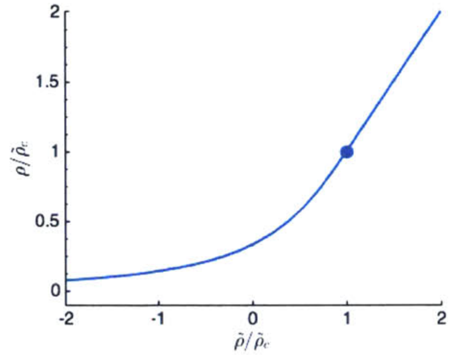


FIGURE 5-1: Surrogate density and temperature functions

The code which was developed for this work originally used conservative states as the working variables, $Q = U$. Thus, the linearization $\partial R/\partial U$ was already available. In using surrogates, $Q = \tilde{Z}$ and now $\partial R/\partial \tilde{Z}$ is needed. We construct $\partial R/\partial \tilde{Z}$ via the chain rule which then only requires small changes in the existing $\partial R/\partial U$ code (to ensure $U\left(Z(\tilde{Z})\right)$ is correctly used). Thus,

$$\frac{\partial R}{\partial \tilde{Z}} = \frac{\partial R}{\partial U}\frac{\partial U}{\partial Z}\frac{\partial Z}{\partial \tilde{Z}} \tag{5.6}$$

where

$$\frac{\partial U}{\partial Z} = \begin{pmatrix} 1 & 0 & 0 \\ u & \rho & 0 \\ E & \rho u & \rho c_v \end{pmatrix} \tag{5.7}$$

and

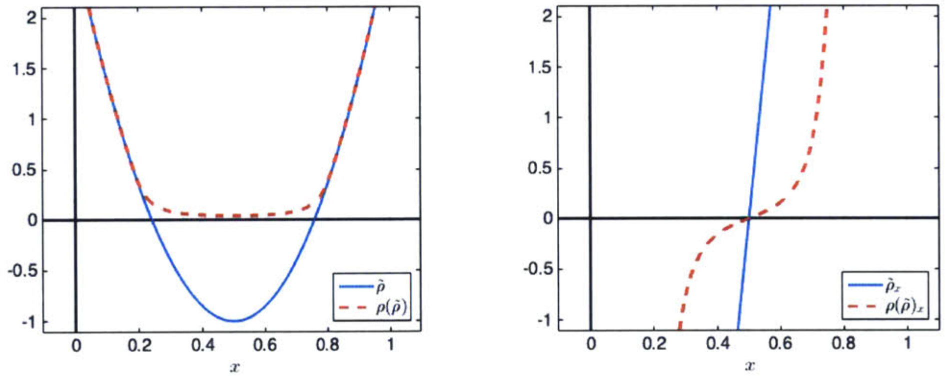
$$\frac{\partial Z}{\partial \tilde{Z}} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \beta \end{pmatrix} \tag{5.8}$$

with

$$\alpha = \frac{\partial \rho}{\partial \tilde{\rho}} \quad , \quad \beta = \frac{\partial T}{\partial \tilde{T}}. \tag{5.9}$$

5.2 Artificial Viscosity and Linearization

A difficulty that arises when combining surrogate variables and artificial viscosity is that the physical variables will tend to have small variations in regions where the surrogates are active (e.g. $\rho < \tilde{\rho}_c$). Then, spatial gradients such as U_x or Z_x will be small. This particularly problematic because regions where the surrogates are active are likely to require artificial dissipation to smooth the solution. Instead, we use the gradient of the surrogate primitive variables, $\tilde{Z}_x$ (or "surrogate" conservative variables, $\tilde{U}_x = U(\tilde{Z})_x$). For a threshold value of 0.5, FIGURE 5-2 illustrates the effect of the surrogates flattening the gradient of density, while the gradient of surrogate density remains large in magnitude.



(a) Density

(b) Density gradient

FIGURE 5-2: Effect of surrogate reducing physical gradient

The physical viscosity uses the defined μ_0, which is constant throughout the domain. The linearization of the physical viscous flux, $F^v(Z, Z_x)$, with respect to $\tilde{Z}$ is

$$\begin{aligned}\frac{dF^v}{d\tilde{Z}} &= \frac{\partial F^v}{\partial \tilde{Z}} + \frac{\partial F^v}{\partial Z_x}\frac{\partial Z_x}{\partial \tilde{Z}} \\ &= \frac{\partial F^v}{\partial Z}\frac{\partial Z}{\partial \tilde{Z}} + \frac{\partial F^v}{\partial Z_x}\frac{\partial}{\partial \tilde{Z}}\left(\frac{\partial Z}{\partial \tilde{Z}}\frac{\partial \tilde{Z}}{\partial x}\right) \\ &= \frac{\partial F^v}{\partial Z}\frac{\partial Z}{\partial \tilde{Z}} + \frac{\partial F^v}{\partial Z_x}\left(\frac{\partial^2 Z}{\partial \tilde{Z}^2}\frac{\partial \tilde{Z}}{\partial x} + \frac{\partial Z}{\partial \tilde{Z}}\frac{\partial}{\partial \tilde{Z}}\left(\frac{\partial \tilde{Z}}{\partial x}\right)\right)\end{aligned} \tag{5.10}$$

where

$$\frac{\partial F^v}{\partial Z} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{4}{3}\mu\frac{\partial u}{\partial x} & 0 \end{pmatrix} \tag{5.11}$$

and

$$\frac{\partial F^v}{\partial Z_x} = K^v = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{4}{3}\mu & 0 \\ 0 & \frac{4}{3}\mu u & k \end{pmatrix}. \tag{5.12}$$

The presence of the second derivative with respect to $\tilde{Z}$ in EQUATION 5.11 suggests that the mapping from surrogate to primitive variables must be $C2$ smooth. When making the artificial viscosity a function of the gradient of the surrogate variables instead of the gradient of the physical variables, the linearization of the surrogate physical viscous flux, $F^v(Z, \tilde{Z}_x)$, simplifies to:

$$\begin{aligned}\frac{dF^v}{d\tilde{Z}} &= \frac{\partial F^v}{\partial \tilde{Z}} + \frac{\partial F^v}{\partial \tilde{Z}_x}\frac{\partial \tilde{Z}_x}{\partial \tilde{Z}} \\ &= \frac{\partial F^v}{\partial \tilde{Z}} + \frac{\partial F^v}{\partial \tilde{Z}_x}\frac{\partial}{\partial \tilde{Z}}\left(\frac{\partial \tilde{Z}}{\partial x}\right)\end{aligned} \tag{5.13}$$

with $\frac{\partial F^v}{\partial \tilde{Z}} = \frac{\partial F^v}{\partial Z}$ and $\frac{\partial F^v}{\partial \tilde{Z}_x} = \frac{\partial F^v}{\partial Z_x}$, since they are only functions of velocity and the velocity gradient, and the surrogate variables use $\tilde{u} = u$.

