Development of SUPG and Stabilized Finite Element Method Solvers for the Two Fluid Plasma Model

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Kenneth Croft

May 2024

I would like to dedicate this dissertation to my parents, Ken and Diane Croft.

Acknowledgements

I would like to acknowledge the support of the University of Tennessee Space Institute and Tennessee Space Grant consortium, and the patience and support of my advisor, Dr. Trevor Moeller.

Abstract

This dissertation describes efforts to employ stabilized finite element method approaches to simulate ideal two fluid plasma dynamics. First, the streamline-upwind/Petrov-Galerkin (SUPG) finite element method, which is well developed and known to be applicable to models containing terms like those in the ideal two fluid plasma model, is employed. Then, in an attempt to address some shortcomings found in that approach, another stabilized finite element method is developed along similar lines, starting from a steady state advection-reaction equation rather than a steady state advection-diffusion equation as was done in the development of the SUPG method. The performance of the SUPG method for the ideal two fluid plasma model is evaluated by simulating the two fluid analog of the Brio-Wu shock tube problem and the Geospace Environment Modeling (GEM) challenge problem, one dimensional and two dimensional transient problems, respectively. The stabilized method based on an advection-reaction equation is ultimately applied to the two fluid analog of the Brio-Wu shock tube problem after some examination in the context of simpler models.

The SUPG finite element method was found to perform well for the two fluid analog of the Brio-Wu shock tube problem at lower charge to mass ratios, but some difficulty was encountered at higher charge to mass ratios. It is possible that this difficulty could be resolved with more grid resolution. The advection-reaction equation based stabilized method was able to simulate the two fluid analog of the Brio-Wu shock tube problem, but ultimately did not appear to be superior to existing methods such as the SUPG finite element method.

Table of Contents

1 Introduction **1**

2 Literature review **3**

3 Mathematical modeling of plasma dynamics **6**

3.1 The multi-fluid plasma model 8

3.2 Electrodynamics modeling 10

3.2.1 The purely hyperbolic Maxwell's equations 12

4 Numerical methods **14**

4.1 Finite difference methods 16

4.2 Finite volume methods 18

4.3 Finite element methods 21

4.3.1 The discontinuous Galerkin finite element method 24

4.3.2 Stabilized finite element methods 25

5 Software implementation **47**

6 Test and validation problems **56**

6.1 The method of manufactured solutions 56

6.2 Scaling and nondimensionalization of the ideal two fluid model 59

6.3 The Brio-Wu shock tube problem 64

6.3.1 The Brio-Wu shock tube problem for the two fluid model 64

6.4 The GEM challenge problem 70

7 Results and discussion **76**

7.1 SUPG results 76

7.1.1 Manufactured solution 76

7.1.2 Two fluid Brio-Wu problem 92

7.1.3 GEM challenge problem 114

7.2 Advection-reaction equation based stabilized FEM results 138

7.2.1 Manufactured solution 138

7.2.2 Two fluid Brio-Wu problem 144

8 Conclusion **150**

Bibliography **153**

Appendices **160**

A Preliminary numerical evaluation of stabilized FEM based on advection-reaction equation 160

A.1 Steady state advection-reaction 160

A.2 Steady state advection-oscillation 161

A.3 Transient advection-reaction/advection-oscillation 164

A.4 steady state advection-reaction two equation system 173

A.5 Transient advection-oscillation as a two equation system 177

Vita **185**

List of Figures

4.1 Two dimensional domain and discretizations 15
4.2 Illustration of method stencils 45

5.1 Overall program flow chart 49
5.2 Program outer solve loop flow chart 50
5.3 Program global system assembly and solution flow chart 51

6.1 Harris equilibrium magnetic field in the GEM challenge problem domain. 73
6.2 GEM challenge problem initial magnetic field and boundary conditions. 74

7.1 One dimensional, transient manufactured solution error convergence plots with SUPG, BDF2, no discontinuity capturing 79
7.2 One dimensional, transient manufactured solution error convergence plots with SUPG, implicit trapezoid time term, no discontinuity capturing 80
7.3 One dimensional, transient manufactured solution error convergence plots with SUPG, implicit trapezoid time term, unswitched discontinuity capturing 81
7.4 One dimensional, transient manufactured solution error convergence plots with SUPG, implicit trapezoid time term, switched discontinuity capturing 82
7.5 One dimensional, transient manufactured solution electron density convergence plots with SUPG, implicit trapezoid time term, no discontinuity capturing 83
7.6 One dimensional, transient manufactured solution electron density convergence plots with SUPG, implicit trapezoid time term, unswitched discontinuity capturing terms 83
7.7 One dimensional, transient manufactured solution electron density convergence plots with SUPG, implicit trapezoid time term, switched discontinuity capturing terms . . 84

7.8 Two dimensional, steady state manufactured solution error convergence plots with SUPG, no discontinuity capturing 86
7.9 Two dimensional, steady state manufactured solution error convergence plots with SUPG and unswitched discontinuity capturing terms 87
7.10 Two dimensional, steady state manufactured solution error convergence plots with SUPG and switched discontinuity capturing terms 88
7.11 Two dimensional, steady state manufactured solution error convergence plots with SUPG, quadratic elements, no discontinuity capturing 89
7.12 Two dimensional, steady state manufactured solution error convergence plots with SUPG, quadratic elements, and unswitched discontinuity capturing terms 90
7.13 Two dimensional, steady state manufactured solution error convergence plots with SUPG, quadratic elements, and switched discontinuity capturing terms 91
7.14 Two dimensional steady state manufactured solution electron density convergence plots with SUPG, switched discontinuity capturing terms, linear elements 93
7.15 Two dimensional steady state manufactured solution electron density convergence plots with SUPG, switched discontinuity capturing terms, quadratic elements 94
7.16 Brio-Wu problem with $\frac{q_i}{m_i} = 1$ species densities computed with the SUPG FEM without discontinuity capturing term for different specific heat ratios. 95
7.17 Comparison of two fluid density results with $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$ to ideal MHD. . . 98
7.18 Brio-Wu problem species densities overlaid to show quasineutrality in results computed with the SUPG FEM without discontinuity capturing terms. 98
7.19 Time evolution of total energy in the domain for Brio-Wu problem with $\frac{q_i}{m_i} = 1$ simulated with the SUPG FEM and various discontinuity capturing terms and time differences. 99
7.20 Time evolution of total energy in the domain for Brio-Wu problem with $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms and time differences. 101
7.21 Electron density oscillations to the left of the domain of interest for Brio-Wu problem with $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms and time differences. 102

7.22 Brio-Wu problem species densities for simulation with $\frac{q_i}{m_i} = 100$ without discontinuity capturing terms 104
7.23 Brio-Wu problem electromagnetic fields for simulation with $\frac{q_i}{m_i} = 100$ without discontinuity capturing terms 104
7.24 Time evolution of total energy in the domain for Brio-Wu problem with $\frac{q_i}{m_i} = 100$ simulated with the SUPG FEM, no discontinuity capturing terms, and implicit trapezoid time differences. 105
7.25 Brio-Wu problem with $\frac{q_i}{m_i} = 1$ species densities comparison SUPG FEM without discontinuity capturing terms and with discontinuity capturing terms in the species fluid equations. 105
7.26 Ion density on $0.45 \leq x \leq 0.55$ at $t = 2.6$ for Brio-Wu problem with $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms. 108
7.27 Electron density on $0.1 \leq x \leq 0.7$ at $t = 2.6$ for Brio-Wu problem with $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms. 108
7.28 Electron density on $0.5 \leq x \leq 0.7$ at $t = 2.6$ for Brio-Wu problem with $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms. 109
7.29 Brio-Wu problem with $\frac{q_i}{m_i} = 1$ electron density computed with SUPG and finite volume schemes comparison 112
7.30 GEM challenge reconnected flux with BDF2 time difference and no discontinuity capturing terms. 116
7.31 GEM challenge simulation failure state electron pressure with no discontinuity capturing and BDF2. 117
7.32 GEM challenge reconnected flux with implicit Euler time difference and no discontinuity capturing terms. 118
7.33 GEM challenge reconnected flux with unswitched discontinuity capturing terms in fluids equations 120
7.34 GEM challenge total energy in domain with unswitched discontinuity capturing terms in the fluids equations 120
7.35 GEM challenge density at $t = 24$, refined mesh, unswitched discontinuity capturing terms in fluid dynamics equations. 121

7.36 GEM challenge electron density at $t = 24$, coarse mesh, unswitched discontinuity capturing terms in fluid dynamics equations. 122
7.37 GEM challenge reconnected flux with switched discontinuity capturing terms in fluids equations 124
7.38 GEM challenge total energy in domain with switched discontinuity capturing terms in the fluids equations 124
7.39 GEM challenge density at $t = 24$ computed with switched discontinuity capturing terms in fluid dynamics equations. 126
7.40 GEM challenge reconnected flux with unswitched system specific discontinuity capturing terms 127
7.41 GEM challenge total energy in domain with unswitched system specific discontinuity capturing terms 127
7.42 GEM challenge reconnected flux with simultaneously switched system specific discontinuity capturing terms 129
7.43 GEM challenge total energy in domain with simultaneously switched system specific discontinuity capturing terms 129
7.44 GEM challenge density and in plane magnetic field at selected times, switched system specific discontinuity capturing viscosity 130
7.45 GEM challenge system specific discontinuity capturing viscosity for PHM equations at $t = 40$ 131
7.46 GEM challenge system specific discontinuity capturing viscosity for ion fluid equations at $t = 40$ 132
7.47 GEM challenge system specific discontinuity capturing viscosity for electron fluid equations at $t = 40$ 133
7.48 GEM challenge reconnected flux with simultaneously switched full model discontinuity capturing terms 135
7.49 GEM challenge total energy in domain with simultaneously switched full model discontinuity capturing terms 135
7.50 GEM challenge solution field at selected times, switched full model discontinuity capturing viscosity 136

7.51 GEM challenge full model discontinuity capturing viscosity at $t = 40$ 137
7.52 One dimensional, transient manufactured solution error convergence plots with SS A-R eq. based stabilization, implicit trapezoid time term, no discontinuity capturing 139
7.53 One dimensional, transient manufactured solution error convergence plots with SS A-R eq. based stabilization, implicit trapezoid time term, unswitched discontinuity capturing 140
7.54 One dimensional, transient manufactured solution error convergence plots with SS A-R eq. based stabilization, implicit trapezoid time term, switched discontinuity capturing 141
7.55 One dimensional, transient manufactured solution electron density convergence plots with SS A-R eq. based stabilization, implicit trapezoid time term, no discontinuity capturing 142
7.56 One dimensional, transient manufactured solution electron density convergence plots with SS A-R eq. based stabilization, implicit trapezoid time term, unswitched discontinuity capturing terms 142
7.57 One dimensional, transient manufactured solution electron density convergence plots with SS A-R eq. based stabilization, implicit trapezoid time term, switched discontinuity capturing terms 143
7.58 Brio-Wu problem with $\frac{q_i}{m_i} = 1$ species densities computed with advection-reaction equation based stabilized FEM without discontinuity capturing term 145
7.59 Brio-Wu problem with $\frac{q_i}{m_i} = 100$ species densities computed with advection-reaction equation based stabilized FEM without discontinuity capturing term 145
7.60 Time evolution of total energy in the domain for Brio-Wu problem with $\frac{q_i}{m_i} = 1$ simulated with advection-reaction equation based stabilized FEM 146
7.61 Brio-Wu problem with $\frac{q_i}{m_i} = 1$ species densities computed with advection-reaction equation based stabilized FEM with simultaneously switched full model discontinuity capturing term 148
7.62 Brio-Wu problem with $\frac{q_i}{m_i} = 100$ species densities computed with advection-reaction equation based stabilized FEM with simultaneously switched full model discontinuity capturing term 148

7.63 Time evolution of total energy in the domain for Brio-Wu problem with $\frac{q_i}{m_i} = 100$ simulated with advection-reaction equation based stabilized FEM 149

A.1 Linear advection-reaction solution with Dirichlet left boundary condition 162
A.2 Linear advection-reaction solution with Neumann left boundary condition 162
A.3 Linear advection-oscillation solution with Dirichlet left boundary condition 163
A.4 Linear advection-oscillation solution with Neumann left boundary condition 163
A.5 Linear advection-oscillation solution with Dirichlet boundary conditions 165
A.6 Transient advection-oscillation solution with periodic boundary conditions, transient based stabilization 167
A.7 Transient advection-oscillation solution with periodic boundary conditions, steady state based stabilization 167
A.8 Transient advection-oscillation solution with Neumann boundary conditions, steady state based stabilization, 32 elements 168
A.9 Transient advection-oscillation solution with Neumann boundary conditions, steady state based stabilization, 16 elements 168
A.10 Transient advection-oscillation solution with Neumann boundary conditions, transient based stabilization, 16 elements 169
A.11 Transient advection-reaction solution with sharp layer simulated using steady state solution based stabilization, 8 elements 169
A.12 Transient advection-reaction solution with sharp layer simulated using SUPG FEM, 8 elements 170
A.13 Transient advection-reaction solution with sharp layer simulated using Galerkin FEM, 8 elements 170
A.14 FEM comparison on transient advection-reaction problem with square wave IC . . . 172
A.15 Simulation of a system of steady state advection-reaction equations with Dirichlet boundary conditions. 175
A.16 Simulation of a system of steady state advection-reaction equations in characteristic variables with Neumann/extrapolated boundary conditions. 176

A.17 Simulation of a system of steady state advection-reaction equations with Neumann/extrapolated boundary conditions, SS A-R eq. based stabilization version 1. 178
A.18 Simulation of a system of steady state advection-reaction equations with Neumann/extrapolated boundary conditions, SS A-R eq. based stabilization version 2. 178
A.19 Simulation of a system of transient advection-reaction equations with periodic boundary conditions, SS A-R eq. based stabilization extension to systems version 1. 180
A.20 Simulation of a system of transient advection-reaction equations with Neumann/extrapolated boundary conditions, SS A-R eq. based stabilization extension to systems version 1. 180
A.21 Simulation of a system of transient advection-reaction equations with periodic boundary conditions, SS A-R eq. based stabilization extension to systems version 2. 182
A.22 Simulation of a system of transient advection-reaction equations with Neumann/extrapolated boundary conditions, SS A-R eq. based stabilization extension to systems version 2. 182

Chapter 1

Introduction

Plasma is essentially ionized gas. [1] It is often considered a separate state of matter, beyond gas, liquid, and solid because its significant ion/free electron content considerably alters its behavior as compared to a nonionized gas, particularly in the presence of electric and magnetic fields. Like a gas, the molecules, atoms, and electrons in a plasma move throughout their enclosure with the actual physical volume of these particles being very small compared to the volume of the enclosure. But, in contrast to particles in a neutral gas, the charged particles of the plasma, the ionized atoms and molecules and free electrons, are not so free to move independently of each other. Displacement of the negatively charged particles, typically free electrons but also possibly negative ions, relative to the positively charged particles will result in an electric potential that will force the particles back towards each other. Additionally, an independent flow of particles of a particular charge constitutes an electric current that will produce a magnetic field. The production of electromagnetic fields due to the motion of the plasma particles and their interaction with externally applied electromagnetic fields results in many peculiar behaviors such as plasma oscillation, ambipolar diffusion, and anisotropic temperature/pressure/diffusion in the presence of magnetic fields. Furthermore, the overall impact of these various electromagnetic interactions on the bulk plasma's behavior can vary wildly depending on the degree of ionization of the plasma, its density/collisionality and temperature, and the presence/strength of any externally applied fields. This adds to the already wide range of behavioral regimes present in neutral fluid dynamics to render plasma dynamics particularly complex to model. In spite of this complexity, the behavior of plasma needs to be understood in many circumstances, usually because it is hot enough to cause

material integrity concerns whether it is encountered as an unavoidable consequence of some other endeavor, as in the case of a hypersonic vehicle traveling through the air at a high speed or a planet or spaceship traveling through the interplanetary medium, or if the presence of the plasma is integral to the operation of the device, such as in the case of an electric thruster or potential fusion power generating device. This has motivated a great deal of work in the development of mathematical models of plasma behavior and numerical methods to enable their application to realistic scenarios, of which the work described here is a small piece. [2–7] In the following chapters, many of these models and numerical methods will be discussed with particular focus on the plasma model and numerical methods employed in the work underlying this paper.

Chapter 2

Literature review

This work involves ideas from the fields of plasma dynamics and numerical methods, in particular the multi-fluid plasma model, the SUPG finite element method, and the idea of stabilized finite element methods. The fields of plasma dynamics and numerical methods are both very mature and the amount of literature related to them is vast, so this literature review will focus mainly on literature related to the particular plasma model and numerical methods relevant to this work.

The multi-fluid description of a plasma model has been around for some time, it appears in text books from as early as the 1960's. [2] However, its direct use as a model for numerical simulation appears to be more recent. Refs. [4,8–10] from the early 2000's are early publications of what seems the most prominent and extensive work on numerical simulations using the model, all of these coming from the Computational Plasma Dynamics Lab at the University of Washington. Their work continues to the present and discusses numerical simulations of the "5 moment" multi-fluid model, which is the one described here, as well as more general multi-fluid models. They have utilized a number of numerical methods with the multi-fluid models, notably the discontinuous Galerkin finite element method [4] and the high resolution wave propagation scheme, a finite volume approach [9]. They do not appear to have attempted to use the SUPG method with the model. Another group of researchers, from two institutions in Belgium, has also implemented a numerical solver for the multi-fluid plasma model using the finite volume method. They published ref. [5] describing that work in 2018. All of these works employed the two fluid analog of the Brio-Wu shock tube problem [11] and the GEM challenge problem [12] as significant validation problems,

refs. [4] and [9] additionally considered some application problems; a z-pinch and lower-hybrid drift instability in a field-reversed configuration, respectively.

Literature related to the component models of the ideal two fluid model, the compressible fluid dynamics equations and purely hyperbolic Maxwell's (PHM) equations, is also of relevance. Two prominent papers on numerical simulations using the PHM equations are refs. [33,34]. They present the equations and many details related to their numerical implementation within finite volume and discontinuous Galerkin algorithms. Ref. [27], a dissertation related to the use of the SUPG method for simulations of electromagnetics using Maxwell's equations was also particularly relevant to this work. Much of the work related to the development of the SUPG method was performed with fluid dynamics as the model of interest. Refs. [17, 20, 22–25] were found particularly useful to this work, providing many of the details necessary to implement the SUPG method solver and, later, the discontinuity capturing terms, and smooth switching of the discontinuity capturing terms. Ref. [62], describing the AUSM flux for developed for the finite volume method also provided critical information used to implement the species fluid equations' boundary conditions.

Finite element methods are one of three major classes of numerical methods for simulating the behavior of phenomena governed by differential or partial differential equations; the other two being finite difference and finite volume methods. All three of these classes of numerical methods are quite mature, being the topic of numerous text books and college courses. [13–16] The earliest work found in the development SUPG finite element method is that of ref. [17], which was published in 1982. That work references some earlier work with "upwinding" finite element methods by introducing artificial diffusion, using Petrov-Galerkin weight functions that shift the weight function associated with a particular node so that it is heavier in upstream elements, and numerical quadrature approaches capable of producing a similar effect. A key novelty of the SUPG approach described in ref. [17] was the extension of the Petrov-Galerkin upwinding approach to multiple spatial dimensions such that the added diffusion was anisotropic, being strongest in the streamline direction and absent in directions perpendicular to the streamline direction. Ref. [17] presented a thorough study of the SUPG method's performance for a multi-dimensional scalar advection-diffusion equation and the incompressible Navier-Stokes equations. Subsequent work, occurring through the 1980's and 1990's investigated and refined the method for transient, compressible flow [18, 19], refined the generalization to multi-dimensional systems of advection-diffusion equations [20], added discontinuity

capturing terms [21, 22], and examined the method's performance for fluid dynamics when the conserved variables are used versus the entropy variables [23]. Work with the SUPG finite element method continues, with the majority of it being in the field of fluid dynamics, such as refs. [24–26]. Another notable area of application is electromagnetics simulations governed by Maxwell's equations [25, 27].

Other notable stabilized finite element methods also evolved in the 1980's and 1990's with the Galerkin/least-squares (GLS) method growing out of the efforts that had originally produced SUPG. [28] That method would eventually be applied to the ideal MHD plasma model as described in ref. [29]. The least squares finite element method was also described around this time in ref. [30] while the (SU+C)G method for advection-diffusion-reaction equations was described in ref. [31]. The former arose from the idea of minimizing the integral of the square of the residual over the domain of the simulation a process that, interestingly, leads to a weighted residual statement with a SUPG-like weight function contribution for equations containing an advective term. The latter added both a SUPG like weight modification and a weight modification that concentrates weight near the associated node. Optimal stabilization parameters were obtained and good results were seen for a scalar equation. Ref. [32] also described an approach for obtaining a weight function modification to complement SUPG for convection-diffusion-reaction equations, but used an additional weight function designed to have a similar effect to added diffusion in the solution gradient direction, rather than the weight concentrating modification of ref. [31].

Ultimately, use of the multi-fluid plasma model with the PHM equations as an electrodynamics model appears to be a state of the art continuum plasma model for situations where electron inertia needs to be retained to appropriately model effects like charge separation, electromagnetic wave propagation, or other phenomena occurring in the scenario of interest. The state of the art approaches to numerically simulating the model are the use of the discontinuous Galerkin finite element method or the finite volume method. Meanwhile, the SUPG finite element method appears to be a contemporary of those numerical method for equations like those of the multi-fluid plasma model (systems of conservation/balance laws for continuum fields) in fluid dynamics and electrodynamics.

Chapter 3

Mathematical modeling of plasma dynamics

The most general plasma model would have to involve the calculation of the trajectories of every particle of every plasma species (neutral and ionized atoms and molecules and free electrons) as each one moves under the effects of the Lorentz force,

$$\boldsymbol{F}_L = q\left(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}\right), \tag{3.1}$$

and collisions with other particles. Or, equivalently, the point in $6n$ dimensional phase space that the n particles of the plasma occupy at each instant of time, from a statistical mechanics point of view. [2] This is the basic idea of the full-particle and statistical-mechanical/kinetic plasma models, which can be simulated numerically via algorithms like the particle-in-cell (PIC) method. [7] While these highly general models offer the most complete and detailed description of plasma dynamics, they are often impractical, even with today's computational resources, due to the number of particles involved. Simplification must often be relied upon to obtain a tractable problem. PIC solvers, themselves, often track "parcels" that represent a collection of many particles located near each other, instead of every individual particle, to reduce the computational cost of PIC at a given number density but this is still insufficient to enable them to handle all plasma dynamics scenarios. As an example of the computational cost involved, ref. [7] describes a magnetic reconnection simulation involving simulating the motion of approximately 2×10^{10} super particles in

a $\sim 262 \times 10^6$ cell mesh for $\sim 9 \times 10^5$ time steps taking roughly 14×10^6 CPU-hours. The authors of that work were able to utilize $\sim 16,000$ cores in parallel to obtain that result in a tolerable amount of time, but such a simulation would likely be out of reach for many.

The continuum models remove the need to consider every particle of a plasma by describing it in terms of a limited number of "macroscopic" properties that are functions of time and space coordinates within the domain of the problem. [2,9] The use of these models in numerical simulations removes the tendency of the computational cost to scale with number density, but this comes at the cost of assumptions that limit the applicability of the model, and thus, any numerical solver built around it. Obviously, the minimum condition necessary for a continuum description is that it must be possible to use the macroscopic properties to compute fluxes and body forces that reflect the real physical behavior of the plasma. This typically requires number densities that result in large numbers of particles within the region of interest or within a typical cell resulting from a numerical method's spatial discretization. A significant amount of interaction between the particles within each cell over the time interval of interest is often also necessary so that a species' thermal velocity distribution can be obtained and used to describe the species' internal energy content in terms of a limited number of temperatures without maintaining knowledge of every particle's motion, among other things. These requirements can be evaluated by considering the Knudsen number [2], a non-dimensional parameter comparing the "mean free path" of a plasma's (or gas's) particles and the length scale of the geometry of interest, potentially a cell length scale within a numerical simulation, and by comparing the particle's collision frequency [2] to the timescale of the problem or the simulation time step size.

The multi-fluid plasma model is among the most detailed plasma models and is laid out in more detail in the following section as it is the model employed in the work described here, the most simplified is likely the ideal MHD model, which reduces the model to 8 governing equations; conservation of mass, momentum, and energy, and magnetic field transport equations through assumptions of quasineutrality, negligible electron mass, infinite speed of light and negligible displacement current after those necessary to arrive at the multi-fluid model. Even after the continuum assumption used to obtain the multi-fluid model, significant opportunities remain to reduce the computational expense of plasma dynamics simulations through model simplification, but it must be understood that the simplifications will typically restrict the model's applicability

because they can lead it to produce incorrect results in scenarios where the assumptions necessary to make the model simplifications are not appropriate. For example, simplifying the ideal two fluid model with the purely hyperbolic Maxwell's equations electrodynamics model into the ideal MHD model reduces the number of macroscopic property fields and governing equations involved from 18 to 8; but, because of the quasi-neutrality, negligible electron mass, and negligible displacement current assumption used to make this simplification, plasma oscillation and electromagnetic wave propagation will not be able to exist in an ideal MHD simulation. [3] This actually compounds the computational savings associated with using the ideal MHD model since plasma oscillation and electromagnetic wave propagation are extremely fast phenomena that make models that don't assume them away extremely numerically stiff, but if these phenomena turn out to have important impacts on the scenario being modeled, that impact will not be shown by an ideal MHD simulation. Thus, in plasma dynamics simulation, it is particularly important to recognize when model simplification is reasonable and exploit it, but also to consider the conditions expected and results obtained to ensure that the conditions necessary for any utilized model simplifications prevail.

3.1 The multi-fluid plasma model

The equations describing the behavior of each species s in the multi-fluid plasma model can be obtained from a kinetic theory model by inserting the particle mass, m_s, momentum, $m_s\boldsymbol{v}_s$, and kinetic energy $\frac{1}{2}m_s\boldsymbol{v}_s \cdot \boldsymbol{v}_s$, as quantities of interest in the Boltzmann equation and integrating over all of velocity space. This process is often called "taking moments" of the Boltzmann equation and is described in ref. [2]. The three equations resulting from taking moments with the three quantities specified are

$$\frac{\partial}{\partial t}\rho_s + \nabla \cdot (\rho_s \boldsymbol{u}_s) = S_{coll\,s}, \tag{3.2}$$

$$\frac{\partial}{\partial t}(\rho_s \boldsymbol{u}_s) + \nabla \cdot (\rho_s \boldsymbol{u}_s \boldsymbol{u}_s) + \nabla \cdot \tilde{\boldsymbol{\Psi}}_s = \frac{q_s}{m_s}\rho_s (\boldsymbol{E} + \boldsymbol{u}_s \times \boldsymbol{B}) + \boldsymbol{A}_{coll\,s}, \tag{3.3}$$

and

$$\frac{\partial}{\partial t}(\rho_s e_{ts}) + \nabla \cdot \left(\rho_s e_{ts}\boldsymbol{u}_s + \tilde{\boldsymbol{\Psi}}_s \cdot \boldsymbol{u}_s + \boldsymbol{Q}_s\right) = \frac{q_s}{m_s}\rho_s \boldsymbol{u}_s \cdot \boldsymbol{E} + M_{coll\,s}, \tag{3.4}$$

with ρ_s being the species mass density, $\boldsymbol{u}_s$ being its mean velocity, $\tilde{\boldsymbol{\Psi}}_s$ being its pressure tensor, $\boldsymbol{Q}_s$ being its heat flux vector, and $\frac{q_s}{m_s}$ being the charge to mass ratio of its particles. The electric

and magnetic fields are represented by $\boldsymbol{E}$ and $\boldsymbol{B}$, respectively, $S_{coll\,s}$, $\boldsymbol{A}_{coll\,s}$, and $M_{coll\,s}$ are source terms accounting for the effects of collisions amongst the particles, and e_{ts} is the species' total energy per unit mass, which would be, for example, $e_{ts} = \frac{3}{2}\frac{P_s}{\rho_s} + \frac{1}{2}\boldsymbol{u}_s \cdot \boldsymbol{u}_s$ for a simple monatomic species with $P_s = \text{Trace}\left(\tilde{\boldsymbol{\Psi}}_s\right)$. These are a particular truncated set of moment equations called the five moment model [4], counting the components of momentum equation as three equations, and bear a clear resemblance to the laws of conservation of mass, momentum, and energy as commonly stated for a compressible fluid, resulting in the model's name. An infinite number of additional moment equations could actually be made by continuing to multiply the particle mass with successively higher order tensor products of the particle velocity vector taking moments of the Boltzmann equation with the results as quantities of interest, and assembling the entire infinite set of moment equations would be necessary to obtain a model that is equivalent to the Boltzmann equation, itself [4], but truncated sets are useful as a simpler model that can often provide the macroscopic plasma information desired at a lower computational cost. [2, 9] It can be seen in eqs. 3.2-3.4, that each successive moment equation introduces terms from the next higher moment equation. (The momentum appears in the continuity equation, the pressure tensor, which depends on the total energy appears in the momentum equation, and the heat flux tensor appears in the energy equation.) This is never resolved, no matter how many moment equations are formed, so closure relationships must be used to express these quantities in terms of others already present in the equations at hand to truncate the set. [2] These closure relationships often rely on assumptions that limit the applicability of models that employ them. Simply obtaining the fully generalized pressure tensor without employing closure relationships would actually require the use of the components of the tensor $m_s\boldsymbol{v}_s\boldsymbol{v}_s$ as quantities of interest for moment equations, rather than simply the kinetic energy, resulting in ten moment equations per species, using the symmetry of $m_s\boldsymbol{v}_s\boldsymbol{v}_s$. [4] Something that could be necessary in a situation where the pressure tensor is indeed anisotropic in ways that can't be modeled well by a closure relationship. In the work described here, an isotropic pressure tensor is assumed such that $\Psi_{sij} = P_s\delta_{ij}$, while the heat flux vector, $\boldsymbol{Q}$, and collisional terms are assumed to be zero. These simplifications imply assumptions that the particles of a particular species do interact with each other enough to maintain an isotropic thermal velocity distribution and local thermodynamic equilibrium, but that the effects of collisions amongst particles of different species are negligible, as are diffusive transport processes within a species.

These simplifications render eqs. 3.2 - 3.4 the equations of the ideal multi-fluid model, which amount to a set of compressible Euler flow equations for each species s with source terms due to the Lorentz force acting as a body force and the monatomic perfect gas internal energy, $e_i = \frac{1}{\gamma - 1}\frac{P}{\rho}$, with $\gamma = \frac{5}{3}$. While it is not a fully general model due to the simplifications discussed, it remains capable of simulating plasma dynamics scenarios where charge separation, electromagnetic wave propagation, electron inertial effects, and/or ion currents are significant factors. It is also a common starting point for many more detailed plasma models. While dealing with an anisotropic pressure tensor can result in the need for additional governing equations, as discussed, many effects such as ionization/recombination reactions, collisional momentum and energy transfer between species, and diffusive terms representing viscous stress and conductive heat transfer can be accounted for with additional terms in the five equations already present in the ideal multi-fluid model. There are also many accounts of simulations employing the ideal two fluid model in the literature [4, 5, 9], which provide an opportunity for code validation and thus make it appealing for an initial implementation of a numerical algorithm, even if a somewhat more detailed model is ultimately needed.

3.2 Electrodynamics modeling

The equations of the multi-fluid plasma model previously stated remain under determined due to the presence of the electromagnetic fields $\boldsymbol{E}$ and $\boldsymbol{B}$ in the source terms due to the Lorentz force and its associated work term in the species energy equations. The particular density and velocity fields of the charged species may result in space charge or electric current, which, in turn, affect the electromagnetic fields. Thus, these fields can not typically be specified across the domain of interest without regard for the plasma's behavior, instead, a set of governing equations for the electromagnetic fields must usually be added to close the model. The most complete closure for the multi-fluid model, or any plasma model, is the addition of Maxwell's equations to the model. The Ampère-Maxwell law,

$$\frac{\partial}{\partial t}\boldsymbol{D} - \nabla \times \boldsymbol{H} = -\boldsymbol{J} \tag{3.5}$$

and Faraday's law,

$$\frac{\partial}{\partial t}\boldsymbol{B} + \nabla \times \boldsymbol{E} = \boldsymbol{0} \tag{3.6}$$

provide enough equations to close the model, with electric displacement and magnetic flux density being related to the electric and magnetic fields by the free space relationships $\boldsymbol{D} = \epsilon_0 \boldsymbol{E}$ and $\boldsymbol{B} = \mu_0 \boldsymbol{H}$, respectively, and the material response of the plasma being modeled through the current density, $\boldsymbol{J}$, which can be obtained from the charged fluids' velocities: [2]

$$\boldsymbol{J} = \sum_{s=1}^{N_s} \frac{q_s}{m_s} \rho_s \boldsymbol{u}_s. \tag{3.7}$$

The Gauss laws,

$$\nabla \cdot \boldsymbol{D} = \rho_c \tag{3.8}$$

and

$$\nabla \cdot \boldsymbol{B} = 0, \tag{3.9}$$

which result from taking the divergence of eqs. 3.5 and 3.6 and using the space charge conservation law,

$$\frac{\partial \rho_c}{\partial t} + \nabla \cdot \boldsymbol{J} = 0, \tag{3.10}$$

are not technically necessary since they should be satisfied by a solution that satisfied those equations, but are typically enforced through "divergence cleaning" methods in numerical algorithms because it has been found that divergence error can accumulate enough to induce non-physical behavior when the curl form of Maxwell's equations is used with a discretized version of the space charge conservation law. [33] Divergence cleaning methods are model additions that allow the Gauss laws to be enforced along with Ampère-Maxwell law and Faraday's law in numerical simulations without creating an over constrained model. The purely hyperbolic Maxwell's (PHM) equations [33, 34], to be discussed in sec. 3.2.1, are an example of an electrodynamics model with hyperbolic divergence cleaning. Another approach to ensuring satisfaction of the Gauss laws in numerical simulations is to employ the potential formulation of Maxwell's equations, numerical solutions of which do also satisfy the Gauss laws due to vector identities. [3]

It can be seen that when eq. 3.8 is included in the plasma model, space charge density, ρ_c, is additionally needed to describe the plasma's interaction with the electromagnetic fields, this easily

obtained from the charged species' density fields under the multi-fluid model: [2]

$$\rho_c = \sum_{s=1}^{N_s} \frac{q_s}{m_s} \rho_s. \tag{3.11}$$

Other electrodynamics models are typically simplified models derived from Maxwell's equations alongside simplifications of the plasma model to a single fluid model. In these models, it is usually assumed that the electrons' mass is negligible and that they, consequently, dominate any conduction current or are even able to reconfigure rapidly enough to prevent space charge from accumulating at all (the assumption of quasi-neutrality). These assumptions greatly simplify the generalized Ohm's law resulting from the species momentum equations and, together with assumptions of negligible displacement current and perfect conductivity can reduce the electrodynamics model to the single magnetic induction equation seen in the ideal MHD model. [35]

3.2.1 The purely hyperbolic Maxwell's equations

The purely hyperbolic Maxwell's equations augment Maxwell's equations with terms involving two additional scalar fields, ψ and ϕ, in order to allow inclusion of the Gauss laws as governing equations in the electrodynamics model without producing an overdetermined set of equations. They are discussed in particular since they are the electrodynamics model employed in the work described here. The PHM equations are

$$\frac{\partial}{\partial t}\boldsymbol{D} - \nabla \times \boldsymbol{H} + \chi \nabla \psi = -\boldsymbol{J} \tag{3.12}$$

and

$$\frac{\partial}{\partial t}\boldsymbol{B} + \nabla \times \boldsymbol{E} + \kappa \nabla \phi = \boldsymbol{0}, \tag{3.13}$$

analogous to the Ampère-Maxwell law and Faraday's law, respectively, along with the modified Gauss laws

$$\frac{\partial}{\partial t}\psi + \chi \nabla \cdot \boldsymbol{D} = \chi \rho_c \tag{3.14}$$

and

$$\frac{\partial}{\partial t}\phi + \kappa \nabla \cdot \boldsymbol{B} = 0. \tag{3.15}$$

The χ and κ appearing in these equations are simply constant parameters that set the added wave speeds of this model. These equations fit into the standard hyperbolic balance law form,

$$\frac{\partial}{\partial t}\boldsymbol{q} + \frac{\partial}{\partial x_d}\boldsymbol{f}_d = \boldsymbol{s}. \tag{3.16}$$

Viewing the equations from this perspective, it is possible to construct a one dimensional flux across an abritrarily oriented surface, obtain the Jacobian of that flux with respect to the conserved variables, then diagonalize that one dimensional flux Jacobian to obtain the characteristic variables of the system of equations and its wave speeds as is done with any other set of hyperbolic balance laws. [15] Using the vacuum relations to relate $\boldsymbol{D}$ and $\boldsymbol{E}$ and $\boldsymbol{B}$ and $\boldsymbol{H}$, the one dimensional flux Jacobian can be found to have all nonzero eigenvalues of $\pm\frac{1}{\sqrt{\epsilon_0\mu_0}}$, $\pm\frac{1}{\sqrt{\epsilon_0\mu_0}}$, $\pm\chi$, and $\pm\kappa$, assuming that κ and χ are nonzero. The quantity $\frac{1}{\sqrt{\epsilon_0\mu_0}}$ appearing in these wave speeds can be recognized as the speed of light, c_0, while χ and κ are the added parameters of the PHM equations, appearing directly as wave speeds. It can be seen from the fact that all of these eigenvalues, which are the wave speeds of the model, are all real and nonzero, that this set of governing equations is, indeed, purely hyperbolic. (The electrodynamics model formed by Ampère-Maxwell and Faraday's laws alone, share the repeated $\pm c_0$ eigenvalues, but have zero eigenvalues for the remaining two.)

The additional fields do require particular conditions alongside common electrodynamics boundary conditions such as the perfectly conducting wall and Silver-Müller absorbing boundary. The details can be found in refs. [33, 34].

Chapter 4

Numerical methods

Numerical methods are simply algorithms for obtaining approximate solutions to math problems that are impossible or, at least, impractically difficult to solve outright. [14] In this case we need to solve a system of partial differential equations or integro-differential equations describing the evolution of the dependent variable fields of the ideal two fluid plasma model equations and PHM equations in order to simulate the behavior of a plasma undergoing a process that we would like to understand. Three main classes of numerical methods exist for solving these types of equations numerically: finite difference methods, finite volume methods, and finite element methods. [13–16] They all involve discretizing the domain of the problem into a finite number of small "cells" or "elements" within which the dependent variable fields are defined as constant or varying in some relatively simple manner, typically as a polynomial function of the spatial coordinates and time if applicable. An example of a two dimensional domain and a quadrilateral and triangle mesh discretizing it are shown in fig. 4.1. When time variation is present, the temporal evolution of the dependent variable fields is typically obtained for the entire spatial domain one time level at a time, sequentially marching from a defined initial condition. These techniques ultimately enable the production of a system of algebraic equations that can be solved for some finite number of "degrees of freedom", usually nodal or cell averaged dependent variable values, that end up defining the approximate solution to the actual governing equations of the model. Solving this system of equations may yield the approximate solution outright for steady state problems involving linear equations, but transient problems usually require repeated assembly and solution of these sets of algebraic equations, each solution yielding the evolution of the approximate solution over some small

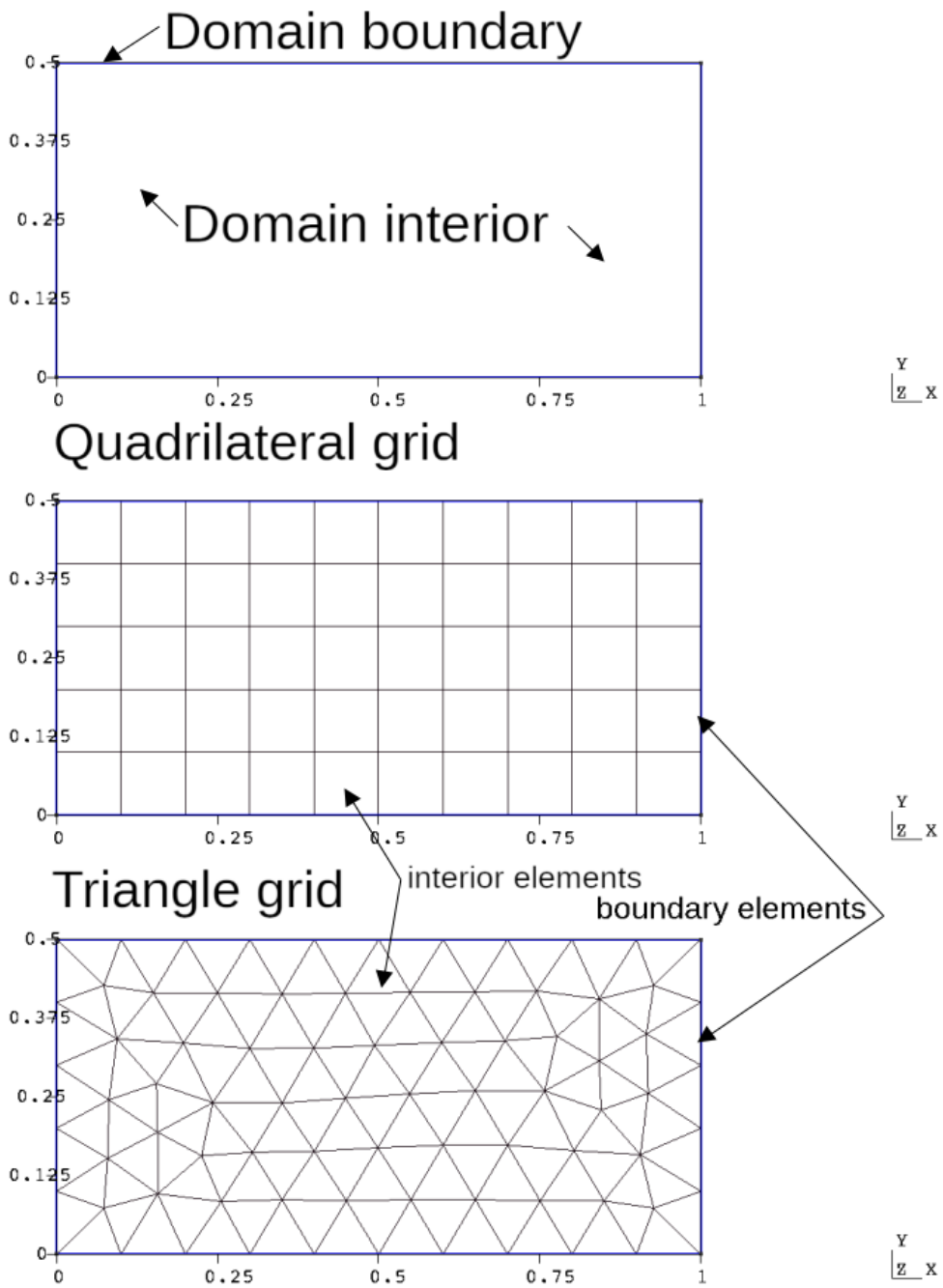


Figure 4.1: Two dimensional domain and discretizations: A simple two dimensional domain and corresponding quadrilateral and triangular discretizations. The interior elements are separated by faces that are line segments, like the boundary elements, in these two dimensional grids. Grid nodes exist only at the element corners in these linear element grids.

increment of time. Further, nonlinear equations of the type addressed by these methods typically require iteration since the system of equations to be solved changes, itself, as the approximate solution evolves. This is much of the reason that the simulation of the plasma model discussed here is so computationally expensive; it is composed of nonlinear governing equations and phenomena that we want to use it to model often contain time varying features, necessitating transient simulations. In the remainder of this chapter, finite difference and finite volume methods will be briefly discussed for comparison to finite element methods, which will be discussed in more detail since they are central to this work.

4.1 Finite difference methods

Finite difference methods typically operate with the values of the dependent variable fields at the grid nodes as their degrees of freedom and are typically constructed from truncated Taylor series approximations of the derivatives in the model's governing equations written based on a typical interior node and its neighbors. An example is likely the simplest means of explaining the method. For the transient advection-reaction equation,

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = sq, \tag{4.1}$$

the derivation of a simple finite difference method begins with the truncated Taylor series,

$$q\left(x_i, t_n + \Delta t\right) = q\left(x_i, t_n\right) + \left.\frac{\partial q}{\partial t}\right|_{x_i,t_n} \Delta t + \mathcal{O}\left((\Delta t)^2\right) \tag{4.2}$$

for q at location x_i and time $t_n + \Delta t$, that is, the value at node i at the next time level, and

$$q(x_i - h, t_n) = q(x_i, t_n) - \left.\frac{\partial q}{\partial x}\right|_{x_i,t_n} h + \mathcal{O}\left(h^2\right) \tag{4.3}$$

for q at location $x_i - h$ and time t_n, that is, the value at the node to the left of node i at the current time level. These Taylor series are simply solved, ignoring the $\mathcal{O}\left(\cdot\right)$ terms, for $\left.\frac{\partial q}{\partial t}\right|_{x_i,t_n}$ and $\left.\frac{\partial q}{\partial x}\right|_{x_i,t_n}$, respectively, resulting in approximations of partial derivatives at the node location and current time that are then substituted into the governing equation to yield an expression that can

be algebraically manipulated to obtain

$$q\left(x_i, t_n + \Delta t\right) = q\left(x_i, t_n\right) + \Delta t\left[-\frac{a}{h}\left(q\left(x_i, t_n\right) - q\left(x_i - h, t_n\right)\right) + sq\left(x_i, t_n\right)\right], \tag{4.4}$$

which can be used to compute the value of q at node i at $t + \Delta t$, given the values of q at nodes i and $i-1$ at time t. Given a boundary condition specifying the value of q at the left-most node of the grid, this formula can be used to compute the value of q at all of the other nodes in the grid at the next time level, allowing time marching of the solution from a known initial condition to the end of the time interval of interest. This is called an explicit finite difference method because it is possible to write the single node update formula expressed in eq. 4.4. It is a quite simple algorithm, but it is only first order accurate due to the dropped terms from the Taylor series and has very limited stability. More stable implicit methods for this governing equation can be constructed by evaluating the spatial derivative and source term based on q at the advanced time level, however evaluation of the spatial derivative at the advanced time level will result in expressions that can not be manipulated into an explicit update formula like eq. 4.4, rather the expressions for each node of the grid will form a system of equations that must be simultaneously solved for q at all of the nodes at the advanced time level. This is the case, in general, for finite difference methods (and the finite volume and finite element methods to be discussed in the following sections) when applied to governing equations like the transient advection-reaction equation; implicit methods offer greater stability, so that larger timesteps can be taken, but usually result in systems of simultaneous equations that increase the computational expense of each timestep. As for the order of accuracy of the method, that can be improved by extending the Taylor series and writing more of them, approximating q at other nearby nodes/time levels, then using them to cancel the $\mathcal{O}\left(h^2\right)$ and $\mathcal{O}\left(\Delta t^2\right)$ terms in the expressions for the partial derivatives, but care must be taken in selecting the neighboring nodes to be used when advection terms like the spatial derivative term in eq. 4.1 are present. They can make it necessary to directionally bias the finite difference derivatives in the "upwind" direction for stability, e.g. the method of eq. 4.4 will be stable for many scenarios where $a \geq 0$ but would not be expected to be stable for scenarios where $a < 0$.