5.3 Euler with Shock

5.3.1 Test Case

The inviscid test case solves the Euler equations, with artificial dissipation for DBC and homotopy. The governing equations are:

$$\frac{\partial}{\partial x}\begin{pmatrix} \rho u A \\ (\rho u^2 + p)A \\ \rho u H A \end{pmatrix} - \begin{pmatrix} 0 \\ p\dfrac{\partial A}{\partial x} \\ 0 \end{pmatrix} - \frac{\partial}{\partial x}\left(F^{v*}\right) = 0 \quad , \quad \forall x \in [0,1]\,. \tag{5.14}$$

with F^{v*} is the artificial viscosity flux, either F^v or F^l. For a diverging nozzle with area varying according to:

$$A(x) = 1 + 9\frac{\ln(1+x)}{\ln(2)} \tag{5.15}$$

shown in FIGURE 5-3, the boundary conditions are set such that the inviscid exact solution has a shock in the middle of the domain at $x = 0.5$, over which the temperature increases by a factor of 10. At $x = 0$, the supersonic inflow Dirichlet boundary condition specifies the state with $M_\infty = 4.37$. At $x = 1$, the subsonic outflow boundary condition specifies the static back-pressure with $p_{back}/p_\infty = 4.07$. FIGURE 5-4 shows the exact analytic solution. For in this exact solution, $Z = \tilde{Z}$ because the exact solution values for $\tilde{\rho}$ and $\tilde{T}$ are above their respective critical values.

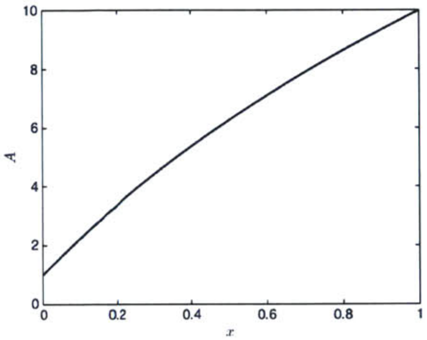
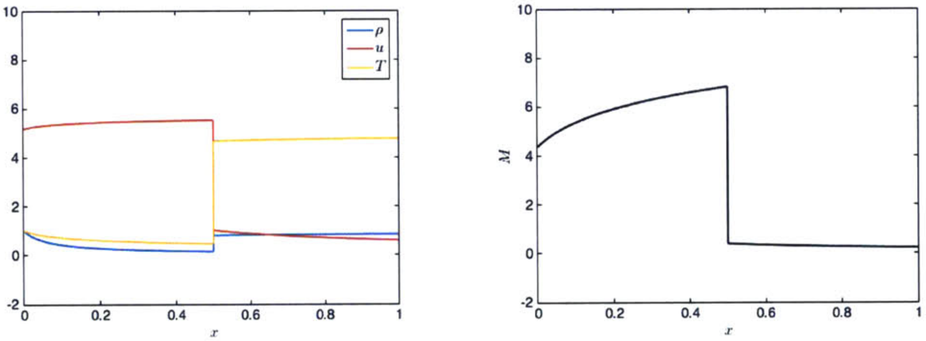


FIGURE 5-3: Area geometry for Euler shock



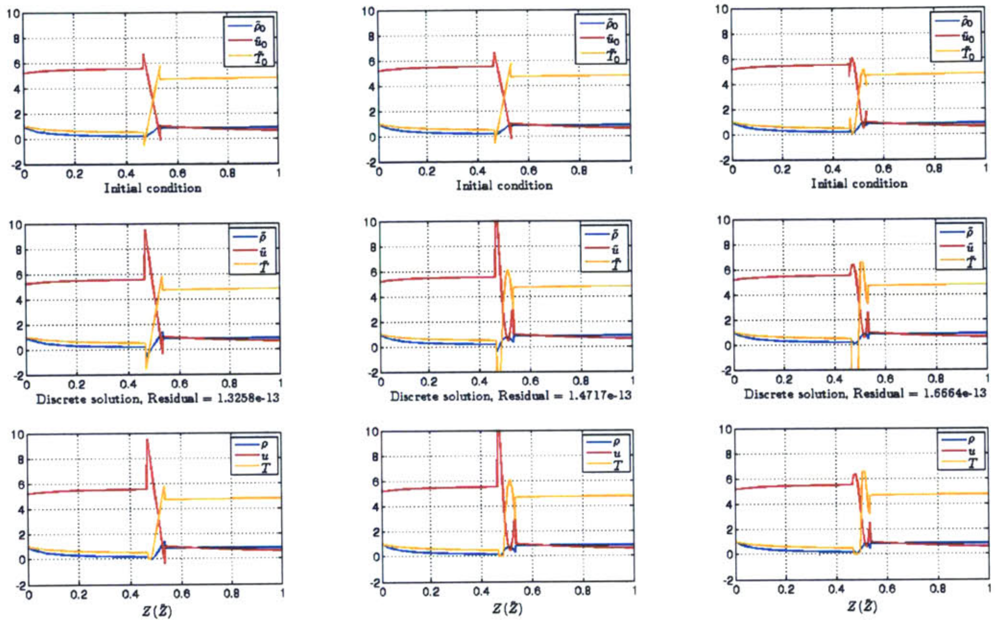
(a) Primitive variables for exact solution

(b) Mach number for exact solution

FIGURE 5-4: Exact solution for Euler shock

5.3.2 Discrete Solution

As done for the Burgers' examples, we check for the existence of discrete solutions. For a grid of 15 elements, FIGURE 5-5 shows the discrete solutions for various polynomial orders, using the L2 projection of the exact solution as the initial condition, and a convergence tolerance of 2×10^{-13}. The X labels indicate the final residual. For the P2 solution, the oscillations of $\tilde{\rho}$ peak off the plot at $\tilde{\rho}_{max} = 13$, while the oscillations of $\tilde{T}$ peak off the plot at $\tilde{T}_{min} = -12$. For the P3 solution, the oscillations of $\tilde{T}$ peak off the plot at $\tilde{T}_{min} = -86$.