Overall, the biggest limitation of finite difference methods is likely that they generally require structured quadrilateral grids in two dimensions and structured hexahedral grids in three dimensions

since unstructured grids are subject to have a variable number of edges connecting to each node, a situation which complicates the construction of finite difference partial derivative approximations. This increases the difficulty of using these methods, in general, because structured grids are often much more difficult to generate than unstructured grids for the complex geometries often encountered in practical applications. Another notable complication in utilizing finite difference methods is that each node's update formula will depend on an increasing number of neighboring nodes as higher order approximations of the derivatives require more Taylor series to be constructed and combined to cancel each others' lower order terms, and that these dependencies need to change as one approaches boundaries, and/or numerous "ghost" nodes beyond the boundaries need to be added to the grid and populated with dependent variable values that cause the finite difference solution to have the specified value/ derivative values at the boundary.

4.2 Finite volume methods

Cell centered finite volume methods essentially treat the grid's elements as control volumes for which the flow of conserved quantities in/out of the control volume's surface balances against their rates of production and accumulation within the control volume. Node centered finite volume methods construct their control volumes by creating control surfaces that enclose small non-overlapping volumes around each node, typically by defining surfaces containing the centroids of the elements and edges surrounding each node, then proceed similarly. These methods begin with an integral statement of the governing equations which can be obtained by integrating a differential form statement of the governing equations, like eq. 4.1, over an arbitrary control volume and using the divergence theorem [36] to replace the volume integral of the divergence of flux terms with surface integrals, or by a direct construction based on physical arguments. For the multi-dimensional, transient advection-reaction equation,

$$\frac{\partial q}{\partial t} + \nabla \cdot (\boldsymbol{a} q) = sq, \tag{4.5}$$

an extension of eq. 4.1, such an integral statement for an arbitrary, fixed control volume V_i, enclosed by surface S_i, is

$$\int_{V_i} \frac{\partial q}{\partial t} \mathrm{d}V + \int_{S_i} \boldsymbol{a} q \cdot \hat{\boldsymbol{n}} \mathrm{d}A = \int_{V_i} s q \mathrm{d}V, \tag{4.6}$$

in which $\hat{\boldsymbol{n}}$ is the unit normal of differential area $\mathrm{d}A$ that points out of V_i. With the equation in this form, the conserved quantity, or conserved variable, can be identified as q, since the equation is literally a balance law defining the time rate of change of the amount of whatever q is the spatial density of in control volume V_i. For two dimensional meshes like those shown in fig. 4.1, a cell centered finite volume method would take each interior element to be a control volume and its control surface would be composed of the line segments that enclose it. The simplest finite volume methods take q to have a constant value within each control volume and the flux to have a constant value over each of the surrounding faces that make up the control surface, so that the integrals of eq. 4.6 are trivial and the balance law for V_i corresponding to cell i of the grid is simply

$$V_i \frac{\Delta q_i}{\Delta t} + \sum_{f=1}^{n_f} \mathcal{F}_n \left(q_i, q_r, \hat{\boldsymbol{n}} \right)\big|_f A_f = V_i s q_i, \tag{4.7}$$

in which $\mathcal{F}_n$ is a numerical flux computed based on the values of q at the face inside cell i and at the face in the cell on the other side of the face, q_i and q_r, respectively, and the face's unit normal vector. If q_i and q_r are taken to be the current time level values of the dependent variable field, eq. 4.7 defines an explicit scheme to compute Δq_i, the change of q in cell i over the timestep. It can be seen that if q_i and q_r in the flux computations are taken to be the advanced time level values, $q_i + \Delta q_i$ and $q_r + \Delta q_r$, an implicit scheme that requires the solution of a system of simultaneous equations for the Δq's will result. The stability and computational expense consequences of that would be expected to be similar to that discussed for finite difference methods. The use of a constant dependent variable value within each cell results in an algorithm with first order spatial accuracy while the temporal order of accuracy will depend on the time integration technique or finite difference approximation of the time derivative that is employed. For these methods, a way of obtaining improved spatial order of accuracy is to define the dependent variable field to be a polynomial function of the spatial coordinates within each volume, then use that polynomial to evaluate the dependent variable values at the control surface face centroids for the flux computations, a process called reconstruction. [15]

One approach to obtaining the necessary polynomial is via a least-squares fit to the dependent variable values at the centroids of the cell and its nearest neighbors. For a linear polynomial field in a two dimensional element like $q(x, y) = ax + by + c + dxy$, it can be seen that four coefficients need to be determined, thus, at least four data points are necessary. This is, conveniently, exactly what is available if the q values at the centroid of a triangular element and its face neighbors are used while a quadrilateral element and its face neighbors would provide one more data point than necessary. A reconstruction polynomial providing quadratic spatial variation would require fitting to even more surrounding q values, drawing the degrees of freedom for more surrounding elements into the balance law equation corresponding to a particular element. A similar effect to that seen when constructing higher order finite difference methods.

The use of a solution representation that can be discontinuous between adjacent cells and the numerical flux, $\mathcal{F}_n$, necessary to compute the flux at each face (right where the solution is subject to be discontinuous) are defining features of finite volume algorithms. For a scalar equation with a linear advection term like the one in eq. 4.5, a simple upwind flux like

$$\mathcal{F}_n\left(q_i, q_r, \hat{\boldsymbol{n}}\right) = \begin{cases} q_i \boldsymbol{a} \cdot \hat{\boldsymbol{n}} & \text{if } \boldsymbol{a} \cdot \hat{\boldsymbol{n}} > 0 \\ q_r \boldsymbol{a} \cdot \hat{\boldsymbol{n}} & \text{if } \boldsymbol{a} \cdot \hat{\boldsymbol{n}} < 0 \end{cases} \tag{4.8}$$

will work, but numerical fluxes often become much more complex. For example, for an equation containing a nonlinear advection (convection) term, the numerical flux can behave similarly if the flux has the same sign on both sides of the face, but becomes more complicated when the flux changes sign across the face. For systems of equations the common approach is to re-express the advection/convection term in terms of characteristic variables for which the advection/convection terms of the governing equations decouple, allowing the scalar method to be applied to each equation separately. [13] For governing equations in multiple spatial dimensions, it is often necessary to express them as one dimensional in the face normal direction to do this, as it is usually not possible to simultaneously decouple the advection/convection terms for each direction. Considerable literature exists on finite volume methods and much of it focuses on the development of numerical fluxes, particularly for the fluid dynamics equations. [13, 15, 37, 38]

Another issue of importance in finite volume methods is flux limiting or slope limiting for higher order reconstructions. The aforementioned approach of using a least squares fit to obtain a linear slope will result in reconstructions that can produce new local extrema at the face centers of cells whose centroid values are already local extrema and cells adjacent to sharp features in the solution. This will result in the growth of numerical oscillations in the solution. Slope limiting approaches seek to prevent that by limiting the slope of the reconstruction polynomial (and possibly its higher derivatives in higher order reconstructions) so that new local extrema are not produced by the reconstruction. Flux limiting approaches have a similar objective and many flux limiting approaches can be expressed equivalenly or nearly equivalently as slope limiting approaches and vice versa. [15]

Overall, finite volume methods appear to be the most mature, widely adopted, and successful numerical approach for hyperbolic conservation laws/ balance laws and other governing equations in which advection/convection like terms are strong.

4.3 Finite element methods

Like the previously described methods, finite element methods for steady state simulations and semi-discrete finite element methods for transient simulations begin with the discretization of the spatial domain of the problem into a mesh of smaller, simpler domains, called the "elements". For transient simulations, space-time finite element methods also exist. These treat the temporal extent of the problem domain in the same manner as the spatial extents, enabling the construction of space-time elements discretizing a "spatiotemporal" domain and allowing for a complete finite element method based discretization of the problem domain rather than employing a finite difference equation to approximate the time derivative as is done in semi-discrete methods. A domain discretization will usually include both interior and boundary elements discretizing the interior of the domain and the boundary of the domain, respectively. For the kinds of governing equations considered here, the interior elements will be line segments in a one dimensional domain and usually triangles or quadrilaterals in two dimensional domains while the boundary elements are simply points in one dimension and line segments in two dimensions. With a discretization established, simple functions can be chosen to approximate the values of the dependent variables within each

element, these are the basis functions. In many finite element method implementations, including the work described here, Lagrange interpolation functions [14] are employed as the basis functions. These prescribe value of the dependent variable(s) within a particular element according to the formula

$$q(\boldsymbol{x}) = \sum_{i=1}^{n_e} N_i(\boldsymbol{x})\, q_i, \tag{4.9}$$

where $\boldsymbol{x}$ is the position inside the element, n_e is the number of nodes the element has, $N_i(\boldsymbol{x})$ is the basis function associated with node i of the element, and q_i is the value of the dependent variable at the location of node i. The Lagrange basis functions are constructed such that each N_i has the value of 1 at the location of node i and is equal to 0 at the locations of the other nodes in the element and the sum of the basis functions is equal to 1 at all locations inside the element. Lagrange basis functions can be constructed with any polynomial order from 1, up, and the polynomial order of the basis functions used often determines the order of accuracy of the finite element method being employed. At this point, provided that the basis functions chosen define an interpolating polynomial with sufficient differentiability, it is possible to evaluate the balance of the model's governing equation(s) at any location in the domain given some nodal values of the dependent variable fields, this value is called the residual. For a one dimensional advection-diffusion-reaction equation like

$$a\frac{\mathrm{d}q}{\mathrm{d}x} - k\frac{\mathrm{d}^2 q}{\mathrm{d}x^2} = sq, \tag{4.10}$$

the residual is simply

$$r = a\frac{\mathrm{d}q}{\mathrm{d}x} - k\frac{\mathrm{d}^2 q}{\mathrm{d}x^2} - sq. \tag{4.11}$$

Clearly, to obtain dependent variable fields that approximate a solution to the governing equations, it is necessary to obtain nodal dependent variable values that minimize the magnitude of the residual throughout the domain. An approach like integrating the square or absolute value of the residual across the domain would only result in one equation per governing equation, each with a variable for every dependent variable value at every node in the mesh. A least-squares approach can be used to obtain a sufficient number of equations, resulting in a finite element method known as the least-squares finite element method or LSFEM. [30] Otherwise, a series of weight functions with appropriate properties can be chosen, multiplied by the residual, and integrated across the

domain to produce the required number of equations. The use of weight function $w_i(x)$ with the residual from eq. 4.11 on the domain, Ω, results in the weighted residual statement,

$$\int_\Omega w_i \left(a\frac{\mathrm{d}q}{\mathrm{d}x} - k\frac{\mathrm{d}^2 q}{\mathrm{d}x^2} - sq \right) \mathrm{d}x = R_i, \tag{4.12}$$

which would be the foundation of any finite element method for the governing equation, eq. 4.10. Use of the basis functions, themselves, as the weight functions results in one of the most widely known and successful finite element methods; the Galerkin finite element method. [16]

One application where the Galerkin finite element method encounters some difficulty is in the case of equations which contain strong advective or convective terms such as an advection-diffusion equation (eq. 4.10 with $s = 0$) when a is large and k is small or zero such that the quantity $\frac{ah}{2k}$ has a large magnitude when the domain of interest is discretized with elements of length h. This results in solutions that can contain very large changes in q over very small distances referred to as "sharp features" or discontinuities. When the Galerkin finite element method is used on a problem like this, it will produce an approximate solution containing node-to-node oscillations that overshoot and undershoot the actual solution considerably. This can be confined to the vicinity of the the underresolved features or can propagate throughout the domain, depending on the model and nature of the sharp feature/discontinuity. In the case of fluid dynamics, in particular, this can result in unphysical solution features like negative densities which cause the method to break down or produce a uselessly inaccurate solution. Furthermore, in inviscid, compressible fluid flow some features are indeed discontinuities under the assumptions of the governing equations, meaning that it will not be possible to produce a mesh fine enough to resolve the feature and some other strategy must be employed to avoid these numerical oscillations. It is possible to simply move to a more complete fluid model, which includes viscous effects, but in many practical fluid flow scenarios the mesh resolution required to render the natural viscous terms strong enough at the element scale to prevent the spurious oscillations is quite excessive compared to what would otherwise be necessary to adequately resolve the solution features. [17] Thus, there has been considerable research into finite element methods that can handle these problems more effectively, the two most prominent of which are the discontinuous Galerkin (DG) method and the streamline-upwind/Petrov-Galerkin (SUPG) method, which can be regarded as a particular stabilized finite element method.

4.3.1 The discontinuous Galerkin finite element method

In contrast to the other finite element methods discussed here, the discontinuous Galerkin (DG) method allows the approximate solution to be completely discontinuous at all element boundaries. Whereas the Galerkin method used Lagrange basis functions with single field values at each grid node, the DG method with Lagrange basis functions would have a potentially different field value for each element around a particular node. Allowing this discontinuity in the field values necessitates the addition of an interface flux residual to the weighted residual statement in order to couple the solutions in the various elements of the grid together. For an equation with a multidimensional advective flux, $\boldsymbol{f} = \boldsymbol{a}q$, this residual would be

$$\sum_{e_{int}} \int_{\partial\Omega_{e_{int}}} w_i \left(\mathcal{F}_n \left(q_i, q_r, \hat{\boldsymbol{n}}\right) - \left(\boldsymbol{a}q_i\right) \cdot \hat{\boldsymbol{n}}\right) \mathrm{d}A = R_{fi} \tag{4.13}$$

where the sum is over all interior elements, the integration is over the surface of each interior element, $\partial\Omega_{e_{int}}$, and $\mathcal{F}_n$ is a numerical flux function, as discussed in the context of finite volume methods, with q_i being the value of q at the element surface inside the element, q_r being the value of q at the surface outside the element, and $\hat{\boldsymbol{n}}$ being the outward pointing normal of the element surface. Applying integration by parts to the advection component of the interior element residuals changes it from

$$\sum_{e_{int}} \int_{\Omega_{e_{int}}} w_i \nabla \cdot \left(\boldsymbol{a}q\right) \mathrm{d}V \tag{4.14}$$

into

$$\sum_{e_{int}} \left(\int_{\partial\Omega_{e_{int}}} w_i \left(\boldsymbol{a}q_i\right) \cdot \hat{\boldsymbol{n}} \mathrm{d}A - \int_{\Omega_{e_{int}}} \nabla w_i \boldsymbol{a}q \mathrm{d}V \right). \tag{4.15}$$

Clearly, the resulting element surface integral cancels with part of the element surface integral from the interface flux residual, leaving only the integral of the numerical flux over the element surfaces. Rather than looping over the faces of the volume elements and computing the same numerical flux twice, once for the element on each side of a particular interior face, the summation can be carried out over the grid faces, resulting in the residual contribution associated with the advective term

being computed as

$$-\sum_{e_{int}}\int_{\Omega_{e_{int}}}\nabla w_i \boldsymbol{a} q \mathrm{d}V+\sum_{f_{int}}\int_{\Gamma_{f_{int}}}\left(w_i^- - w_i^+\right)\mathcal{F}_n\left(q^-, q^+, \hat{\boldsymbol{n}}\right)\mathrm{d}A+\sum_{f_{bry}}\int_{\Gamma_{f_{bry}}} w_i \mathcal{F}_n\left(q_i, q_b, \hat{\boldsymbol{n}}\right)\mathrm{d}A = R_{ai}, \tag{4.16}$$

where $\Gamma_{f_{int}}$ is the spatial extent of interior grid face f_{int} and $\Gamma_{f_{bry}}$ is the same for a boundary face, and w_i^- and q^- are the value of weight i and the field q on the side of the face that the face normal vector points away from while w_i^+ and q_i^+ are the same on the side of the face that the normal points toward. The boundary face integral term stated here has been written assuming the normal vector points out of the simulation domain and, in its numerical flux, q_i is the field value inside the domain, while q_b is the specified boundary value. The same technique can be applied to a diffusive flux term, if present in the model of interest, so that it can be assembled face-wise using a suitable numerical flux, as well. [39]

DG methods have been noted to require slope limiting or the addition of nonlinear artificial diffusion within elements to prevent numerical oscillations in their solutions [39], like finite volume, finite difference methods, and the continuous and stabilized finite element methods.

4.3.2 Stabilized finite element methods

Stabilized finite element methods seek improved performance by modifying the weight functions while maintaining the continuous basis functions of the original Galerkin method. They are often constructed by applying the finite element method to a model that is simple enough that an analytical solution can be obtained for it with arbitrary weight functions, or weight function perturbations, then solving for the weight functions, or perturbations, necessary to make the equations of the finite element method balance for a nodally exact solution. [17, 31] They can also be constructed based on the idea of modeling the solution components that are not representable in terms of the basis functions on the mesh employed, as in the subgrid scale method. [40] The streamline-upwind/Petrov-Galerkin, or SUPG, finite element method is one of the earliest developed, most mature, and most successful of these methods. [17, 18, 20–23, 25–27, 41] Other notable stabilized finite element methods include the Galerkin/least-squares method [28], subgrid scale method [40], and (SU+C)G method [31] which seem to have come along later as attempts to refine the SUPG method or adapt it for use with models that have different solution quality

issues. The following sections will describe the development of the SUPG method and the similar stabilized method based on an advection-reaction equation developed as part of this work.

The streamline-upwind/Petrov-Galerkin (SUPG) finite element method

In ref. [17], Brooks and Hughes motivate the construction of the original SUPG weight function modification through a comparison between the global matrices obtained when employing the Galerkin FEM, and centered finite differences with artificial viscosity on a uniform mesh to solve the steady state, one dimensional, scalar advection diffusion equation. For this equation, the global matrices resulting from the use of the Galerkin FEM with linear Lagrange basis functions and the use of the centered finite differences,

$$\left.\frac{\mathrm{d}q}{\mathrm{d}x}\right|_{x_i} \approx \frac{q_{i+1} - q_{i-1}}{2h} \tag{4.17}$$

$$\left.\frac{\mathrm{d}^2q}{\mathrm{d}x^2}\right|_{x_i} \approx \frac{q_{i+1} - 2q_i + q_{i-1}}{h^2} \tag{4.18}$$

on a uniform mesh of element length/grid spacing h, result in essentially the same system of equations. (The global matrix entries differ by a constant factor of h.) Thus, they noted that the addition of artificial diffusion as already widely studied in the context of the finite difference scheme should work similarly with the Galerkin FEM. They then noted that when integration by parts is applied to the artificial diffusion term of the Galerkin FEM's weighted residual statement, it can be re-expressed as the advection term with a different weight function as shown in the following sequence of equations:

$$\int_\Omega N_i \left(a\frac{\mathrm{d}q}{\mathrm{d}x} - k\frac{\mathrm{d}^2q}{\mathrm{d}x^2} - \tilde{k}\frac{\mathrm{d}^2q}{\mathrm{d}x^2} \right) \mathrm{d}x = 0 \tag{4.19}$$

$$\int_\Omega \left(N_i \left(a\frac{\mathrm{d}q}{\mathrm{d}x} - k\frac{\mathrm{d}^2q}{\mathrm{d}x^2} \right) + \frac{\mathrm{d}N_i}{\mathrm{d}x}\tilde{k}\frac{\mathrm{d}q}{\mathrm{d}x} \right) \mathrm{d}x - \int_{\partial\Omega} N_i\tilde{k}\frac{\mathrm{d}q}{\mathrm{d}x}\mathrm{d}x = 0 \tag{4.20}$$

$$\int_\Omega \left(\left(N_i + \frac{\mathrm{d}N_i}{\mathrm{d}x}\frac{\tilde{k}}{a} \right) a\frac{\mathrm{d}q}{\mathrm{d}x} - N_i k\frac{\mathrm{d}^2q}{\mathrm{d}x^2} \right) \mathrm{d}x - \int_{\partial\Omega} N_i\tilde{k}\frac{\mathrm{d}q}{\mathrm{d}x}\mathrm{d}x = 0 \tag{4.21}$$

where $\tilde{k}$ is the artificial diffusion coefficient. Using this weight for the diffusion term as well results in the interior weighted residual statement

$$\int_{\Omega} \left(N_i + \frac{\mathrm{d}N_i}{\mathrm{d}x} \frac{\tilde{k}}{a} \right) \left(a \frac{\mathrm{d}q}{\mathrm{d}x} - k \frac{\mathrm{d}^2 q}{\mathrm{d}x^2} \right) \mathrm{d}x = 0, \tag{4.22}$$

which is then used as the basis for the SUPG method. This weight function fits the common stabilized FEM weight function prototype,

$$w_i = N_i + P_i \tag{4.23}$$

with N_i, the i^{th} basis function, being the weight that would be used in the Galerkin FEM, and P_i being the weight modification, here,

$$P_i = \frac{\mathrm{d}N_i}{\mathrm{d}x} \frac{\tilde{k}}{a}. \tag{4.24}$$

Applying integration by parts to the Galerkin weighted portion of this WRS and integrating over an element domain with the two weights that are non-zero in that element domain results in the element matrix,

$$\boldsymbol{A}_e = \begin{bmatrix} \frac{a}{2} + \frac{k}{h} + \frac{\tilde{k}}{h} & \frac{a}{2} - \frac{k}{h} - \frac{\tilde{k}}{h} \\ -\frac{a}{2} - \frac{k}{h} - \frac{\tilde{k}}{h} & -\frac{a}{2} + \frac{k}{h} + \frac{\tilde{k}}{h} \end{bmatrix}. \tag{4.25}$$

Assembling this to the global matrix for the elements to the left and right of any interior node i will result in the global matrix row associated with the weight function that is nonzero at node i expressing the equation

$$\left(-\frac{a}{2} - \frac{k}{h} - \frac{\tilde{k}}{h} \right) q_{i-1} + \left(2\frac{k}{h} + 2\frac{\tilde{k}}{h} \right) q_i + \left(\frac{a}{2} - \frac{k}{h} - \frac{\tilde{k}}{h} \right) q_{i+1} = 0, \tag{4.26}$$

where q_i is the value of q at node i. The one dimensional, steady state linear advection-diffusion equation considered is simple enough that an analyical solution can be obtained, it is

$$q\left(x\right) = c_1 \mathrm{e}^{\frac{a}{k}x} + c_2. \tag{4.27}$$

Thus, the exact values of q at nodes $i-1$, i, and $i+1$ should be

$$q_{i-1} = c_1 \mathrm{e}^{\frac{a}{k}x_i} \mathrm{e}^{-\frac{a}{k}h} + c_2 \tag{4.28}$$

$$q_i = c_1 \mathrm{e}^{\frac{a}{k}x_i} + c_2 \tag{4.29}$$

$$q_{i+1} = c_1 \mathrm{e}^{\frac{a}{k}x_i} \mathrm{e}^{\frac{a}{k}h} + c_2. \tag{4.30}$$

These values can be substituted into eq. 4.26 to produce an equation that can be solved for the $\tilde{k}$ that causes the global FEM equations for weights associated with interior nodes to balance for a nodally exact solution. In this case, the terms that include the integration constant, c_2, cancel and the remaining terms all have the common factor $c_1 \mathrm{e}^{\frac{a}{k}x_i}$ which can be divided away thanks to the zero right-hand side. This results in a very nice equation for a general $\tilde{k}$ in terms of element Peclet number, $\mathcal{P}$, that does not depend on the integration coefficients, which would change with boundary conditions, or the location in the domain (x_i):

$$\tilde{k} = \frac{ah}{2}\left(\coth\left(\mathcal{P}\right) - \frac{1}{\mathcal{P}}\right), \mathcal{P} = \frac{ah}{2k}. \tag{4.31}$$

The term in parethesis is much like a smoothed sign function of the element Peclet number, implying

$$\tilde{k} = \frac{|a|\,h}{2} \tag{4.32}$$

in the pure advection limit, since h should always be positive.

In addition to incorporating the stabilizing diffusion as a weight function modification, a key feature of the SUPG method is including it only in the flow direction. For a scalar equation, the necessary formulation can be determined by transforming the governing equation into a coordinate system in which one direction is parallel to the advection direction and adding an anisotropic stabilizing diffusion tensor which is only nonzero in the flow direction-flow direction component. The resulting WRS for two dimensional advection-diffusion equation,

$$a_x \frac{\partial q}{\partial x} + a_y \frac{\partial q}{\partial y} - k\left(\frac{\partial^2 q}{\partial x^2} + \frac{\partial^2 q}{\partial y^2}\right) = 0, \tag{4.33}$$

transformed back into a coordinate system that is not necessarily aligned with $\boldsymbol{a} = a_x\hat{\boldsymbol{i}} + a_y\hat{\boldsymbol{j}}$, is

$$\int_\Omega \left(N_i + \left(a_x \frac{\partial N_i}{\partial x} + a_y \frac{\partial N_i}{\partial y} \right) \frac{\tilde{k}}{|\boldsymbol{a}|^2} \right) \left(a_x \frac{\partial q}{\partial x} + a_y \frac{\partial q}{\partial y} - k \left(\frac{\partial^2 q}{\partial x^2} + \frac{\partial^2 q}{\partial y^2} \right) \right) \mathrm{d}x\mathrm{d}y = 0. \tag{4.34}$$

Some freedom exists in the calculation of $\tilde{k}$ for the multidimensional case due to h now needing to be the length of the element in the flow direction, a quantity which could be defined in multiple ways. The original reference ([17]) provides formulas for computing it based on directional element Peclet numbers, but a different approach is largely used at this time. Also, later work, such as [20], the quantity $\frac{\tilde{k}}{|\boldsymbol{a}|^2}$ is replaced with the symbol τ resulting in the more commonly seen SUPG modification to weight function n,

$$P_n = a_d \frac{\partial N_n}{\partial x_d} \tau, \tag{4.35}$$

in which summation over the repeated index, d, is implied. Ref. [20] extends the method to systems of equations in multiple spatial dimensions, beginning with generalization to a system of transient advection equations in one dimension. For a system of transient, one dimensional advection equations, it is possible to simply diagonalize the system, then use the scalar SUPG method with the pure advection $\tilde{k}$ shown in eq. 4.32. For a system of m transient, one dimensional advection equations like

$$\frac{\partial}{\partial t}\boldsymbol{q} + \boldsymbol{A}\frac{\partial}{\partial x}\boldsymbol{q} = \boldsymbol{0}, \tag{4.36}$$

where $\boldsymbol{q}$ is the $m \times 1$ vector of dependent variable fields and $\boldsymbol{A}$ is the $m \times m$ advection matrix, or the convective flux Jacobian in the case of a system of transient convection equations, this results in the weight function modification

$$\boldsymbol{P}_n = \boldsymbol{A}\frac{\partial N_n}{\partial x}\boldsymbol{\tau}, \tag{4.37}$$

with

$$\boldsymbol{\tau} = \frac{h}{2} \left| \boldsymbol{A} \right|^{-1}. \tag{4.38}$$

where, if the modal matrix of $\boldsymbol{A}$ is $\boldsymbol{R}$ and $\boldsymbol{\Lambda}$ is the matrix corresponding diagonal matrix of the eigenvalues of $\boldsymbol{A}$ found during the diagonalization process,

$$\left| \boldsymbol{A} \right| = \boldsymbol{R} \left| \boldsymbol{\Lambda} \right| \boldsymbol{R}^{-1}. \tag{4.39}$$

Unfortunately, it is not possible to generalize the method to multidimensional systems of equations in such a rigorous manner due to the fact that it is usually not possible to simultaneously diagonalize the advection matrices, let alone the advection matrices and diffusion matrices. As a result this extension is usually made based on constructing generalized weight function modifications that simplify to the preceding methods when applied to a scalar equation and when applied to a one dimensional system. [20] A weight function modification that does this for multidimensional systems of equations with advection/ convection terms with advection matrices/ flux Jacobians, $\boldsymbol{A}_d$, for each spatial direction d is [24, 25, 27]

$$\boldsymbol{P}_n = \boldsymbol{A}_d \frac{\partial N_n}{\partial x_d} \boldsymbol{\tau} \tag{4.40}$$

with

$$\boldsymbol{\tau}^{-1} = \sum_{i=1}^{n_{Ne}} \left| \boldsymbol{A}_d \frac{\partial N_i}{\partial x_d} \right| \tag{4.41}$$

Use of the absolute value of the advection matrices in this definition makes it clear that it is intended to be optimal for diffusionless models. It can still be used with models that contain diffusive terms, but will lose order of accuracy as mesh resolution becomes high enough for the diffusive terms to be significant at the grid scale. [42] Due to this, there has been considerable research into ways to compute a $\boldsymbol{\tau}$ that performs better when diffusive terms become significant. [24, 42] Since the ideal two fluid plasma model and PHM equations considered in this work do not naturally contain diffusive terms, the SUPG method defined by eqs. 4.40-4.41 has been used here.

Other stabilized finite element methods

The weight function modification of the SUPG method is not the only possibility. The method of substituting the values of an exact solution to the governing equations at the nodes, then solving for weight functions/ weight function modifications that balance the typical equation expressed by the row of the global system corresponding to the weight that is nonzero at an arbitrary interior node can be utilized with any weight modification function or even an arbitrary set of weight functions. In general, this will yield equations constraining integrals of these weight functions over the prototype element. E. g. deriving an FEM that produces nodally exact solutions for a field governed by a one dimensional linear advection-diffusion equation on a mesh of uniformly sized linear Lagrange elements with the weight functions left as unspecified functions of the natural coordinate, ξ: $w_1(\xi)$

and $w_2(\xi)$, results in an equation containing the integrals

$$\int_{-1}^{1} w_1(\xi)\,\mathrm{d}\xi, \int_{-1}^{1} w_2(\xi)\,\mathrm{d}\xi, \int_{-1}^{1} \xi w_1(\xi)\,\mathrm{d}\xi, \text{ and } \int_{-1}^{1} \xi w_2(\xi)\,\mathrm{d}\xi. \tag{4.42}$$

Since satisfaction of this one equation should produce an FEM that provides nodally exact solutions, suitable constant weights of the form $w_1(\xi) = c_1$ and $w_2 = 1 - c_1$ can be specified and the equation can be solved for the single unknown constant, c_1, defining the weights. More complex governing equations can result in multiple equations that need to be satisfied to produce a nodally exact scheme, necessitating more complex weights. The biggest drawback of this approach is that the methods ultimately end up being derived based on governing equations that are simplified to the point that an analytical solution can be obtained. Clearly, these are not the equations that we ultimately want to apply the numerical methods to, and it is possible that what works for those simplified equations will not work on the more complex equations that we want to use the method to solve numerically. Another issue that can arise is that the derived weight functions/ weight function modifications can be too complex or too poorly behaved to be practical. For example, the weight functions for the advection-diffusion scheme just discussed could just as easily have been specified to be $w_1(\xi) = c_1 f(\xi)$ and $w_2(\xi) = 1 - w_1(\xi)$, where $f(\xi)$ is some arbitrarily chosen function, and it would still be possible to solve the single constraining equation for c_1. But, a poor choice of $f(\xi)$ could result in c_1 needing to be infinity for some element Peclet numbers. An example of another method obtained in this manner is found in ref. [31], which describes a stabilized method called (SU+C)G, derived for a scalar advection-diffusion-reaction equation to be solved with an FEM using the SUPG weight function modification and a weight function modification defined by $\gamma\left(\xi^2 - 1\right)/4$, in which the parameter γ, is also obtained from the relations necessary to obtain nodally exact solutions. In that case the additional weight modification function was specifically chosen to enable concentration of the weight near the nodes, a behavior that the authors believed to be desirable when the reaction term is strong. This method with two free stabilization parameters (the usual SUPG τ and the new γ) was possible/necessary due to the more complex analytical solution of the advection-diffusion-reaction equation. Another example is the Galerkin/least-squares method presented in ref. [28], which uses the LSFEM's weight functions as weight modifications added to the Galerkin FEM.

A stabilized finite element method for systems of advection-reaction equations

Likely the most simplified model analogous to the ideal two fluid plasma dynamics equations coupled to the PHM equations is the one dimensional, steady state advection-reaction equation;

$$a\frac{\mathrm{d}q}{\mathrm{d}x} = sq, \tag{4.43}$$

particularly the case with an imaginary s, termed the advection-oscillation equation. [4] For a stabilized FEM employing linear Lagrange elements of uniform length, h, with weight functions,

$$w_i = N_i + a\frac{\mathrm{d}N_i}{\mathrm{d}x}\tau, \tag{4.44}$$

as in the SUPG method, the global matrix is tri-diagonal with the interior row corresponding to the weight that is nonzero at node i having sub-diagonal, diagonal, and super-diagonal entries $a_{i,i-1}$, $a_{i,i}$, and $a_{i,i+1}$, respectively, as shown in eqs. 4.45-4.47 below.

$$a_{i,i-1} = -\frac{a}{2} - \frac{hs}{6} - \tau\left(\frac{a^2}{h} + \frac{as}{2}\right) \tag{4.45}$$

$$a_{i,i} = -\frac{2hs}{3} + \tau\left(\frac{2a^2}{h}\right) \tag{4.46}$$

$$a_{i,i+1} = \frac{a}{2} - \frac{hs}{6} + \tau\left(-\frac{a^2}{h} + \frac{as}{2}\right) \tag{4.47}$$

As in the case of the advection-diffusion equation, this typical row of the global system expresses an equation that can be solved for a value of τ that balances it when the nodal values of the FEM solution are equivalent to the exact solution at the location of node i and its neighbors, nodes $i-1$ and $i+1$. This value of τ is

$$\tau = \frac{h}{a}\left(\frac{1+\mathrm{e}^{\theta}}{2\left(\mathrm{e}^{\theta}-1\right)-\theta\left(1+\mathrm{e}^{\theta}\right)} + \frac{\theta\left(\mathrm{e}^{\theta}+2+\sqrt{3}\right)\left(\mathrm{e}^{\theta}+2-\sqrt{3}\right)}{\left(6\left(\mathrm{e}^{\theta}-1\right)-3\theta\left(\mathrm{e}^{\theta}+1\right)\right)\left(1-\mathrm{e}^{\theta}\right)}\right) \tag{4.48}$$

where $\theta = \frac{sh}{a}$. The technique can also be applied to the transient advection-reaction equation,

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = sq. \tag{4.49}$$

An elementary solution to this equation can be obtained via separation of variables:

$$q\left(x,t\right)=c_2\mathrm{e}^{c_1\left(t-\frac{x}{a}\right)+\frac{s}{a}x} \tag{4.50}$$

with constants c_1 and c_2 that are fixed by the specification of q at two particular locations and times. (A solution that can accomodate a complete initial condition, $q_0(x)$, can be found by the method of characteristics to be $q\left(x,t\right)=q_0\left(x-at\right)\mathrm{e}^{st}$.) Again on a uniform mesh of element size h, and using the implicit trapezoid method with a timestep of Δt, attempting to derive a semi-discrete stabilized FEM with a SUPG style weight function modification for the transient equation based on the elementary solution of eq. 4.50 results in the following equation for τ:

$$\tau=\frac{h}{a}\left(\frac{-3f_1g_1g_2+\left[\frac{2}{C}f_2-\theta f_1\right]h_1h_2}{6f_1g_2^2+3\theta f_1g_1g_2-\frac{6}{C}f_2g_1g_2}\right), \tag{4.51}$$

where θ is as previously defined and the additional symbols are defined by eqs. 4.52 - 4.59 below.

$$C=\frac{a\Delta t}{h} \tag{4.52}$$

$$Z=c_1\Delta t \tag{4.53}$$

$$f_1=\mathrm{e}^Z+1 \tag{4.54}$$

$$f_2=\mathrm{e}^Z-1 \tag{4.55}$$

$$g_1=\mathrm{e}^{\frac{Z}{C}-\theta}+1 \tag{4.56}$$

$$g_2=\mathrm{e}^{\frac{Z}{C}-\theta}-1 \tag{4.57}$$

$$h_1=\mathrm{e}^{\frac{Z}{C}-\theta}+2+\sqrt{3} \tag{4.58}$$

$$h_2=\mathrm{e}^{\frac{Z}{C}-\theta}+2-\sqrt{3} \tag{4.59}$$

It can be seen that one of the boundary/initial condition dependent constants, c_1, remains in the equation for τ, precluding the development of a general method for any scenario governed by a transient advection-reaction equation. The method can indeed perform well when this constant is known (and the solution is really eq. 4.50), but this will not be the case in practice and the method did not handle well with arbitrary values of the constant in preliminary tests on general initial-boundary value problems, so it was not pursued further in this work. Instead, the weight

modification derived based on the steady state equation has been employed in the hopes that it will perform acceptably in transient simulations as well, as is the case for the SUPG method derived based on an advection-diffusion equation.

The extension of the steady state scalar method to steady state systems of advection-reaction equations is fairly straightforward. Multiplying both sides by the inverse of the advection matrix and using the diagonalization of the product of the inverse of the advection matrix and the reaction matrix (eq. 4.70), a decoupled set of equations can be constructed, then the scalar method can be applied to each decoupled equation. Beginning with the system of n linear advection-reaction equations in n unknowns,

$$\mathbf{A}\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{q} = \mathbf{S}\mathbf{q}, \tag{4.60}$$

in which $\mathbf{A}$ and $\mathbf{S}$ are $n \times n$ matrices and $\mathbf{q}$ is an n component column vector, multiplying both sides by the inverse of the advection matrix and moving the reaction term to the left hand side results in

$$\mathbf{I}\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{q} - \mathbf{A}^{-1}\mathbf{S}\mathbf{q} = \mathbf{0}. \tag{4.61}$$

Assuming $\mathbf{A}^{-1}\mathbf{S}$ is diagonalizable, it can be written as

$$\mathbf{A}^{-1}\mathbf{S} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}. \tag{4.62}$$

Making that substitution in eq. 4.61, then multiplying both sides of the result by $\mathbf{R}^{-1}$ on the left and using

$$\mathbf{v} = \mathbf{R}^{-1}\mathbf{q} \tag{4.63}$$

yields the system of equations,

$$\mathbf{I}\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{v} - \mathbf{\Lambda}\mathbf{v} = \mathbf{0}, \tag{4.64}$$

in which $\mathbf{\Lambda}$ is diagonal and the equations are, thus, decoupled. The scalar method can be applied to each equation of the decoupled system resulting in the weighted residual statement,

$$\int_\Omega \left(N_i\mathbf{I} + \mathbf{I}\frac{\mathrm{d}N_i}{\mathrm{d}x}\tilde{\boldsymbol{\tau}}\right)\left(\mathbf{I}\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{v} - \mathbf{\Lambda}\mathbf{v}\right)\mathrm{d}x = \mathbf{0} \tag{4.65}$$

in which $\tilde{\boldsymbol{\tau}}$ is a diagonal matrix in which the entry on the j^{th} row is defined by eq. 4.48 with $a = 1$ and $\theta = \lambda_j h$ where λ_j is the entry on the j^{th} row of $\boldsymbol{\Lambda}$. Inserting the product $\mathbf{R}^{-1}\mathbf{A}^{-1}\mathbf{A}\mathbf{R}$, which resolves to an identity matrix, between the weight and the residual allows the substitutions done to obtain the diagonalized system to be undone resulting in the weighted residual statement,

$$\int_{\Omega} \left(N_i \mathbf{R}^{-1}\mathbf{A}^{-1} + \frac{\mathrm{d}N_i}{\mathrm{d}x}\tilde{\boldsymbol{\tau}}\mathbf{R}^{-1}\mathbf{A}^{-1} \right) \left(\mathbf{A}\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{q} - \mathbf{S}\mathbf{q} \right) \mathrm{d}x = \mathbf{0}, \tag{4.66}$$

for the original system of equations (eq. 4.60). An $\mathbf{A}\mathbf{A}^{-1}$ can be inserted to the left of $\tilde{\boldsymbol{\tau}}$ to obtain the $\mathbf{A}$ of the SUPG style weight modification while the $\mathbf{A}^{-1}$ is absorbed into the $\boldsymbol{\tau}$ definition. A choice now exists for the method used to express the weight function as the Galerkin FEM weight function plus a modification. One approach is to rewrite the resulting weight as $\mathbf{W}_i = N_i\mathbf{I} + (\mathbf{W}_i - N_i\mathbf{I})$, where $\mathbf{I}$ is an identity matrix, then take the $(\mathbf{W}_i - N_i\mathbf{I})$ to be $\mathbf{P}_i$. For a system of n linear advection-reaction equations in n unknowns, the resulting weight matrix is

$$\mathbf{W}_i = N_i\mathbf{I}_{n\times n} + N_i\boldsymbol{\tau}_1 + \frac{\mathrm{d}N_i}{\mathrm{d}x}\mathbf{A}\boldsymbol{\tau}_2, \tag{4.67}$$

and $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are $n \times n$ matrices defined by eqs. 4.68- 4.73 below.

$$\boldsymbol{\tau}_1 = \mathbf{R}^{-1}\mathbf{A}^{-1} - \mathbf{I} \tag{4.68}$$

$$\boldsymbol{\tau}_2 = \mathbf{A}^{-1}\tilde{\boldsymbol{\tau}}\mathbf{R}^{-1}\mathbf{A}^{-1} \tag{4.69}$$

$$\mathbf{A}^{-1}\mathbf{S} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1} \tag{4.70}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n) \tag{4.71}$$

$$\tilde{\boldsymbol{\tau}} = \mathrm{diag}(\tau_1, \tau_2, ..., \tau_n) \tag{4.72}$$

$$\tau_i = h\left(\frac{1 + \mathrm{e}^{\lambda_i h}}{2\left(\mathrm{e}^{\lambda_i h} - 1\right) - \lambda_i h\left(1 + \mathrm{e}^{\lambda_i h}\right)} + \frac{\lambda_i h\left(\mathrm{e}^{\lambda_i h} + 2 + \sqrt{3}\right)\left(\mathrm{e}^{\lambda_i h} + 2 - \sqrt{3}\right)}{\left(6\left(\mathrm{e}^{\lambda_i h} - 1\right) - 3\lambda_i h\left(\mathrm{e}^{\lambda_i h} + 1\right)\right)\left(1 - \mathrm{e}^{\lambda_i h}\right)} \right) \tag{4.73}$$

It can be seen that, because the matrix multiplying N_i in eq. 4.66 is not an identity matrix, a second weight function modification of $N_i\boldsymbol{\tau}_1$ has been picked up due to the extension to a system of equations.

Another approach is to multiply the WRS on the left by $\mathbf{A}\mathbf{R}$. This causes the matrix that is multiplied by the basis function in the weight to become an identity matrix, reducing that term to

the Galerkin FEM weight. This results in a SUPG style weight modification, only, rather than two weight modifications. Using the same notation as above, the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ matrices obtained in this manner are

$$\boldsymbol{\tau}_1 = \mathbf{0} \tag{4.74}$$

and

$$\boldsymbol{\tau}_2 = \mathbf{R}\tilde{\boldsymbol{\tau}}\mathbf{R}^{-1}\mathbf{A}^{-1}. \tag{4.75}$$

The methods would be expected to behave similarly, if not identically, since they ultimately just express the stabilized scheme for the diagonalized equations as stabilized schemes for the original equations and the $\mathbf{AR}$ factor is just a matrix of scalars for linear equations. However, preliminary tests revealed an important difference: It appears that when the method resulting from the first approach is used with a problem in which $\mathbf{A}^{-1}\mathbf{S}$ has complex eigenvalues, both $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are subject to contain complex numbers, however that does not seem to occur when the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ of the second approach is used. The problem with these complex $\boldsymbol{\tau}$'s is that they cause the global system's left-hand side matrix and right-hand side to contain complex entries. To deal with that, either the dependent variables will have to be treated as complex numbers, making the whole global system complex valued or, if the dependent variables are assumed to be real valued, they must simultaneously satisfy two global systems; one with the real left-hand side matrix and right-hand side, and one with the imaginary left-hand side matrix and right-hand side. Preliminary testing with the advection-oscillation equation expressed as a 2 equation, 2 unknown system, that is

$$a\frac{\mathrm{d}z}{\mathrm{d}x} = \mathrm{i}sz \tag{4.76}$$

with $z = z_r + \mathrm{i}z_i$ expressed in the form of eq. 4.60 with

$$\mathbf{q} = \begin{bmatrix} z_r \\ z_i \end{bmatrix}, \tag{4.77}$$

$$\mathbf{A} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}, \tag{4.78}$$

and

$$\mathbf{S} = \begin{bmatrix} 0 & -s \\ s & 0 \end{bmatrix}, \tag{4.79}$$

indicated that the simple approach of adding or differencing the two global systems obtained under the real valued dependent variable fields assumption to form a single global system did not work reliably. For some values of a and s, using a real valued τ_n^* obtained by $\tau_n^* = \text{real}(\tau_n) + \text{imag}(\tau_n)$ (effectively assembling the real global system and imaginary global system, with i factored out, added together) worked while using $\tau_n^* = \text{real}(\tau_n) - \text{imag}(\tau_n)$ did not and vice versa. Treating the entire global system as complex valued did work, but obviously adds computational expense when the dependent variable fields are known to be real valued. That approach would also require substantial reworking of the previously developed ideal two fluid plasma dynamics code already built on the assumption that the dependent variable fields, and indeed the entire global system, are real valued. Fortunately, the alternative τ_n definitions of eq. 4.74 and 4.75 appear to be real valued for real valued dependent variables and governing equations consistently enough to be usable even with a nonlinear model like the ideal two fluid plasma model. More details of the numerical evaluation of these methods with the linear advection-reaction equation and small systems of linear advection equations are presented in appendix A.

An issue that immediately arises with extension to the ideal two fluid plasma model is that $\mathbf{A}$ will not always be invertible. To address this situation, the modal matrix of $\mathbf{A}$, $\mathbf{P}$, can be used to re-express the system in terms of the variables $\mathbf{v} = \mathbf{P}^{-1}\mathbf{q}$. With $\mathbf{A} = \mathbf{P}\mathbf{K}\mathbf{P}^{-1}$ where $\mathbf{K}$ is the diagonal matrix of $\mathbf{A}$'s eigenvalues, and $\mathbf{M} = \mathbf{P}^{-1}\mathbf{S}\mathbf{P}$, thus eq. 4.60 can be written as

$$\mathbf{K}\frac{\mathrm{d}}{\mathrm{d}x}\mathbf{v} = \mathbf{M}\mathbf{v}. \tag{4.80}$$

If $\mathbf{A}$ has zero eigenvalues, $\mathbf{K}$ and $\mathbf{P}$ can be written with those zero eigenvalues grouped together so that the system of equations can be block-partitioned as

$$\begin{bmatrix} \mathbf{K}'' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}x} \begin{bmatrix} \mathbf{v}' \\ \mathbf{v}^* \end{bmatrix} = \begin{bmatrix} \mathbf{M}'' & \mathbf{M}'^* \\ \mathbf{M}^{*\prime} & \mathbf{M}^{**} \end{bmatrix} \begin{bmatrix} \mathbf{v}' \\ \mathbf{v}^* \end{bmatrix}, \tag{4.81}$$

where the parts with the $'$ superscript are associated with the equations containing the nonzero eigenvalues of $\mathbf{A}$ and those with the $*$ superscript are associated with the zero eigenvalues of $\mathbf{A}$. (The $\mathbf{0}$'s are simply matrices of zeroes.) From here, both sides can be multiplied on the left by

$$\begin{bmatrix} \mathbf{I}'' & -\mathbf{M}'^{*}\mathbf{M}^{**-1} \\ \mathbf{0} & \mathbf{I}^{**} \end{bmatrix}, \tag{4.82}$$

where the $\mathbf{I}$'s are identity matrices of sizes determined by the number of nonzero/zero eigenvalues of $\mathbf{A}$, resulting in

$$\begin{bmatrix} \mathbf{K}'' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}x} \begin{bmatrix} \mathbf{v}' \\ \mathbf{v}^{*} \end{bmatrix} = \begin{bmatrix} \mathbf{M}'' - \mathbf{M}'^{*}\mathbf{M}^{**-1}\mathbf{M}^{*'} & \mathbf{0} \\ \mathbf{M}^{*'} & \mathbf{M}^{**} \end{bmatrix} \begin{bmatrix} \mathbf{v}' \\ \mathbf{v}^{*} \end{bmatrix}. \tag{4.83}$$

This has decoupled the system into a set of differential equations, which fit the form of eq. 4.60 with a diagonal, invertible $\mathbf{A}$, to which the previously described stabilized FEM can be applied directly, and a set of algebraic equations. For the numerical simulations described here, we have simply used Galerkin weighting for the algebraic equations. It is also possible that $\mathbf{M}^{**}$ could be uninvertible. To address this it may be possible to further decompose the system and try to appropriately address the parts of the system associated with zero/nonzero eigenvalues of $\mathbf{M}^{**}$, but this has not been pursued at this time. In the ideal two fluid model, this seems to come up in quiet regions of still, quasi-neutral plasma where the absence of flow and body force collapses many parts of the advection and reaction terms to zeroes. In the transient case, the nonzero time derivative term remains and this has the trivial result that the fields should not change over the timestep, but the stabilization matrix calculation becomes an issue because it is based on the steady state equations and many of those will have degenerated to $0 = 0$ in these conditions. To move numerical computations forward in these cases, the pseudoinverse [43] of $\mathbf{M}^{**}$ is employed.

Transforming the scheme resulting from applying the scalar scheme to the differential equations of eq. 4.83 and the Galerkin method to the algebraic equations, and multiplying on the left by the factors necessary to recover the Galerkin weight function results in

$$\boldsymbol{\tau}_1 = \mathbf{0} \tag{4.84}$$

and

$$\boldsymbol{\tau}_2 = \mathbf{A}^{-1}\mathbf{P}\begin{bmatrix} \mathbf{K}''\mathbf{R}\tilde{\boldsymbol{\tau}}\mathbf{R}^{-1}\mathbf{K}''^{-1} & -\mathbf{K}''\mathbf{R}\tilde{\boldsymbol{\tau}}\mathbf{R}^{-1}\mathbf{K}''^{-1}\mathbf{M}'^{*}\mathbf{M}^{**-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{P}^{-1} \tag{4.85}$$

for use with the weight of eq. 4.67 with the advection-reaction system in its original form, eq. 4.60. In eq. 4.85, $\mathbf{R}$ and the λ_i's for $\tilde{\boldsymbol{\tau}}$, which is still computed via eqs. 4.72, 4.73, come from the diagonalization,

$$\mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1} = \mathbf{K}''^{-1}\left(\mathbf{M}'' - \mathbf{M}'^{*}\mathbf{M}^{**-1}\mathbf{M}^{*'}\right). \tag{4.86}$$

It should be noted that it would be possible to utilize the pseudoinverse of $\mathbf{A}$ to avoid the need to block the system like this in the first place, and that was attempted, but did perform well enough to even yield reportable results for the ideal two fluid plasma model. Ultimately, even with this approach of blocking the system of equations and using the stabilized scheme only for the portion of the equations with nonzero advective term eigenvalues, some special treatments were found necessary. For one, it was found necessary to use a linear approximation of τ_i across $\lambda_i = 0$, where the expression of eq. 4.73 has parts that become $\frac{1}{0}$ and $\frac{0}{0}$ It was also found necessary to use the asymtotic limits of τ_i for large λ_i's, where the exponential functions overflow, causing numerical errors even when the the value should be reasonable. (Exponential function overflow was found to cause issues even along the real axis where eq. 4.73 smoothly asymtotes to $\frac{1}{3}$.) Additionally, eq. 4.73 does truly blow up at some values along the imaginary axis, that was addressed, in this work, by limiting the magnitude of τ_i to one. Allowing it to become larger than that tended to cause numerical issues rather than solution improvement. These measures were not considered until the simpler pseudoinverse of $\mathbf{A}$ approach had been abandoned, so they may improve the performance of that approach as well.

Another issue that arises with the ideal two fluid plasma model is that the source term is also nonlinear and can't be expressed as $\mathbf{Sq}$ in only one way. It can be expressed in the form $\mathbf{S}(\mathbf{q})\,\mathbf{q}$ where the coefficient matrix, $\mathbf{S}(\mathbf{q})$, is a function of the dependent variables $(\mathbf{q})$ in several different ways. The electrodynamics variables could be left as the coefficients in $\mathbf{S}(\mathbf{q})$, positioned to multiply by the appropriate species fluid dynamics variables to form the species fluid dynamics equations' source terms, or vice versa. Also, if the rows of the source term Jacobian associated with the species fluid dynamics equations' source terms were all multiplied by $\frac{1}{2}$, to form $\mathbf{S}(\mathbf{q})$, then $\mathbf{S}(\mathbf{q})\,\mathbf{q}$ would

equal the source term. More generally, if the entries on the rows of the Jacobian associated with the species fluid dynamics equations' source terms that are multiplied by the species fluid variables in the product $\mathbf{S}(\mathbf{q})\,\mathbf{q}$ were multiplied by some scalar, α, between 0 and 1, and the entries, on the same rows, that are multiplied by the electrodynamics variables were multiplied by $1-\alpha$, to form $\mathbf{S}(\mathbf{q})$ then $\mathbf{S}(\mathbf{q})\,\mathbf{q}$ would again equal the source term. In this work, the Jacobian of the source term was used so that the $\mathbf{S}$ ultimately used in the computation of $\boldsymbol{\tau}_2$ was just

$$S_{ij} = \frac{\partial s_i}{\partial q_j}. \tag{4.87}$$

The resulting product, $\mathbf{S}(\mathbf{q})\,\mathbf{q}$, does not equal the source term of the equations, but all of the obvious strategies for forming $\mathbf{S}(\mathbf{q})$ matrices that do result in $\mathbf{S}(\mathbf{q})\,\mathbf{q}$ equalling the source term result in matrices that are comprised of scalar multiples of the terms in the source term Jacobian matrix, where the scalars just depend on how one wishes to break up the species fluid dynamics equations' source terms amongst the variables used to compute them.

The semi-discrete formulation for transient models

The semi-discrete formulation, sometimes referred to as the "method of lines" [44], is employed in the software ultimately developed for this work, so it will be briefly described here. In a semi-discrete formulation, the same spatially varying basis functions used in a steady state method are employed and the degrees of freedom vary in time, thus

$$q(\boldsymbol{x},t) = \sum_{i=1}^{n_e} N_i(\boldsymbol{x})\, q_i(t). \tag{4.88}$$

Using indicial notation, the weighted residual statement that would be the foundation of a stabilized finite element method using a continuous solution representation of the form of eq. 4.88 for the transient linear advection equation,

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0, \tag{4.89}$$

would then be

$$\int_\Omega w_i \left(N_j \frac{\mathrm{d}q_j}{\mathrm{d}t} + a\frac{\mathrm{d}N_j}{\mathrm{d}x} q_j \right) \mathrm{d}x = R_i. \tag{4.90}$$

Since neither q_j, nor $\frac{\partial q_j}{\partial t}$ vary with x, they can be brought outside the integrals to obtain

$$\left(\int_\Omega w_i N_j \mathrm{d}x\right) \frac{\mathrm{d}q_j}{\mathrm{d}t} + \left(\int_\Omega w_i a \frac{\mathrm{d}N_j}{\mathrm{d}x}\right) q_j = R_i. \tag{4.91}$$

The global system of equations can now be seen to be a system of ordinary differential equations (ODEs) for the q_j's. One approach to solving this system of ODEs is to approximate the time derivative with a finite difference, this is the approach applied in this work. With Δq_j being the change of q_j from time level n to time level $n+1$, with the time levels being Δt apart, using the implicit Euler time difference would result in

$$\left[\frac{1}{\Delta t}\left(\int_\Omega w_i N_j \mathrm{d}x\right) + \left(\int_\Omega w_i a \frac{\mathrm{d}N_j}{\mathrm{d}x}\right)\right] \Delta q_j + \left(\int_\Omega w_i a \frac{\mathrm{d}N_j}{\mathrm{d}x}\right) q_{j_n} = R_i. \tag{4.92}$$

Since the coeffients are not dependent on the q_j's due to the fact that the governing equation is linear, these equations can all be directly solved for the Δq_j's that make each $R_i = 0$, resulting in a system of linear equations like

$$\left[\frac{1}{\Delta t}\left(\int_\Omega w_i N_j \mathrm{d}x\right) + \left(\int_\Omega w_i a \frac{\mathrm{d}N_j}{\mathrm{d}x}\right)\right] \Delta q_j = -\left(\int_\Omega w_i a \frac{\mathrm{d}N_j}{\mathrm{d}x}\right) q_{j_n}. \tag{4.93}$$

For a nonlinear equation, an iterative approach like the multivariable Newton method could be setup to solve for the Δq_j's that make each $R_i = 0$:

$$\left[\frac{\partial R_i}{\partial\left(\Delta q_j\right)}\right]\Big|_{\Delta q_{j_m}} \Delta\left[\Delta q_j\right] = -\left.R_i\right|_{\Delta q_{j_m}}, \tag{4.94}$$

in which Δq_{j_m} is the value of Δq_j on the m^{th} iteration and $\Delta q_{j_{m+1}} = \Delta q_{j_m} + \Delta\left[\Delta q_j\right]$. The use of different time differences such as BDF2 or the implicit trapezoid rule result in similar equations, and the pseudotime stepping method [45] can also be used to construct an iterative scheme for nonlinear equations. The pseudotime stepping method actually results in a very similar system of equations to the multivariable Newton method.

A discontinuity capturing term for use with SUPG/stabilized finite element methods

When used with nonlinear, multidimensional, and/or transient governing equations, the SUPG method is often unable to fully eliminate node-to-node oscillations in the numerical solution around under-resolved features or sharp discontinuities. An approach to avoiding these node-to-node oscillations, that has been studied considerably, is the inclusion of an additional discontinuity capturing term. An explicit artificial viscosity term is often employed and has been incorporated into the solver developed as part of this work. The particular term employed in this work is defined by eqs. 4.95-4.98, below, and is largely based on ref. [23]. The term is essentially a Galerkin weighted isotropic diffusion term that has had integration by parts applied to it. This discontinuity capturing term was originally developed for compressible fluid flow and the $\tilde{A}_0^{-1}$ in eq. 4.97, is, thus, the transformation between the conserved variables and entropy variables for the equations of fluid dynamics, specifically, in the reference ([23]), but the idea is intended to be extensible to hyperbolic conservation laws in general. In ref. [46], written based on part of the work described here, the alternative discontinuity capturing viscosity definition of eq. 4.99, which does not include $\tilde{A}_0^{-1}$, was tested for one dimensional compressible Euler flow and the one dimensional ideal two fluid plasma model and was found to perform comparably to the original definition for the Sod shock tube problem (compressible Euler flow) and found to be slightly more robust in simulations of the two fluid analog of the Brio-Wu shock tube problem (ideal two fluid plasma dynamics).

$$R_{DCni} = \sum_{e=1}^{n_e} \int_{\Omega_e} D_{ij} \frac{\partial N_n}{\partial x_d} \frac{\partial Q_j}{\partial x_d} dV \tag{4.95}$$

$$D_{ij} = \nu_D \delta_{ij} \tag{4.96}$$

$$\nu_D = \left(\frac{A_i \frac{\partial Q}{\partial x_i} \cdot \tilde{A}_0^{-1} A_j \frac{\partial Q}{\partial x_j}}{\frac{\partial \xi_l}{\partial x_j} \frac{\partial Q}{\partial x_j} \cdot \tilde{A}_0^{-1} \frac{\partial \xi_l}{\partial x_k} \frac{\partial Q}{\partial x_k}} \right)^{1/2} \tag{4.97}$$

$$A_i = \frac{\partial F_{x_i}}{\partial Q} \tag{4.98}$$

$$\nu_D = \left(\frac{\left(A_i \frac{\partial Q}{\partial x_i} \right) \cdot \left(A_j \frac{\partial Q}{\partial x_j} \right)}{\left(\frac{\partial \xi_l}{\partial x_j} \frac{\partial Q}{\partial x_j} \right) \cdot \left(\frac{\partial \xi_l}{\partial x_k} \frac{\partial Q}{\partial x_k} \right)} \right)^{1/2} \tag{4.99}$$

Since the defined viscosity coefficient is $\mathcal{O}(h)$, including this term reduces the method to first order spatial accuracy and often results in severely overdiffuse solutions away from discontinuities. To alleviate this, ν_D is often multiplied by a switching function that is designed to reduce its magnitude away from discontinuities. [25, 47, 48] Smooth switching functions that gradually reduce the discontinuity capturing viscosity to zero over a range of values of the discontinuity sensor function are recommended in the literature over simple binary switching, which can cause difficulty converging the numerical solution. In this work, the smooth switching function defined by

$$S(f) = \begin{cases} 0 & \text{if } f < f_s \\ \left(\frac{f-f_s}{f_e-f_s}\right)^2 \left(3 - 2\frac{f-f_s}{f_e-f_s}\right) & \text{if } f_s \leq f \leq f_e \\ 1 & \text{if } f > f_e \end{cases} \tag{4.100}$$

has been used with a discontinuity sensor function defined by

$$f(Q, \nabla Q) = \frac{\nabla e_i \cdot \nabla e_i}{\left(\frac{e_i}{h}\right)^2} \tag{4.101}$$

to switch discontinuity capturing terms added to the model. This switching function simply defines a cubic spline with a value of 0 and slope of 0 at f_s and value of 1 and slope of 0 at f_e. Thus, the values f_s and f_e are tunable parameters setting the value where the discontinuity capturing viscosity will begin to become nonzero and the value at which it will attain its full value, respectively. In the discontinuity sensor function, h is, again, the element length scale and e_i is the internal energy of a specific fluid. This function is analogous to $\left(\frac{\nabla T \cdot \nabla T}{T^2}\right) h^2$ in a constant specific heat gas, and the term in parenthesis in that expression is associated with entropy production in a fluid flow [49], a feature typically indicative of a shock in inviscid flow, so this sensor function is expected to become increasingly positive around shocks but remain near zero elsewhere in the flow, giving it utility as a discontinuity sensor. Many discontinuity sensor function have been studied in the context of compressible fluid flow. [47, 48, 50] In this work, the sensor function of eq. 4.101 has been used to independently switch discontinuity capturing terms in the ion and electron fluid equations and the sum of this function's value for all the fluids has been used to simultaneously switch discontinuity capturing terms used across the model.

The potential advantages of SUPG and stabilized FEMs

The primary reasons that the SUPG method and other stabilized FEMs that use a continuous solution representation are believed to have the potential to reduce the computational costs associated with complex, stiff models like multi-fluid plasma dynamics are their smaller global systems for implicit schemes, as compared to DG methods and the fact that they can achieve higher order accuracy without expanding the number of neighboring elements or cells upon whose degrees of freedom a particular global system equation depends (called the method's stencil), an effect seen when constructing higher order finite volume and finite difference methods. A comparison of the methods in the context of second order accurate methods for a scalar field on a two dimensional cartesian mesh with n_{cx} cells in the x direction and n_{cy} cells in the y direction illustrates the situation. Using the SUPG method results in $(n_{cx}+1)\times(n_{cy}+1)$ degrees of freedom (DOFs), or less if any boundaries have Dirichlet boundary conditions, and using a cell centered finite volume method results in $n_{cx}\times n_{cy}$ DOFs. However, using the DG method results in $n_{cx}\times n_{cy}\times 4$ DOFs assuming the field in each element is represented by a polynomial providing spatial variation of the form $q(x,y)=ax+by+c+dxy$, comparable to that provided by the SUPG method. As can be seen, for the usually large n_{cx}, n_{cy} used in practical simulations, the number of DOFs in the global system is comparable for the SUPG/stabilized FEMs and the cell centered finite volume method, but both require $\approx 4\times$ fewer DOFs than the DG method. Stabilized methods' potential advantage over finite volume and finite difference methods is due to the fact that their stencil does not expand as higher order accuracy is sought. (This potential advantage is actually possessed by all finite element methods, including the DG method when compared to finite volume and finite difference methods.) Stencils associated with the SUPG method, finite volume method, and DG method are presented in fig. 4.2 to illustrate the situation. The reconstruction necessary to obtain a second order accurate finite volume method brings DOFs from every cell that shares a face with a cell that shares a face with cell i,j (its "neighbors" and "neighbors-of-neighbors") into the stencil while the SUPG/stabilized and DG FEM's only include the neighbors' DOFs in their stencils. From examination of the stencils one can see that the row of the global system obtained using the weight function associated with node i,j in the SUPG/stabilized method will contain 9 nonzero entries, whereas the row obtained from the balance law for volume i,j in the finite volume method's global

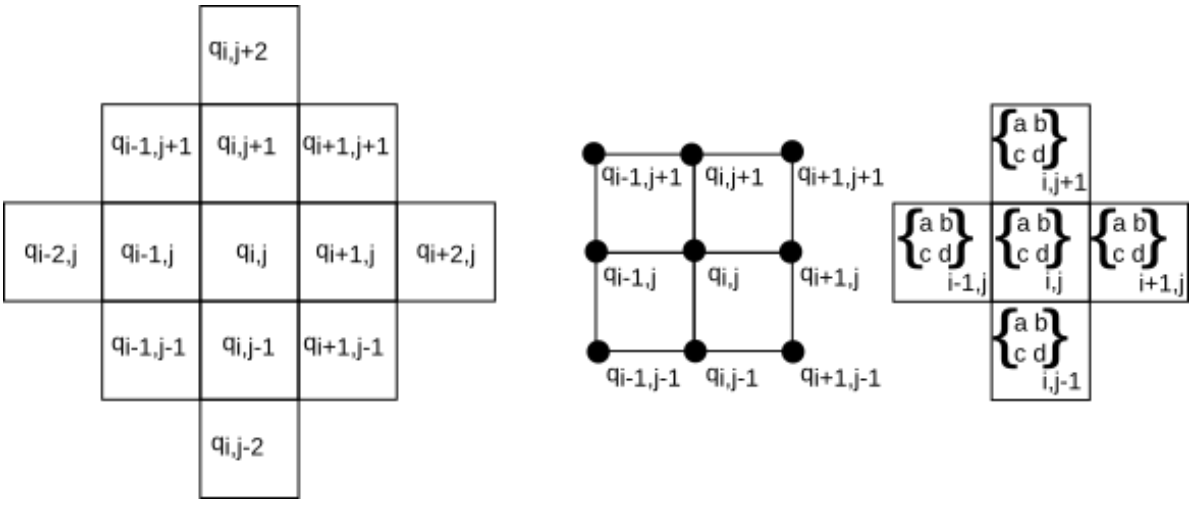


Figure 4.2: Illustration of method stencils: Stencils showing the DOFs around cell or node i,j that appear in the global system row(s) associated with DOFs at cell/node i,j for (left to right) second order cell centered finite volume schemes, SUPG/stabilized FEMs, and the DG FEM.

system will contain 13 nonzero entries and the 4 rows associated with the weight functions that are nonzero in element i, j for a DG method will contain 20 nonzero entries each. The number of nonzero entries that need to be handled on each row of the global system obviously impacts the speed with which the global system can be solved to obtain the solution or next timestep values, giving an advantage to methods that result in fewer of them, if they can obtain similar accuracy at the same grid resolution. Beyond the second order methods illustrated in the figure, FEMs of higher order of accuracy continue to include only DOFs from immediately neighboring elements (those sharing a common face or common node), while higher order finite volume methods and finite difference methods continue to draw DOFs from neighbors, neighbors-of-neighbors, etc. into their stencils. The number of DOFs in the neighboring elements does climb as higher order elements are used in FEMs, mitigating some of their advantage here, in terms of nonzero elements per row in the global system, but only requiring information from immediately neighboring elements has further advantages as the algorithms are parallelized and grid partitioning becomes necessary. It could also be noted that SUPG and stabilized methods do not require assembly of numerical fluxes from the interior faces like the DG FEM and finite volume methods, however, this flux evaluation is added to the volume elements and the computation of the weight function modification required for the SUPG and stabilized methods can be costly, possibly costly enough to offset the numerical flux calculations carried out on the interior faces in the DG FEM and finite volume method for many models.

Chapter 5

Software implementation

In order to investigate the performance of the SUPG finite element method and the stabilized method based on an advection-reaction equation, author implemented a semi-discrete finite element method simulation code that uses the implicit Euler, BDF2, or implicit trapezoid time difference for the governing equations' time derivatives. Both Newton's method and pseudotime stepping are available to obtain the change in the nodal solution values over a timestep. When pseudotime stepping is employed, a spatially varying pseudotimestep size is used, with the timestep size being set to result in a specified pseudotime CFL number based on conditions and surrounding mesh size at each node. Once two consecutive residual norms are available, the CFL number is adapted based on the formula,

$$CFL_{m+1} = CFL_m \left(\frac{R_{max(m)}}{R_{max(m-1)}} \right)^f , \tag{5.1}$$

adopted from ref. [51], in which CFL_m is the pseudotime CFL number used on the last iteration, $R_{max(m)}$ is the largest residual norm for the previous iteration, $R_{max(m-1)}$ is the largest residual norm from the iteration before that, and f is a factor that can vary depending on whether the residual norms are increasing or decreasing and is set in the input to control how aggressively the pseudotimestep size is increased when the solution appears to be converging and decreased when the solution appears to be diverging. The "largest norm" for the iteration arises because there are multiple governing equations in the model. The residuals of the global system of equations for a multi-equation model can be thought of as doubly indexed as expressed in eq. 4.95, where one index (n) specifies the node that the weight function is associated with, while the other (i) specifies

which row of the weight matrix for the system of governing equations was used to compute it. In this work, the norms are L^2 norms across all of the nodes for each row of the weight matrix, excluding those nodes and equations for which Dirichlet boundary conditions are specified. The largest of those is what is used in eq. 5.1.

The code was implemented in the C programming language and the Open MPI [52] library was used to implement the inter-process communication necessary for parallel execution of simulations. It starts by reading a fixed set of fixed format text input files that specify the way the grid should be partitioned, the grid file name, the model to be solved (Euler and PHM models were also developed within the solver), the time difference to be used, the model properties and parameters, the boundary conditions and initial condition, whether or not to use the discontinuity capturing term, linear solver parameters and convergence criteria, and the output frequency. The discontinuity capturing term's parameters, such as the switch width and threshold, are set at compile time. The initial and boundary condition files support the specification of constant value initial/boundary conditions or solely interior conditions dependent boundary conditions over different portions of the domain/boundary that are tagged in the grid file. Complex spatially varying initial conditions and spatially and temporally varying boundary conditions must be defined at compile time, they can then be used through the usual input files by specifying the appropriate variable initial/boundary condition type and the index of the desired function of spatial coordinate and time. The solver uses grids supplied in ASCII formats produced by the open source meshing software, GMSH [53], and produces output files in the ASCII vtk format [54]. Since the vtk format begins with the grid information for the entire grid, and this is lost once the grid is partitioned, files containing only the primary variable values at each node are output throughout the simulation, then once the simulation is complete, the master process re-reads the grid file and cycles through the nodal data files writing the corresponding vtk files. Thus, the partitioning and file output steps remain serialized, while the computation, which is the vast majority of the work, is parallelized. Given that most problems of interest for this model are transient, for grid sizes practical with the computational resources available for this work, these serialized operations are typically a negligible fraction of the compute time. Flow charts describing the code's operation are presented in figs. 5.1- 5.3. Fig. 5.1 is the overall program flow chart, fig. 5.2 then describes the "Solve steady state problem or time march transient problem" process seen in fig. 5.1 in more detail. Finally, fig. 5.3 describes

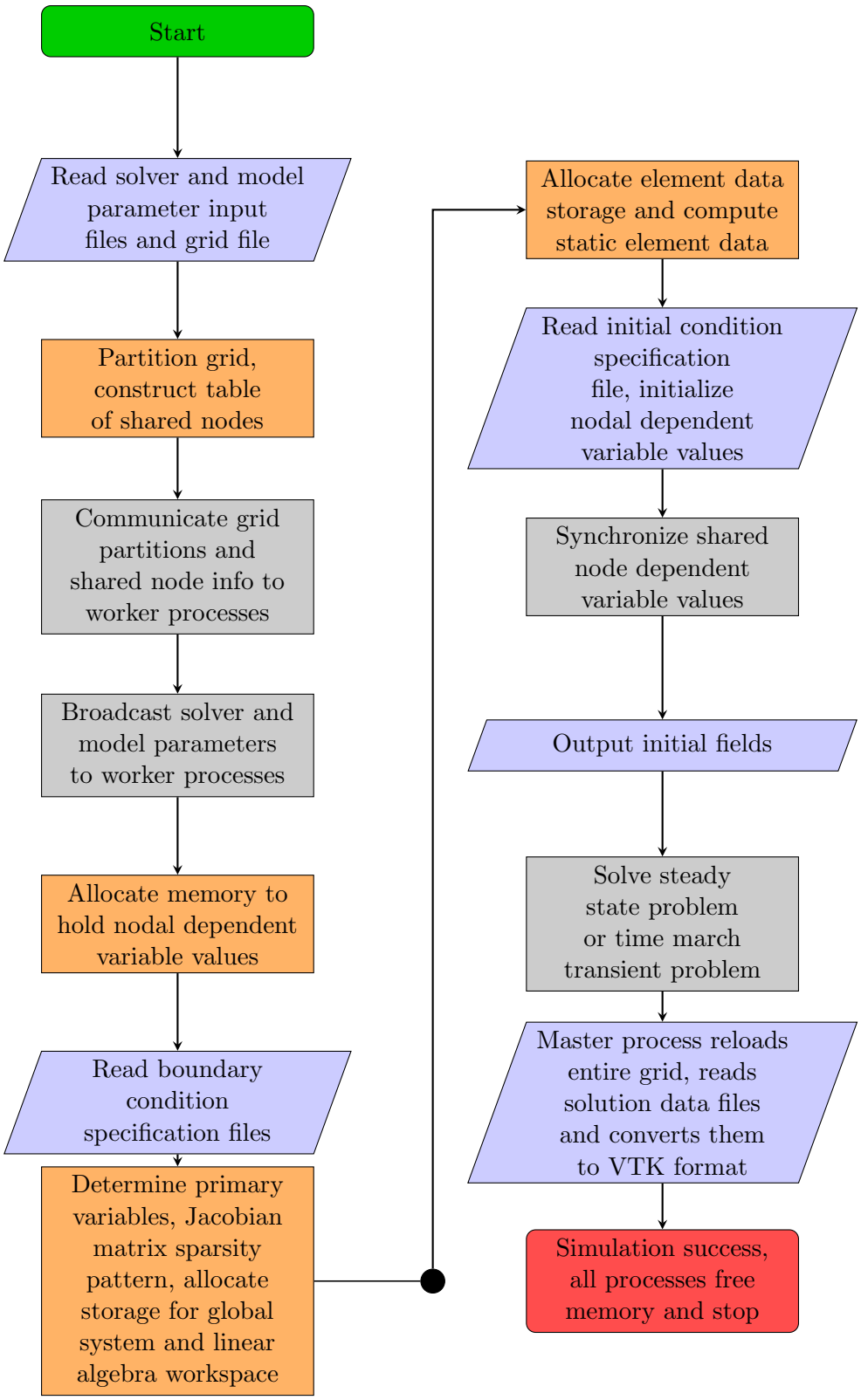


Figure 5.1: An overall program flow chart for the FEM simulation code developed for this work. Fig. 5.2 further details the "Solve steady state problem or time march transient problem" process.

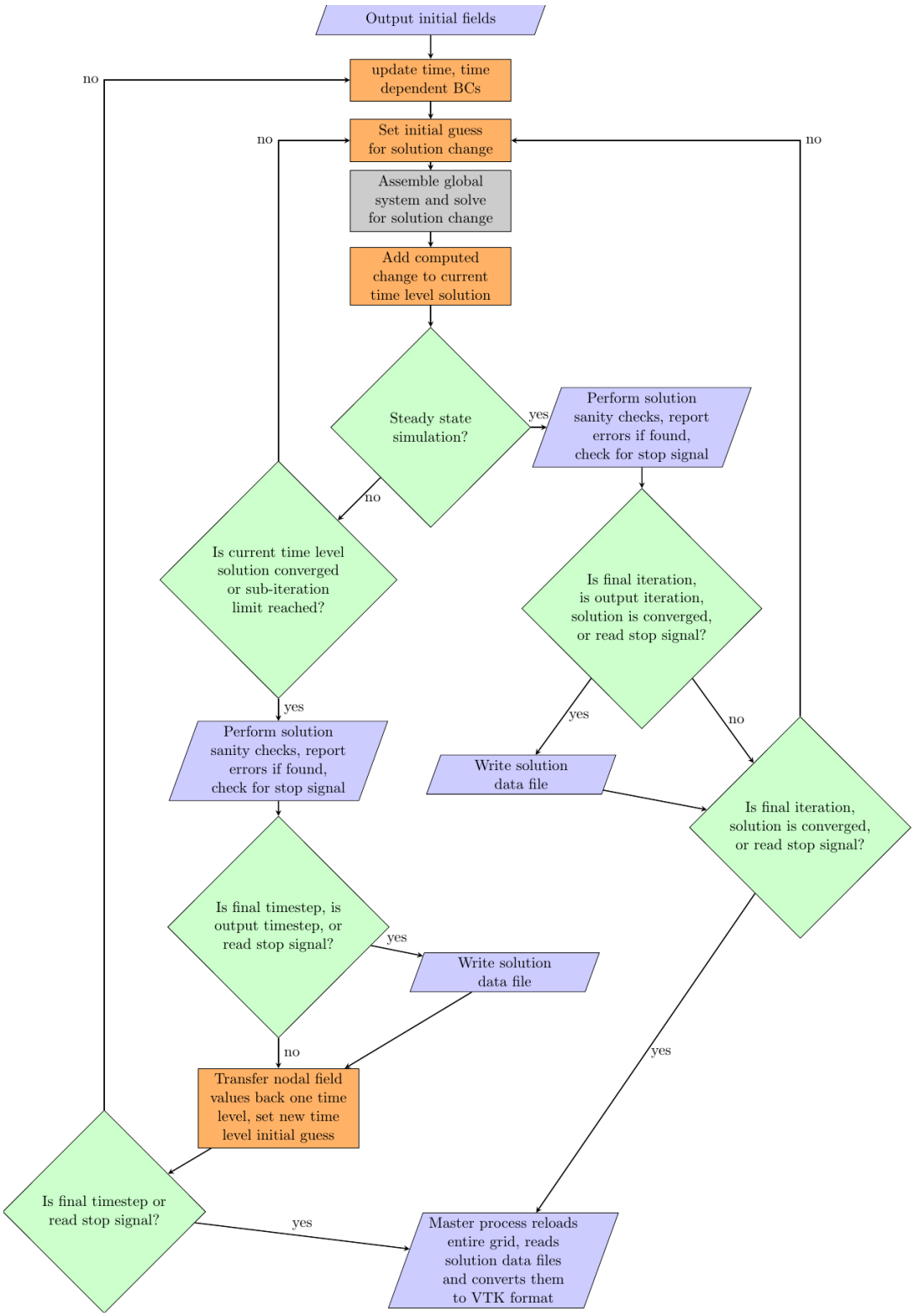


Figure 5.2: A flow chart detailing the simulation solution/ time marching process of the developed code. The process labelled "Assemble global system and solve for solution change" is further detailed in fig. 5.3.

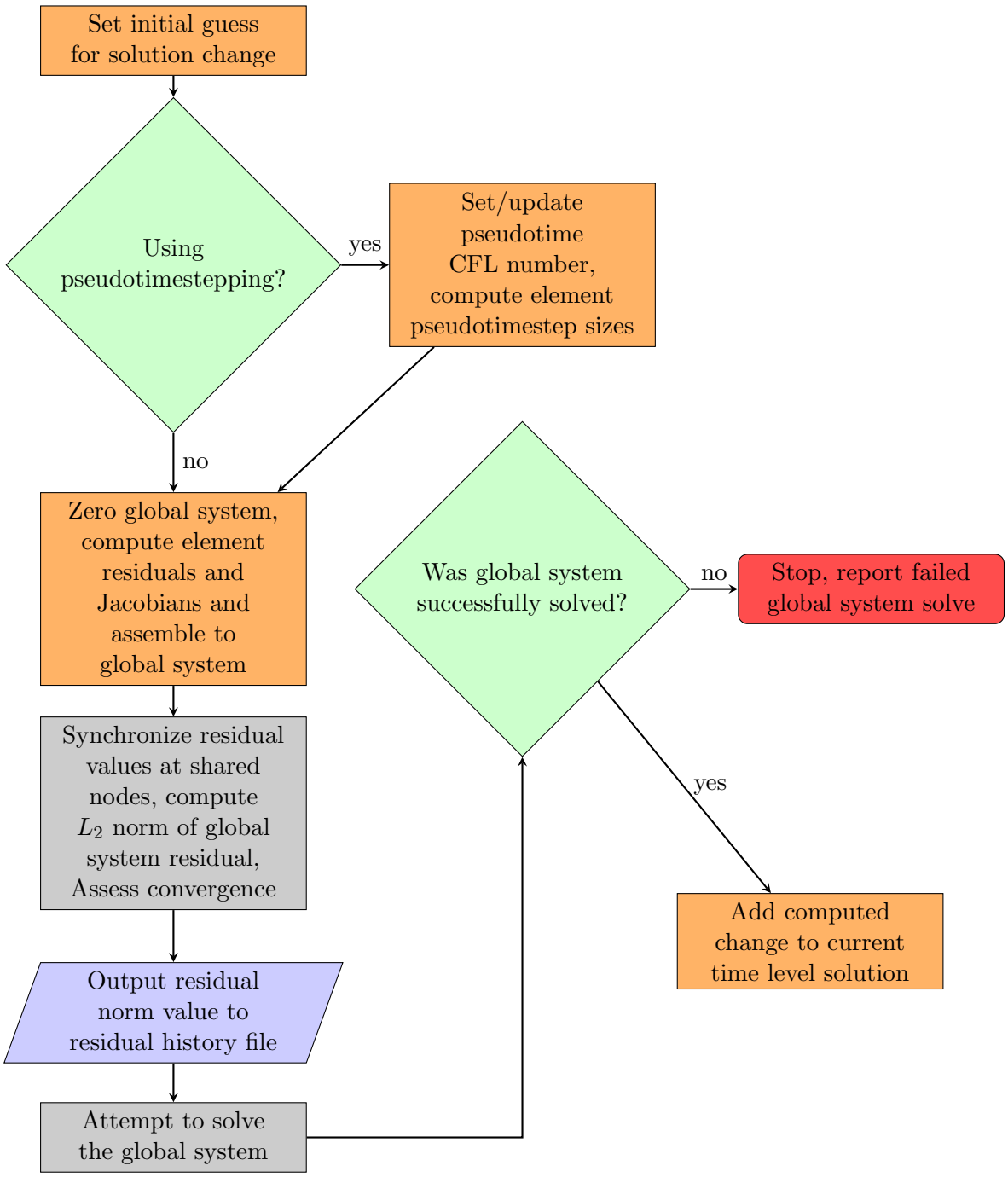


Figure 5.3: A flow chart detailing the process of assembling and solving the global system to obtain the model field updates.

the "Assemble global system and solve for solution change" process seen in fig. 5.1 in more detail. In these figures, the green and red blocks indicate the program entry and exit points, respectively, while orange boxes are processes, gray boxes are processes requiring communication, and purple boxes are processes that perform input and/or output. The green diamonds are decision points.

Domain decomposition is used to parallelize the code and the grid partitions assigned to each process are constructed to include all of the elements surrounding the nodes assigned to the process so that each process can fully assemble the rows of the global system left-hand side associated with the degrees of freedom at the nodes assigned to it without needing to communicate with other processes. These grid partitions are obtained using a striping grid partitioner that proceeds along any of the three cartesian axes as follows:

1. Determine the maximum and minimum coordinate of any node in the striping direction (x_i), x_{imax} and x_{imin}.

2. Form $n \times n_p$ bins associated with equal sized spatial intervals between x_{imin} and x_{imax}, where n_p is the number of partitions desired in direction, x_i, and n is a parameter greater than one. (In practice, $n = 20$ has been used.)

3. Place all of the points in the bins based on their x_i coordinate.

4. Assign bins to a partition until it has approximately N/n_p points, where N is the total number of points in the mesh, then proceed to the next partition, continue until all points are assigned. These will be the nodes for which the process assigned the partition will "own" the degrees of freedom and will fully assemble the rows of the global system associated with the weights that correspond to their basis functions.

5. Add all elements that contain at least one of the owned nodes to each partition grid. Elements along a partition boundary will be duplicated in adjacent partitions, allowing each process to form the full row the global system for weights corresponding to its owned nodes without communication.

6. Add any nodes of elements in the partition that are not already in the partition. The process that owns these nodes must be stored and this information must be made available to the

process that receives the partition, and the process that owns the duplicated nodes because information associated with these nodes will need to be communicated as the global system is solved and the update is determined.

The striping partitioner routine was made compatible with previously partitioned meshes, so that it can be used recursively to obtain checkerboard partitioning. The refined binning step slightly improves the balance of the number of nodes assigned to each partition in meshes with local refinement versus simply dividing the interval into n_p bins and assigning one bin to each partition.

The global matrix setup routine was designed to omit weights and equations associated with nodes for which Dirichlet boundary conditions are specified, but in practice it was found that these boundary conditions are seldom used with multi fluid/PHM plasma models. Specified primary variable boundary conditions are instead usually enforced through a Riemann flux computed with the specified primary variable values as the boundary values.

Communication of the residuals at nodes for which a process does not have all of the surrounding elements (nodes not owned by the process) is necessary, as is communication of each process's share of the residual norms so that the complete norm can be computed, reported, and used in pseudotime CFL number adaptation computations if pseudotimestepping is being used. Most of the inter-process communication is required within the custom developed parallel GMRES [55] solver used to solve the global system.

The software implementation was built around the SUPG method originally, and the stabilized method based on the steady state advection-reaction equation was later implemented into it. The $\boldsymbol{\tau}$ definitions of eqs. 4.74 and 4.75 fit nicely into the SUPG code since the term containing $\boldsymbol{\tau}_2$ is analogous to the SUPG weight function modification and $\boldsymbol{\tau}_1$ is simply a zero matrix.

The initial implementation utilized a loosely coupled formulation in which each fluid species' equations and the PHM equations were assembled into separate global systems that could be solved independently. This neglects the dependence of the residuals associated with one set of governing equations on the primary variables of the other governing equations. For example, the dependence of the current term in the Ampère-Maxwell law on the electron and ion momenta used to calculate it would not be represented in the left hand side matrix of the global system resulting from the PHM equations because $\boldsymbol{D}$, $\boldsymbol{B}$, ϕ, and ψ are the primary variables of the PHM equations as

used in this work, while the fluid species momenta are primary variables of their respective fluid equations. It was thought that this approach would result in less computational expense due to the dropped off-diagonal left-hand side entries. The approach was abandoned for a fully coupled approach resulting in a single global system in hopes of alleviating the lack of robustness found when simulating the two fluid Brio-Wu problem with higher charge to mass ratios and when simulating the GEM challenge problem without discontinuity capturing terms. Ultimately, the fully coupled approach did not help with those issues, but there was not a notable increase in computational cost when switching to the fully coupled solution strategy because the added work of handling the additional off diagonal entries was largely offset by quicker convergence of the iterative methods used to determine the solution update for each timestep.

The initial implementation also looped over weight functions (weight matrices for systems of equations) during residual computation, and over weight functions, differentiating basis functions, and differentiating variable outside the integration point loop for each element. This strategy had come about due to viewing the residuals to be computed on a weight function by weight function basis, rather than the complete element matrices approach later introduced. The downside of this is that the basis function gradients, fluxes, flux Jacobians, SUPG weight function modification, and discontinuity capturing viscosity either have to be recomputed every time an integration point is visited under the different weight matrices, or they have to be stored so that they can be reused by residual/Jacobian computations occuring later at the same integration point. With a system the size of the two fluid - PHM model, this created a lot of redundant computations. Storage for these data was eventually added to the code's element data storage data structures and a loop through the elements was performed prior to assembly to compute them so that they could be used repeatedly without recomputation. This greatly improved the code's speed, but added considerable storage costs. The ultimate solution was to re-implement the assembly routine to place all of the element residual and Jacobian matrix entries at once and implement a single element residual and Jacobian computation routine so that all of the element matrix entries can be computed inside a single loop through the integration points. With the final approach implemented in the code, only one or two loops through the elements are needed per iteration, the additional loop is required to set the pseudotime step size field when using constant CFL number pseudotime stepping as the iterative solution method. This is because the pseudotime step size is set nodally based on conditions in all

of the elements surrounding a node and the lengths of the edges surrounding it, and neighboring element information is not available within the element matrix assembly routines. Also, when using pseudotime stepping, it was possible to adapt the pseudotime CFL number used based on the current timestep's residual when using the separated residual and Jacobian assembly, since an implicit Euler pseudotime difference was employed, resulting only in additional terms in the left hand side of the global system, and the residual was assembled first. With the later simultaneous assembly method, pseudotime CFL adaptation is now performed based on the previous iteration's residual.

Chapter 6

Test and validation problems

Three test and validation problems were used to establish that the numerical algorithms described in this work have been properly implemented and to characterize their performance. First, the method of manufactured solutions [56] is used to verify that the implemented algorithm converges to a known solution with nontrivial spatial, and possibly temporal, variation at an appropriate rate. The Brio-Wu shock tube problem for the two fluid model [4, 5, 9, 11], then provides a one dimensional, transient test case for which there are comparable results in the literature. Finally, the Geospace Environment Modeling (GEM) challenge problem [4,5,12,57] provides a two dimensional, transient test case with many comparable results in the literature. These validation techniques and test problems are described in more detail in the remainder of this chapter.

6.1 The method of manufactured solutions

The method of manufactured solutions, described in ref. [56], provides a means to test an implementation of a numerical algorithm against a known, exact solution with nontrivial spatial and temporal variation. This is achieved by simply specifying the solution fields as nontrivial functions of space (and time, if desired) then plugging those functions into the governing equations. It is unlikely that the governing equations will balance for these arbitrarily specified fields, but this is not an issue, the imbalance can be added to the governing equations as a spatially (and possibly temporally) varying source term that will cause the equations to be satisfied by the specified fields. As an example, consider the transient linear advection-reaction equation augmented with spatially and

temporally varying source term $s_m(x,t)$,

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = sq + s_m(x,t), \tag{6.1}$$

and the specified field,

$$q(x,t) = c_1 + c_2\cos(x) + c_3\sin(t), \tag{6.2}$$

in which the c_i's are arbitrary constants. If this specified field function and its partial derivatives with respect to x and t are substituted into the governing equation (eq. 6.1) and the result is solved for $s_m(x,t)$, it is found that

$$s_m(x,t) = c_3(\cos(t) - \sin(t)) - c_2(a\sin(x) + s\cos(x)) - c_1 s \tag{6.3}$$

is the spatially and temporally varying source term necessary to make eq. 6.2 the solution of eq. 6.1.