(a) P1 15 elements

(b) P2 15 elements

(c) P3 15 elements

FIGURE 5-5: Discrete solutions for Euler shock case

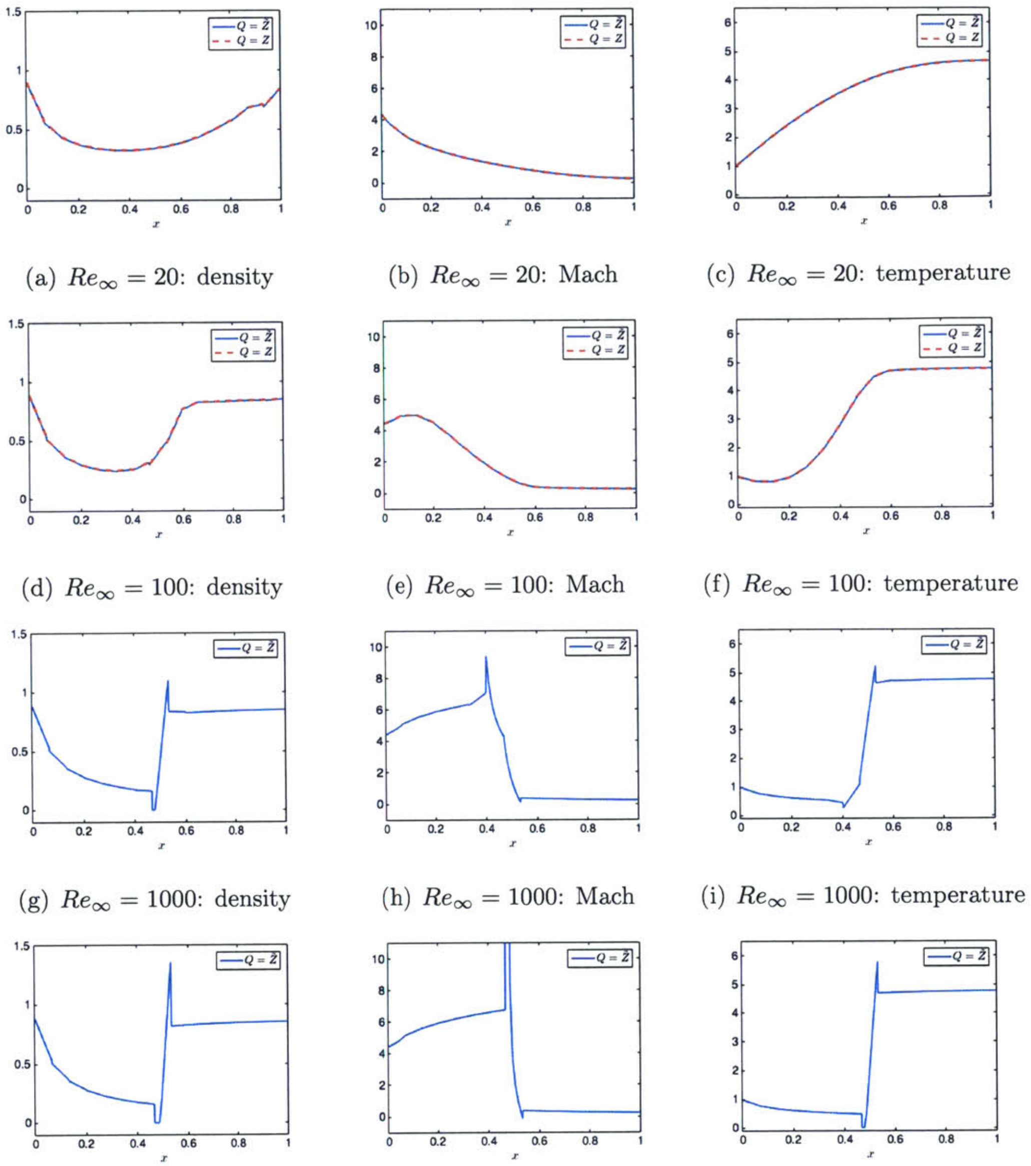
5.3.3 Demonstration of Surrogate Variables

Converging to inviscid solution with primitive and surrogate variables

The problem is solved for a 15-element uniform grid with a uniform initial condition based on the outflow state of the exact solution. While it is common to start with an initial condition based on the inflow state, it was found that supersonic initial conditions were not likely to converge for this test problem, since the flux upwinding largely prevents the subsonic outflow condition from allowing a shock to travel inward. The physical viscosity model is used as the artificial viscosity for all subsequent results shown.

Because FIGURE 5-5 shows that there are oscillations which go negative in the discrete solution, it is expected that the primitive variables alone will not be able to converge to a fully inviscid solution on this grid. To see how far the primitive variables could be used before the surrogate variables become necessary, the dissipation-based continuation method, starting with a viscosity such that the freestream Reynolds number equals 20 ($Re_\infty = \rho_\infty u_\infty L_\infty / \mu_\infty$) is used to approach the solution from a smooth, physical state. FIGURE 5-6 and FIGURE 5-7 show the solutions for density, Mach number, and temperature, for $Re_\infty = 20, 100, 1000$, and finally the $Re_\infty = \infty$ inviscid case, for P1 and P2. In these figures, $Q = Z$ and $Q = \tilde{Z}$ what the solution variables were.

The primitive variables, even with DBC combined with PTC, are unable to converge with $Re_\infty = 1000$ and $Re_\infty = \infty$ for P1. The $Re = \infty$ case is also unable to converge using primitive variables for P2. The line search is unable to find an update fraction that does not result in a nonphysical state for all of the unconverged cases. Using the same solution method (DBC combined with PTC), the primitive surrogate variables are able to converge each case shown in FIGURE 5-6 and FIGURE 5-7, including the fully inviscid solutions.