Thanks to the analytical solution being available, it is possible to obtain accurate calculations of the error present in the numerical solution and observe whether or not the numerical solution is converging to the analytical solution at the expected rate. In this work, this is done by integrating the absolute value of the difference between the numerical solution and analytical solution over the domain of the simulation (and time interval, in the case of transient simulations). For the one dimensional, transient governing equation just discussed, simulated on the domain $x_1 \leq x \leq x_2$, for $0 \leq t \leq t_f$, the formula would be

$$E_{num} = \int_0^{t_f}\int_{x_1}^{x_2} |q_{numerical}(x,t) - q(x,t)|\,\mathrm{d}x\mathrm{d}t. \tag{6.4}$$

In the code developed for this work, this is evaluated via numerical integration using Gaussian quadrature within the elements of mesh the numerical solution was computed on, with enough integration points to exactly integrate a polynomial of at least one higher order than the method's shape functions, to ensure that the error of the numerical integration is much smaller than the error being integrated. For transient manufactured solutions, the elements are extruded into space time elements with the numerical solution at the nodes being interpolated to the integration point time

via linear Lagrange interpolation functions for solutions obtained using implicit Euler and implicit trapezoid time differences, and using the implied quadratic variation for solutions obtained using BDF2. Once the all of the nodal values are interpolated to the integration point time level, the spatial basis functions are used as usual to interpolate to the spatial coordinates of the integration points.

To compute the observed order of accuracy of the scheme, at least two solutions must be computed, with one being computed on a grid uniformly refined relative to the other (both spatially and temporally, if a transient simulation). If the element and timestep size of the refined grid is $\frac{1}{r}$ times that of the coarse grid, then the observed order of accuracy is [56]

$$p \approx \log_r \left(\frac{E_{numc}}{E_{numr}} \right), \tag{6.5}$$

where E_{numc} and E_{numr} are the error integrals for the coarse and refined simulations, respectively.

The choice of the specified solution is fairly free, but a few points should be considered when choosing it. [56] One is that the specified solution should vary in all of the spatial dimensions present in the model being validated and in time for transient models, and its variation should be complex enough that the numerical methods being used can not solve it exactly. The sine and cosine functions are good for this purpose; it is simple to compute their derivatives when deriving the source terms, but they also do not run out of nonzero derivatives. Another is that, for multi field models, the variation should be different for each field. If two fields are specified to have exactly the same solution, a coding error where the one field is used in place of the other would not be detected. It is also necessary to ensure that the specified solution does not become so nonphysical that it causes issues for the physical model. For example, specifying a density or pressure solution that becomes negative could cause the speed of sound to become imaginary, this would likely stop most numerical algorithms trying to solve the compressible Euler equations. In this work, manufactured solution simulations have only been pursued with Dirichlet boundary conditions or Riemann flux based boundary conditions where the boundary value is set to the exact solution, however, it is possible to derive manufactured solutions that are compatible with other boundary conditions so that the implementation of those boundary conditions can also be tested. [56]

6.2 Scaling and nondimensionalization of the ideal two fluid model

Before discussing the final two test and validation problems, a discussion of the nondimensionalization of the ideal two fluid model is necessary because it plays an important part in their formulation and interpretation. The formulation of a nondimensional form of a set of governing equations can provide important insights into the conditions when various terms in the equations are important or so insignificant that they can safely be neglected. For example, in ref. [4], the nondimensionalizations employed cause ratios of reference conditions plasma parameters like the Larmor radius to reference length ratio and the Debye length to Larmor radius ratio to appear in the nondimensionalized equations. The presence of these ratios as scaling factors on certain terms of the equations makes it possible to determine whether or not those terms are likely to be important in modeling the scenario at hand, enabling one to identify conditions where simplifications are possible, or where more detailed models will be necessary. It can also be beneficial to operate on nondimensionalized problems numerically, as an appropriate rescaling can make the relative magnitude of the residuals and Jacobian entries that end up in the global system more similar, potentially removing some avoidable roundoff error and making convergence easier to judge. It also helps with models that have tuneable parameters like the threshold of a shock sensor or a threshold for treating a small singular value as zero when computing a pseudoinverse, since these don't have to be set for multiple unit systems. As a result of this, the use of nondimensional governing equations in numerical solvers is common and many test problems are described in nondimensional terms. Unfortunately, this can result in some non-uniformity in the presentation of comparable work in the literature. For example, in ref. [9], the ideal two fluid model is presented without nondimensional scaling groups as in eqs. 3.2-3.4 and eqs. 3.12-3.15, but in ref. [5], the equations are written with nondimensional scaling groups containing the ratio of the reference conditions Debye length and Larmor radius to the reference length and the ion-electron mass ratio in the source terms. Additionally, in ref. [9] and previous publications associated with this work ([6, 46, 58]) variation of the strength of the source terms relative to the convective terms in two fluid Brio-Wu shock tube problems is described as variation of the ion charge to mass ratio whereas in refs. [4, 5, 10] it is described as variation of the nondimensional Larmor radius. Ultimately, some consideration of the way the the equations have been nondimensionalized is necessary to determine which cases are comparable or how the

initial/boundary conditions and properties must be scaled to produce the same scenario under a different nondimensionalization.

Not considering the source terms, the remainder of the fluid dynamics equations for a species in an ideal two fluid model are simply the Euler equations. The nondimensional form of these equations will not contain any nondimensional groups of reference quantities if the reference pressure, time, and specific energy are chosen so based on the formulas,

$$P_r = \rho_r {u_r}^2, \tag{6.6}$$

$$t_r = l_r / u_r, \tag{6.7}$$

and

$$e_r = {u_r}^2 \tag{6.8}$$

once an appropriate reference length, l_r, density, ρ_r, and velocity, u_r, have been chosen. [13] What satisfying these equations does is make each the nondimensional scaling groups appearing in the nondimensional Euler equations equal one. As can be seen, since only three equations involving six reference quantities are necessary to remove the nondimensional scaling groups from that portion of the governing equations, there remains some latitude in the selection of the reference quantities. In this case, three of the six reference quantities appearing in eqs. 6.6-6.8 can be chosen to give the nondimensionalized fields of the scenario at hand similar magnitudes or to give the nondimensionalized fields and/or equations additional nice properties. For example, in compressible aerodynamics computations, the free stream sound speed is usually used as the reference velocity so that the nondimensional velocity at the boundaries is just the Mach number of the flow. [13] The free stream ion sound speed is a common choice in ideal MHD simulations, as well. [59]

Now considering the source terms of the speccies fluid equations, the scaling factors on the additional Lorentz force and electric field work terms appearing in the ideal two fluid species equations can also be cancelled out to ones if the reference charge, q_r, particle mass, m_r, electric field strength, E_r, and magnetic flux density, B_r, are chosen to satisfy

$$E_r = \frac{m_r u_r^2}{l_r q_r} \tag{6.9}$$

and

$$B_r = \frac{E_r}{u_r}. \tag{6.10}$$

Satisfying these constraints on the reference quantities in addition to those of eqs. 6.6-6.8 will reduce all of the nondimensional scaling groups in the nondimensionalized species fluid equations to ones.

When the PHM equations are used with vacuum permittivity and permeability are used as the electrodynamics model, a number of additional relationships amongst the reference quantities must also be maintained to cancel additional nondimensional scaling groups present in those equations. The relations

$$D_r = \epsilon_r E_r \tag{6.11}$$

and

$$B_r = \mu_r H_r \tag{6.12}$$

are an obvious starting point, necessary to eliminate the scaling groups from the electric field strength - electric displacement and magnetic field strength - magnetic field flux relationships. The relations

$$\rho_{cr} = \frac{q_r}{m_r}\rho_r \tag{6.13}$$

and

$$J_r = \frac{q_r}{m_r}\rho_r u_r \tag{6.14}$$

relating the reference space charge and reference current to the reference particle charge and mass, fluid density, and velocity are also quite rational choices and eliminate the scaling groups in the equations necessary to compute the current and space charge from the plasma species properties. This leaves the following relations to be satisfied by the reference quantities to eliminate all of the

nondimensional scaling groups from the ideal two fluid - PHM model:

$$\frac{1}{\mu_r \epsilon_r} = u_r^2 \tag{6.15}$$

$$\psi_r = D_r \tag{6.16}$$

$$\phi_r = B_r \tag{6.17}$$

$$\kappa_r = u_r \tag{6.18}$$

$$\chi_r = u_r \tag{6.19}$$

$$\epsilon_r = \rho_r \left(\frac{q_r l_r}{m_r u_r} \right). \tag{6.20}$$

These relationships are only one expression of the relationships necessary for a nondimensional scaling groups free statement of the model. It is possible to assemble other equivalent sets of relationships that achieve the same result, such as those in refs. [58] and [8]. The constraints expressed in eqs. 6.6-6.8, 6.10, 6.11, 6.12, and 6.15 - 6.19 are also necessary for a nondimensionalization with the source terms scaled by factors containing the nondimensional Larmor radius and Debye length, like the one used in refs. [5, 10]. The reference quantities of that nondimensionalization must additionally satisfy

$$\epsilon_r = \epsilon_0, \tag{6.21}$$

$$m_r = m_i, \tag{6.22}$$

$$\rho_r = \rho_i, \tag{6.23}$$

$$q_r = q_e, \tag{6.24}$$

$$n_r = \frac{\rho_r}{m_r}, \tag{6.25}$$

and

$$T_r = \frac{P_r}{n_r k_B}, \tag{6.26}$$

so that the nondimensionalized governing equations of the ideal two fluid models are

$$\frac{\partial}{\partial t} \rho_s + \nabla \cdot (\rho_s \boldsymbol{u}_s) = 0, \tag{6.27}$$

$$\frac{\partial}{\partial t}\left(\rho_s \boldsymbol{u}_s\right) + \nabla \cdot \left(\rho_s \boldsymbol{u}_s \boldsymbol{u}_s\right) + \nabla P_s = \frac{1}{\hat{r}_g}\frac{q_s}{m_s}\rho_s\left(\boldsymbol{E} + \boldsymbol{u}_s \times \boldsymbol{B}\right), \tag{6.28}$$

and

$$\frac{\partial}{\partial t}\left(\rho_s e_{ts}\right) + \nabla \cdot \left(\left(\rho_s e_{ts} + P_s\right)\boldsymbol{u}_s\right) = \frac{1}{\hat{r}_g}\frac{q_s}{m_s}\rho_s \boldsymbol{u}_s \cdot \boldsymbol{E} \tag{6.29}$$

for species s with PHM equations

$$\frac{\partial}{\partial t}\boldsymbol{D} - \nabla \times \boldsymbol{H} + \chi \nabla \psi = -\frac{1}{\hat{r}_D^2 \hat{r}_g}\boldsymbol{J}, \tag{6.30}$$

$$\frac{\partial}{\partial t}\boldsymbol{B} + \nabla \times \boldsymbol{E} + \kappa \nabla \phi = \boldsymbol{0}, \tag{6.31}$$

$$\frac{\partial}{\partial t}\psi + \chi \nabla \cdot \boldsymbol{D} = \frac{1}{\hat{r}_D^2 \hat{r}_g}\chi \rho_c, \tag{6.32}$$

and

$$\frac{\partial}{\partial t}\phi + \kappa \nabla \cdot \boldsymbol{B} = 0. \tag{6.33}$$

In these equations, the nondimensional Larmor radius is the reference conditions ion Larmor radius, computed taking the reference velocity as the thermal velocity and reference charge as the particle charge, divided by the reference length;

$$\hat{r}_g = \frac{m_r u_r}{q_r B_r l_r}, \tag{6.34}$$

and the nondimensional Debye length is the reference conditions Debye length divided by the reference conditions Larmor radius;

$$\hat{r}_D^2 = \frac{\epsilon_0 k_B T_r}{q_r^2 n_r} \times \left(\frac{q_r B_r}{m_r u_r}\right)^2. \tag{6.35}$$

Ultimately, it is not necessary to assign a real, dimensional equivalent problem when translating a nondimensionalized problem to a different nondimensionalization, but the fact that both should describe the same real problem can be used to translate between the nondimensionalizations using relations like

$$q_r \hat{q} = q_R \bar{q}, \tag{6.36}$$

where q_r and q_R are different reference quantities for the field or property q and $\hat{q}$ and $\bar{q}$ are its nondimensional values under different two different nondimensionalizations.

6.3 The Brio-Wu shock tube problem

The Brio-Wu shock tube problem is a one dimensional Riemann problem for the ideal magnetohydrodynamics (MHD) equations originally stated in ref. [11] where it was used as a test case for an upwind finite difference scheme that the authors had developed for the ideal MHD equations. It has since become a standard test case for numerical algorithms for plasma models, not only for the ideal MHD model, but also for algorithms working with more detailed plasma models, where it can be desirable to demonstrate that an algorithm can reproduce a known result, even if it is for a scenario that does not exercise all of its terms. There is an issue with the test case for models retaining electron inertia and the full Maxwell's equations, like the two fluid plasma model used in this work, however. For these models, electromagnetic wave propagation is possible and plasma oscillation occurring at frequencies and length scales comparable to the electron plasma frequency and Debye length scale are possible and typically must be resolved to obtain an accurate solution. But under the ideal MHD assumptions of massless electrons, quasineutrality, and infinite speed of light, the Debye length of the plasma is zero while the electron plasma frequency is infinite and electromagnetic wave propagation is not possible. [4] The consequence of this that the grid spacing and timestep size necessary to obtain an accurate two fluid plasma simulation both go to zero in conditions approaching those where the MHD assumptions would be valid, leading to an explosion of computational expense. Because of this, it is very difficult to obtain MHD like behavior with the ideal two fluid plasma model, but there are multiple examples of simulations of the two fluid analog of the Brio-Wu shock tube problem in the literature with which results can be compared to assess the performance of the algorithms employed in this work. [4, 5, 9]

6.3.1 The Brio-Wu shock tube problem for the two fluid model

The original statement of the Brio-Wu shock tube problem amounts to the Sod shock tube problem from gas dynamics [60] in the presence of a magnetic field which has a constant component in the direction along which conditions vary and a transverse component with a jump at the diaphragm

location. (The same location as the jumps in the fluid variable fields.) The originators employed a specific heat ratio of $\gamma = 2$ in their simulations, because the Roe matrix for the ideal MHD equations is the Jacobian of an appropriately averaged state for that specific heat ratio, and ran the simulation on the domain, $0 \leq x \leq 800$, to final time, $t_f = 80$. The initial conditions to the left and right of the initial discontinuity at $x = 400$ were

$$Q_L = \begin{bmatrix} \rho \\ u_x \\ u_y \\ u_z \\ P \\ B_x \\ B_y \\ B_z \end{bmatrix}_L = \begin{bmatrix} 1.0 \\ 0 \\ 0 \\ 0 \\ 1.0 \\ 0.75 \\ 1.0 \\ 0 \end{bmatrix} \tag{6.37}$$

and

$$Q_R = \begin{bmatrix} \rho \\ u_x \\ u_y \\ u_z \\ P \\ B_x \\ B_y \\ B_z \end{bmatrix}_R = \begin{bmatrix} 0.125 \\ 0 \\ 0 \\ 0 \\ 0.1 \\ 0.75 \\ -1.0 \\ 0 \end{bmatrix}. \tag{6.38}$$

It is necessary to determine the relationships between the reference quantities employed in different nondimensionalized model/problem statements in order to formulate equivalent problems and compare solutions. No references to reference quantities or dimensions are made in the description of the ideal MHD equations or the Brio-Wu shock tube problem in ref. [11]. They are simply stated as a nondimensional set of governing equations and problem containing no nondimensional scaling groups. This kind of statement can be achieved with both the ideal MHD equations and the ideal two fluid plasma model by choosing reference quantities that make all of the nondimensional scaling

groups equal to one. For the ideal MHD equations, this requires the use of reference quantities that satisfy

$$t_r = \frac{l_r}{u_r} \tag{6.39}$$

and

$$P_r = \rho_r u_r^2 = \frac{B_r^2}{\mu_0} = \mathcal{E}_r, \tag{6.40}$$

where the t_r, l_r, u_r, P_r, ρ_r, B_r, $\mathcal{E}_r$, are the reference time, length, velocity, pressure, mass density, magnetic flux density, and energy per unit volume, respectively, and μ_0 is the permeability of free space. The relationships between the reference quantities that are necessary for the ideal two fluid model are discussed in section. 6.2. From the problem specification, it is clear that the reference mass density used is that of the left initial state and that the reference length used is the cell width. From eqs. 6.39 and 6.40 it can be determined that the reference length and time can be changed by the same factor without causing a need to adjust any other reference quantities, thus if the domain length is taken to be the reference length, rather than the cell width, the final simulation time can simply be multiplied by the same factor to obtain a solution that is comparable. That is, the solution at $t = 0.1$ on $0 \leq x \leq 1$ should be identical to to solution at $t = 80$ on $0 \leq x \leq 800$ except that the x axis is labelled differently. This makes it possible to translate the original problem statement to the unit domain more commonly seen in later work, including the ideal two fluid simulations of refs. [4,5,9]. Constructing the equivalent two fluid problem begins with assuming that the two fluid plasma is quasineutral and in thermal equilibrium in its initial condition, as an ideal MHD plasma would be. These conditions dictate that the ion and electron number densities be equivalent and their partial pressures be equivalent. Together with the previously described relationships required for a nondimensional scaling group free statement of the ideal two fluid model (sec. 6.2) and the selection of the speed of light as the reference velocity for the two fluid model, this necessitates the following relationships between the nondimensional two fluid density, pressure, magnetic flux density, domain length, and simulation time interval and the nondimensional MHD analogs for the

Brio-Wu problem:

$$\rho = \hat{\rho}, \tag{6.41}$$

$$P = \frac{{u_r}^2}{{c_0}^2}\hat{P}, \tag{6.42}$$

$$B = \frac{u_r}{c_0}\hat{B}, \tag{6.43}$$

$$x = \frac{t_r}{t_{r2}}\frac{u_r}{c_0}\hat{x}, \tag{6.44}$$

and

$$t = \frac{t_r}{t_{r2}}\hat{t}, \tag{6.45}$$

with the MHD variables denoted with hats and the two fluid reference quantities having the subscript $r2$ while the MHD reference quantities have the subscript r. Thus, the initial conditions

for the two fluid analog of the Brio-Wu shock tube problem are

$$\begin{bmatrix} \rho_i \\ u_{i_x} \\ u_{i_y} \\ u_{i_z} \\ P_i \\ \rho_e \\ u_{e_x} \\ u_{e_y} \\ u_{e_z} \\ P_e \\ D_x \\ D_y \\ D_z \\ B_x \\ B_y \\ B_z \\ \phi \\ \psi \end{bmatrix}_L = \begin{bmatrix} \frac{1}{1+\frac{m_e}{m_i}} \\ 0 \\ 0 \\ 0 \\ 0.5\frac{{u_r}^2}{{c_0}^2} \\ \frac{1}{1+\frac{m_i}{m_e}} \\ 0 \\ 0 \\ 0 \\ 0.5\frac{{u_r}^2}{{c_0}^2} \\ 0 \\ 0 \\ 0 \\ 0.75\frac{u_r}{c_0} \\ 1.0\frac{u_r}{c_0} \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} \rho_i \\ u_{i_x} \\ u_{i_y} \\ u_{i_z} \\ P_i \\ \rho_e \\ u_{e_x} \\ u_{e_y} \\ u_{e_z} \\ P_e \\ D_x \\ D_y \\ D_z \\ B_x \\ B_y \\ B_z \\ \phi \\ \psi \end{bmatrix}_R = \begin{bmatrix} \frac{0.125}{1+\frac{m_e}{m_i}} \\ 0 \\ 0 \\ 0 \\ 0.05\frac{{u_r}^2}{{c_0}^2} \\ \frac{0.125}{1+\frac{m_i}{m_e}} \\ 0 \\ 0 \\ 0 \\ 0.05\frac{{u_r}^2}{{c_0}^2} \\ 0 \\ 0 \\ 0 \\ 0.75\frac{u_r}{c_0} \\ -1.0\frac{u_r}{c_0} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.46}$$

to the left and right of the initial discontinuity at the center of the domain, respectively. Because of eq. 6.44, it is necessary to choose $t_{r2} = \frac{u_r}{c_0} t_r$ when the same reference length is used for both nondimensionalizations, meaning that the final simulation time for the two fluid problem will be $\frac{c_0}{u_r}$ times the final simulation time of the ideal MHD problem. The ratios $\frac{u_r}{c_0}$ and $\frac{m_e}{m_i}$ are prevalent in this problem specification and would be expected to be specified along with any two fluid plasma simulation results as they are essential to constructing equivalent simulations. In the MHD limit, both go to zero, thus the smaller these ratios are made, the more MHD-like the two fluid result should become, but the two fluid model will become increasingly numerically stiff in those conditions. Two fluid simulations seen in the literature typically use $\frac{u_r}{c_0}$ ratios of $\frac{1}{100}$ (ref. [5],

for example), while electron-ion mass ratios of $\frac{1}{100}$ and $\frac{1}{1836.2}$ (the electron-proton mass ratio) are common. With these ratios set, the final parameter that must be specified to fix a two fluid Brio-Wu problem is the ion charge-mass ratio, $\frac{q_i}{m_i}$. In ideal MHD conditions, this ratio would be large and examination of the species fluid equations, eqs. 3.2- 3.4, and the electric current density and space charge density, the source terms in the electrodynamics equations, defined by eqs. 3.7 and 3.11, respectively, shows that the source terms become stronger relative to the convective terms as this ratio increases. In simulations employing the nondimensional scaling group free statement of the governing equations, this ratio is typically what is varied to try to show a tendency toward MHD behavior as it increases. [9] Values in the range from 0 to 1000, and equivalents under the nondimensional Larmor radius based nondimensionalization are seen in the literature. [4, 5, 9, 10] With an ion charge-mass ratio of 0, the electron charge-mass ratio must also be 0 for quasineutrality and the species fluid equations and electrodynamics equations are all completely decoupled. The species fluid fields simply behave as those for two different neutral gas Riemann problems with the much lighter electron fluid expanding much further than the ion fluid. The source terms of the electrodynamics equations are also inactive when the species charge-mass ratios are all 0 and the initial discontinuity in the magnetic flux density collapses inducing a jump in the electric field to produce a constant intermediate state. (χ and κ are frequently chosen to be the speed of light in the PHM equations, so there are only two distinct wave speeds, $\pm c_0$.) At $\frac{q_i}{m_i} = 1000$, ideal two fluid mass density solutions bear some resemblance to the ideal MHD solution, but sizeable oscillations are present in the ideal two fluid solutions. For simulations at the intermediate ion charge-mass ratios, the results bear little resemblance to the ideal MHD solution, thus comparison to other two fluid simulation results are the most useful for code validation and performance evaluation. Most references provide only electron and ion mass or number densities at the final time. Ref. [5] additionally provides some species velocities, ref. [8] appears to have the most complete set of two fluid simulation results for this problem; it includes results for all of the MHD variables for various nondimensional Larmor radii and a convergence study of the final time bulk fluid and electrodynamic variables solutions for $\hat{r}_g = \frac{1}{300}$ with $\frac{m_i}{m_e} = 1836$, $\frac{c_0}{u_r} = 100$, and $\hat{r}_D = \frac{1}{100}$.

While the domain of interest of the Brio-Wu shock tube will be the same as that of the ideal MHD problem, it should be noted that a much larger domain must often be simulated with the ideal two fluid model to avoid the possibility of boundary conditions influencing the solution because

of to the potential for electromagnetic disturbances that propagate at the speed of light. (The Brio-Wu shock tube is a Riemann problem concerned only with the evolution of the solution from the initial discontinuity, the boundaries only exist to truncate the domain for computation.) This is particularly necessary at low charge-mass ratios. At higher charge-mass ratios, as MHD behavior is approached, the electromagnetic disturbances should be so heavily damped that significant solution evolution does not occur outside the original MHD problem domain, potentially allowing for the use of a shorter computational domain. The Brio-Wu problem simulations done during this work were actually done on domains extending beyond the reach of any electromagnetic waves leaving the initial discontinuity, or $c_0 t_f$, in each direction.

A slight variation of this problem is also examined in some references, such as ref. [4], in which the magnetic field is zero everywhere in the initial condition. In that reference, it is referred to as the "two fluid electrostatic plasma shock". This is an interesting test case because it removes the influence of the magnetic field of the Brio-Wu problem so that the expanding plasma behavior can be compared to that of an expanding neutral gas and the effects of charge separation can be examined. Since the electron gas is much lighter than the ion gas, it would like to expand much more quickly, but it is coupled to the ion gas by the electromagnetic fields that are generated when it tries to do that. This results in a solution with the rarefaction, contact discontinuity, and shock features typically seen in a neutral gas, but there is also a sharp increase in ion density around the shock and some oscillations behind the tail of the rarefaction due to the effects of the finite Debye length. [4]

6.4 The GEM challenge problem

The Geospace Environment Modeling (GEM) challenge problem was introduced as a computationally tractable magnetic reconnection scenario for assessing the ability of plasma dynamics models and numerical algorithms to appropriately model magnetic reconnection phenomena like what occurs in the Earth's magnetosphere. [12] The problem conditions, including a very low ion-electron mass ratio of 25, make it accessible to full particle simulation, and a variety of models and numerical methods were used to simulate it as part of the original effort. Full particle simulations were among the approaches used [12, 57] and provide reference solutions indicating what should happen

if sufficient detail is retained in a fluid model. Hall MHD simulations were also included in the original effort, including some that included electron mass, resulting in a similar model to the ideal two fluid model used in this work. [12] The most comparable results will be those of refs. [4, 5, 9] which employed the same plasma model, but different numerical methods.

The initial condition is based on a smoothly distributed, dense, current carrying plasma layer supporting a magnetic field reversal in a configuration known as Harris equilibrium. [12] For the nondimensional scaling group free nondimensionalization employed by the solver developed as part of this work, with c_0 chosen as the reference speed, the initial condition is

$$\begin{bmatrix} \rho_i \\ u_{i_x} \\ u_{i_y} \\ u_{i_z} \\ P_i \\ \rho_e \\ u_{e_x} \\ u_{e_y} \\ u_{e_z} \\ P_e \\ D_x \\ D_y \\ D_z \\ B_x \\ B_y \\ B_z \\ \phi \\ \psi \end{bmatrix}_{t=0} = \begin{bmatrix} \operatorname{sech}^2(2z) + \frac{1}{5} \\ 0 \\ \frac{1}{6}\operatorname{sech}^2(2z) / \left(\operatorname{sech}^2(2z) + \frac{1}{5}\right) \\ 0 \\ \frac{1}{240}\left(\operatorname{sech}^2(2z) + \frac{1}{5}\right) \\ \frac{1}{25}\left(\operatorname{sech}^2(2z) + \frac{1}{5}\right) \\ 0 \\ -\frac{1}{30}\operatorname{sech}^2(2z) / \left(\operatorname{sech}^2(2z) + \frac{1}{5}\right) \\ 0 \\ \frac{1}{1200}\left(\operatorname{sech}^2(2z) + \frac{1}{5}\right) \\ 0 \\ 0 \\ 0 \\ \frac{1}{10}\tanh(2z) - \left[\frac{1}{100}\frac{\pi}{12.8}\cos\left(\frac{\pi}{12.8}x\right)\sin\left(\frac{\pi}{12.8}z\right)\right] \\ 0 \\ \left[\frac{1}{100}\frac{\pi}{12.8}\sin\left(\frac{\pi}{12.8}x\right)\cos\left(\frac{\pi}{12.8}z\right)\right] \\ 0 \\ 0 \end{bmatrix}, \tag{6.47}$$

and the simulation is performed on the domain $-12.8 \leq x \leq 12.8$, $-6.4 \leq z \leq 6.4$ from $t = 0$ to $t = 400$. This time interval corresponds to the $0 \leq t \leq 40$ time interval that results were

reported for in the original results summary [12], the difference in magnitude is due to the different nondimensionalization. The reference conditions Alfvén is $0.1c_0$, consistent with previous reports of simulations of this problem using the ideal two fluid model. [9] The simulation is performed with an ion charge to mass ratio of 1, both species are treated as perfect gases with specific heat ratios of $\frac{5}{3}$, and an ion to electron mass ratio of 25 is used. The electron charge to mass ratio is specified to result in quasineutrality when the ion and electron number density are equal. The problem has periodic boundary conditions at $x = -12.8$ and $x = 12.8$ and perfect electric conductor wall boundary conditions at $z = -6.4$ and $z = 6.4$.

In the Harris equilibrium configuration, the spatial variation of the density, magnetic field, and current result in a Lorentz body force and pressure gradient that balance each other, preventing the denser current carrying layer of plasma from expanding into the adjacent lower density plasma. The current layer is also consistent with the direction change in the magnetic field and the configuration is a steady state solution of the ideal two fluid model, but it is an unstable equilibrium. If the dense current carrying plasma layer picks up any width variation in the direction of the magnetic field, the forces acting on it will tend to break it up into plasmoids associated with magnetic islands between which the previously non-intersecting magnetic field lines above and below the current layer reconnect. The GEM challenge problem begins with the magnetic field perturbed from the Harris equilibrium configuration in order to push the current sheet immediately toward fast reconnection around large magnetic islands. [12] The perturbation is the terms containing the $\frac{1}{100}$ factor in the B_x and B_z specifications of eq. 6.47, the remaining terms of the initial condition are the Harris equilibrium conditions, which have one dimensional variation in the z direction only. Figs. 6.1 and 6.2 illustrate the features of the Harris equilibrium and the perturbed magnetic field of the GEM challenge problem initial condition.

An important point of comparison between the solutions generated as part of this work and those in the literature is the time history of the reconnected flux, defined by [4]

$$F_r(t) = \frac{1}{2} \int_{-12.8}^{12.8} |B_z(x, z = 0, t)| \, \mathrm{d}x. \tag{6.48}$$

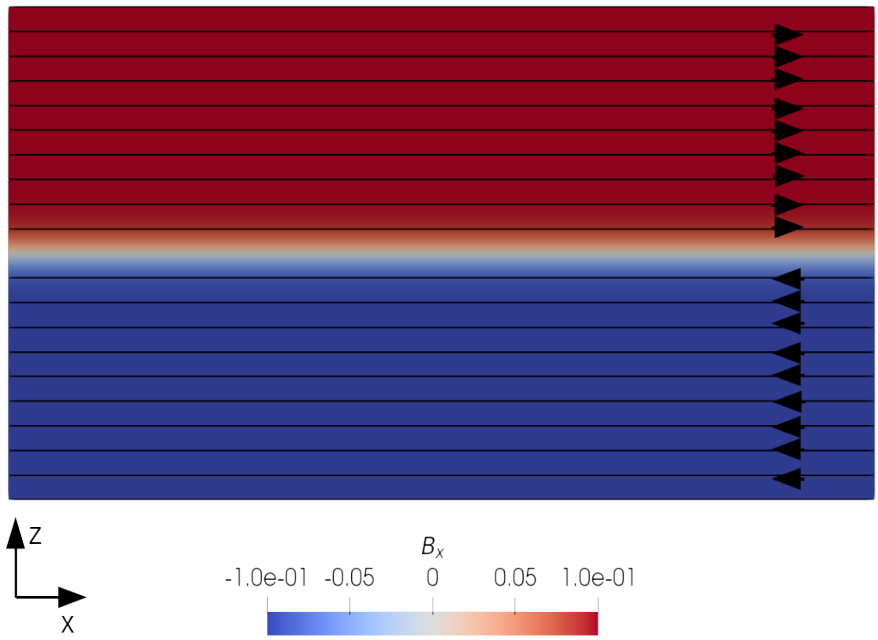


Figure 6.1: The magnetic field of the Harris equilibrium configuration that the GEM challenge problem initial condition is based on. The color scale represents the numerical value of the x component of the magnetic flux density while the lines follow the magnetic field vector field, the arrow heads indicate the direction of the field. A thin, horizontal layer of denser plasma exists across the center of the domain carrying a current in the positive y direction (into the page) that is consistent with the magnetic field change.

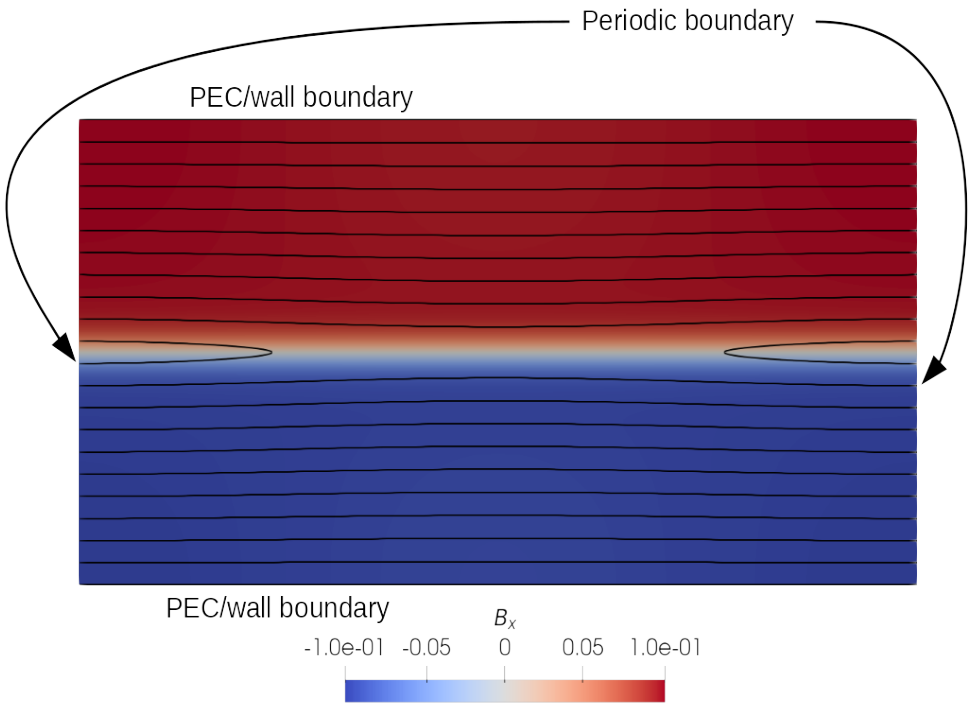


Figure 6.2: GEM challenge problem initial magnetic field and boundary conditions. The color scale represents the numerical value of the x component of the magnetic flux density while the lines follow the magnetic field vector field.

This can be thought of as a count of the number of magnetic field lines crossing the $z = 0$ line and, thus, reconnected across the current layer. It is included in the summary of the results of the original effort in ref. [12] as well as the subsequent simulations of the problem using the ideal two fluid plasma model reported in refs. [4, 5, 9]. Based on results in the literature, the ideal two fluid plasma model is expected to reasonably model this problem, though it typically shows an earlier reconnection onset and higher final reconnected flux than full particle simulation. [4,5,9] Also, based on the results in the literature, it is known that somewhat underresolved or excessively diffusive solutions will feature the emergence of an additional magnetic island and show a much higher peak reconnected flux because of it. [4,9] However, even more underresolved/excessively diffuse solutions can result in suppression of reconnection. [12] Not only is the amount of reconnection interesting, the peak rate is also an important feature. One of the particular goals of the original GEM challenge was to determine what physics needed to be included in simplified models to achieve appropriate reconnection rates in this kind of scenario. [12] Another interesting integrated quantity is the total energy in the domain:

$$\mathcal{E}_t(t) = \int_{-6.4}^{6.4} \int_{-12.8}^{12.8} \left(\frac{1}{2} \boldsymbol{D} \cdot \boldsymbol{E} + \frac{1}{2} \boldsymbol{B} \cdot \boldsymbol{H} + \rho_i e_{ti} + \rho_e e_{te} \right) \mathrm{d}x \mathrm{d}z. \tag{6.49}$$

Due to the boundary conditions, no energy should be able to enter or leave the domain, so this quantity should remain constant over any time interval, but numerical simulations tend to gain or lose some energy. [9] Ref. [9] reports a 0.7% energy loss using the high resolution wave propagation finite volume method with the source term handled by an ODE solver via Strang splitting, while ref. [5] shows good energy conservation in a plot for their simulations using a fully implicit finite volume method with iterative time advancement, but does not indicate a specific amount of energy gain or loss.

Chapter 7

Results and discussion

The results for this work are presented and discussed in two main sections. In the first, results for work using the SUPG method and some different approaches to the discontinuity capturing term are presented; in the second, results of work with the advection-reaction equation based stabilization are presented and discussed. Test cases for the SUPG work include multi dimensional manufactured solutions, the two fluid analog to the Brio-Wu shock tube, and the GEM challenge problem. The stabilized method based on the advection-reaction equation is only tested on a one dimensional manufactured solution and the two fluid Brio-Wu problem since the method was not extended to multiple spatial dimensions in this work.

7.1 SUPG results

7.1.1 Manufactured solution

Steady state three dimensional manufactured solution simulations were performed early in this work and some results are reported in ref. [58]. For those simulations, $\boldsymbol{D}$, $\boldsymbol{B}$, ψ, ϕ and ρ, $\boldsymbol{u}$, and e_t for each fluid species were specified using equations of the form

$$f(x, y, z) = a\cos(bx)\sin(cy)\cos(dz) + e, \tag{7.1}$$

where a, b, c, d, and e are constants chosen to have different values for each field. The constants were arbitrarily chosen, but it was ensured that the density and internal energy of each fluid species

was positive everywhere and the use of the sine and cosine functions was interchanged for some fields. It was also ensured that the acoustic velocities and flow velocities of the fluids were less than the speed of light. A hexahedral domain of $0.02 \leq x \leq 0.17$, $0.1 \leq y \leq 0.25$, and $0.15 \leq z \leq 0.3$ was used for those simulations. In later work, one and two dimensional steady state manufactured solutions and one dimensional transient manufactured solutions were developed and simulated to test the dedicated one and two dimensional element residual and Jacobian computations that were added to reduce the computational expense of lower dimensional simulations. The results of these simulations will be presented here. These simulations used specified solutions based on equations of the form

$$f(x,y) = a + \left(b + c \left(\frac{1 + \cos(\pi x)}{2} \right) \left(\frac{1 + \cos(\pi y)}{2} \right) \right) \tag{7.2}$$

for the two dimensional manufactured solution and

$$f(x,t) = a + \left(b + c \left(\frac{1 + \cos(\pi x)}{2} \right) \left(\frac{1 + \cos(\pi t)}{2} \right) \right) \tag{7.3}$$

for the one dimensional transient manufactured solution. These functions have no gradient normal to boundaries placed at $x = -1$, $x = 1$ and $y = -1$, $y = 1$ and the solution at $t = 2$ is the same as the solution at $t = 0$. These later simulations were carried out on domains of $-1 \leq x \leq 1$, $-1 \leq y \leq 1$, and a time interval of $0 \leq t \leq 2$, where applicable. For the simulations described here, boundary conditions are applied via specified fluxes obtained using Riemann fluxes with the exterior conditions set to the exact solution. Also, the discontinuity capturing terms were constructed by considering the full set of governing equations as a single set of balance laws when applying eqs. 4.95, 4.96, and 4.99 and using the sum of eq. 4.101 for all of the fluid species as the sensor function when switching the discontinuity capturing terms. In the cases with switched discontinuity capturing terms, a threshold of $f_s = 0$ and switch width of $f_e - f_s = 1 \times 10^{-3}$ were used.

Transient, one dimensional manufactured solution

The transient one dimensional manufactured solution problem was simulated on 4, 8, 16, 32, 64, and 128 linear element grids with equivalent numbers of timesteps using the SUPG scheme

alone, with unswitched discontinuity capturing terms, and with switched discontinuity capturing terms. The error integrals for these simulations, computed as described in sec. 6.1, are plotted against element size in figs. 7.1, 7.2, 7.3, and 7.4 for simulation with BDF2 time difference and no discontinuity capturing terms, simulation with implicit trapezoid rule time difference and no discontinuity capturing terms, simulation with implicit trapezoid rule time difference and unswitched discontinuity capturing terms, and simulation with implicit trapezoid rule time difference and switched discontinuity capturing terms, respectively. From figs. 7.1 and 7.2, it can be seen that the SUPG method without discontinuity capturing terms and using either second order time difference achieves approximately second order accuracy, and from fig. 7.3, it can be seen that the use of the unswitched discontinuity capturing term reduced the method to approximately first order accuracy. From fig. 7.4 it can be seen that when the discontinuity capturing terms are switched, it can be seen that the order of accuracy increases from first to second order as the solution becomes resolved and the discontinuity capturing terms get deactivated. Again in fig. 7.4, it can be seen that when switched discontinuity capturing terms are used, very rapid convergence is seen as they deactivate, but once they are substantially deactivated, the convergence rate returns to that expected for the underlying SUPG scheme. Only results for simulations with the discontinuity capturing terms and implicit trapezoid rule time differences are shown here, but those with the BDF2 time differences behave similarly. This is believed to be appropriate behavior for the numerical methods employed.

Plots of the final time electron density fields computed using the implicit trapezoid time difference without discontinuity capturing terms, with unswitched discontinuity capturing terms, and with switched discontinuity capturing terms are presented in figs. 7.5, 7.6, and 7.7, respectively, to provide and example of the solution quality obtained. In fig. 7.5 it can be seen that, without discontinuity capturing terms, the method produces numerical solutions with extrema outside the range of the true solution, but the convergence to the true solution occurs quite rapidly. Without discontinuity capturing terms, the numerical solutions are indistinguishable from the true solution at the scale of fig. 7.5 by 16 elements and timesteps. With the unswitched discontinuity capturing terms the convergence to the true solution is more gradual, as seen in fig. 7.6, but the solution is quite good by 128 elements and timesteps, showing only a slight shift. Switching the discontinuity capturing terms can be seen to speed convergence to the true solution by comparing figs. 7.6 and 7.7.