(a) $Re_\infty = 20$: density

(b) $Re_\infty = 20$: Mach

(c) $Re_\infty = 20$: temperature

(d) $Re_\infty = 100$: density

(e) $Re_\infty = 100$: Mach

(f) $Re_\infty = 100$: temperature

(g) $Re_\infty = 1000$: density

(h) $Re_\infty = 1000$: Mach

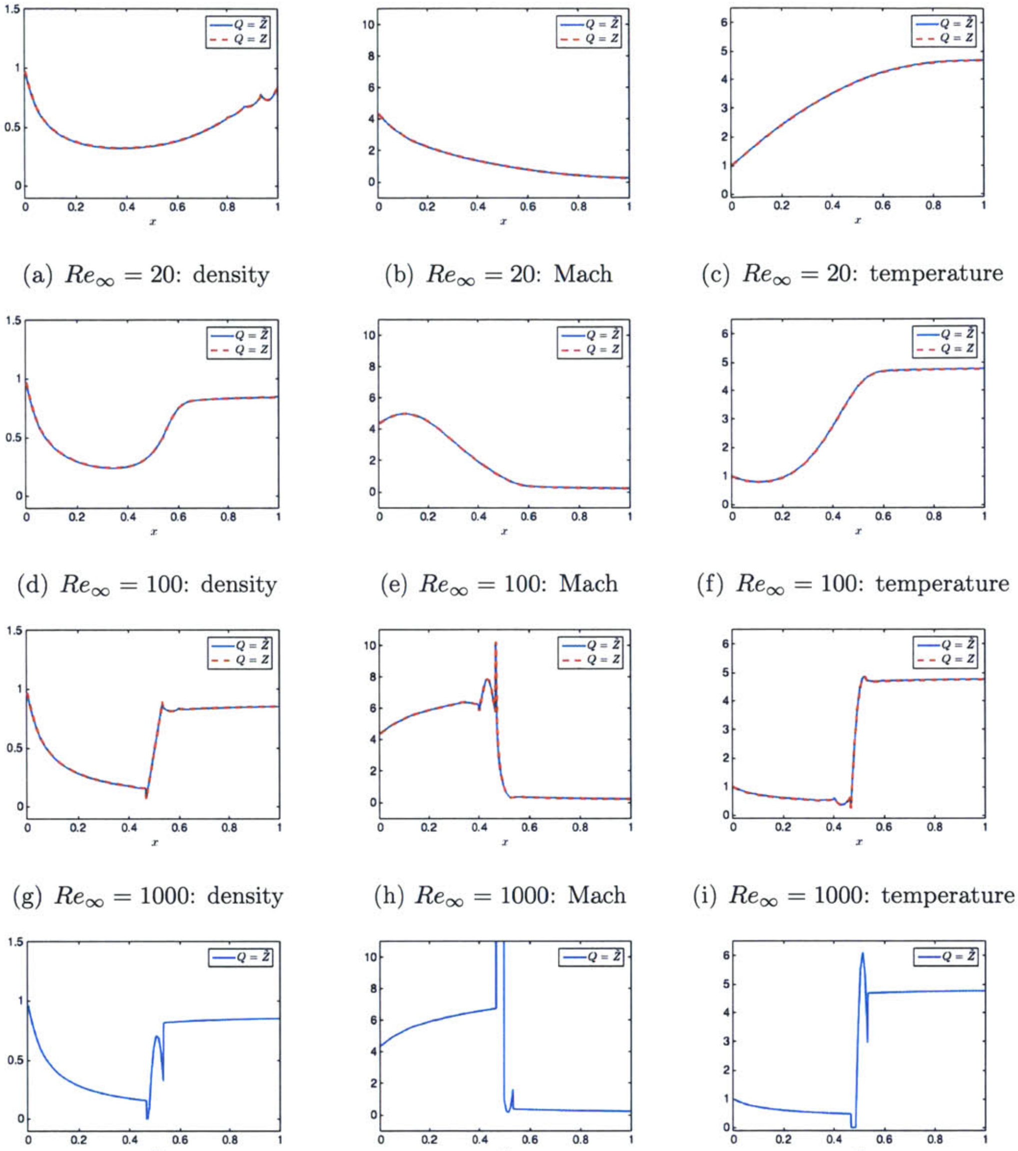
(i) $Re_\infty = 1000$: temperature

(j) Inviscid: density

(k) Inviscid: Mach number

(l) Inviscid: temperature

FIGURE 5-6: Surrogate vs primitive variables: P1



(a) $Re_\infty = 20$: density (b) $Re_\infty = 20$: Mach (c) $Re_\infty = 20$: temperature

(d) $Re_\infty = 100$: density (e) $Re_\infty = 100$: Mach (f) $Re_\infty = 100$: temperature

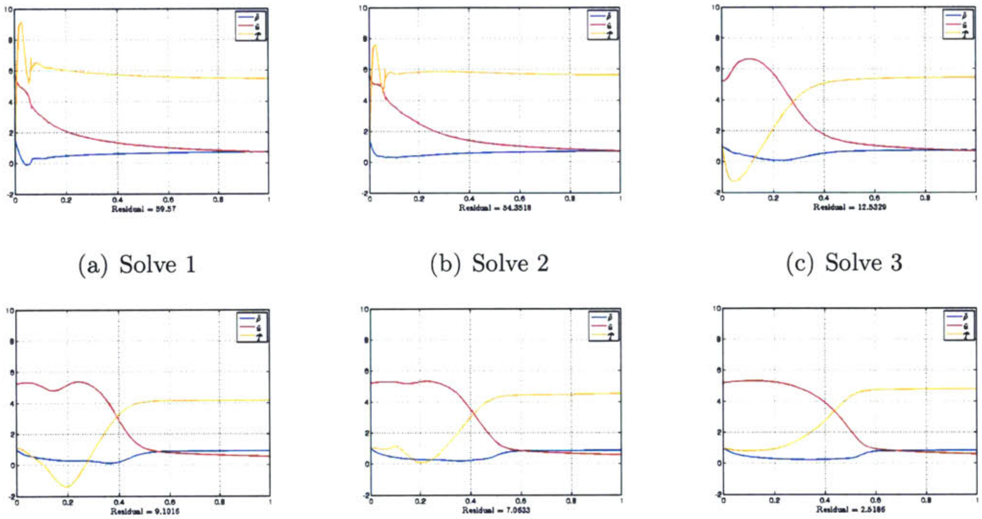
(g) $Re_\infty = 1000$: density (h) $Re_\infty = 1000$: Mach (i) $Re_\infty = 1000$: temperature