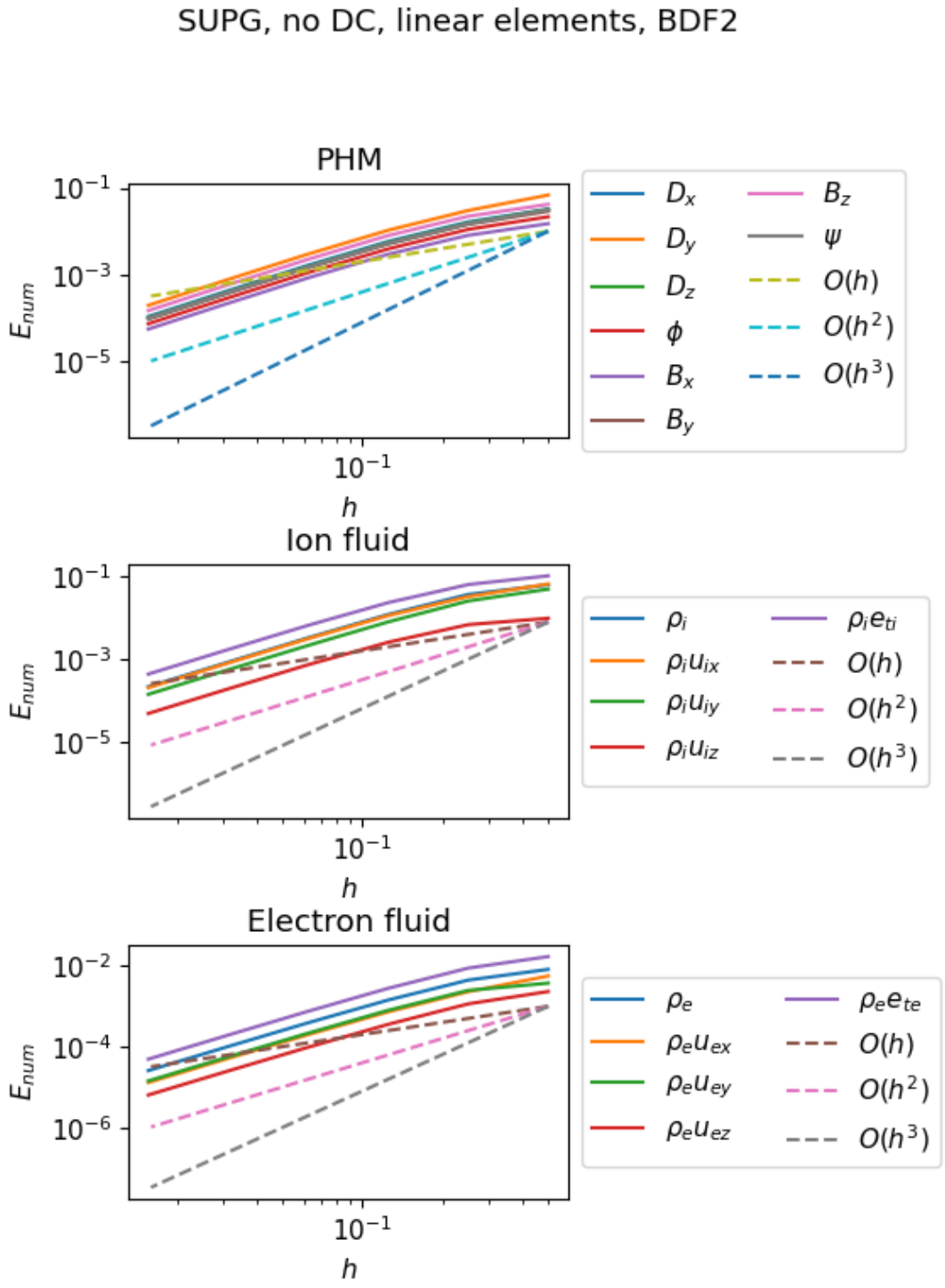


Figure 7.1: Plots of discretization error when using the SUPG method with linear elements, no discontinuity capturing term, and BDF2 time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

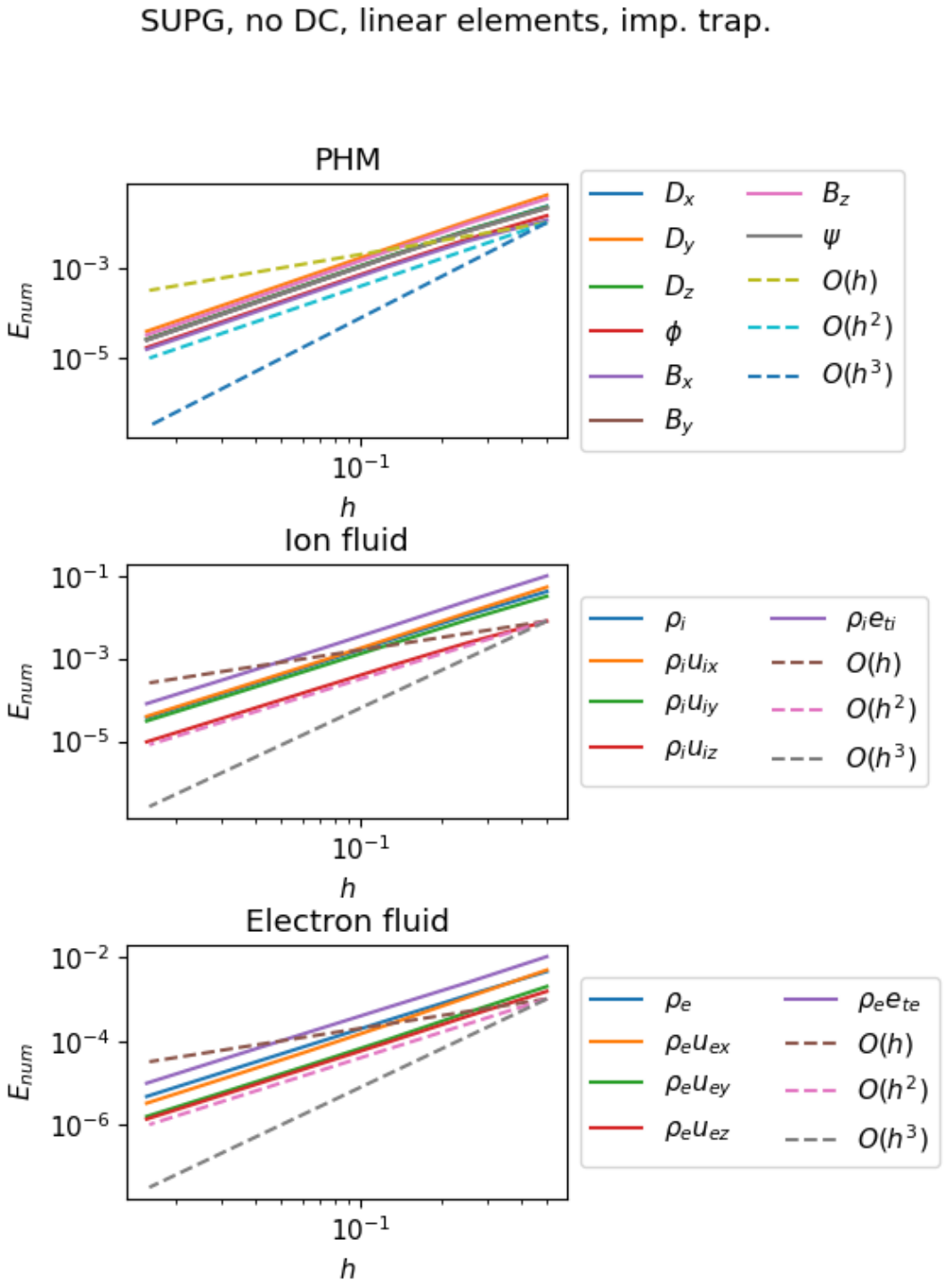


Figure 7.2: Plots of discretization error when using the SUPG method with linear elements, no discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

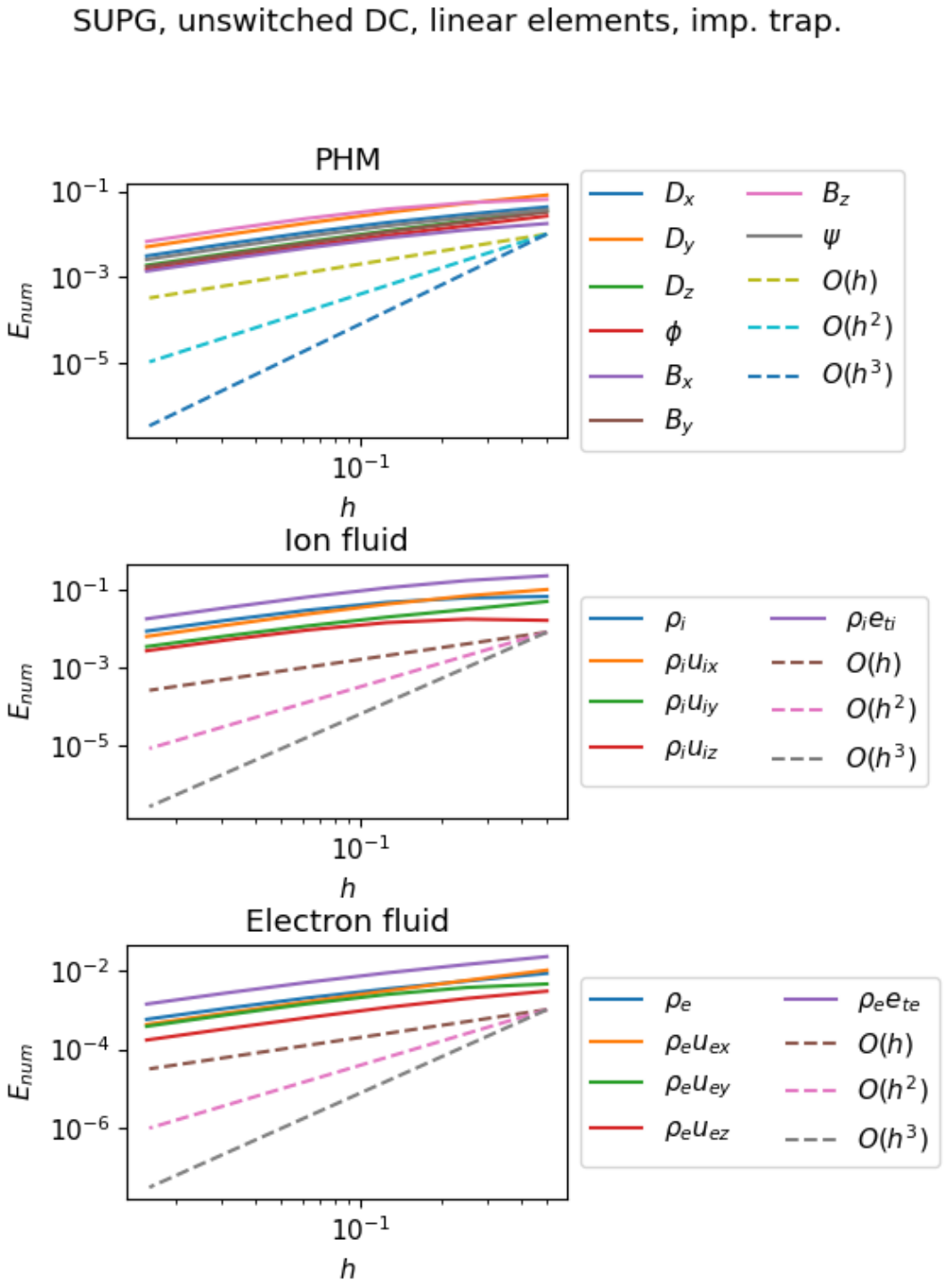


Figure 7.3: Plots of discretization error when using the SUPG method with linear elements, unswitched discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

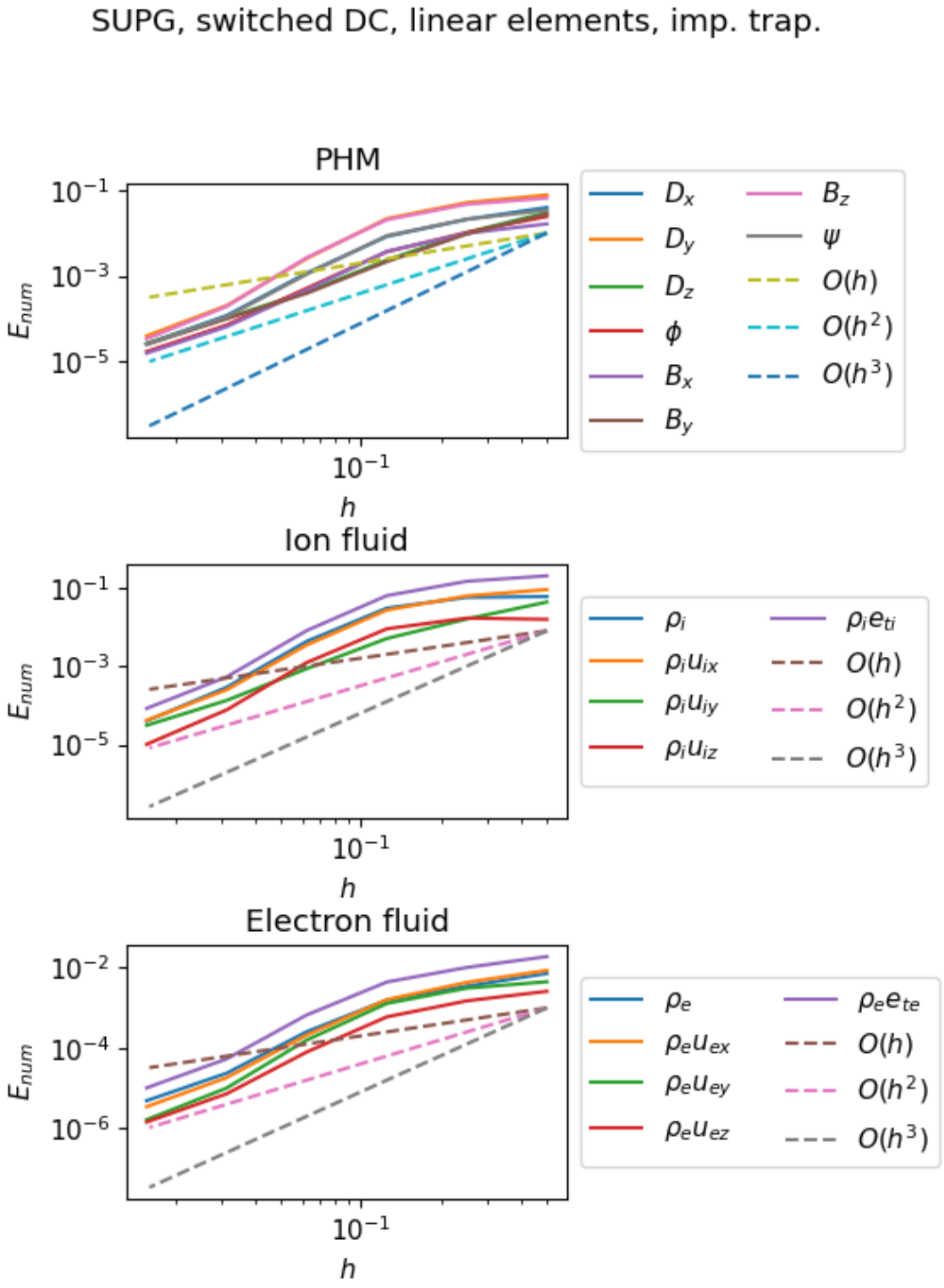


Figure 7.4: Plots of discretization error when using the SUPG method with linear elements, switched discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

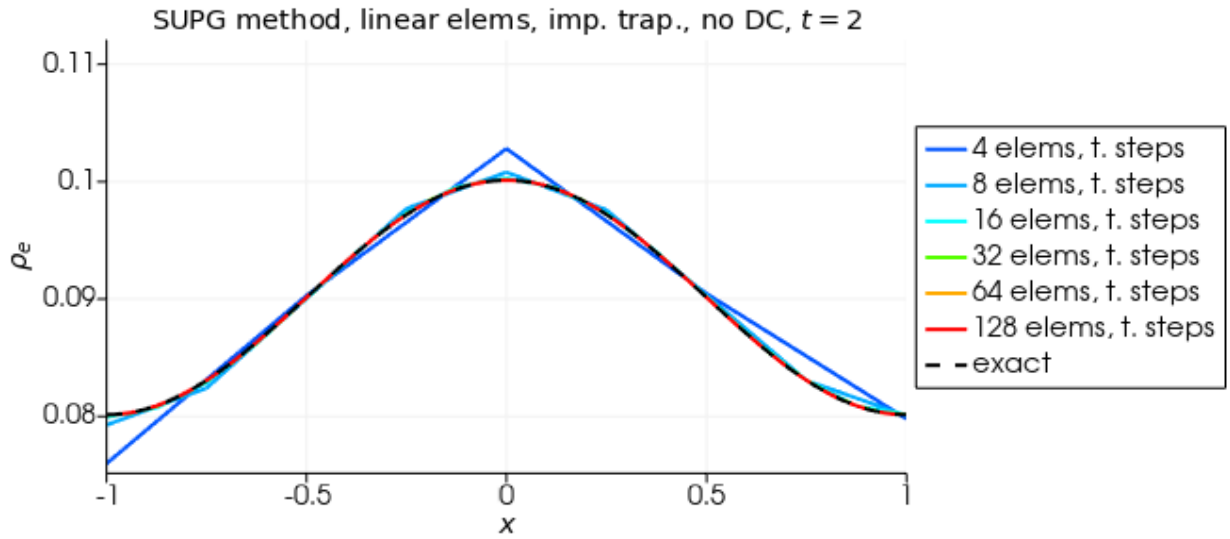


Figure 7.5: Plot of final time electron density fields obtained using the SUPG method with linear elements, no discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem.

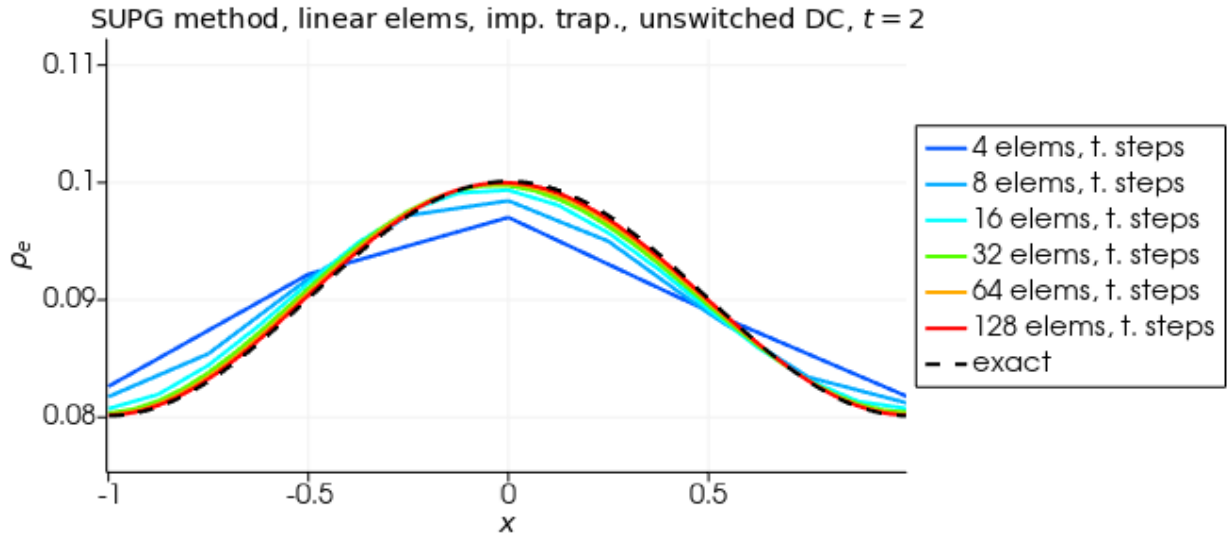


Figure 7.6: Plot of final time electron density fields obtained using the SUPG method with linear elements, unswitched discontinuity capturing terms, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem.

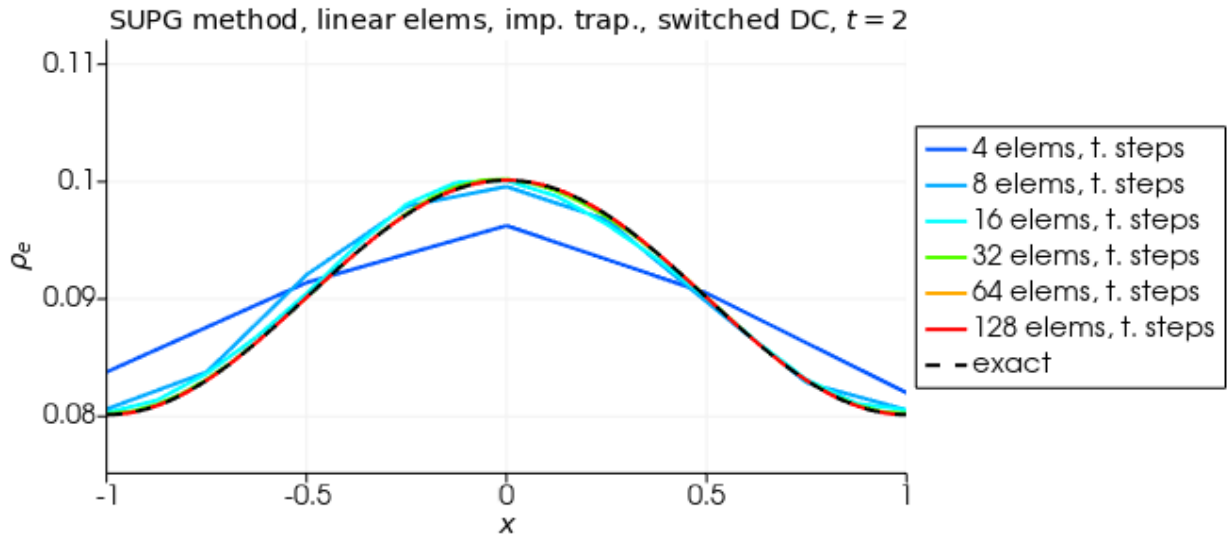


Figure 7.7: Plot of final time electron density fields obtained using the SUPG method with linear elements, switched discontinuity capturing terms, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem.

Steady state two dimensional manufactured solution

The steady state two dimensional manufactured solution was simulated using 4×4, 8×8, 16×16, 32×32, and 64×64 linear quadrilateral element grids using SUPG without discontinuity capturing terms, with discontinuity capturing terms, and with switched discontinuity capturing terms. The same strategies were also applied with 8 node quadratic quadrilateral elements for grids up to 32×32. For the simulations with linear elements, the resulting error integrals are plotted against element size in figs. 7.8, 7.9, and 7.10. For the simulations with quadratic elements, they are plotted in figs. 7.11, 7.12, and 7.13. (The element size is taken to be the side length since all grids employed square elements.) Without discontinuity capturing terms it can be seen that simulations using both the linear and quadratic elements converge at about the appropriate rates, second and third order, respectively, as seen in figs. 7.8 and 7.11, respectively. With unswitched discontinuity capturing terms, simulations with both linear and quadratic elements show approximately first order convergence, as shown in fig. 7.9 for linear elements and fig. 7.12 for quadratic elements. The convergence rate in both cases starts less than one, but appears to be approaching one with refinement. For the simulation with linear elements, the convergence rates on the final refinement are about 1 for the PHM equations, about 0.9 for the electron fluid equations and about 0.8 for the ion fluid equations as seen in fig. 7.9. For simulation with quadratic elements, the convergence rates on the final refinement are about 1 for the PHM equations, about 0.8 for the electron fluid equations, and between 0.4 and 0.8 for the ion fluid equations as seen in fig. 7.12. This is still quite low for the ion equations, but comparable to what was seen for linear elements for the 16×16 to 32×32 refinement, a refinement involving similar numbers of degrees of freedom. Since the error is dominated by the discontinuity capturing term in this case, it is believed that the appropriate behavior would be obtained with continued refinement as was seen with linear elements. As can be seen in fig. 7.10, with switched discontinuity capturing terms, the simulations with linear elements started out at about a first order convergence rate, then achieved very rapid convergence as the discontinuity capturing terms deactivated. On the final refinement, the convergence rate does seem to be moderating toward the second order rate as it should for the underlying SUPG scheme. In fig. 7.13, it can be seen that the simulation with quadratic elements and switched discontinuity capturing terms shows approximately second order accuracy even on the initial refinement, when

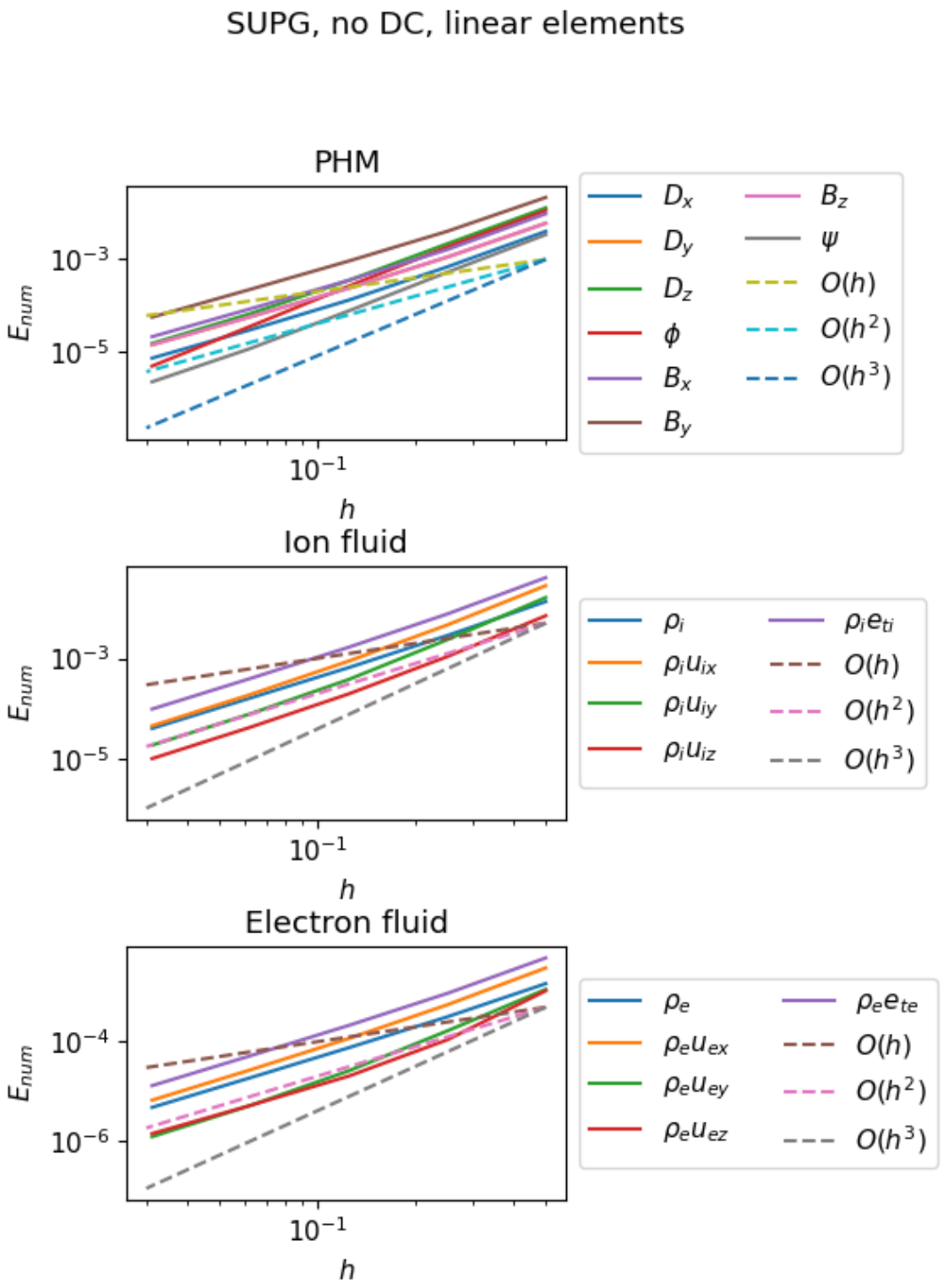


Figure 7.8: Plots of discretization error when using the SUPG method with linear quadrilateral elements and no discontinuity capturing terms on a steady state manufactured solution problem. Both axes are logarithmic.

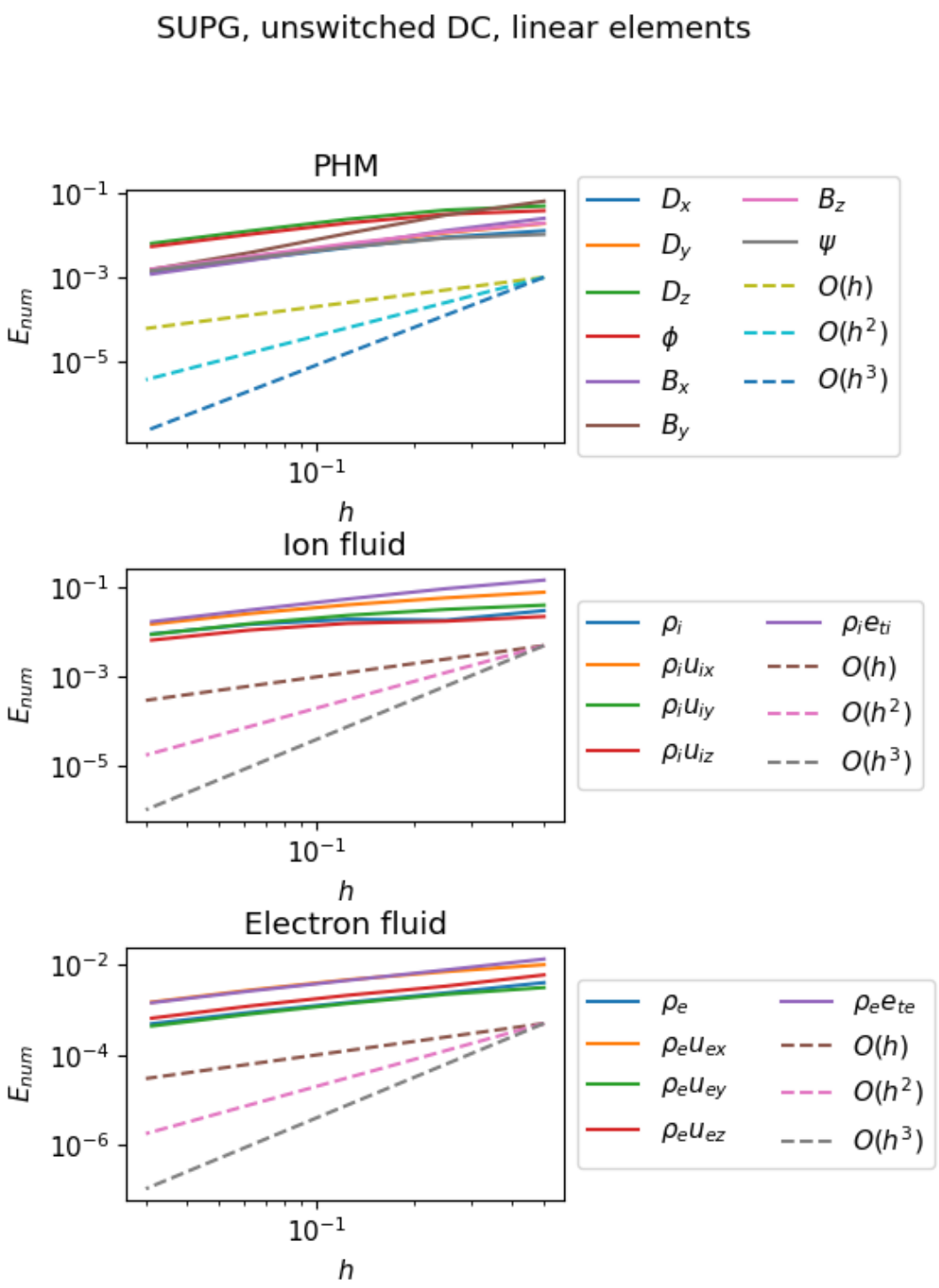


Figure 7.9: Plots of discretization error when using the SUPG method with linear quadrilateral elements and unswitched discontinuity capturing terms on a steady state manufactured solution problem. Both axes are logarithmic.

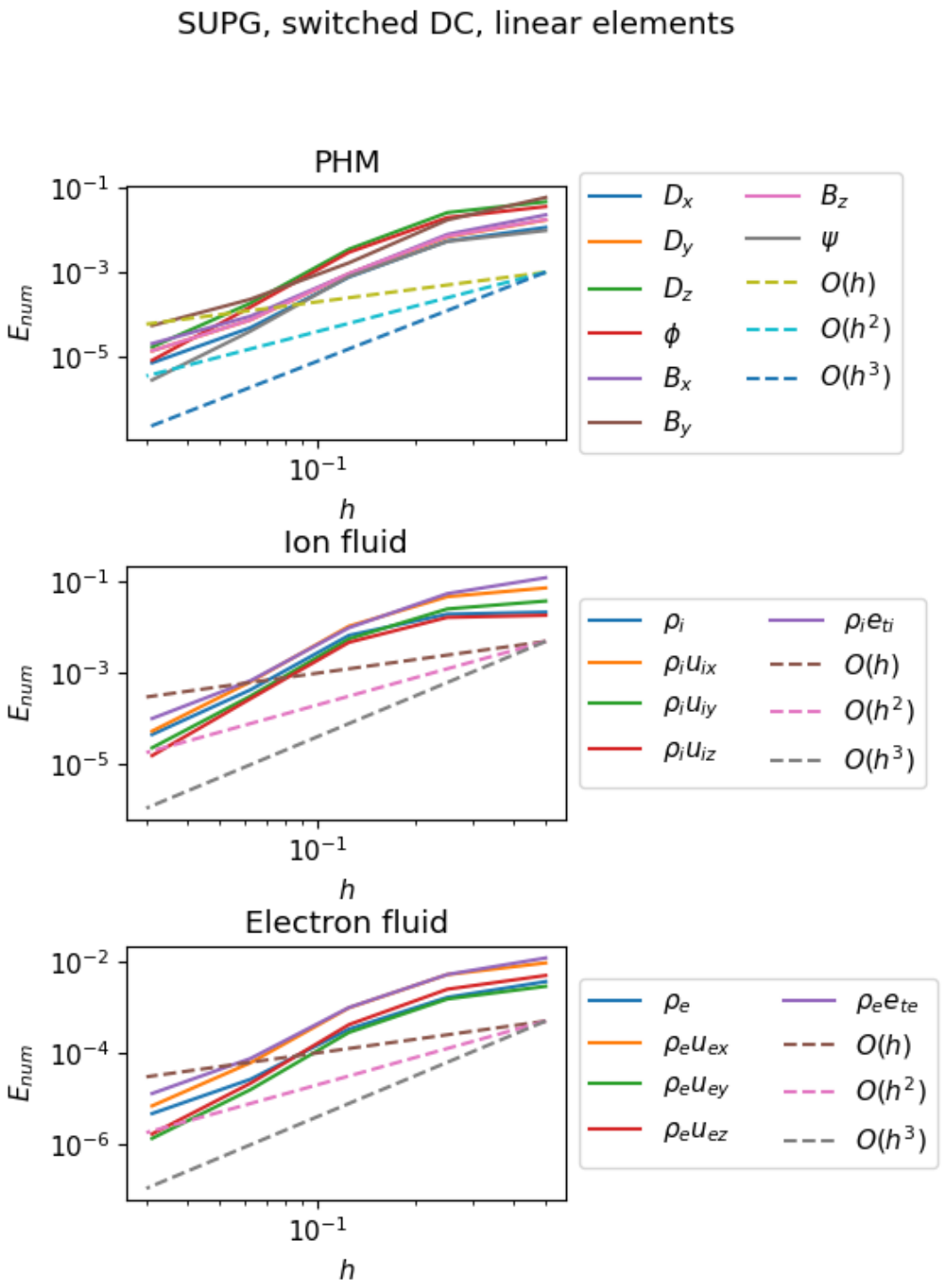


Figure 7.10: Plots of discretization error when using the SUPG method with linear quadrilateral elements and switched discontinuity capturing terms on a steady state manufactured solution problem. Both axes are logarithmic.

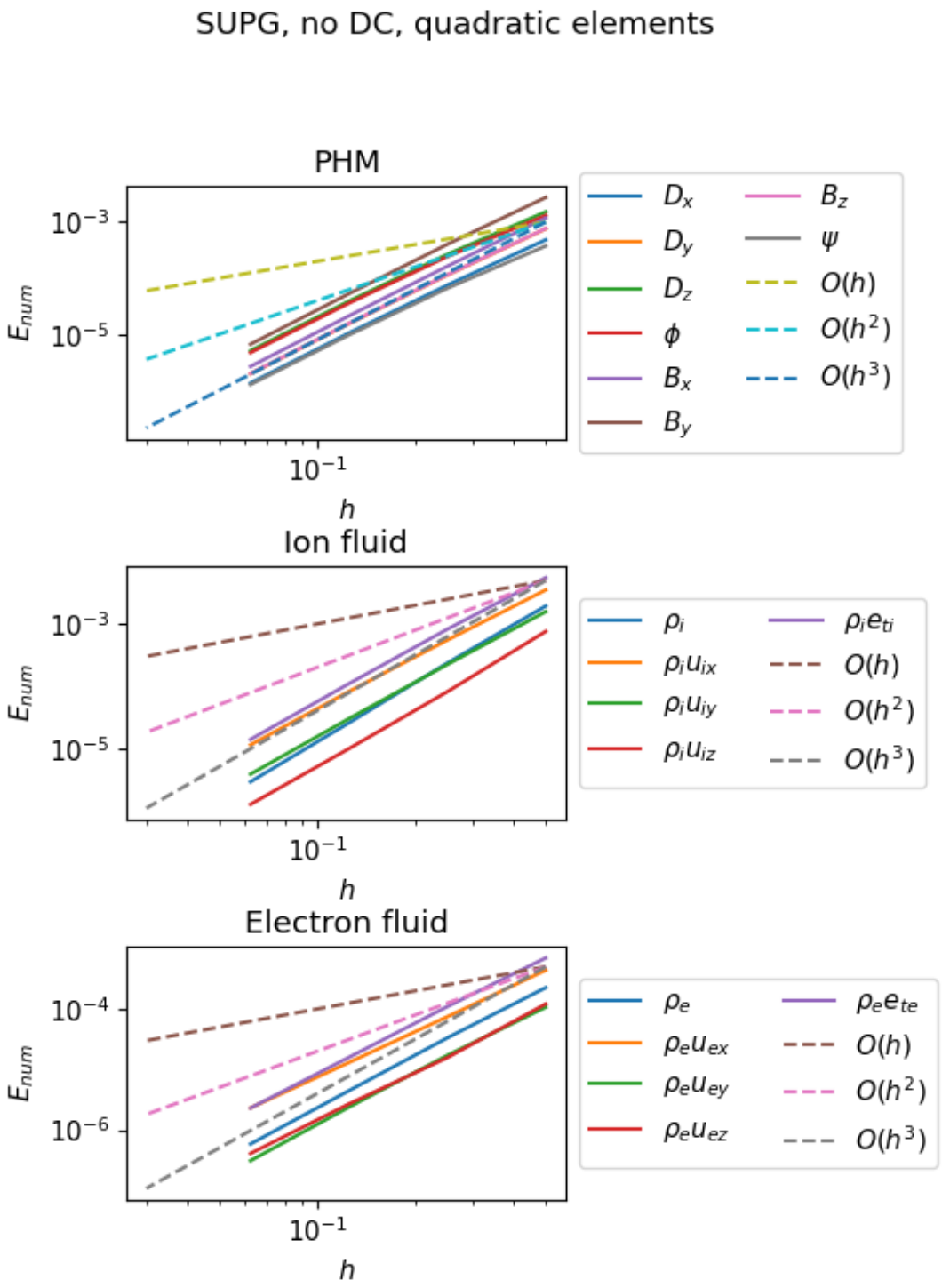


Figure 7.11: Plots of discretization error when using the SUPG method with 8 node quadratic quadrilateral elements and no discontinuity capturing terms on a steady state manufactured solution problem. Both axes are logarithmic.

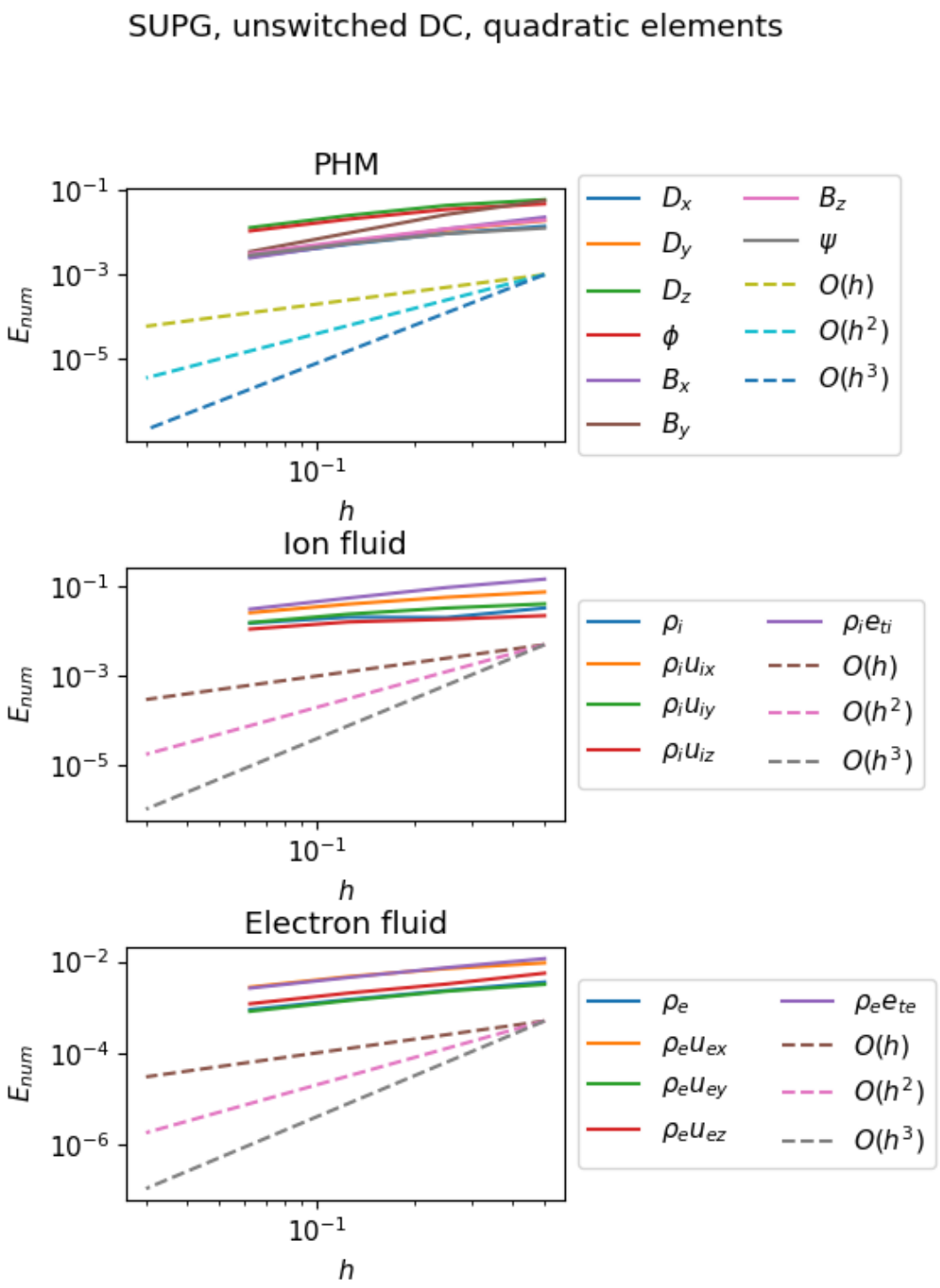


Figure 7.12: Plots of discretization error when using the SUPG method with 8 node quadratic quadrilateral elements and unswitched discontinuity capturing terms on a steady state manufactured solution problem. Both axes are logarithmic.

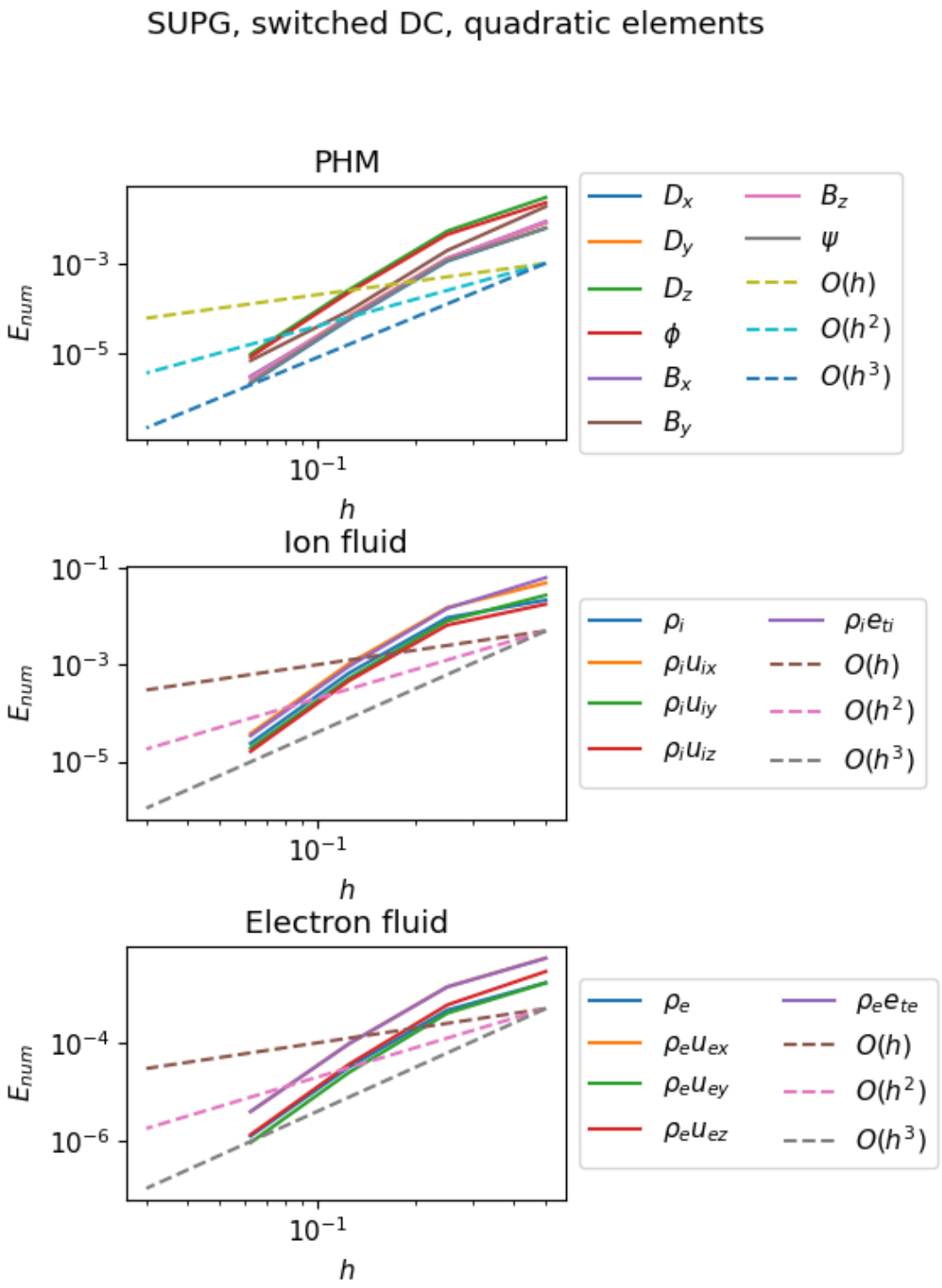


Figure 7.13: Plots of discretization error when using the SUPG method with 8 node quadratic quadrilateral elements and switched discontinuity capturing terms on a steady state manufactured solution problem. Both axes are logarithmic.

the grid is still very coarse and the discontinuity capturing terms would be expected to be active. Those simulations still experience rapid convergence on the 16×16 to 32×32 refinement that was the last refinement performed with the quadratic elements. Since appropriate behavior was seen for the base SUPG scheme with quadratic elements, it is believed that this would moderate toward the appropriate third order convergence rate with continued refinement, once the solution becomes sufficiently resolved for the discontinuity capturing terms to fully deactivate. The electron density fields produced with the switched discontinuity capturing terms and different numbers of elements are presented in figs. 7.14 and 7.15 for simulations with linear and quadratic elements, respectively, to illustrate the solution quality. With either linear or quadratic elements, the solution is quite good on the 32×32 element meshes. There are a few ripples evident in the 16×16 element quadratic solution, but generally, the quadratic element solutions are of comparable quality to the linear element solutions with twice the number of elements in each direction. It should be noted that the version of the software used for solution visualization (Paraview [61]) renders the solution in quadratic elements by discretizing them into smaller linear elements and interpolating linearly inside of those.