(j) Inviscid: density (k) Inviscid: Mach number (l) Inviscid: temperature

FIGURE 5-7: Surrogate vs primitive variables: P2

Negative transient surrogate values

It is shown that the surrogate variables are needed to converge on the solution with nonphysical oscillations. In doing so, it was observed that transient solutions using $Q = \tilde{Z}$ sometimes passed through states with negative surrogate densities and temperatures, even when the converged solution was completely positive and free of oscillations. FIGURE 5-8 shows an example of such a progression of transient states for P3 on a 15-element grid. The initial condition is again the uniform subsonic-outlet IC, with $Re_\infty = 100$. The system is not yet converged after the sixth linear solve shown, but the subsequent solves remain positive in density and temperature, for a total of 10 linear solves.



(a) Solve 1

(b) Solve 2

(c) Solve 3

(d) Solve 4

(e) Solve 5

(f) Solve 6

FIGURE 5-8: Surrogate variables passing through negative temperatures

To investigate how well either choice of variables converges without artificial viscosity continuation, we now compare the use of surrogate primitive variables to the use of regular primitive variables, supported with a physicality line search. Having been convinced that the primitive variables are unable to converge $Re_\infty = \infty$, the $Re_\infty = 20$, $Re_\infty = 100$ and $Re_\infty = 1000$ are compared. $Re_\infty = 1000$ is the "borderline" case where the primitive variables, using DBC, converge for P2 but not for P1. Now we see what happens without DBC. TABLE 5.1 reports the number of linear solves taken to converge with a residual tolerance of 10^{-11}, with X indicating that the solve failed. "N" indicates the use of the Newton solver with no additional continuation method; "PTC" indicates that pseudo-transient continuation was also used.

		$Re_\infty = 20$		$Re_\infty = 100$		$Re_\infty = 1000$	
		N	PTC	N	PTC	N	PTC
P1	Primitive	7	14	10	16	X	X
	Surrogate	7	14	11	14	X	33
P2	Primitive	7	15	9	16	X	X
	Surrogate	10	14	11	14	X	91
P3	Primitive	7	15	9	16	X	53
	Surrogate	10	14	10	14	69	36

TABLE 5.1: Number of linear solves for P1–3, 15 elements

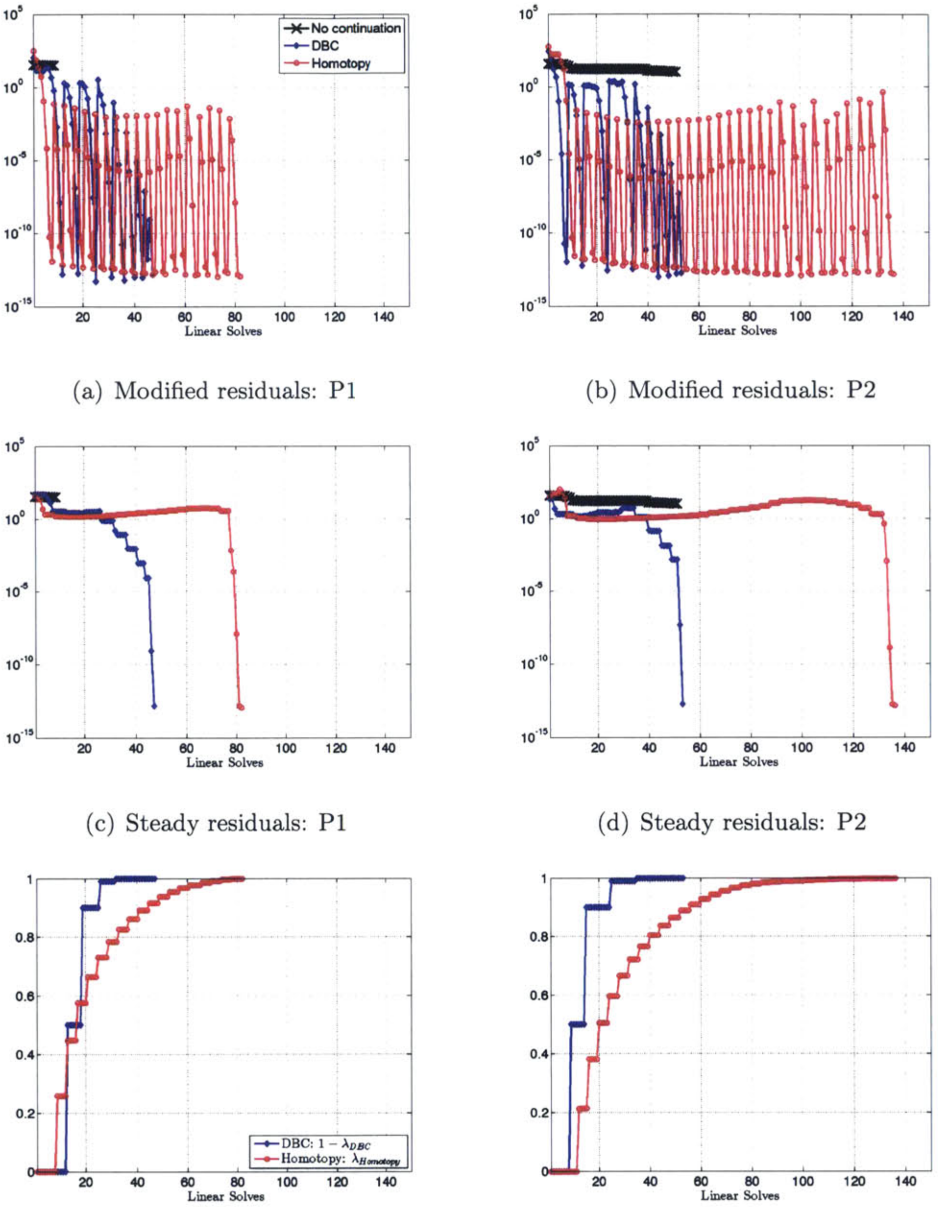
Each of the cases converges for $Re_\infty = 20$ and $Re_\infty = 100$, with or without PTC, for both the primitive and the surrogate variables. We can conclude that allowing the transient solutions to pass through negative surrogate values of density and temperature, such as those depicted by FIGURE 5-8, has little effect on the overall solver performance for these smoother cases. However, the higher Reynolds number cases using the primitive variables are often observed to fail before the shock was set up at in the center of the domain; where these cases fail is where the surrogate variables go negative in the transient.

For $Re_\infty = 1000$, 5 of the 12 cases converge; the primitive variables account for only 1 of those 5 cases, while the surrogate variables account for the remaining 4. Without any continuation method, none of the primitive variable cases converge; the only case to converge is the P3 surrogate case. When applying PTC, the only case to converge with primitive variables is P3, while all of the surrogate variable cases do converge. Overall, surrogate variables combined with PTC yielded the best results for the borderline $Re_\infty = 1000$ case. These results demonstrate three things. First, the relative success of P3 over P1 and P2 demonstrates the benefit of using increased resolution in general. Second, PTC is shown to successfully aid in convergence of difficult problems. Third, the use of surrogate variables is shown to aid in the convergence of the problems in which PTC alone is not enough.

Continuation methods with surrogate variables

We have shown that the discrete solution to this inviscid shock problem exist, and that they require the surrogate variables to converge. We have also shown that the surrogate variables are beneficial to convergence from the uniform initial condition, although convergence was less likely for lower-order cases without pseudo-transient continuation. We now apply the diffusion-based continuation methods to the surrogate variables for the fully inviscid case.

The problem is solved with and without the artificial viscosity continuation methods on a 15-element uniform grid, with the uniform initial condition based on the outflow of the exact solution. The residual and continuation parameter histories are shown in FIGURE 5-9. "Modified residual" refers to the residuals that include artificial viscosity, i.e., the ones which Newton's method is driving to zero on each sub-problem: EQUATION 3.12 for DBC and EQUATION 3.29 for homotopy. "Steady residual" refers to the unmodified residual without any artificial dissipation. The continuation parameter for DBC is plotted as $(1-\lambda_{DBC})$ for better comparison to the homotopy continuation parameter. The sawtooth patterns of the modified residuals show the convergence of each sub-problem, with the increasing jumps indicating the start of the subsequent sub-problem.



(a) Modified residuals: P1

(b) Modified residuals: P2

(c) Steady residuals: P1

(d) Steady residuals: P2

(e) Continuation parameters: P1

(f) Continuation parameters: P2

FIGURE 5-9: Residual histories for Euler shock case using primitive variables

FIGURE 5-9 shows that none of the cases are able to converge without the use of some continuation method. While the DBC method requires more linear solves to converge each sub-problem, it converges faster overall than the homotopy method, which slows down as it approaches the fully inviscid solution with $\lambda = 1$. FIGURE 5-10 shows the Mach number distribution at converged sub-problems of the homotopy method over the course of the P2 solve. The legend indicates the value of the continuation parameter and the running total of linear solves taken. This slowing down near $\lambda = 1$ could be due to the fixed predictor step size, δs; step size algorithms for controlling δs should be explored to improve this continuation method. The DBC method in contrast is able to skip ahead to a more inviscid problem, although the first three sub-problems for P2 indicate that DBC could have performed better with less drastic changes the continuation parameter.

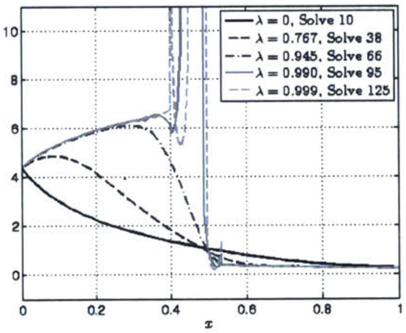


FIGURE 5-10: Evolution of Mach number distribution using homotopy

The DBC method takes 54 linear solves to converge the P2 $Re_\infty = \infty$ case– we note that this is less than the number of linear solves taken by the PTC method to converge the P2 $Re_\infty = 1000$ case–91 solves. While these cases are at two different Reynolds numbers and cannot be directly compared, it is postulated that the DBC method would converge the $Re_\infty = 1000$ case in fewer than 54 solves, as it is more viscous than the $Re_\infty = \infty$ case, thereby outperforming the PTC method. Since PTC can be implemented at each DBC sub-problem, it is further postulated that a combination of PTC and DBC with the surrogate variables could converge even faster and more robustly.

Chapter 6

Conclusion

6.1 Summary and Conclusions

This thesis investigated methods to enhance the robustness of shock capturing for high-order DG discretization. Dissipation-based continuation methods were applied as a form of globalization for both Burgers' and the Euler equations. For the Euler equations, a method for eliminating nonphysical states using surrogate primitive variables was presented to allow for convergence of problems with shocks exhibiting large oscillations, even without artificial viscosity. Through the use of surrogate variables combined with continuation methods, this work has demonstrated ability to converge to a strong shock solution that was otherwise not possible to realize.

Dissipation-based continuation was used as an alternative to pseudo-transient continuation and p-sequencing. Global artificial viscosity was introduced to the initial problem, and gradually decreased to recover the original low-viscosity or inviscid problem by varying a continuation parameter, λ, which scaled the amount of artificial viscosity. The DBC method used a pre-determined continuation schedule for decreasing λ from 1 to 0 by a factor of ten on each sub-problem, where $\lambda = 0$ recovered the original problem. DBC was generally the most efficient method among those tested, applied to both Burgers' and the Euler equations, but there is some uncertainty with using a pre-defined continuation schedule.

Because of these reservations with the DBC method, a homotopy method was introduced in which the continuation parameter became an additional variable. A predictor-corrector method was used to follow the solution path over the range of λ. The homotopy method was able to globalize problems for Burgers' equation that used known Dirichlet boundary conditions; the method was also successful in converging the inviscid shock test case for the Euler equations. While about on par for the test cases for Burgers' equation, the homotopy continuation method was computationally more expensive than the DBC method in nearly all cases shown for the Euler shock test case.