7.1.2 Two fluid Brio-Wu problem

The two fluid Brio-Wu problems performed for this work were all performed with a MHD reference velocity to speed of light ratio of $\frac{u_{rMHD}}{c_0} = \frac{1}{100}$ and an electron-ion mass ratio of $\frac{m_e}{m_i} = \frac{1}{100}$. Most were performed with the fluid specific heat ratios of each fluid being $\gamma = 2$ in hopes of being able to use charge to mass ratios high enough to show a good comparison with ideal MHD solutions, but that was ultimately not achieved. Some simulations with $\frac{q_i}{m_i} = 1$ were repeated with $\gamma = \frac{5}{3}$ for detailed comparison with the ideal two fluid simulation results available in the literature. The effect of the difference in the fluid specific heat ratios is surprisingly subtle, it obviously has an impact on the wave speeds and jump magnitudes coming from the fluid equations, but this is not enough to change overall structure of the solution. This can be seen by comparing the density fields obtained with the different specific heats in fig. 7.16. The overall solutions are not the same but they are, at least, qualitatively comparable. Simulations on grids of 1000 to 4000 linear elements per unit length were performed, these grid sizes appear to provide sufficient resolution for ion charge to mass ratios of $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$. Without discontinuity capturing terms, simulations

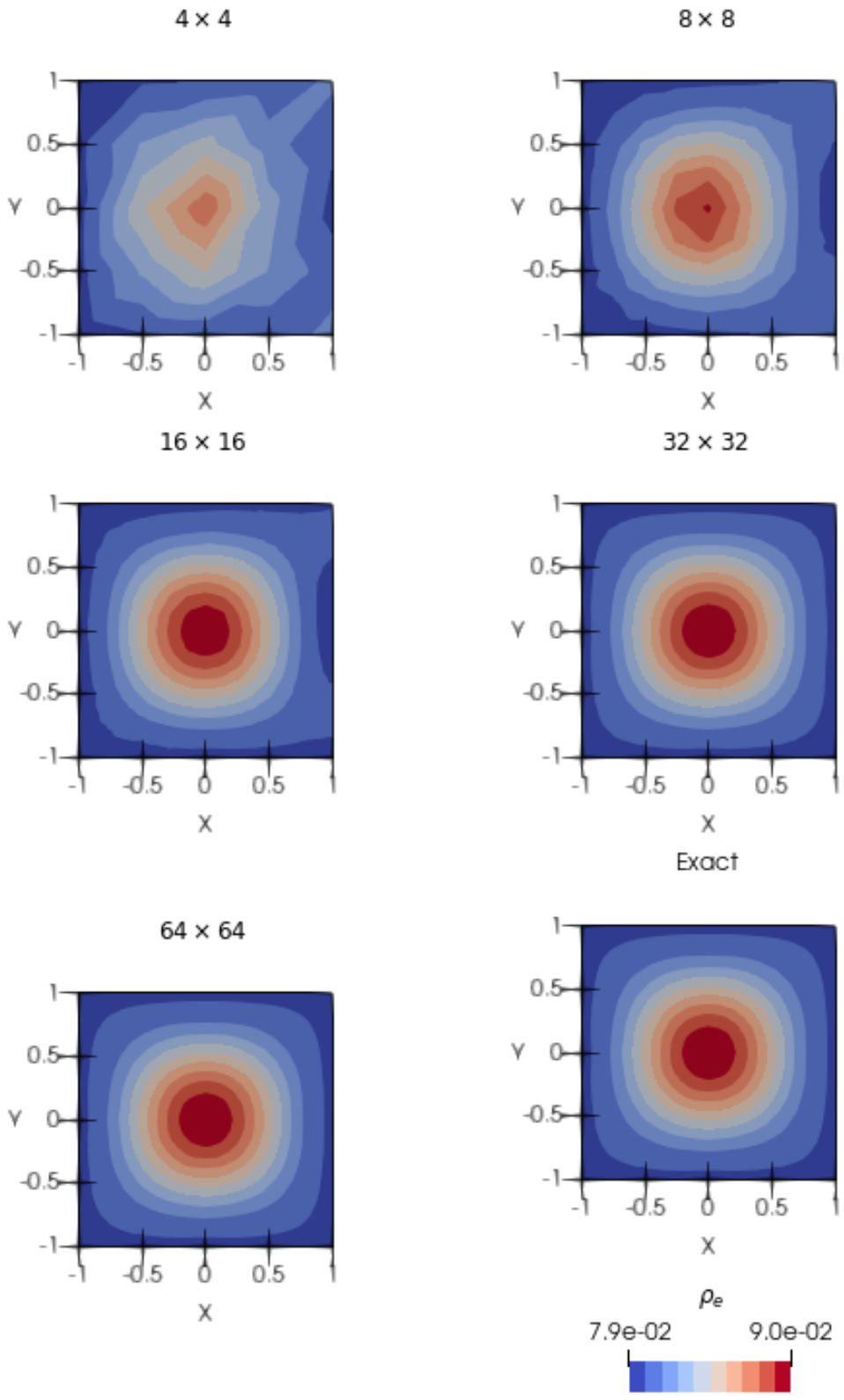


Figure 7.14: Plots of electron density fields obtained using the SUPG method with increasing numbers of linear quadrilateral elements, switched discontinuity capturing terms on a two dimensional manufactured solution problem.

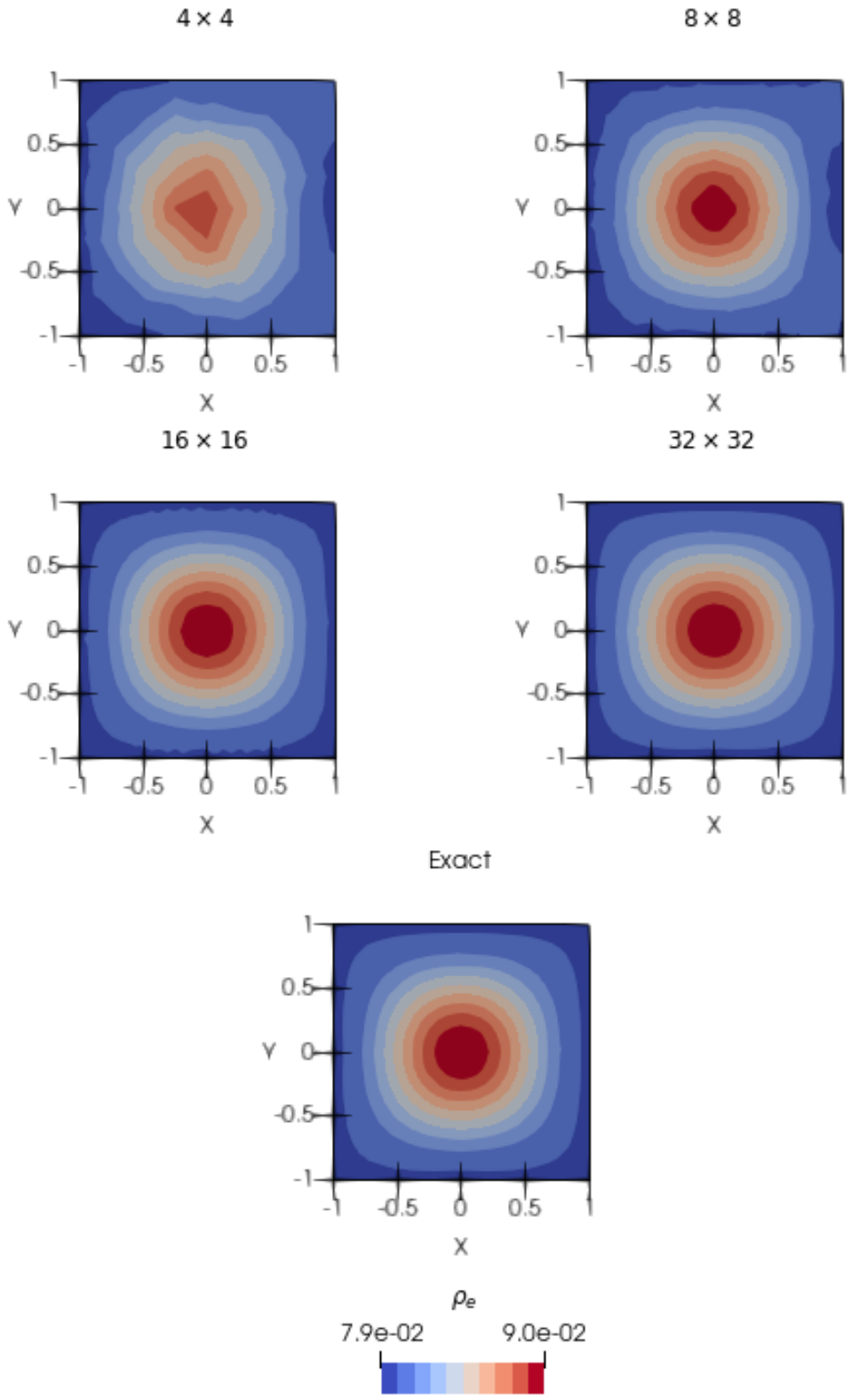


Figure 7.15: Plots of electron density fields obtained using the SUPG method with increasing numbers of 8 node quadratic quadrilateral elements, switched discontinuity capturing terms on a two dimensional manufactured solution problem.

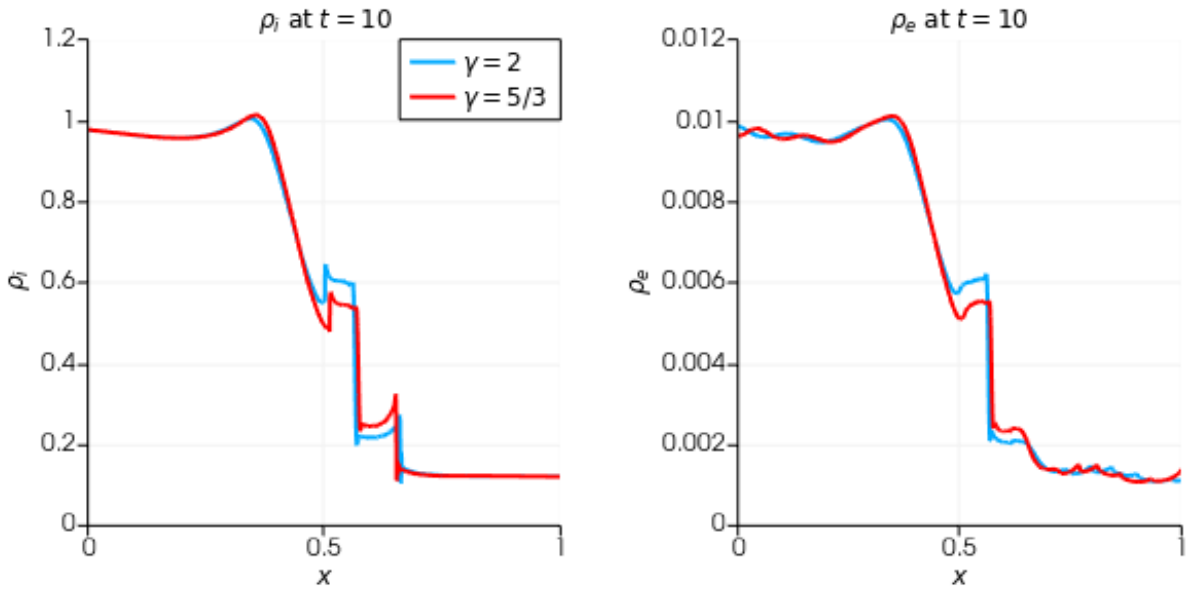


Figure 7.16: Ion and electron densities computed using the SUPG FEM without a discontinuity capturing term plotted for $0 \leq x \leq 1$ at $t = 10$ for Brio-Wu problem with $\frac{q_i}{m_i} = 1$ with $\gamma_i = \gamma_e = 2$ and $\gamma_i = \gamma_e = \frac{5}{3}$.

at both charge to mass ratios are largely grid independent across that refinement. Some changes are seen in some small scale features in the $\frac{q_i}{m_i} = 10$ simulations, but the $\frac{q_i}{m_i} = 1$ simulations are probably sufficiently resolved at even lower resolutions. As previously stated, increasing resolution is necessary as the charge to mass ratio is increased if Debye length scale plasma oscillation like activity must be resolved for an accurate solution because the Debye length decreases as the charge to mass ratio is increased. A much finer grid appears to be necessary for simulations with an ion charge to mass ratio of $\frac{q_i}{m_i} = 100$. Simulations at that ion charge to mass ratio were attempted with grids containing as many as 100000 elements per unit length, but the simulation domain had to be truncated to make that resolution tractable. Except as noted, second order implicit time differences were used for all of the simulations reported here, either BDF2 or the implicit trapezoid rule, and timestep sizes were chosen to provide a CFL number of one based on the speed of light in most cases, this also provides a reasonable number of timesteps per plasma oscillation period at the grid resolutions used. Simulations were performed without discontinuity capturing terms, with unswitched discontinuity capturing terms added to the species fluid equations, with unswitched, system specific discontinuity capturing terms added to each set of govering equations, and with simultaneously switched discontinuity capturing terms added to all of the governing equation sets as the code evolved. In the simulations with simultaneously switched discontinuity capturing terms added to all of the governing equation sets, two approaches to the computation of the discontinuity capturing viscosity were employed. In the first approach, eqs. 4.95, 4.96, and 4.99 were applied independently to the PHM equations, ion fluid equations, and electron fluid equations to obtain discontinuity capturing viscosities specific to each portion of the model. This is referred to as the independent system or system specific discontinuity capturing term approach. In the second approach, the equations were applied considering the entire set of governing equations as one set of balance laws, resulting in a single discontinuity capturing viscosity. This is referred to as the full model discontinuity capturing term approach. In both cases, the smooth switching function of eq. 4.100 was used with threshold $f_s = 0$ and switch width 1×10^{-3}, and the discontinuity sensor function was the sum of eq. 4.101 computed for the ion fluid and electron fluid. The results of these approaches for problems with $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$ were previously reported in refs. [46] and [6] and their presentation here will largely echo that work. In later work, a simple serial, one dimensional finite volume code was also developed for the ideal two fluid - PHM model to see if some of the

behaviors observed in earlier work were specific to the SUPG FEM. This solver employed the AUSM flux [62] for both fluids and a Riemann flux for the PHM equations. A "theta" time scheme [63] was employed, capable of producing an implicit Euler ($\Theta = 1$), implicit trapezoid ($\Theta = \frac{1}{2}$), or explicit Euler ($\Theta = 0$) time term and the multivariable Newton method was used to obtain the update iteratively. (The implicit methods are used almost exclusively, the iterative scheme just converges immediately if the explicit scheme is used.) Limited least squares reconstruction was used to obtain second order accuracy in this code. Some ideal MHD results are also presented for comparison, these were obtained with a very simple first order finite volume scheme employing explicit Euler timestepping and a HLL flux [38].

Simulation without discontinuity capturing terms

Simulations of the two fluid Brio-Wu problem with fluid specific heat ratios of $\gamma_i = \gamma_e = 2$ and $\gamma_i = \gamma_e = \frac{5}{3}$ and ion charge to mass ratios of $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$ were performed without discontinuity capturing terms. These simulations were all performed on an extended domain of $-9.5 \leq x \leq 10.5$. Good agreement with the published results of refs. [4, 5, 9] was seen with $\gamma_i = \gamma_e = \frac{5}{3}$ and $\frac{q_i}{m_i} = 1$ even though the ion to electron mass ratio was much higher in refs. [4, 9]. As mentioned, the results obtained with $\gamma_i = \gamma_e = 2$ were also qualitatively similar. The species density fields from simulations with $\frac{q_i}{m_i} = 1$ and the different specific heat ratios are shown together on $0 \leq x \leq 1$ in fig. 7.16, for comparison. The ion densities obtained in the original MHD problem domain are presented alongside an ideal MHD mass density field in fig. 7.17. From that figure, it is clear that the behavior is still significantly different from ideal MHD, refs. [4, 5, 9] achieve more MHD like behavior with conditions comparable to $\frac{q_i}{m_i} = 1000$, and higher ion to electron mass ratios in refs. [4, 9]. The faster moving slow shock seen in the two fluid results here is similar to the results obtained in ref. [4]. The species densities are presented again in fig. 7.18, scaled so that the ion and electron density overlap on the plot when quasineutrality is achieved. It can be seen from the plots in that figure that significant charge separation occurs in the $\frac{q_i}{m_i} = 1$ scenario, while much less occurs when $\frac{q_i}{m_i} = 10$, indicating a move toward conditions where the quasineutrality assumption of ideal MHD is valid.

For simulations with $\gamma_i = \gamma_e = 2$ and $\frac{q_i}{m_i} = 1$, time histories of the total amount of energy in the domain are presented in fig. 7.19 , along with the results obtained using various discontinuity

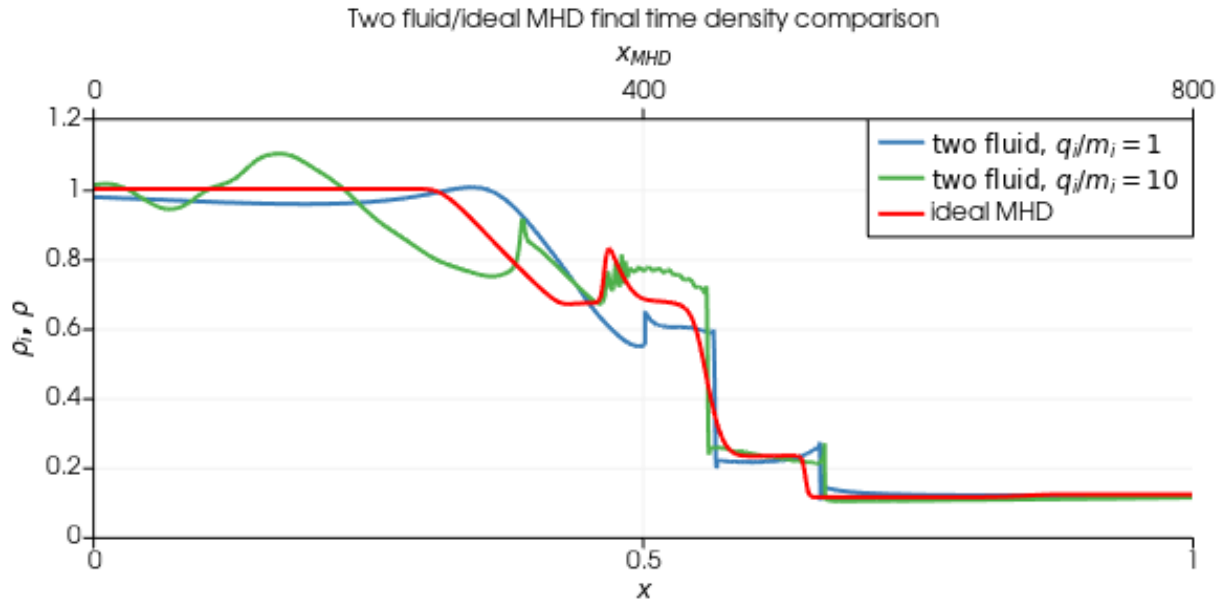


Figure 7.17: Final time two fluid ion density at computed with $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$ compared to final time ideal MHD density.

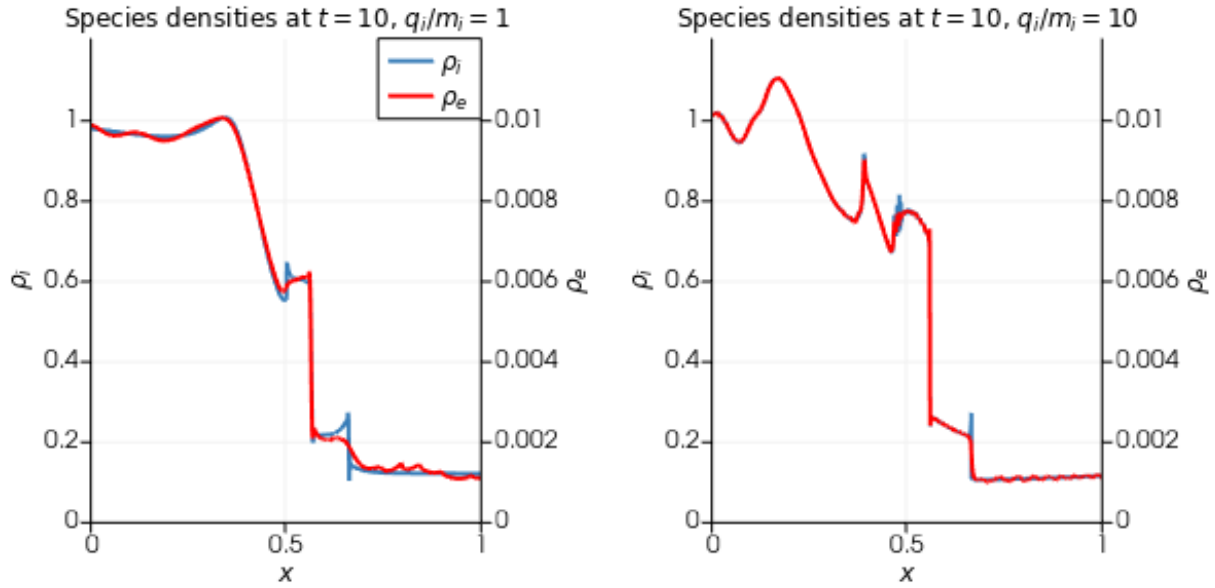


Figure 7.18: Final time ion and electron densities computed using the SUPG FEM without a discontinuity capturing term plotted for $0 \leq x \leq 1$ for Brio-Wu simulations with $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$ with $\gamma_i = \gamma_e = 2$.

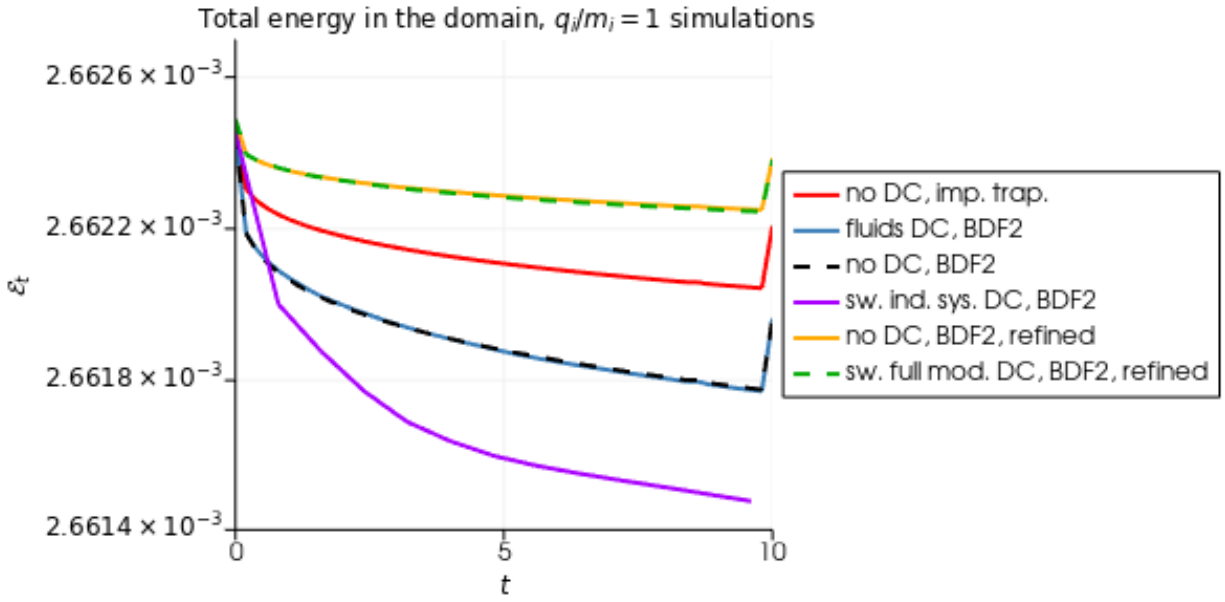


Figure 7.19: Time evolution of total energy in the domain for the Brio-Wu problem with $\frac{q_i}{m_i} = 1$, $\gamma_i = \gamma_e = 2$ simulated with the SUPG FEM with different discontinuity capturing terms, time differences and grid resolutions. Data sets labelled "refined" are for results obtained on 4000 element per unit length grids with 40000 timesteps, the others were obtained on 1000 element per unit length grids with 10000 timesteps. Note that the last data point for the simulation with the switched system specific discontinuity capturing term is missing in this plot.

capturing terms and time differences and different grid resolutions. From the figure, it can be seen that the amount of energy in the domain slightly decreased over the course of the simulation except right at the end in all cases, including those in which a discontinuity capturing term was not employed. (Since the domain boundaries are far enough from the initial discontinuity that the solution should not change there and the conditions there should not result in any energy flow in or out of the domain, the total amount of energy in the domain should remain constant for this problem over this time interval.) The jump in energy at the very end that was seen in every case for which output was available on the final step is believed to be the result of the solution around the fast moving waves being smeared enough to reach the boundaries and cause a small energy influx. Comparing the results for simulations without discontinuity capturing terms in fig. 7.19, it can be seen that the amount of energy loss was higher with the BDF2 time difference than with the implicit trapezoid rule at the same grid resolution, and that grid refinement reduced the amount of energy loss. The time evolution of the energy in the domain for simulations with $\frac{q_i}{m_i} = 10$ is similarly shown in fig. 7.20. Considering the simulations performed without discontinuity capturing terms, the simulation that used the BDF2 time difference and the grid refined to 4000 elements per unit length in $0 \leq x \leq 1$ had a slight energy loss over the course of the simulation. The simulation that used the implicit trapezoid time difference and the uniform 1000 elements per unit length grid resulted in slightly more energy loss and saw a brief period of slight energy accumulation late in the simulation. In fig. 7.21, it can be seen that the coarser grid simulation with the implicit trapezoid time difference did have larger amplitude electron density oscillations in the plasma to the left of the original MHD problem domain. Some oscillations here are likely appropriate since the electromagnetic waves emanating from the domain of interest disturb the plasma here, but in simulations with discontinuity capturing terms, these can grow large enough to cause conditions that crash the algorithm.

Simulations with $\gamma_i = \gamma_e = 2$ and $\frac{q_i}{m_i} = 100$ were also attempted without discontinuity capturing terms but none were able to survive the entire time interval. These simulations were attempted on much higher resolution grids of 10000 and 100000 elements per unit length on the, much shorter, original MHD problem domain of $0 \leq x \leq 1$. It was hoped that, at this high charge to mass ratio, the behavior would be MHD like enough that the electromagnetic disturbances would be damped out before reaching these closer boundaries, which are necessary to make the problem

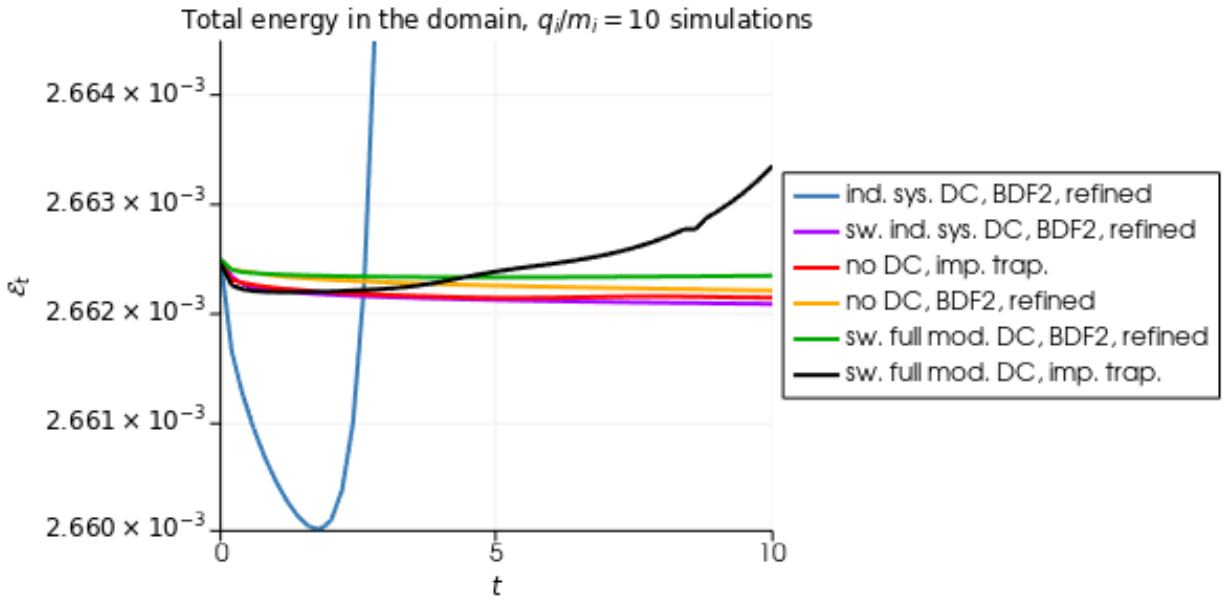


Figure 7.20: Time evolution of total energy in the domain for the Brio-Wu problem with $\frac{q_i}{m_i} = 10$, $\gamma_i = \gamma_e = 2$ simulated with the SUPG FEM with different discontinuity capturing terms, time differences and grid resolutions. Data sets labelled "refined" are for results obtained on 4000 element per unit length grids with 40000 timesteps, the others were obtained on 1000 element per unit length grids with 10000 timesteps. For the simulation without discontinuity capturing terms and BDF2 labelled "refined" only $0 \leq x \leq 1$ was refined to 4000 elements per unit length, the remainder of the grid was coarsened back to 1000 elements per unit length.

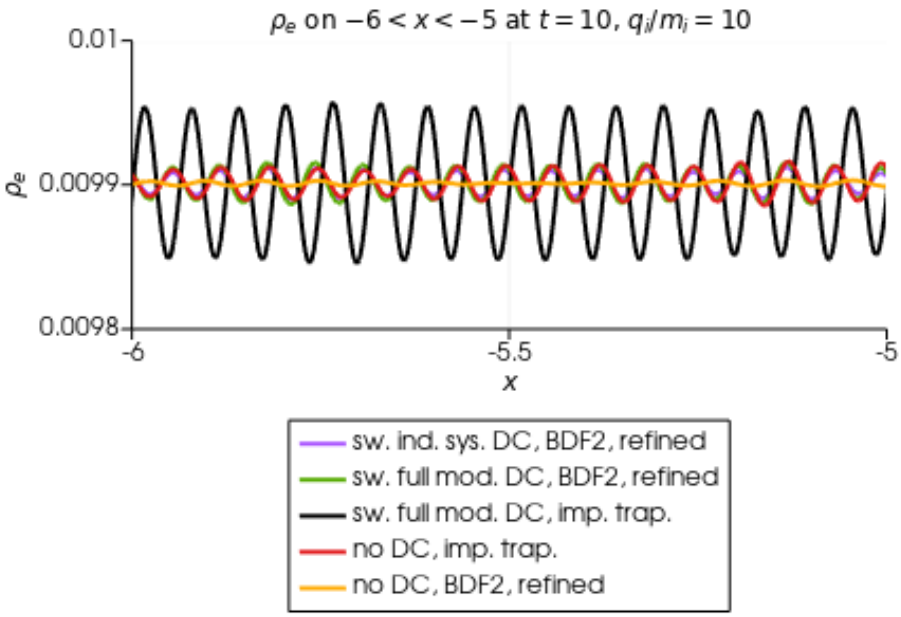


Figure 7.21: Electron density oscillations to the left of the domain of interest ($0 \leq x \leq 1$) for the Brio-Wu problem with $\frac{q_i}{m_i} = 10$, $\gamma_i = \gamma_e = 2$ simulated with the SUPG FEM with different discontinuity capturing terms, time differences and grid resolutions. Data sets labelled "refined" are for results obtained on 4000 element per unit length grids with 40000 timesteps, the others were obtained on 1000 element per unit length grids with 10000 timesteps. For the simulation without discontinuity capturing terms and BDF2 labelled "refined" only $0 \leq x \leq 1$ was refined to 4000 elements per unit length, the remainder of the grid was coarsened back to 1000 elements per unit length.

tractable with the available computational resources, but that was not the case. Some significant electromagnetic waves do leave the discontinuity at the beginning of the simulation and persist to the boundary. These seem to pass out of the domain without any immediate ill effect, and the last available species density fields, shown in fig. 7.22, show little disturbance near the boundaries, giving the impression that the assumption is holding. But there are other features that indicate the assumption remains inappropriate at this charge to mass ratio. Later there are considerable variations in the electromagnetic fields and electron momenta for which it is not clear if they are the result of boundary feedback. Additionally, fig. 7.23 shows the electric and magnetic fields at the last available time, to the left, there is little disturbance in the magnetic fields, but to the right there is significant change all the way to the boundary. The electric field is also disturbed all the way to the boundaries. It is ultimately at the boundary where a negative electron fluid internal energy, which is nonphysical for the equation of state used, and results in an imaginary sound speed unless the electron density is also negative, occurs crashing the algorithm for this problem. The time evolution of the total energy of the domain has less meaning in this case since the domain is not long enough that all of the effects should remain inside the domain, but this should be close to the case if MHD like conditions prevail and little to no solution change at the boundaries is seen. It is presented in fig. 7.24. It can be seen that there is a dip in the amount of energy in the domain in both simulations at about $t = 0.5$ when the first effects would reach the boundary. It can then be seen that the coarser mesh simulation experiences a great deal more energy accumulation than the fine mesh simulation, as was seen in some of the lower charge to mass ratio results.

Simulation with unswitched discontinuity capturing terms added to the fluid equations

Simulations with unswitched discontinuity capturing terms computed based on eqs. 4.95, 4.96, and 4.99 added to each species' set of fluid dynamics equations were also performed with $\gamma_i = \gamma_e = 2$ and $\frac{q_i}{m_i} = 1$. For this low charge to mass ratio, this approach effectively smoothed away the numerical oscillations around the discontinuities that were seen in the solutions without discontinuity capturing terms. As would be expected with an unswitched discontinuity capturing term, some unwanted damping of small features was seen. The final time species densities are presented in fig. 7.25 alongside the results obtained without discontinuity terms to illustrate the effect. For the grid resolutions used, the solution is quite well resolved away from the discontinuities

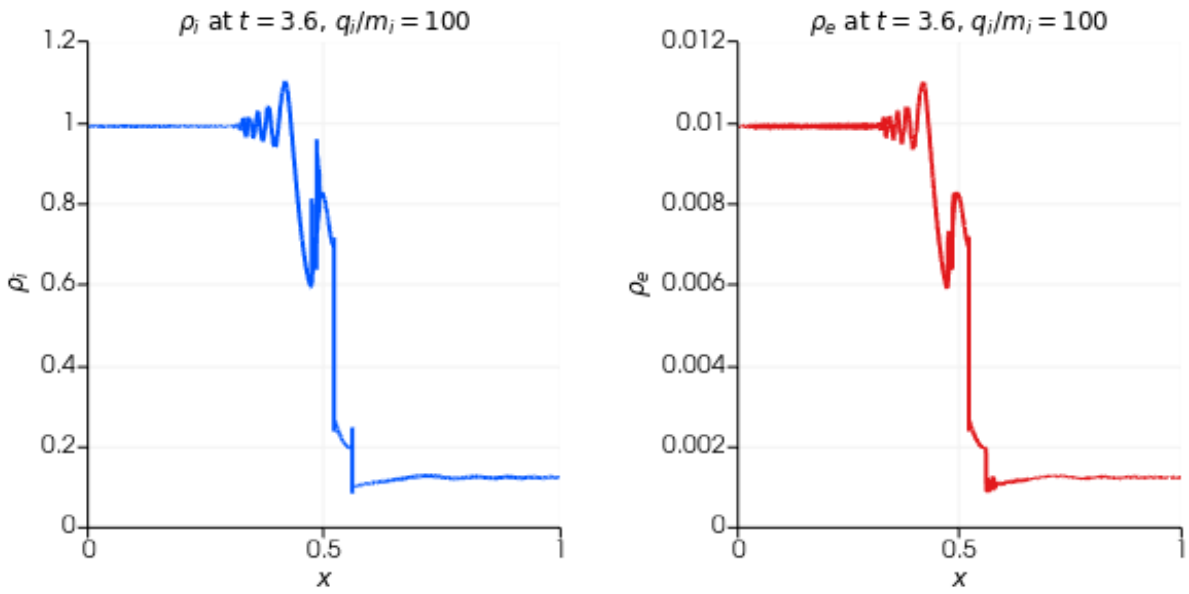


Figure 7.22: Ion and electron densities computed using the SUPG FEM without a discontinuity capturing term for the Brio-Wu problem with $\frac{q_i}{m_i} = 100$ and $\gamma_i = \gamma_e = 2$ at $t = 2.6$.

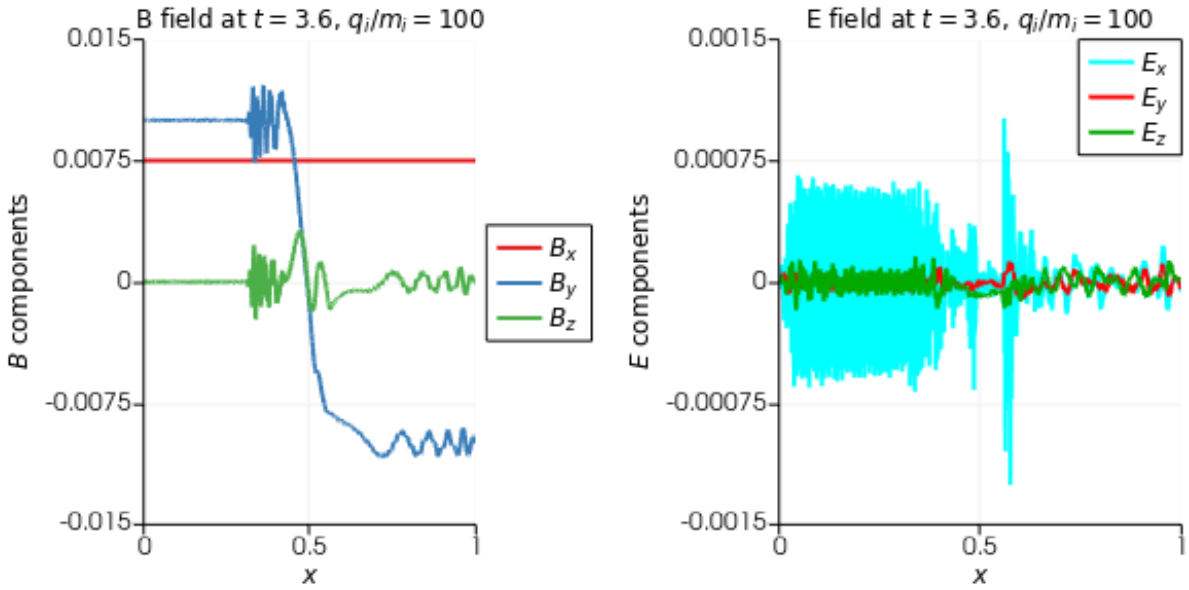


Figure 7.23: Electric and magnetic field components computed using the SUPG FEM without a discontinuity capturing term for the Brio-Wu problem with $\frac{q_i}{m_i} = 100$ and $\gamma_i = \gamma_e = 2$ at $t = 2.6$.

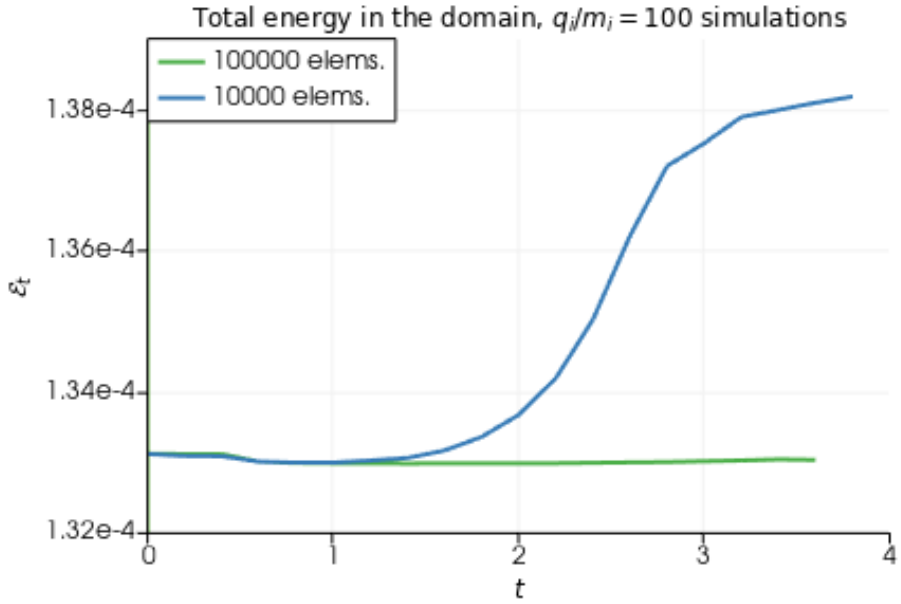


Figure 7.24: Time evolution of total energy in the domain for the Brio-Wu problem with $\frac{q_i}{m_i} = 100$, $\gamma_i = \gamma_e = 2$ simulated with the SUPG FEM with no discontinuity capturing terms, implicit trapezoid time differences, and two grid resolutions. The timestep size was changed also to maintain a CFL number of one based on the speed of light on each grid. For this problem only the domain $0 \leq x \leq 1$ was simulated due to the high number of elements per unit length required.

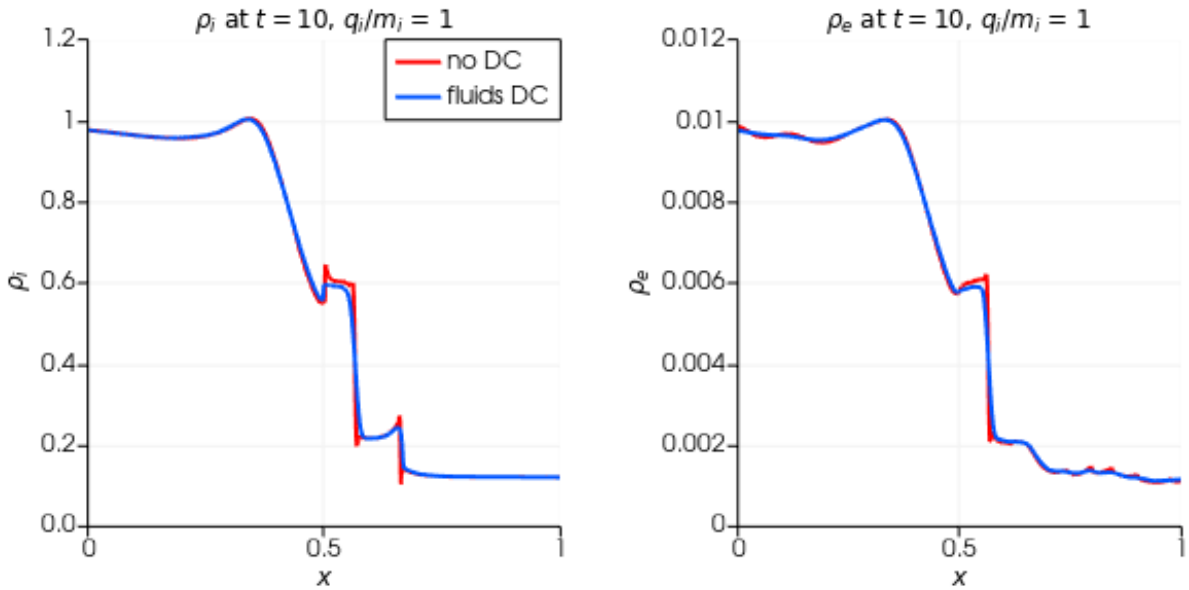


Figure 7.25: Ion and electron densities computed using the SUPG FEM without a discontinuity capturing term and with unswitched discontinuity capturing terms added to the species' fluid equations plotted for $0 \leq x \leq 1$ at $t = 10$ for Brio-Wu problem with $\frac{q_i}{m_i} = 1$ with $\gamma_i = \gamma_e = 2$.

even with the discontinuity capturing term active and only a slight loss of energy in the domain is seen, comparable to that obtained without discontinuity capturing terms, as seen in fig. 7.19. Again, a small jump is seen at the last data point due to the numerical method smearing the fields, allowing some changes to reach further than expected due to the governing equations' wave speeds.

Unfortunately, in later work on the GEM challenge problem, this approach was found to result in the growth of large scale oscillations in the relatively quiet background plasma and seemingly related nonphysical energy production in simulations coarser grids. In ref. [64], the issue was studied using a "magnetic field Riemann problem" which had the magnetic field jump of the Brio-Wu problem, but constant, quiescent conditions for the fluid species' initial condition. It was found that these large, smooth oscillations and nonphysical energy production also occurred in the one dimensional setting when these unswitched discontinuity capturing terms were used and were worsened by coarse grids. The problem was later discovered to be occurring outside the typical domain of interest in Brio-Wu problem simulations and electrostatic Riemann problems with higher charge to mass ratios and with other discontinuity capturing terms, as well. Since the higher charge to mass ratio reduces the spatial scale of the small plasma oscillations that occur in these problems when the electron fluid is excited by the fast electromagnetic waves that sweep through the domain, it is necessary to use a finer grid to prevent the growth of these oscillations and the associated nonphysical energy accumulation. It is, thus, believed that the issue of nonphysical energy production and oscillation growth discussed in later sections would be seen with this approach, as well, on this problem with coarser grids or in simulations with higher charge to mass ratios.

Simulation with unswitched system specific discontinuity capturing terms added to each system of governing equations

Simulations with unswitched discontinuity capturing terms computed based on applying eqs. 4.95, 4.96, and 4.99 to each set of species fluid equations and the PHM equations, separately were attempted with $\frac{q_i}{m_i} = 10$ with $\gamma_i = \gamma_e = 2$. This approach proved severely overdiffusive and was one of the first Brio-Wu problem setups to be found to fail due to the growth of oscillations in the solution away from the discontinuities, where trouble had usually been encountered in simulations without discontinuity capturing terms. In fig. 7.20 it can be seen that the simulation employing this approach lost energy heavily until about $t = 1.8$, then it started experiencing a lot of nonphysical

energy production, which continued until the simulation failed between $t = 6.2$ and $t = 6.4$. By that time oscillations like those seen in fig. 7.21 for the electron density from a simulation with the switched, full model discontinuity capturing terms, had grown incredibly large. Ultimately it was in the vicinity of oscillations like these that the nonphysical conditions that crashed the algorithm occurred with this approach. The ion and electron densities obtained using this approach are shown at $t = 2.6$ in figs. 7.26 and 7.27, respectively for a selected portion of the problem domain. Fig. 7.28 presents a close up of the small waves in the electron density ahead of the right moving discontinuity at $x \approx 0.54$. The timing of these plots is chosen because it is approximately at the point in time where the solution with this approach has stopped losing energy in the domain and gained energy back up to about the original amount. (See fig. 7.20.) The solution without discontinuity capturing terms is believed to be appropriate except some localized overshooting/ undershooting near the discontinuities at $x \approx 0.52$ and 0.54. In the ion density plot, it can be seen that the addition of this discontinuity capturing term effectively smoothed the discontinuity at $x \approx 0.52$, but also blunted the, likely physical, spike in the solution behind the discontinuity at $x \approx 0.54$. It also completely removed the wavy features around $x \approx 0.5$. In the electron density solution, again the discontinuity at $x \approx 0.52$ is well handled, but more undesirable side effects are seen. The small waves in the electron fluid that run ahead of the discontinuity at $x \approx 0.54$ are completely damped out and the short waves seen between $x \approx 0.3$ and 0.4 are absent, but longer wavelength oscillations can be seen developing at the far left end of the solution which are absent from the other solutions. These additional oscillations are the ones that eventually grow large enough to crash the algorithm.