For the Euler equations, surrogate variables were introduced in order to converge solutions with nonphysical oscillations, i.e., strong shocks, in the absence of grid refinement, as well as more sophisticated shock capturing techniques. The conversion from any real values of the surrogate variables maintained physical states of the primitive variables. For the test case shown with a $M = 6.8$ shock, surrogate variables were needed to converge the solution in the presence of nonphysical oscillations. The transient solutions were also allowed to pass through states with negative surrogate densities and temperatures; while this had little effect on the convergence of smooth (high-viscosity) problems, it did allow the less viscous problems reach the converged solutions.

For higher Reynolds numbers and in the absence of any continuation method, the surrogate variables showed more robust convergence at higher order than the primitive variables. When combined with pseudo-transient continuation, the use of surrogate variables showed superior convergence to the primitive variables for the same high Reynolds numbers. The artificial viscosity continuation methods were then applied to the surrogate variables for the inviscid shock problem. The test cases were all able to converge with the use of either the pseudo-transient, diffusion-based, or homotopy continuation methods. It is expected that a combination of surrogate variables with diffusion-based continuation as well as pseudo-transient continuation would be the most robust and efficient method to converge the invsicid shock problem.

6.2 Future Work

Interface with grid adaptation

The next step is to introduce surrogate variables to an already-existing adaptive grid framework. The surrogate variables and other continuation methods will be used to converge on unrefined grids. As the grid refines around shocks, the surrogate variables will allow the cycle to continue in the presence of Gibbs oscillations while local artificial viscosity and grid refinement will eventually remove the oscillations altogether.

More Euler and Navier Stokes testing

A side by side comparison was done of the Euler solver with surrogate variables to a solver without surrogate variables that relies on a line search to preserve physicality, but this was done only for one initial condition and one test case. We would like to see how the surrogate method performs for a larger variety of initial conditions and test problems, as well as for higher polynomial orders. Additionally, direct comparisons of the dissipation-based and pseudo-transient continuation methods applied to the surrogate variables for the same Reynolds number would help to better establish the relative merits of either method beyond what is currently only postulated.

Investigation of artificial viscosity methods

In order to preserve solution accuracy, methods should be investigated to only add dissipation where needed, by use of a sensor that activates the artificial viscosity. Other shock capturing research has successfully used shock switches with artificial viscosity methods. Furthermore, the artificial viscosity should scale with the mesh size, in addition to the switch.

The use of artificial Laplacian viscosity should also be carefully examined. Since the artificial viscosity initially dominates the total residual in diffusion-based continuation, the artificial viscosity should ideally be linear or close to linear, such that the Newton-Raphson solver quickly converges to a solution. Laplacian viscosity is linear, full rank, and its diffusion matrix is acted on by the gradient of the solution variables. Additionally, these properties would potentially allow the homotopy method to start from solving just the Laplacian for its first solve, which would likely increase robustness with respect to the initial conditions.

Expand homotopy framework

A basic homotopy framework has been implemented, but it is missing step size control. Decreasing the predictor step size can be very useful towards the end of the solution process when there is very little artificial dissipation left in the problem and oscillations arise, but using an accordingly small step size throughout the whole solve slows down the process unnecessarily. Including step size control will increase the efficiency and robustness of the homotopy procedure by selecting a better choice for the predicted λ.

Furthermore, it would be beneficial to explore different options for $G(Q)$ for the Euler and Navier Stokes equations that do not include the convective residual. Homotopy showed promise with Burgers' equation, in part because it matched the boundary conditions of the exact answer and because it always solved a linear system for $G(Q_0)$. It is expected that the homotopy process would perform better for the Euler equations if it solved a diffusion equation for $G(Q)$. However, a diffusion equation would require three boundary conditions at the outflow, while the real system requires only one (static pressure) for the shock case considered here.

Boundary conditions

This leads us to consider boundary conditions that vanish with viscosity. Of interest are the Berg-Nordstrom [8] boundary conditions, which are dual-consistent and stable for the Euler and Navier Stokes equations. For subsonic outflow with nonzero viscosity, these give three boundary conditions. In the inviscid limit, only one condition remains that sets static pressure.

Bibliography

[1] J. C. Alexander and James A. Yorke. The homotopy continuation method: Numerically implementable topological procedures. *Transactions of the American Mathematical Society*, 242:271–284, 1978.

[2] E. L. Allgower. A survey of homotopy methods for smooth mappings. In E.L Allgower, K Glashoff, and H.-O Peitgen, editors, *Numerical Solution of Nonlinear Equations*, volume 878 of *Lecture Notes in Mathematics*, pages 1–29. Springer-Verlag, 1981.

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

[4] G.E. Barter and D.L. Darmofal. Shock capturing with higher-order, PDE-based artificial viscosity. AIAA 2007-3823, 2007.

[5] F. Bassi and S. Rebay. A high-order discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

[6] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In Karniadakis Cockburn and Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, pages 197–208. Springer, Berlin, 2000.

[7] F. Bassi and S. Rebay. Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 40:197–207, 2002.

[8] Gens Berg and Jan Nordström. Duality based boundary conditions and dual consistent finite difference discretizations of the navier–stokes and euler equations. *Journal of Computational Physics*, 259:135–153, 2014.

[9] R. P. Brent. Chapter 4: An algorithm with guaranteed convergence for finding a zero of a function. In *Algorithms for Minimization without Derivatives*, pages 47–60. Prentice-Hall, 1973.

[10] R. P. Brent. Chapter 5: An algorithm with guaranteed convergence for finding a minimum of a function of one variable. In *Algorithms for Minimization without Derivatives*, pages 61–80. Prentice-Hall, 1973.

[11] David Brown and David W. Zingg. Advances in homotopy continuation methods in computational fluid dynamics. AIAA 2013-2370, 2013.

[12] G.V. Candler, M.D. Barnhardt, T.W. Drayna, I. Nompelis, D.M. Peterson, and P. Subbareddy. Unstructured grid approaches for accurate aeroheating simulations. AIAA 2007-3959, 2007.

[13] G. F. Carey and R. Krishnan. Continuation techniques for a penalty approximation of the navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 48:265–282, 1985.

[14] Marco Ceze and Krzysztof J. Fidkowski. Pseudo-transient continuation, solution update methods, and cfl strategies for dg dsicretization of the rans-sa equations. AIAA 2013–2686, 2013.

[15] Marco Ceze and Krzysztof J. Fidkowski. Constrained pseudo-transient continuation. *International Journal of Numerical Methods for Heat & Fluid Flow*, 102:1683–1703, 2015.

[16] G. Chavent and G. Salzano. A finite element method for the 1D water flooding problem with gravity. *Journal of Computational Physics*, 42:307–344, 1982.

[17] S. H. Choi, D. A. Harney, and N. K. Book. A robust path tracking algorithm for homotopy continuation. Technical Report 6, 1996.

[18] B. Cockburn, S. Hou, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.

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

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

[21] B. Cockburn and C. W. Shu. The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.

[22] David L. Darmofal, Steven R. Allmaras, Masayuki Yano, and Jun Kudo. An adaptive, higher-order discontinuous galerkin finite element method for aerodynamics. AIAA CFD Conference, June 2013.

[23] B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjögreen. On godunov-type methods near low densities. *Journal of Computational Physics*, 92:273–295, 1991.

[24] Krzysztof J. Fidkowski. A high-order discontinuous Galerkin multigrid solver for aerodynamic applications. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2004.

[25] Koji Fujiwara, Yoshifumi Okamoto, Akihisa Kameari, and Akira Ahagon. The newton–raphson method accelerated by using a line search—comparison between

energy functional and residual minimization. *IEEE Transaction on Magnetics*, 41(5):1724–1727, 2005.

[26] Marshall C. Galbraith. *A Discontinuous Galerkin Chimera Overset Solver.* PhD thesis, University of Cincinnati, School of Aerospace Systems, December 2013.

[27] G. Gassner, C. Altmann, F. Hindenlang, M. Staudenmeier, and C.-D. Munz. Explicit discontinuous galerkin schemes with adaptation in space and time. In H. Deconinck, editor, *VKI LS 2009-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009.* Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

[28] Peter A. Gnoffo and Jeffery A. White. Computational aerothermodynamics simulations issues on unstructured grids. AIAA 2004-2371, 2004.

[29] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230:4248–4267, 2011.

[30] Wenrui Hao, Jonathan D. Hauenstein, Chi-Wang Shu, Andrew J. Sommese, Zhiliang Xu, and Yong-Tao Zhang. A homotopy method based on weno schemes for solvingsteady state problems of hyperbolic conservation laws. *Journal of Computational Physics*, 250:332–346, 2013.

[31] Ami Harten and Stanley Osher. Uniformly high order accurate non-oscillatory schemes. i. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.

[32] Jason E. Hicken, Howard Buckley, Michal Osusky, and David W. Zingg. Dissipation-based continuation: a globalization for inexact-newton solvers. AIAA 2011-3237, 2011.

[33] Jason E. Hicken and David W. Zingg. Globalization strategies for inexact-newton solvers. AIAA 2009-4139, 2009.

[34] Yixuan Hu, Carlee Wagner, David L. Darmofal, Marshall Galbraith, and Steven R. Allmaras. Application of a higher-order adaptive method to rans test cases. AIAA Paper 2015–1530, 2015.

[35] Joseph W. Jerome. Approximate newton methods and homotopy for stationary operator equations. *Constructive Approximation*, 1(1):271–285, 1985.

[36] Wataru Kuroki, Kiyotaka Yamamura, and Shingo Furuki. An efficient variable gain homotopy method using the spice-oriented approach. *IEEE Transactions on Circuits and Systems*, 54(7):621–625, 2007.

[37] Timur Linde and Philip L. Roe. Robust euler codes. AIAA 1997-2098, 1997.

[38] Hong Luo, Joseph D. Baum, and Rainald Löhner. A hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. *J. Comput. Phys.*, 225:686–713, 2007.

[39] James M. Modisette. *An Automated Reliable Method for Two-Dimensional Reynolds-averaged Navier-Stokes Simulations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, September 2011.

[40] Todd A. Oliver. *A Higher-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-averaged Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2008.

[41] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. AIAA 2006-0112, 2006.

[42] Jianxian Qiu and Chi-Wang Shu. Hermite weno schemes and their application as limiters for runge-kutta discontinuous Galerkin method: One-dimensional case. *Journal of Computational Physics*, 193(1):115–135, 2004.

[43] Saeed Khaleghi Rahimian, Farhang Jalali, J.D. Seader, and R.E. White. A new homotopy for seeking all real roots of a nonlinear equation. *Computers and Chemical Engineering*, 35:403–411, 2011.

[44] Nor Hanim Abd. Rahman, Arsmah Ibrahim, and Mohd Idris Jayes. Newton homotopy solution for nonlinear equations using maple14. *Journal of Science and Technology*, 3(2):69–75, 2011.

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

[46] G. R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Math. Comp.*, 50:75–88, 1988.

[47] D. S. Riley and K. H. Winters. A numerical bifurcation of natural convection in a tilted two-dimensional porous cavity. *J. Fluid Mech.*, 215:309–329, 1990.

[48] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

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

[50] Steven A. Stage. Comments on an improvement to the brent's method. *International Journal of Experimental Algorithms*, 4(1):1–16, 2013.

[51] Amy Cha-Tien Sun. *Global optimization using the Newton homotopy-continuation method with application to phase equilibria.* PhD dissertation, University of Pennsylvania, 1993.

[52] Bram van Leer. Towards the ultimate conservative difference scheme. V - a second-order sequel to godunov's method (for ideal compressible flow). *Journal of Computational Physics*, 32:101–136, 1979.

[53] Cheng Wang, Xiangxiong Zhang, Chi-Wang Shu, and Jianguo Ning. Robust high order discontinuous galerkin schemes for two-dimensional gaseous detonations. *Journal of Computational Physics*, 231:653–665, 2012.

[54] L. T. Watson. Globally convergent homotopy algorithms for nonlinear systems of equations. *Nonlinear Dynamics*, 1(2):143–191, 1990.

[55] Meilin Yu and Z. J. Wang. Homotopy continuation for correction procedure via reconstruction – discontinuous galerkin (cpr-dg) methods. AIAA 2015-0570, 2015.

[56] Zhengqiu Zhang. An improvement to the brent's method. *International Journal of Experimental Algorithms*, 2(1):21–26, 2011.

[57] Valentin Zingan, Jean-Luc Guermond, Jim Morel, and Bojan Popov. Entropy viscosity method for nonlinear conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 253:479–490, 2013.