Simulation with switched system specific discontinuity capturing terms added to each system of governing equations

Simulations with switched discontinuity capturing terms, in which the base discontinuity capturing viscosity was computed based on applying eqs. 4.95, 4.96, and 4.99 to each set of fluid species equations and the PHM equations independently, were performed with $\gamma_i = \gamma_e = 2$ and $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$. In both cases, the method can be seen to have resulted in slightly more energy loss than simulations without discontinuity capturing terms on comparable grids in figs. 7.19 and 7.20. In fig. 7.21 it can be seen that, for the $\frac{q_i}{m_i} = 10$ simulation, slightly more oscillation in the quiescent plasma

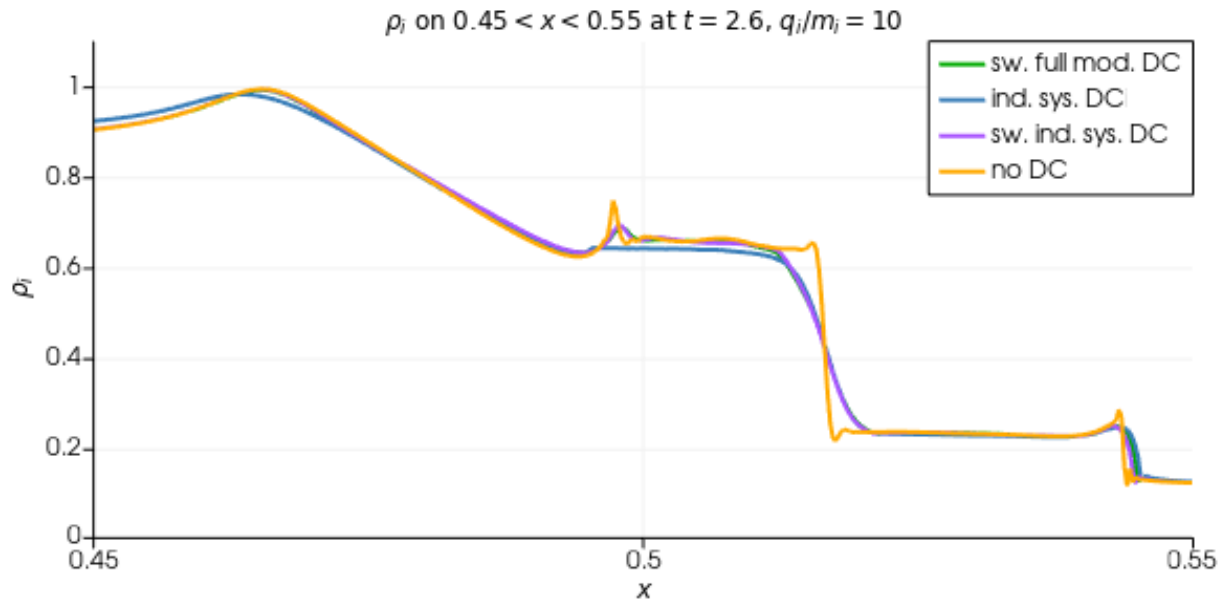


Figure 7.26: Ion density on $0.45 \leq x \leq 0.55$ at $t = 2.6$ for Brio-Wu problem with $\gamma_i = \gamma_e = 2$, $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms. All results used BDF2 time difference and 4000 elements per unit length in the area shown.

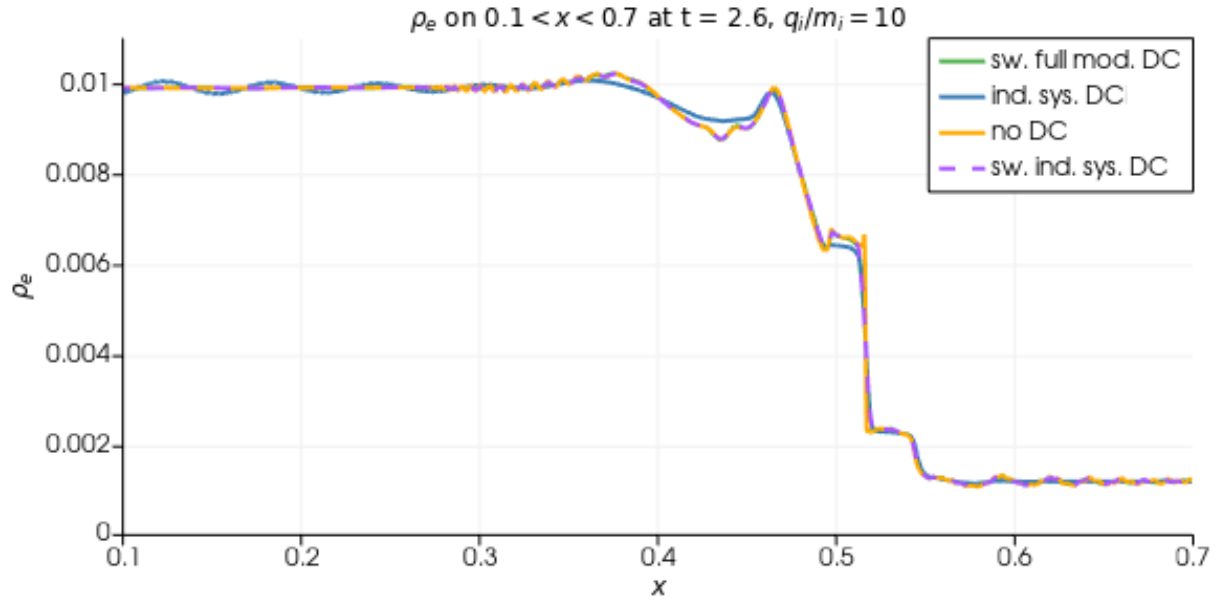


Figure 7.27: Electron density on $0.1 \leq x \leq 0.7$ at $t = 2.6$ for Brio-Wu problem with $\gamma_i = \gamma_e = 2$, $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms. All results used BDF2 time difference and 4000 elements per unit length in the area shown.

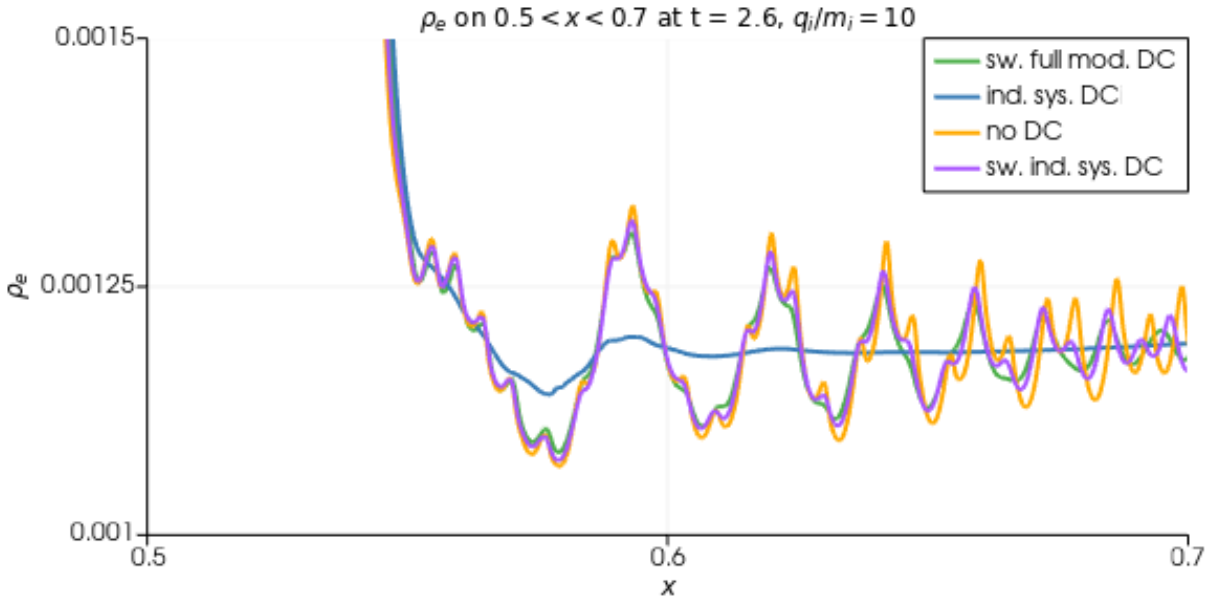


Figure 7.28: Electron density on $0.5 \leq x \leq 0.7$ at $t = 2.6$ for Brio-Wu problem with $\gamma_i = \gamma_e = 2$, $\frac{q_i}{m_i} = 10$ simulated with the SUPG FEM and various discontinuity capturing terms. All results used BDF2 time difference and 4000 elements per unit length in the area shown.

to the left of the domain of interest is seen versus the simulation without discontinuity capturing terms on a comparable grid, but some oscillation in this area could be physical and it does not seem to be growing out of line with what is seen with the other approaches. The base discontinuity capturing viscosities computed in this way are typically slightly lower than the full model viscosity (to be discussed in the next section) for the species fluid equations, but significantly higher for the PHM equations. As reported in ref. [6], the discontinuity sensor and switching function do a fair job of minimizing the discontinuity capturing viscosity away from the discontinuities and quite good performance is seen for the method in simulations with ion charge to mass ratios of 1 and 10. Evidence of this can be seen in the species density plots of figs. 7.26 and 7.27, where it can be seen that many of the small scale features of the $\frac{q_i}{m_i} = 10$ solution obtained without discontinuity capturing terms are maintained, but smoothing of the large discontinuities is achieved comparable to that seen with the unswitched system specific discontinuity capturing viscosities. In fig. 7.28, it can be seen that, of the approaches considered, this method results in the least damping of the small scale waves ahead of the right moving shock at $x \approx 0.54$. Unfortunately, it did not enable successful simulations at higher ion charge to mass ratios at the grid resolutions accessible with the computational resources available.

Simulation with a switched full model discontinuity capturing term

Simulations with the switched full model discontinuity capturing term were successfully ran for scenarios with $\gamma_i = \gamma_e = 2$ and $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 10$ and were attempted with $\frac{q_i}{m_i} = 100$. From the species density plots of figs. 7.26 and 7.27, it can be seen that the method ultimately performs similarly to the use of system specific base discontinuity capturing viscosities already considered. In the close up of the small waves in the electron density ahead of the right moving shock in the $\frac{q_i}{m_i} = 10$ solution, shown in fig. 7.28, it can be seen that these small waves are slightly more damped with this approach than they were with the switched system specific viscosities. From the evolution of the total energy in the domain presented in figs. 7.19 and 7.20 it appears that this approach was one of the best performing in terms of conserving the amount of energy in the domain, when the solution resolution was sufficient. Some nonphysical energy accumulation was seen in the simulation with $\frac{q_i}{m_i} = 10$ on a 1000 element per unit length grid. The fact that this energy accumulation was seen and that the simulation grew much larger oscillations in the

quiescent plasma in the region shown in fig. 7.21 compared to the simulation on the same grid without discontinuity capturing terms indicates that the approach could likely be improved by a sensor function capable of reducing the discontinuity capturing viscosity more rapidly away from the discontinuities. On the other hand, from figs. 7.26, 7.27, and 7.28, it would appear that the discontinuity sensor has done a respectable job, since the discontinuities are smoothed about as much as with the unswitched discontinuity capturing terms, but the small waves are damped much less than they were with the unswitched discontinuity capturing terms. Additionally, plots of the viscosities obtained at the final time of the $\frac{q_i}{m_i} = 10$ simulation and their unswitched values are presented in ref. [6] and indicate that the viscosity is indeed reasonably modulated. It has been noted that, regardless of the base viscosity employed, the sensor function employed tends to impede convergence if the switch threshold, f_s, is moved away from zero. A sensor function that allows the discontinuity capturing term to remain completely deactivated over a greater range of conditions may enable all of these switched discontinuity capturing terms to perform better.

Examination of the numerical oscillation behavior in comparison to a finite volume solution

Reports of the growth of oscillations in the quiescent plasma far from the "interesting" part of this problem and the seemingly concurrent energy accumulation were not seen in the literature, raising the question as to whether or not this is an issue particular to the numerical methods employed. The only mention of difficulty seen was in ref. [9], where it was noted that seemingly excessive grid resolution had been used in simulations of this problem due to a need to resolve fine spatial scales. There was also a mention of more waves emerging in the solution as less diffusive methods and higher resolution grids were employed in ref. [4]. To investigate this, a simple, serial, one dimensional finite volume solver was developed for the model and used to simulate the problem with $\frac{q_i}{m_i} = 1$ and $\gamma_i = \gamma_e = 2$ for comparison to the results previously described. The $\frac{q_i}{m_i} = 1$ scenario was chosen and the grid was coarsened to see if the same issues seen with higher ion charge to mass ratios could be obtained in a less computationally expensive scenario. In the plot of results obtained using SUPG in fig. 7.29 , it can be seen that on a grid of 50 elements per unit length (1000 elements total) the high frequency oscillation seen in the 1000 element per unit length (20000 elements total) grid is damped out and the larger wavelength oscillations are beginning to

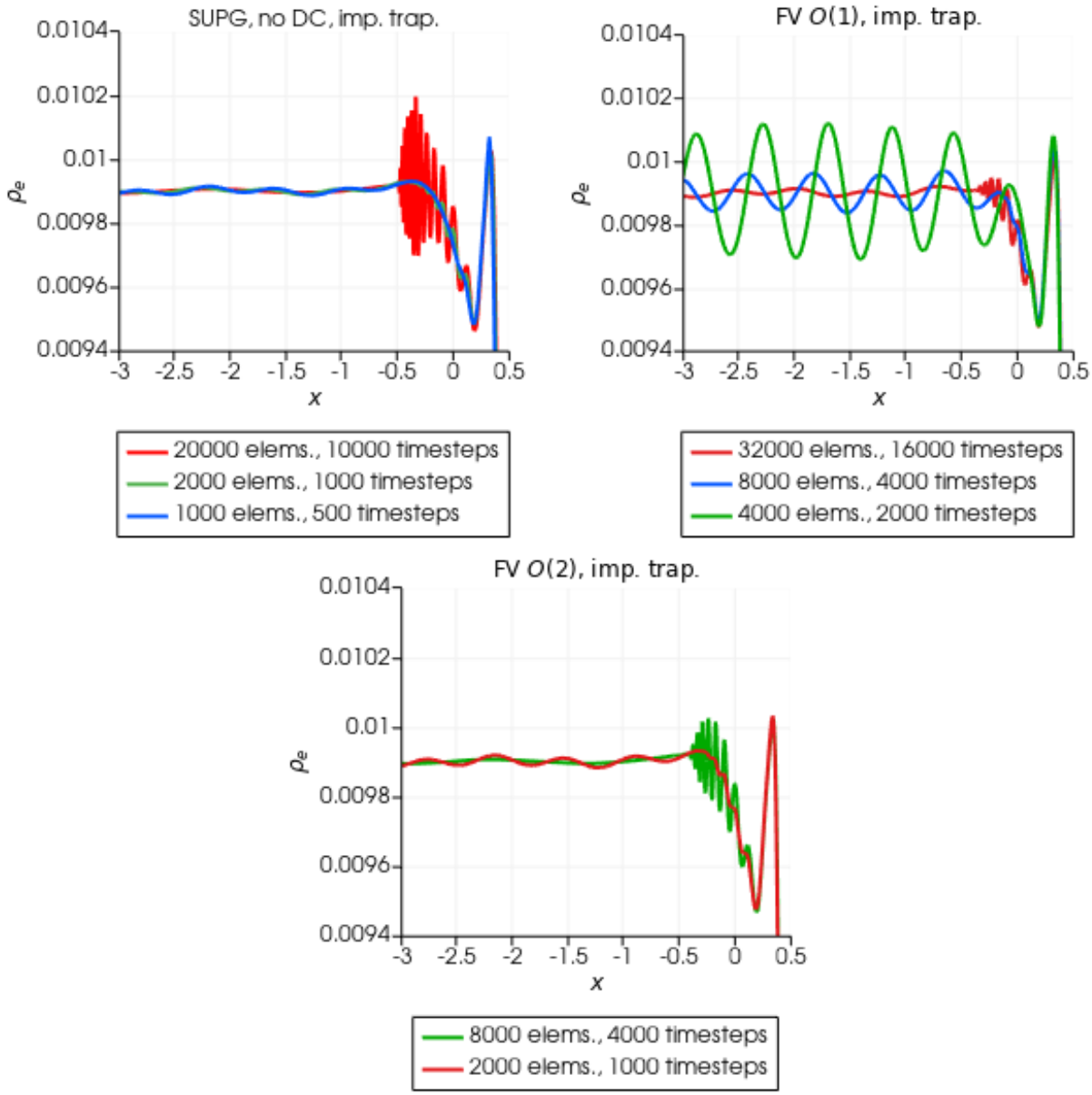


Figure 7.29: Electron density computed on grids of differing resolution using the SUPG FEM without a discontinuity capturing term and first and second order finite volume schemes plotted for $-3 \leq x \leq 0.5$ at $t = 10$ for Brio-Wu problem with $\frac{q_i}{m_i} = 1$ with $\gamma_i = \gamma_e = 2$.

appear in the electron fluid further to the left, where the finer grid solutions suggest that it should remain quiescent. In the remaining plots of fig. 7.29, the finite volume approach can be seen to produce the same result; as the grid is coarsened, the short wavelength waves in $-0.5 \leq x \leq 0.25$ become damped and the problematic longer wavelength waves in the quiescent fluid further left grow larger. A common thread is that all of the results showing this behavior have been obtained with implicit time differences, and these in particular, the implicit trapezoid rule. However, earlier work with the SUPG FEM had found them to occur when the BDF2 time difference is used. The implicit Euler time difference does help damp these out. Even with the first order finite volume method, use of the implicit Euler time difference will damp these oscillations out on grids as coarse as 100 elements per unit length, but that comes with the drawbacks of reducing the method to first order accuracy in time and the heavy numerical diffusion of the implicit Euler time difference. Another interesting feature is that while the more diffusive time difference helped, more diffusive algorithms on the spatial side seem to make the oscillations worse. Both the use of discontinuity capturing terms with SUPG and the use of a first order, rather than second order finite volume scheme result in these oscillations becoming larger for simulations of comparable grid resolution and timestep size.

Examination of performance in comparison to a finite volume scheme

Unfortunately, no references could be found that reported un-normalized run times or the hardware specifics necessary for performance comparisons. However, since a serial finite volume code was ultimately implemented in this work, some examination of the relative performance of these algorithms is possible, though only on problems small enough to be handled by the one dimensional, serial finite volume code developed. Both approaches were used for the two fluid analog of the Brio-Wu problem with $\frac{q_i}{m_i} = 1$ and $\gamma_i = \gamma_e = 2$ with the implicit trapezoid time difference on $-9.5 \leq x \leq 10.5$ with 100 elements per unit length and 1000 timesteps. Newton iteration was used in the finite volume code whereas pseudotimestepping was used in the SUPG code and the SUPG code was ran on 4 processes, whereas the finite volume code is only capable of serial execution. Linear elements were used in the SUPG simulations, limited least squares linear reconstruction was used in the finite volume code. The same desktop computer with an Intel ®Core ™ i7-8700 CPU with 6 cores and a 3.2 GHz base frequency was used in both cases.

On the same grid, the solutions with each method appeared to be of comparable quality. The SUPG solution without discontinuity capturing terms was slightly less diffusive than the finite volume solution with limited least squares reconstruction while the SUPG solution with the switched full model discontinuity capturing term was somewhat more diffusive than the finite volume scheme. In terms of computational cost, the run times for the SUPG code were about 2100 seconds without discontinuity capturing terms and 2500 seconds with the switched full model discontinuity capturing terms and the simulation without discontinuity capturing terms completed in 7854 total iterations while the one with the aforementioned discontinuity capturing terms completed in 7285 total iterations. The finite volume code, on the other hand took 7480 total iterations to complete and those were performed in about 6800 seconds. In this case, the SUPG method without discontinuity capturing terms was about 20% slower overall than the finite volume method and with discontinuity capturing terms it was about 50% slower overall. On a per-iteration basis, the result is roughly the same, though SUPG without discontinuity capturing terms looks slightly better due to having resulted in more iterations than the finite volume method while the SUPG method with discontinuity capturing terms appears slightly better due to the same effect.

Unfortunately, this is still not a complete "apples-to-apples" comparison of the algorithms because the SUPG code is parallelized and has the componentry necessary to handle multi-spatial dimensional problems on non-trivial unstructured grids, whereas the finite volume code is serial and uses only simply constructed one dimensional grids. The additional features of the SUPG code, thus, add some computational cost that has no benefit on this simple problem. A simple one-dimensional serial SUPG code would certainly erase some of the deficit seen here.

7.1.3 GEM challenge problem

The GEM challenge problem was simulated using the SUPG method and evolving discontinuity capturing term calculations many times over the course of this work. The results and discussion presented here largely echo previously published descriptions of this work found in refs. [64] and [6]. Time levels and timestep sizes are stated scaled to the more common time interval, $0 \leq t \leq 40$, rather than that resulting from the nondimensional scaling group free, speed of light reference velocity based nondimensionalization actually used to simplify comparison to existing results found in refs. [4, 9, 12]. The data plots are generated using Paraview [61] visualize data written to VTK

format files by the solver, the plots of reconnected flux and total energy in the domain are also post-processed from the solver generated data using Paraview. Some temporal detail is lost due to the fact that the data sets are too large to save for every timestep, data was typically output every 2 time units.

Simulation without a discontinuity capturing term

Simulations using the SUPG method with BDF2 the time difference and no discontinuity capturing term were performed on linear quadrilateral grids with element counts of 100×200, 200×400, and 400×800 with uniform node spacing and the larger element count in the x direction such that each grid was ultimately composed of equal sized square elements. Simulations on the 100×200 grid were carried out with timestep sizes of both 0.002 and 0.004, while the simulations on the 200×400 and 400×800 grid were carried out with a timestep size of 0.002, only. Some variations to the algorithm, such as returning to the Galerkin formulation when $\boldsymbol{\tau}^{-1}$ inversion failed versus use of the pseudoinverse, and the use of a fully coupled versus loosely coupled treatment of the global system, had a minor impact on the failure time of the simulations but all of these simulations failed just as the reconnection rate was ramping up, typically in the range of $t = 11$ to $t = 16$, with the variation due more to grid and timestep size than the algorithm modifications. All of these simulations failed due to the emergence of node to node oscillations in the solution around the thinning current layer between the magnetic islands that resulted in conditions where one of the fluids' internal energies became negative. (This is an unphysical situation for the constant specific heat internal energy relations used.) The reconnected flux histories for these simulations can be seen in fig. 7.30, an example of the conditions at failure can be seen in fig. 7.31.

The added numerical diffusion of the implicit Euler time difference did not improve the situation substantially, these simulations were also stopped near the onset of fast reconnection due to the development of unphysical conditions resulting from similar oscillations around the thinning current layer. These simulations were also carried out on 100×200, 200×400, and 400×800 element grids with a timestep of $\Delta t = 0.002$. The resulting reconnected flux time histories are shown in fig. 7.32; it can be seen that the flux reconnection behavior obtained is very similar to that obtained using the BDF2 time difference.

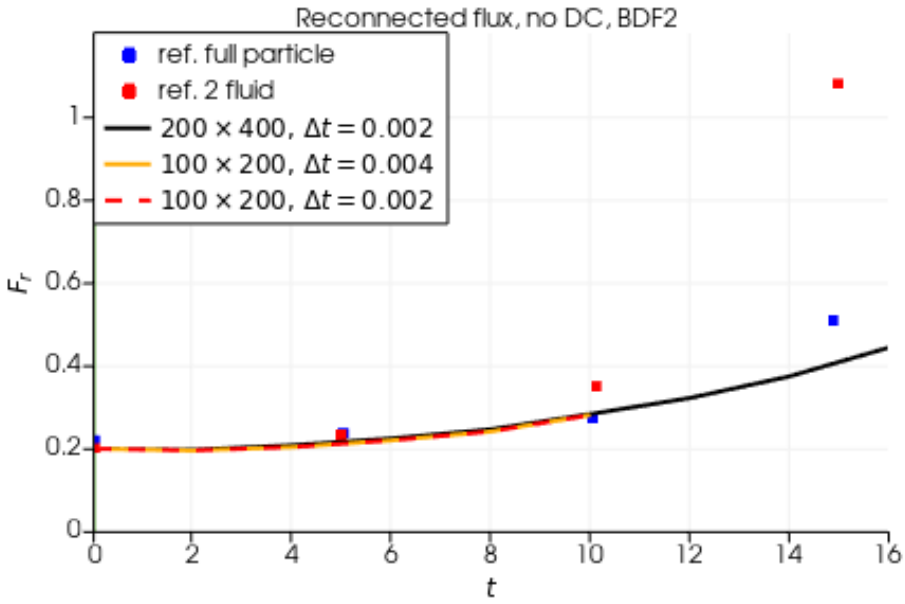


Figure 7.30: Reconnected flux time histories for simulations using the SUPG method with BDF2 time difference and without discontinuity capturing terms.

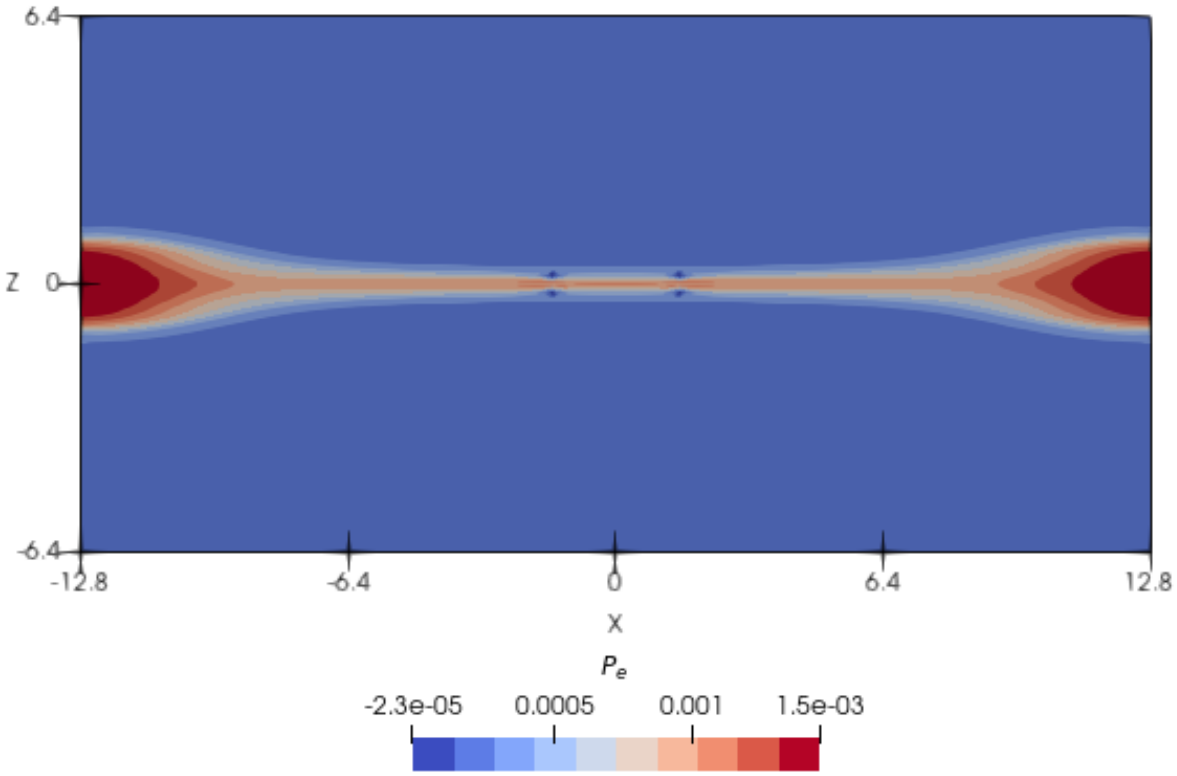


Figure 7.31: GEM challenge electron pressure field at $t = 16.704$ showing negative pressures in four small areas where node to node oscillations around the thinning current layer caused unphysical conditions that crashed the algorithm for the simulation with no discontinuity capturing terms and BDF2 on a 200×400 grid with $\Delta t = 0.002$.

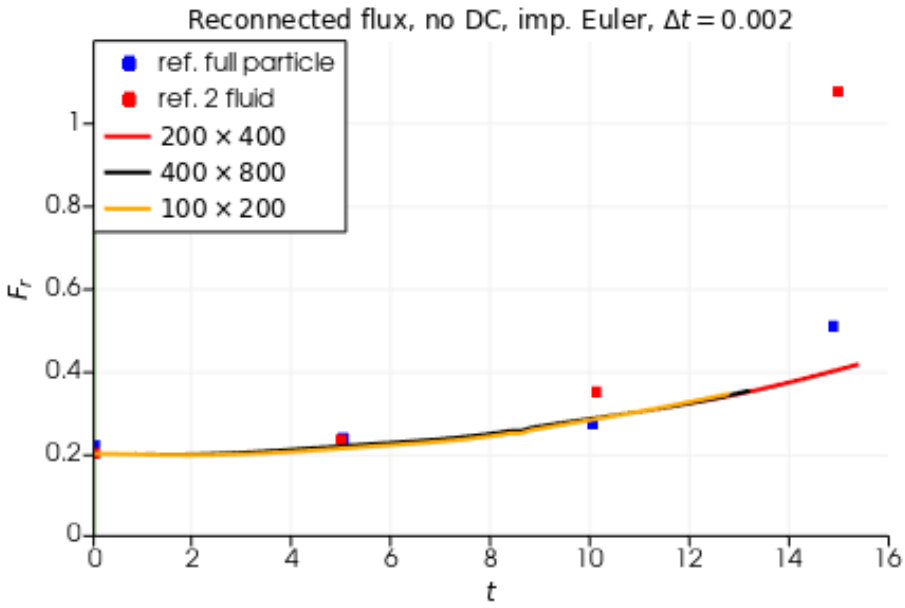


Figure 7.32: Reconnected flux time histories for simulations using the SUPG method with implicit Euler time difference and without discontinuity capturing terms. The data series labelled "ref. full particle" were measured from the figure in ref. [12], the data series labelled "ref. 2 fluid" were measured from the "512 × 256 DG3" data series of fig. 5.17 in ref. [4].

Simulation with discontinuity capturing terms added to the fluid equations

The first approach taken to discontinuity capturing was to add terms as defined by eqs. 4.95, 4.96, and 4.99 to each species' fluid equations. These terms were computed independently based only on the conditions of the fluid to which the term was being added. Simulations were performed using the term unswitched, or exactly as defined by eqs. 4.95, 4.96, and 4.99 and switched; with the viscosity being modulated by the smooth switching function of eq. 4.100 based on the discontinuity sensor function of eq. 4.101 with threshold $f_s = 0$ and switch width $f_e - f_s = 1 \times 10^{-3}$.

Simulations with the unswitched viscosities were attempted on 200×400 and 400×800 linear quadrilateral element grids with timestep sizes of $\Delta t = 0.002$ and $\Delta t = 0.004$ using both the implicit Euler time difference and BDF2. Simulations using the BDF2 time term did not reach the onset of fast reconnection, but a complete duration simulation was obtained using the implicit Euler time term on the 400×800 grid with $\Delta t = 0.002$. The simulation with the larger timestep on the 400×800 grid came very close to completion, failing at $t = 37.344$ due to the internal energy of the electron fluid becoming negative. On the coarser mesh, the simulations employing the implicit Euler time term both failed at $t \approx 24$. The time histories of the reconnected flux for these simulations can be seen in fig. 7.33 while the time history of the total energy in the domain can be seen in fig. 7.34. For these simulations, the onset of fast reconnection appears to be coming slightly later than what was seen in refs. [4, 9, 12] and the expected final reconnected flux value of approximately 3.4 [12] was significantly overshot. A similar overshoot was seen in coarser mesh simulation results presented in ref. [4]. This overshoot is often associated with the formation of a secondary magnetic island, a feature which is apparent in fig. 7.35. Based on the behavior seen with the DG FEM in ref. [4], it is reasonable to believe this is happening because sufficient resolution has not been achieved and that it would be resolved with further grid refinement and/or the use of higher order elements. From fig. 7.34, it can be seen that some energy loss occurs during fast reconnection in the finer mesh simulations, but a small energy accumulation occurs in the coarse mesh simulation with $\Delta t = 0.004$ and a large energy accumulation occurs in the coarse mesh simulation with $\Delta t = 0.002$. The energy accumulation seen in these coarser mesh solutions appears to be associated with the growth of large scale oscillations in what should be relatively quiet, lower density plasma between the current layer and the $\pm z$ boundaries. Fig. 7.36 presents a solution

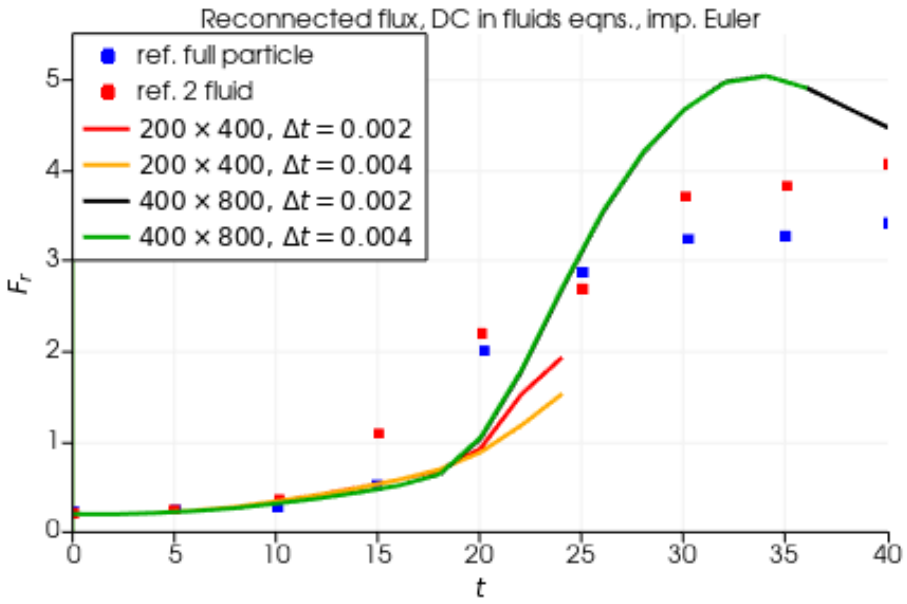


Figure 7.33: Reconnected flux histories for simulations employing unswitched discontinuity capturing terms in the species fluid equations. The data series labelled "ref. full particle" were measured from the figure in ref. [12], the data series labelled "ref. 2 fluid" were measured from the "512 × 256 DG3" data series of fig. 5.17 in ref. [4].

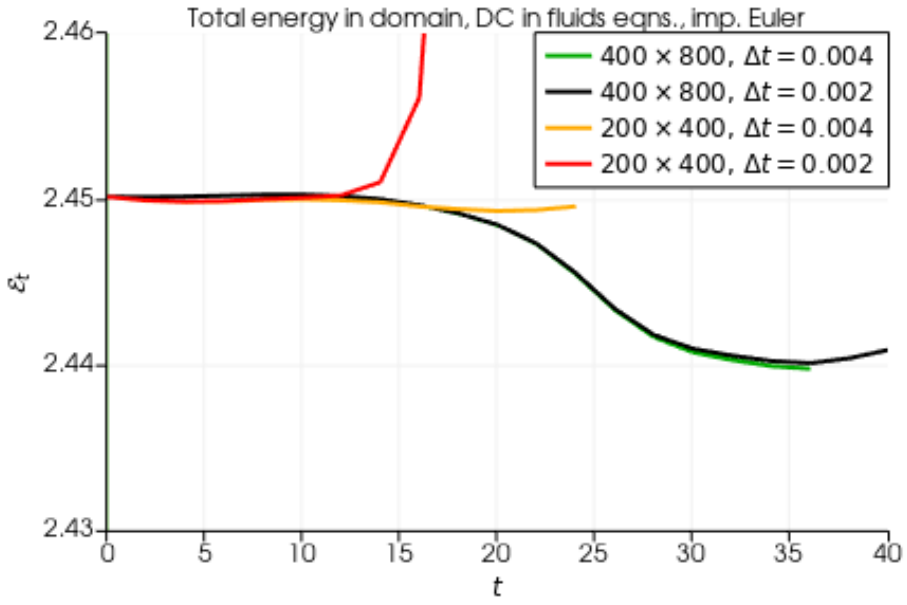


Figure 7.34: Evolution of the total energy in the domain during simulations employing unswitched discontinuity capturing terms in the species fluid equations.

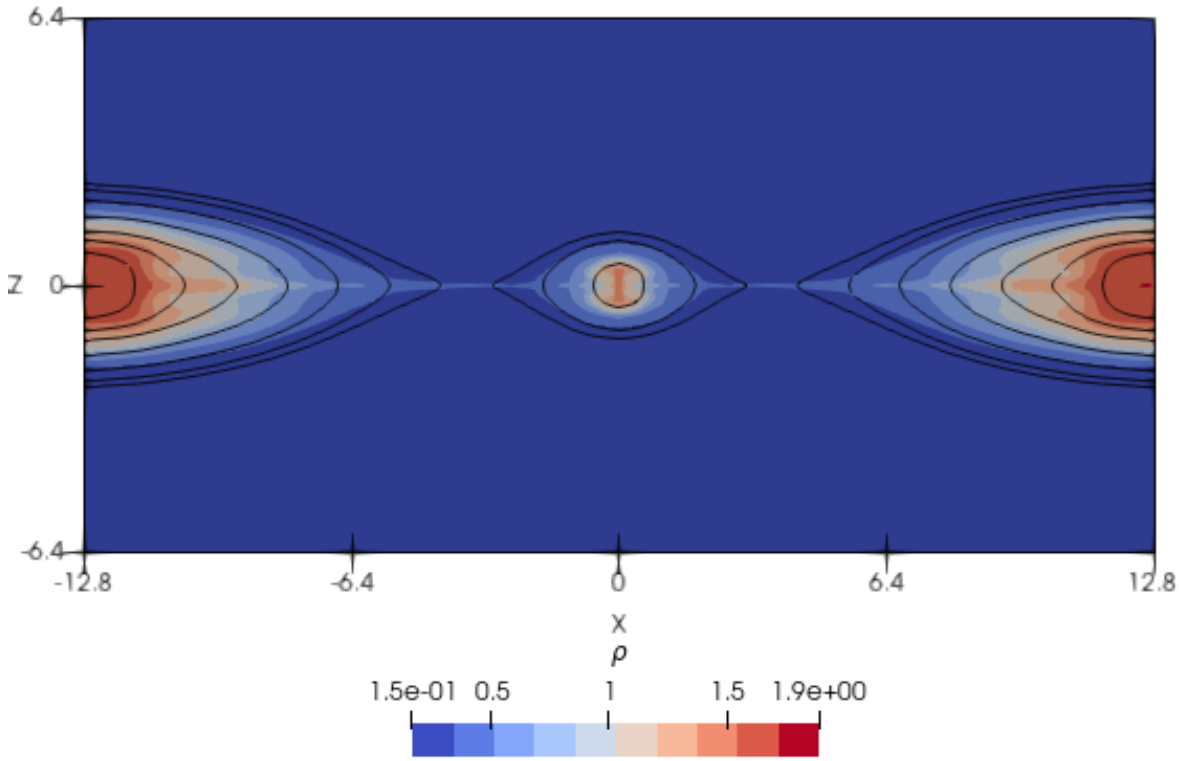


Figure 7.35: Density and in-plane magnetic field at $t = 24$ from simulation on a 400×800 mesh with $\Delta t = 0.002$ using the implicit Euler time term and unswitched discontinuity capturing terms in the fluids equations. The color field indicates the plasma density, the black lines are vector field tracers seeded along $z = 0$ and following the in-plane components of B.

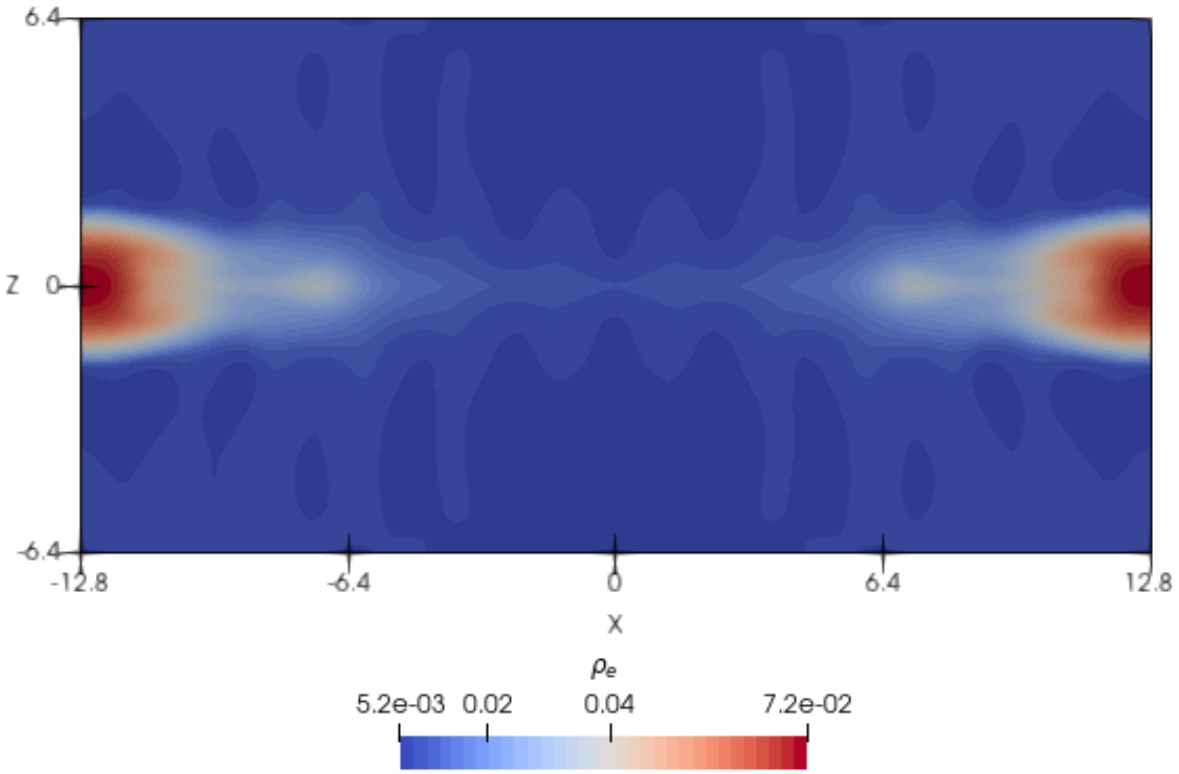


Figure 7.36: Electron density at $t = 24$ from simulation on a 200×400 mesh with $\Delta t = 0.002$ using the implicit Euler time term and unswitched discontinuity capturing terms in the fluids equations.

snapshot in which the oscillations are evident in the electron density. The electron density is one of the fields in which these oscillations are strongest. The growth of these oscillations appears to be what eventually leads to the unphysical conditions that stopped the simulations, and a similar behavior was seen in the simulations that used the BDF2 time difference. The phenomenon appears comparable to the oscillation growth seen in the quiet plasma outside the original MHD domain of the Brio-Wu problem and was found to occur in another simple Riemann problem involving an initial discontinuity in the transverse magnetic field only that was explored in an appendix of ref. [64].

The simulations with switched discontinuity capturing terms added to each species' fluid equations were performed on a 200×400 linear quadrilateral element mesh with the implicit Euler time term and a timestep of $\Delta t = 0.002$ and with the BDF2 time term and timesteps of $\Delta t = 0.002$ and $\Delta t = 0.001$. The simulation employing the implicit Euler time term was successful in completing the problem time interval, but both simulations employing the BDF2 time term failed at $t \approx 24$. The time evolution of the reconnected flux and total energy in the domain for these simulations can be seen in figs. 7.37 and 7.38, respectively. The simulation that used the implicit Euler time difference can be seen to have performed similarly to the same scheme with unswitched discontinuity capturing terms on the much finer 400×800 grid in terms of both the reconnected flux evolution and the evolution of the total energy in the domain. Slightly more energy loss is seen, but this may be due to the presence of unphysical energy production in parts of the solution obtained with unswitched discontinuity capturing terms, rather than worse dissipation in these simulations. The large scale background oscillations seen in the coarser mesh simulations with unswitched discontinuity capturing terms are not apparent in any of these results; however, the slight energy gain seen at the end of the simulation that employed implicit Euler time differences indicates that the phenomenon may still be present, but just growing more slowly due to the improved solution resolution. Plots of the discontinuity capturing viscosities obtained from the BDF1 simulation results were presented in ref. [64] and indicate that the sensor function and switching function did a fair job of reducing the viscosity away from the areas of large solution gradients in the vicinity the collapsing current layer and on the edges of the magnetic islands. In the results of the simulations that employed the BDF2 time difference, the development of two secondary magnetic islands along the current layer was observed prior to failure, rather than one

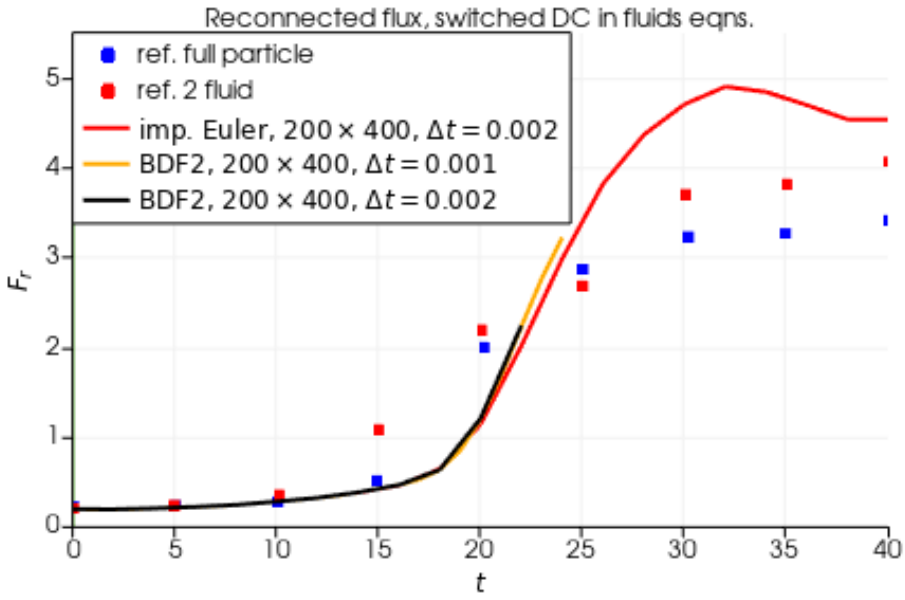


Figure 7.37: Reconnected flux histories for simulations employing switched discontinuity capturing terms in the species fluid equations. The data series labelled "ref. full particle" were measured from the figure in ref. [12], the data series labelled "ref. 2 fluid" were measured from the "512×256 DG3" data series of fig. 5.17 in ref. [4].

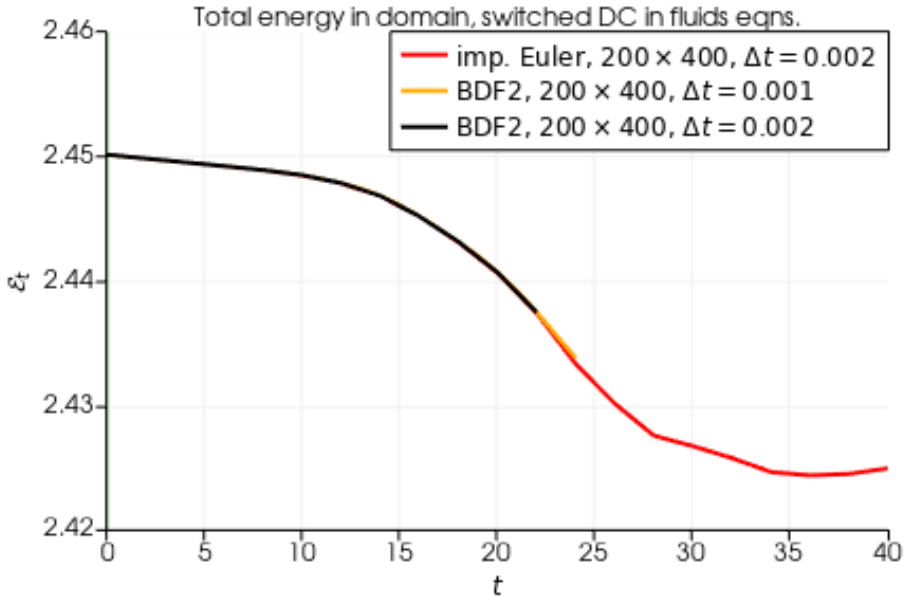


Figure 7.38: Evolution of the total energy in the domain during simulations employing switched discontinuity capturing terms in the species fluid equations.

additional magnetic island that was seen in the implicit Euler time difference simulations. This is illustrated in fig. 7.39, where two areas of increased density are seen along the current layer, rather than the one that was seen in fig. 7.35. The unphysical conditions that caused these simulations to end were seen in a small region at the center of the domain and don't appear to be associated with large scale oscillations like those seen with the unswitched discontinuity capturing terms or the node-to-node oscillations across the current layer that were seen without any discontinuity capturing terms.

Simulation with system specific discontinuity capturing terms added to the species fluid equations and the PHM equations

The next approach to the discontinuity capturing terms that was explored was the addition of terms defined by eqs. 4.95, 4.96, and 4.99 to all three sets of governing equations, both sets of fluid species equations and the PHM equations, with each set of equations having its own ν_D computed based on its flux Jacobians and solution gradients. This approach was used on 200×400 and 400×800 linear quadrilateral grids with BDF2 and a timestep of $\Delta t = 0.002$. The scheme was able to complete the simulation interval, but proved incredibly diffusive; flux reconnection and total energy evolutions are shown in figs. 7.40 and 7.41, respectively. While schemes that are overdiffusive can result in excessive reconnection, due to the formation of the secondary magnetic island as seen in the previously presented results, after a point excessive diffusion can suppress reconnection [12], and this is seen in these results. Much of the loss of reconnected flux is due to decay of the magnetic field strength for this approach, and while there was an overall loss of energy in the domain, it can be seen that both simulations experienced periods of energy generation and the large scale oscillations outside the current layer and magnetic islands were seen in these results. This is like those noted in the results of simulations with unswitched discontinuity capturing terms in the species fluid equations.

When applying switching to this discontinuity capturing viscosity, there are two fundamental approaches, one is to switch each set of governing equations' discontinuity capturing terms independently, or to switch them all simultaneously. The latter approach was chosen here as the independently switched terms in the fluid equations had not proven particularly robust. For a sensor function, the value of eq. 4.101 was computed for both fluid species and the results were summed and used as a

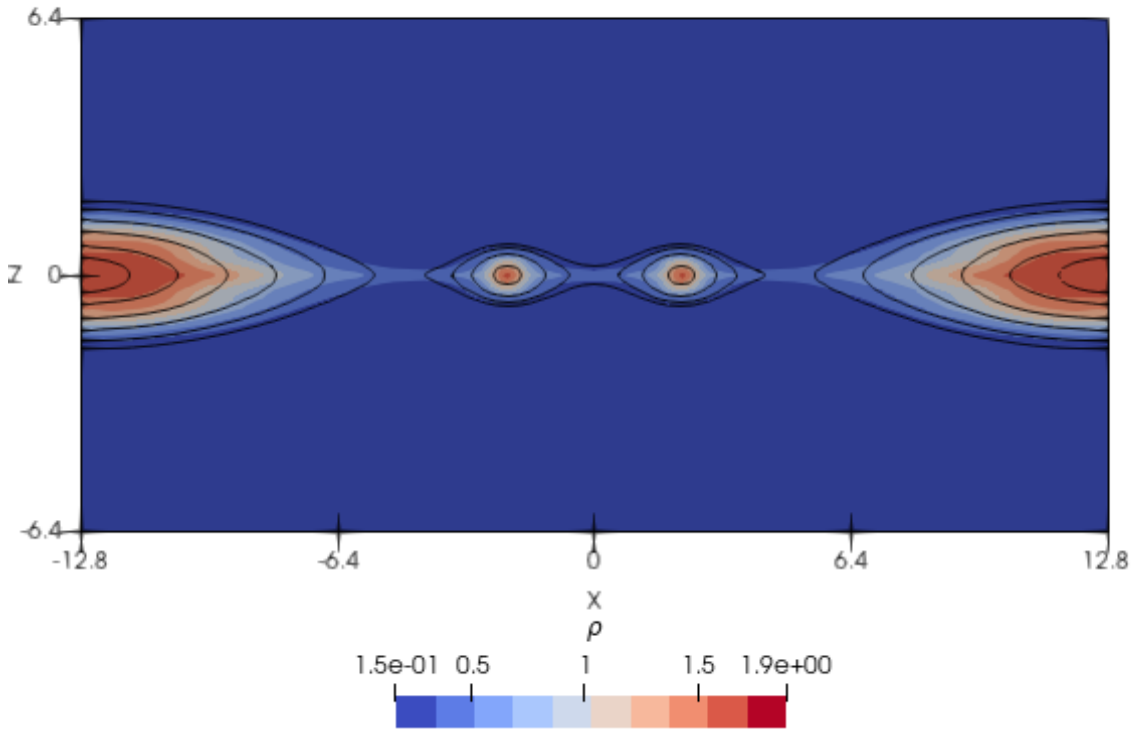


Figure 7.39: GEM challenge density at $t = 24$ obtained using BDF2 and the switched discontinuity capturing terms in the fluid equations showing the development of two secondary magnetic islands. The color field indicates the plasma density, the black lines are vector field tracers seeded along $z = 0$ and following the in-plane components of B.

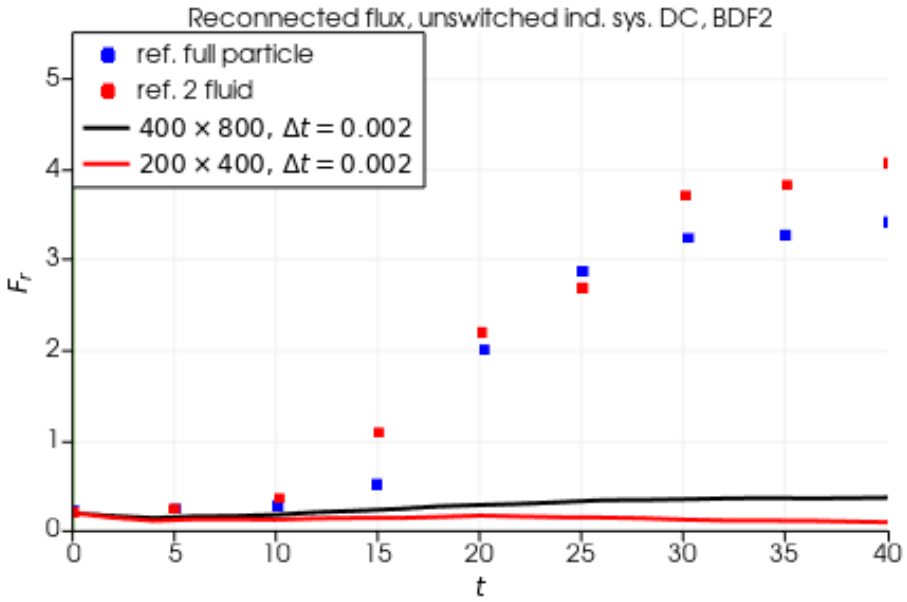


Figure 7.40: Reconnected flux evolution for simulations using BDF2 time terms and discontinuity capturing terms defined by eqs. 4.95, 4.96, 4.99 independently computed for each set of species fluid equations and the PHM equations. The data series labelled "ref. full particle" were measured from the figure in ref. [12], the data series labelled "ref. 2 fluid" were measured from the "512×256 DG3" data series of fig. 5.17 in ref. [4].

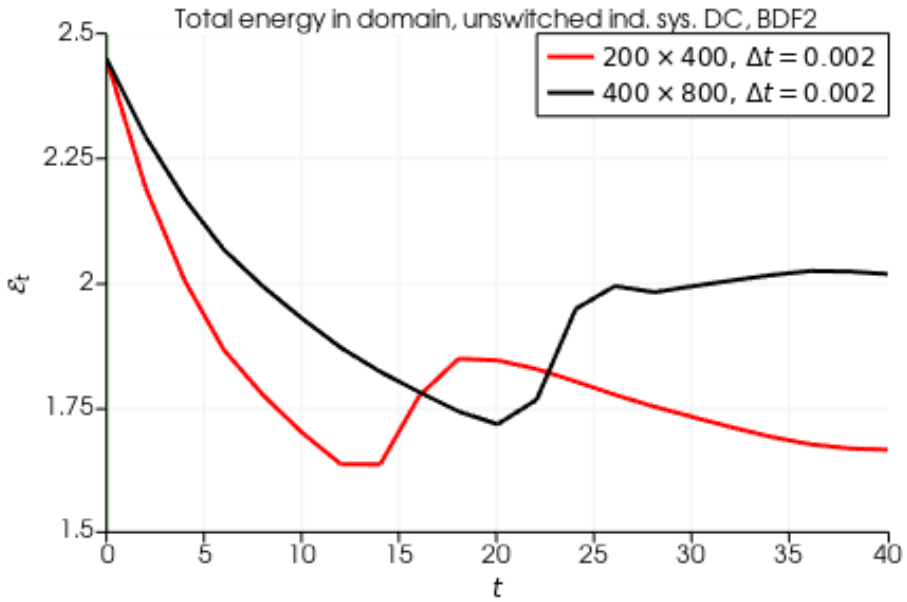


Figure 7.41: Time evolution of total energy in the domain for simulations using BDF2 time terms and discontinuity capturing terms defined by eqs. 4.95, 4.96, 4.99 independently computed for each set of species fluid equations and the PHM equations.

single sensor function for the whole model. The same threshold and switch width that was used with the fluid species discontinuity capturing terms is employed here ($f_s = 0$, $f_e - f_s = 1 \times 10^{-3}$). With this approach, conditions in either fluid warranting a full strength discontinuity capturing viscosity would result in the use of full strength viscosity. Simulations using BDF2 time terms, a timestep of $\Delta t = 0.002$, and 100×200, 200×400, and 400×800 linear quadrilateral grids were undertaken, and the method was able to complete the full time interval in each case. The time evolution of the reconnected flux and total energy in the domain can be seen in figs. 7.42 and 7.43, respectively.

From inspection of these results, it can be seen that the method is quite dissipative and under predicts the amount of flux reconnection, except on the finest grid. The reconnected flux prediction is quite good on the finest mesh, but this is not believed to be grid independent because there is such a significant change from the 200×400 mesh solution and there is still a significant energy loss. On a positive note, the artificial energy production and large scale background oscillations are not seen except in the coarsest grid solution. The fine grid solution obtained by this approach is one of the best obtained in this work and can be seen in fig. 7.44 to not contain a secondary magnetic island and to contain some more complex small scale structure like that seen in the results presented in refs. [4, 5, 9].

Figs. 7.45, 7.46, and 7.47 present the switched and unswitched values of the discontinuity capturing viscosity for each set of governing equations for the final time computed based on the solution computed with switched system specific viscosities on the 200×400 grid with $\Delta t = 0.002$.

It can be seen that the switching function has greatly reduced the viscosity except in a few concentrated areas for both the PHM equations and electron equations. For the ion equations, the viscosity was not very high over most of the domain in the first place, but the switching function did further localize it. The switching function has the same value for each viscosity in this case due to the simultaneous switching approach; thus the differences in the apparent activation regions would be due to differences in the base viscosities.

Simulation with a switched full model discontinuity capturing term

The final approach explored here is the use of a switched discontinuity capturing term in which a single discontinuity capturing viscosity is computed by applying eq. 4.99 considering both species' fluid equations and the PHM equations as one combined set of balance laws. This viscosity is

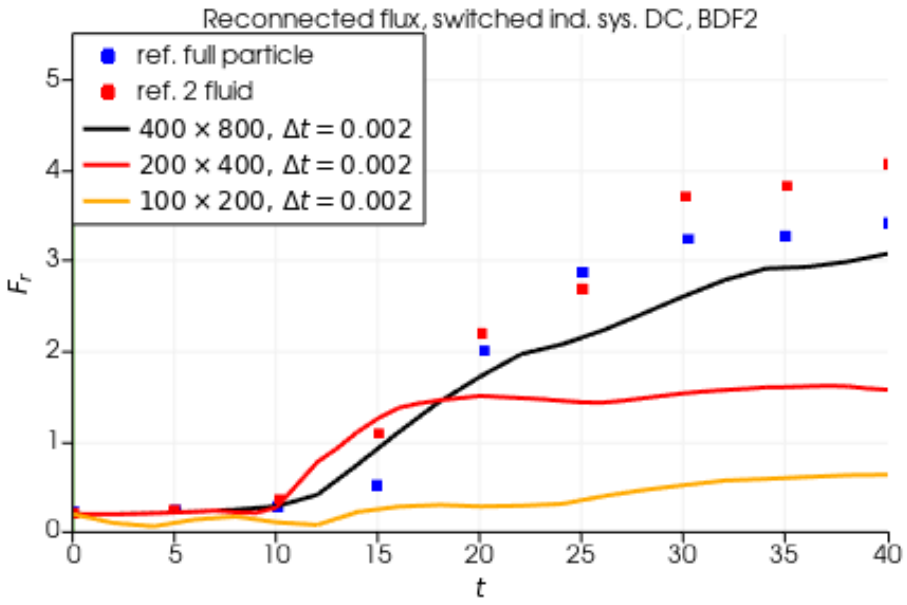


Figure 7.42: Reconnected flux evolution for simulations using BDF2 time terms and simultaneously switched discontinuity capturing terms defined by eqs. 4.95, 4.96, 4.99 independently computed for each set of species fluid equations and the PHM equations and smoothly switched using eq. 4.100 based on the sum of functions defined by eq. 4.101 computed for each fluid species. The data series labelled "ref. full particle" were measured from the figure in ref. [12], the data series labelled "ref. 2 fluid" were measured from the "512 × 256 DG3" data series of fig. 5.17 in ref. [4].

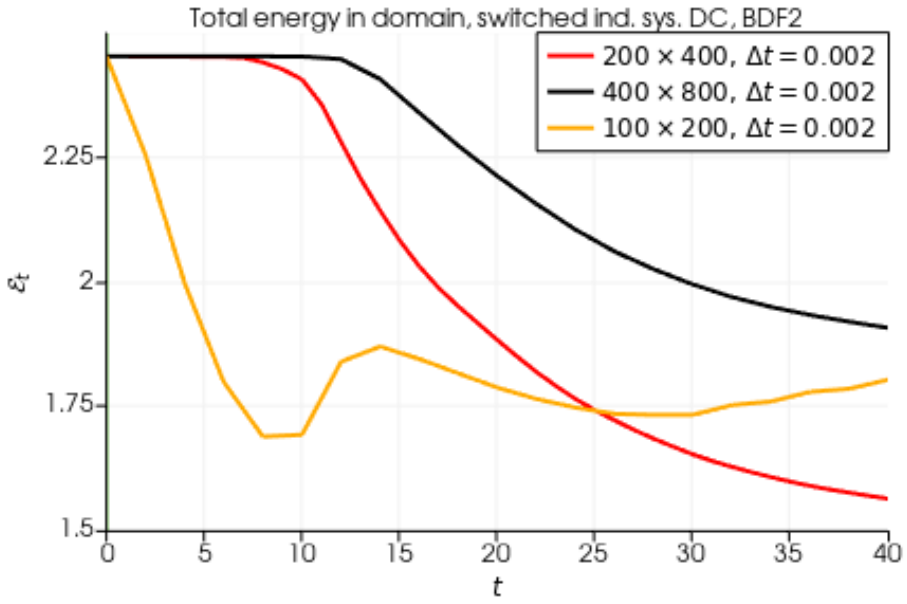


Figure 7.43: Time evolution of total energy in the domain for simulations using BDF2 time terms and simultaneously switched discontinuity capturing terms defined by eqs. 4.95, 4.96, 4.99 independently computed for each set of species fluid equations and the PHM equations and smoothly switched using eq. 4.100 based on the sum of functions defined by eq. 4.101 computed for each fluid species.

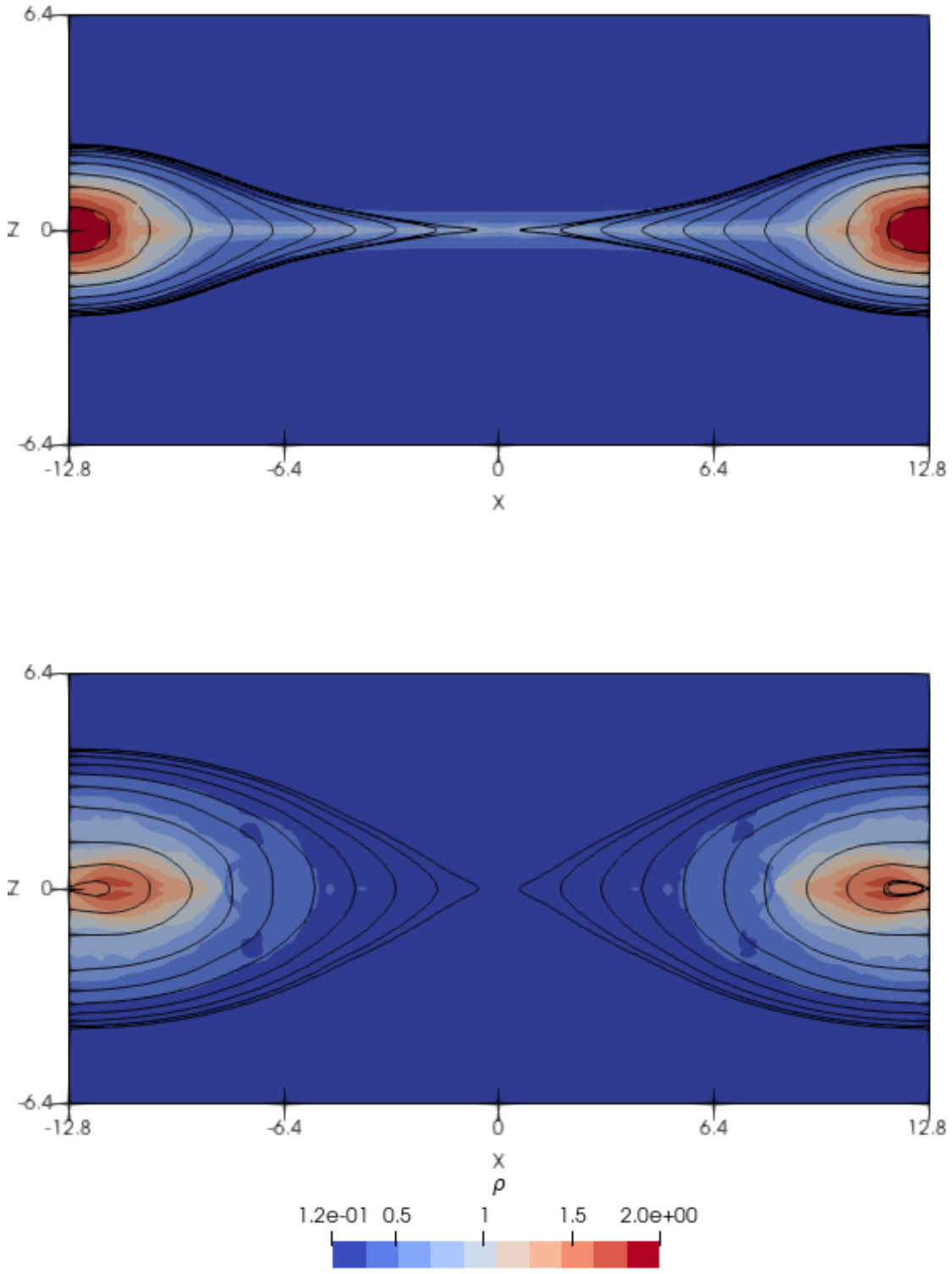


Figure 7.44: GEM challenge density and in plane magnetic field field at times $t = 20$ (top) and $t = 40$ (bottom) computed using switched system specific discontinuity capturing terms on a 400×800 mesh with $\Delta t = 0.002$. The color field indicates the plasma density while the black lines are vector field tracers seeded along $z = 0$ and following the B field.

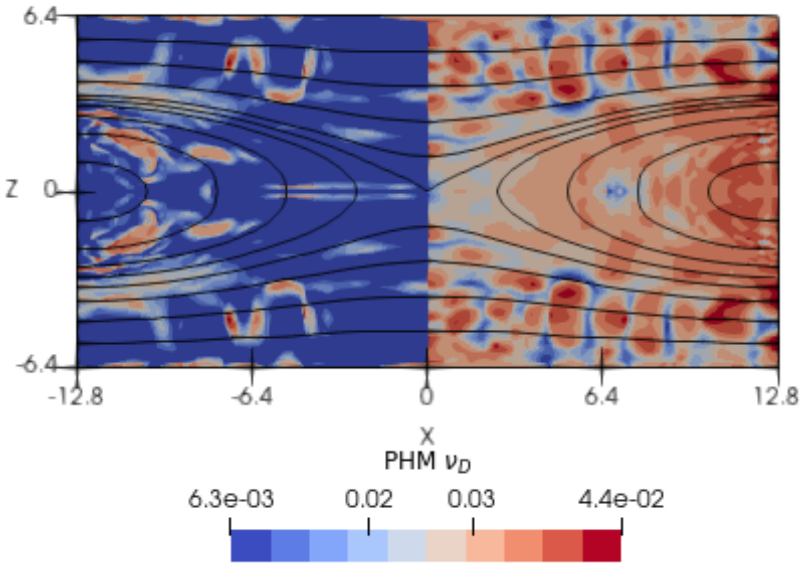


Figure 7.45: System specific viscosity for PHM equations switched (left) and unswitched (right) computed at $t = 40$ based on solution computed with switched system specific discontinuity capturing term on a 200×400 element grid with $\Delta t = 0.002$ and corresponding plasma density field. The color field indicates the discontinuity capturing viscosity magnitude while the black lines are vector field tracers seeded along $z = 0$ and $x = 0$ and following the in plane components of the B field.

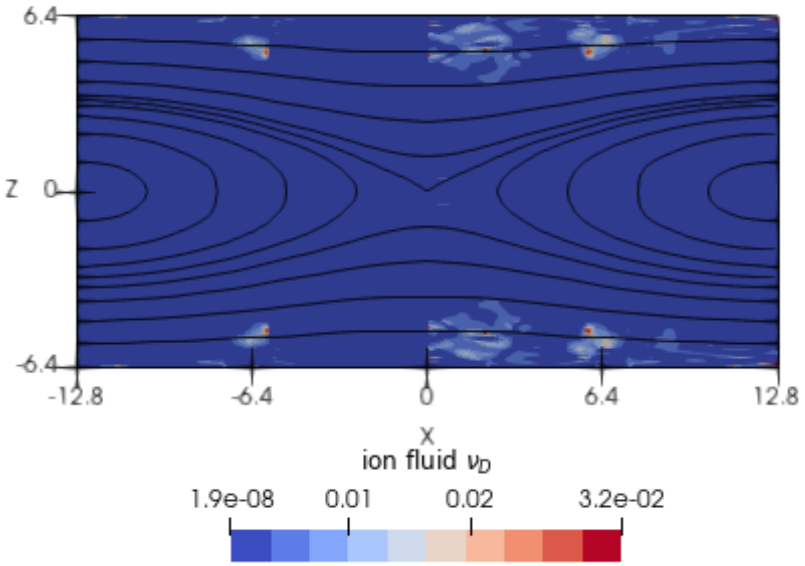


Figure 7.46: System specific viscosity for ion fluid equations switched (left) and unswitched (right) computed at $t = 40$ based on solution computed with switched system specific discontinuity capturing term on a 200×400 element grid with $\Delta t = 0.002$ and corresponding plasma density field. The color field indicates the discontinuity capturing viscosity magnitude while the black lines are vector field tracers seeded along $z = 0$ and $x = 0$ and following the in plane components of the B field.

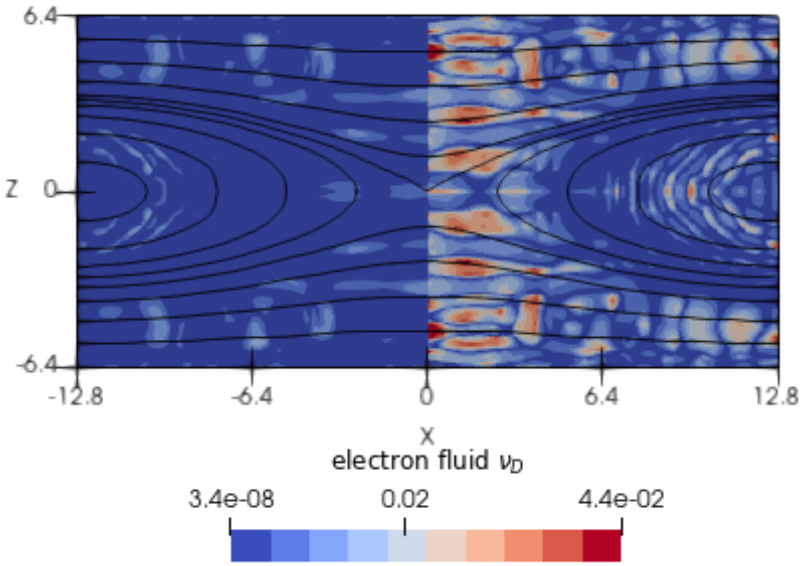


Figure 7.47: System specific viscosity for electron fluid equations switched (left) and unswitched (right) computed at $t = 40$ based on solution computed with switched system specific discontinuity capturing term on a 200×400 element grid with $\Delta t = 0.002$ and corresponding plasma density field. The color field indicates the discontinuity capturing viscosity magnitude while the black lines are vector field tracers seeded along $z = 0$ and $x = 0$ and following the in plane components of the B field.

switched using eq. 4.100 with threshold $f_s = 0$ and switch width $f_e - f_s = 1 \times 10^{-3}$ as in the previously discussed cases. Again the sum of the results of eq. 4.101 over all of the fluid species is used as the sensor function. The test simulations were performed using the BDF2 time difference with a timestep of $\Delta t = 0.002$ on grids of 100×200, 200×400, and 400×800 linear quadrilaterals, as was done when switched system specific discontinuity capturing viscosities were considered. The time evolution of the reconnected flux and total energy in the domain that resulted can be seen in figs. 7.48 and 7.49. Plots of the solution on the 400×800 grid at $t = 20$ and $t = 40$ are presented in fig. 7.50 to illustrate the solution structure obtained. From fig. 7.50, it can be seen that a secondary magnetic island did develop in this solution on the finest mesh, indicating that it may still be overly diffuse, though neither of the coarser mesh simulations with this approach developed a secondary island. Interestingly, the finest mesh solution obtained using this approach did not develop the complex structure that was seen when system specific viscosities was used and that has been seen in most results in the literature. [4, 5, 9] The reconnected flux time evolution shows that the solution on the 200×400 grid produced the appropriate behavior, but this did not hold with grid refinement. The expected final amount of reconnected flux was overshot considerably in the finest mesh solution, likely due to the formation of the secondary magnetic island. The plot of the evolution of the total energy in the domain indicates considerably less energy dissipation with this approach versus using simultaneously switched system specific viscosities. Some energy accumulation was again apparent in the coarsest grid solution, but no net accumulation of energy was seen at any time in either of the two finer grid solutions.

Fig. 7.51 presents the switched and unswitched values of the discontinuity capturing viscosity computed based on the final time solution on the 200×400 grid with $\Delta t = 0.002$ computed with the switched full model viscosity. Some vector field tracers following the B field are also included in the plot. It can be seen that the switching function has greatly reduced the discontinuity capturing viscosity over most of the domain, and particularly in the background plasma outside the islands and reconnection zone. Interestingly, it appears that the discontinuity capturing term activation is greatest along the separatrix of the magnetic field, even though there are seemingly areas of steeper gradient in the fluid property fields further into the magnetic islands on the boundaries.

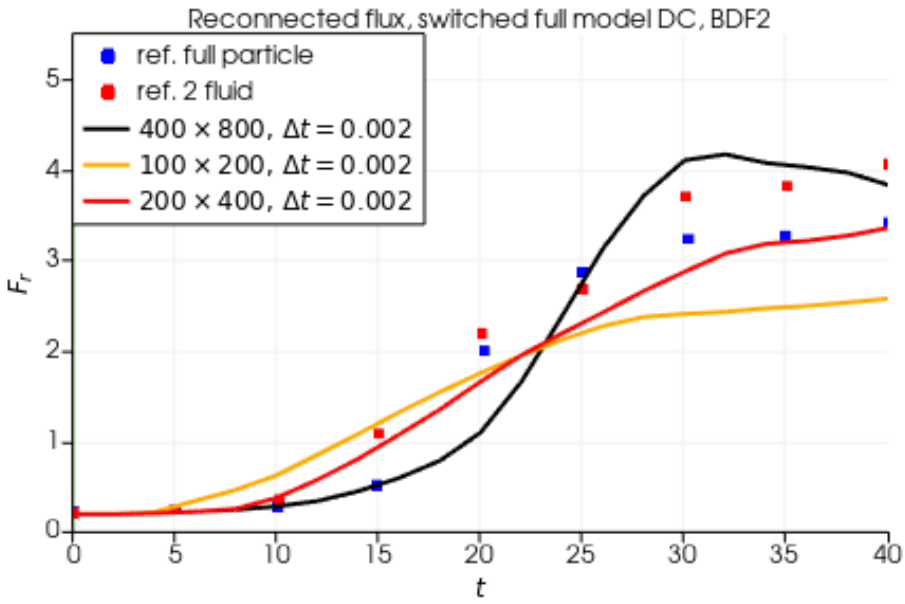


Figure 7.48: Reconnected flux evolution for simulations using BDF2 time terms and simultaneously switched discontinuity capturing terms defined by eqs. 4.95, 4.96, 4.99 computed based on the full set of governing equations and smoothly switched using eq. 4.100 based on the sum of functions defined by eq. 4.101 computed for each fluid species. The data series labelled "ref. full particle" were measured from the figure in ref. [12], the data series labelled "ref. 2 fluid" were measured from the "512 × 256 DG3" data series of fig. 5.17 in ref. [4].

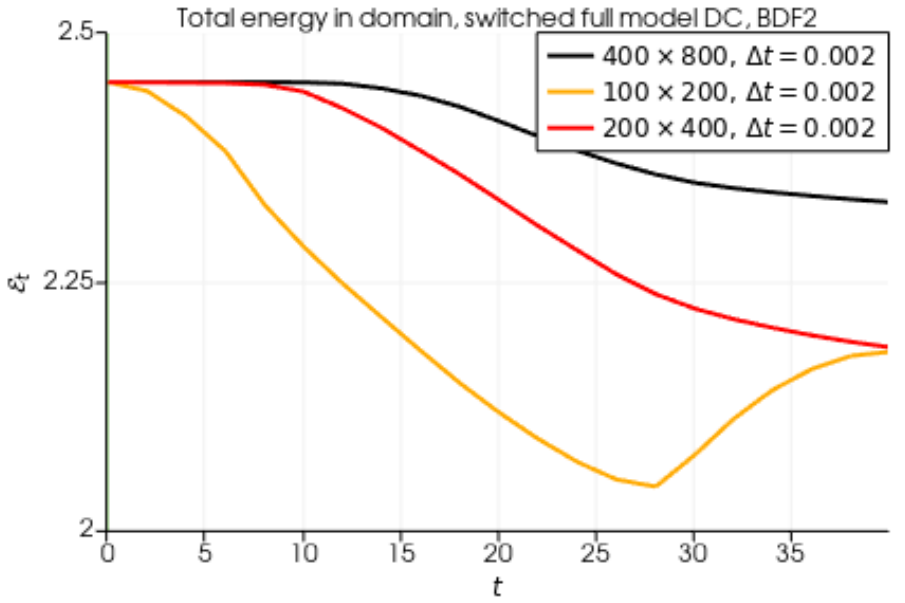


Figure 7.49: Time evolution of total energy in the domain for simulations using BDF2 time terms and simultaneously switched discontinuity capturing terms defined by eqs. 4.95, 4.96, 4.99 computed based on the full set of governing equations and smoothly switched using eq. 4.100 based on the sum of functions defined by eq. 4.101 computed for each fluid species.

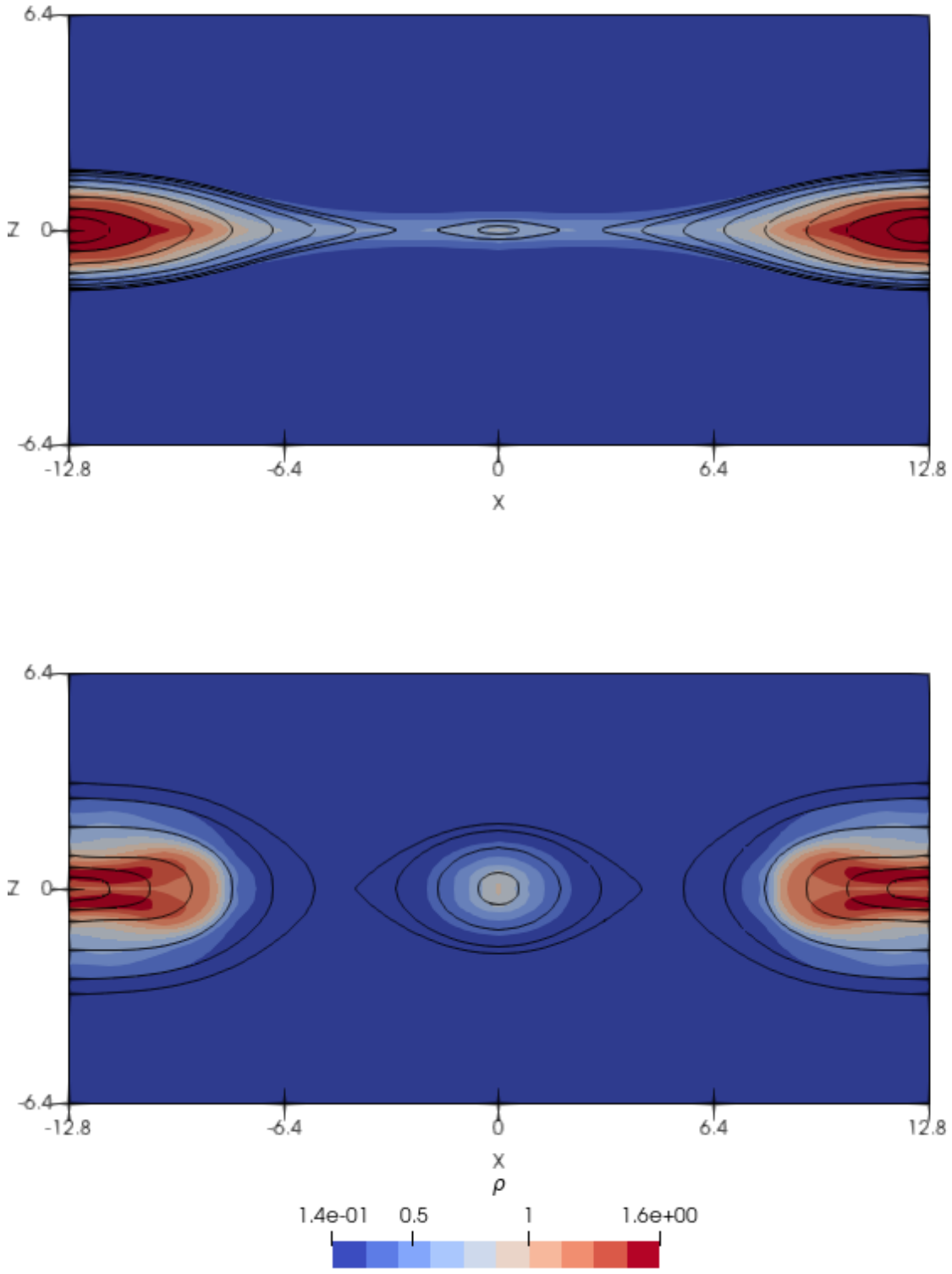


Figure 7.50: Solution field at times $t = 20$ (top) and $t = 40$ (bottom) computed using switched full model discontinuity capturing terms on a 400×800 mesh with $\Delta t = 0.002$. The color field indicates the plasma density while the black lines are vector field tracers seeded along $z = 0$ and following the in plane components of B field.

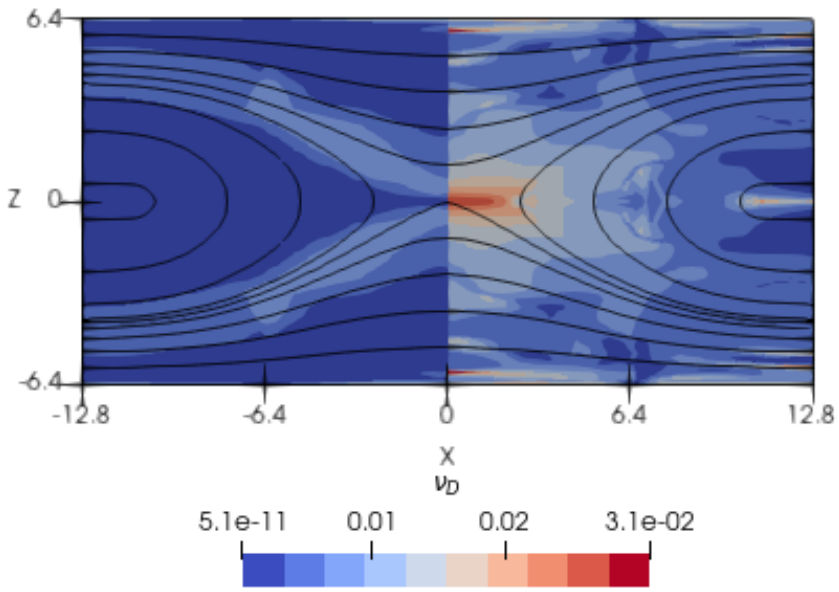


Figure 7.51: Full model viscosity switched (left) and unswitched (right) computed at $t = 40$ based on solution computed with switched full model discontinuity capturing term on a 200×400 element grid with $\Delta t = 0.002$ and corresponding plasma density field. The color field indicates the discontinuity capturing viscosity magnitude while the black lines are vector field tracers seeded along $z = 0$ and $x = 0$ and following the in plane components of the B field.

7.2 Advection-reaction equation based stabilized FEM results

7.2.1 Manufactured solution

Transient one dimensional manufactured solution simulations like those described in sec. 7.1.1 were also performed using the advection-reaction equation based stabilized method. The same domain and time interval was simulated and the same specified solutions were employed. The simulations were performed with the implicit trapezoid rule time difference on the same 4 to 128 uniformly refined linear element grids with the same timestep sizes, again, refined alongside the spatial grid. As with the SUPG simulations, the boundary conditions are applied via specified fluxes obtained using Riemann fluxes with the exterior conditions set to the exact solution. The results for simulations with discontinuity capturing terms presented here also employ a single discontinuity capturing viscosity computed by considering all of the governing equations as a single set of balance laws and with switching based on the sum of the results of eq. 4.101 for each fluid species, as described in sec 7.1.1 for the SUPG manufactured solution simulations.

The resulting error integrals are plotted against the element size in figs. 7.52, 7.53, and 7.54 for simulation without the discontinuity capturing terms, with unswitched discontinuity capturing terms, and with switched discontinuity capturing terms, respectively. As in the case of the SUPG method, an approximately second order convergence rate is seen when the advection-reaction equation based stabilized method is used without discontinuity capturing terms, and an approximately first order convergence rate is seen when the unswitched discontinuity capturing terms are added. These are believed to be appropriate convergence rates for these methods. With switched discontinuity capturing terms, rapid convergence is again seen as the discontinuity capturing terms deactivate with grid refinement, and once they are effectively deactivated, the method returns to the approximately second order convergence rate of the base method. Figs. 7.55, 7.56, and 7.57 present the final time electron density fields computed without discontinuity capturing terms, with unswitched discontinuity capturing terms, and with switched discontinuity capturing terms, respectively, as examples of the solution quality obtained. It can be seen that without discontinuity capturing terms, the method is subject to produce solutions with extrema outside the range of the true solution when the solution is underresolved, but that this improves with grid refinement. This is

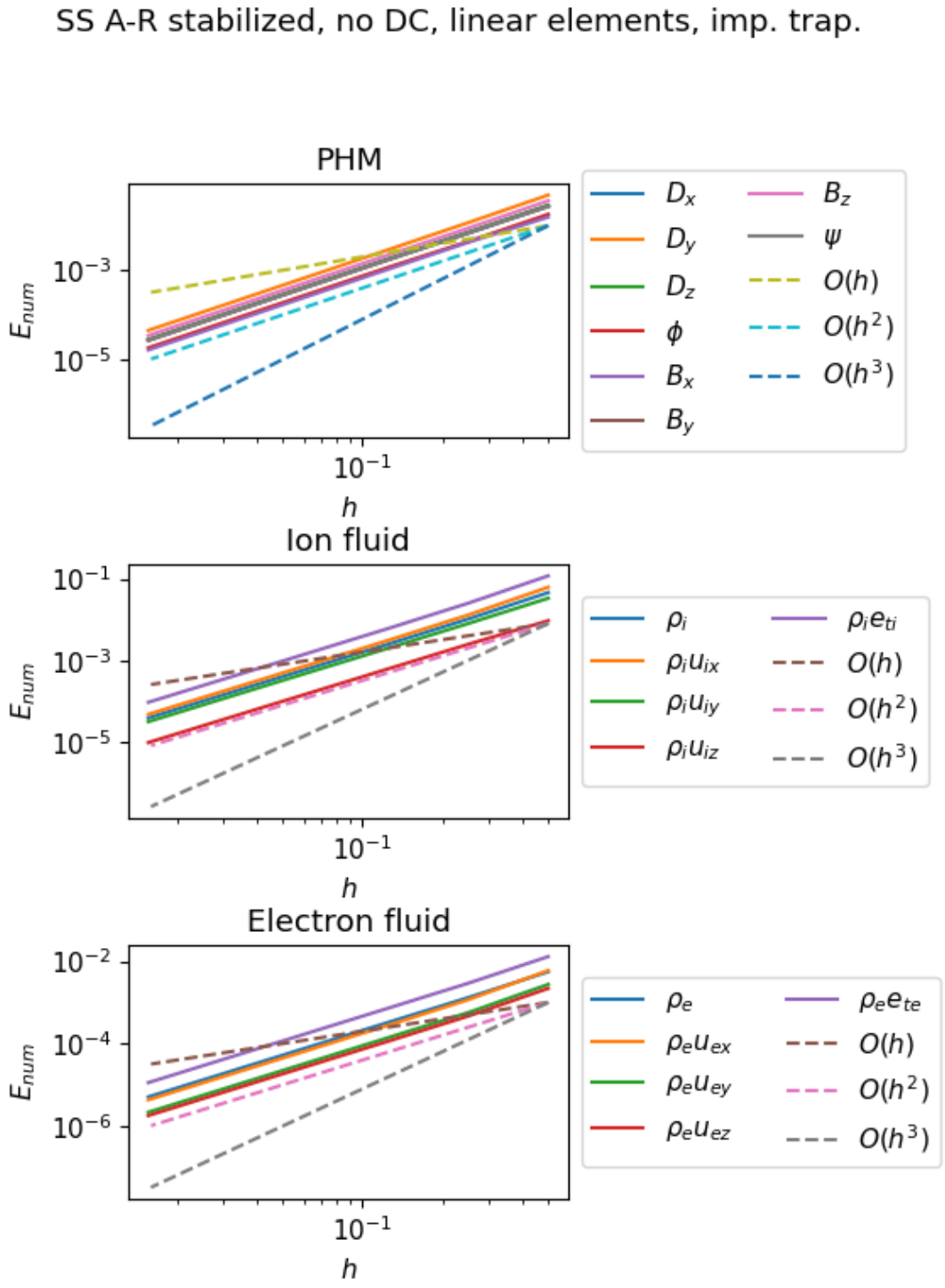


Figure 7.52: Plots of discretization error when using the SS A-R eq. based stabilized method with linear elements, no discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

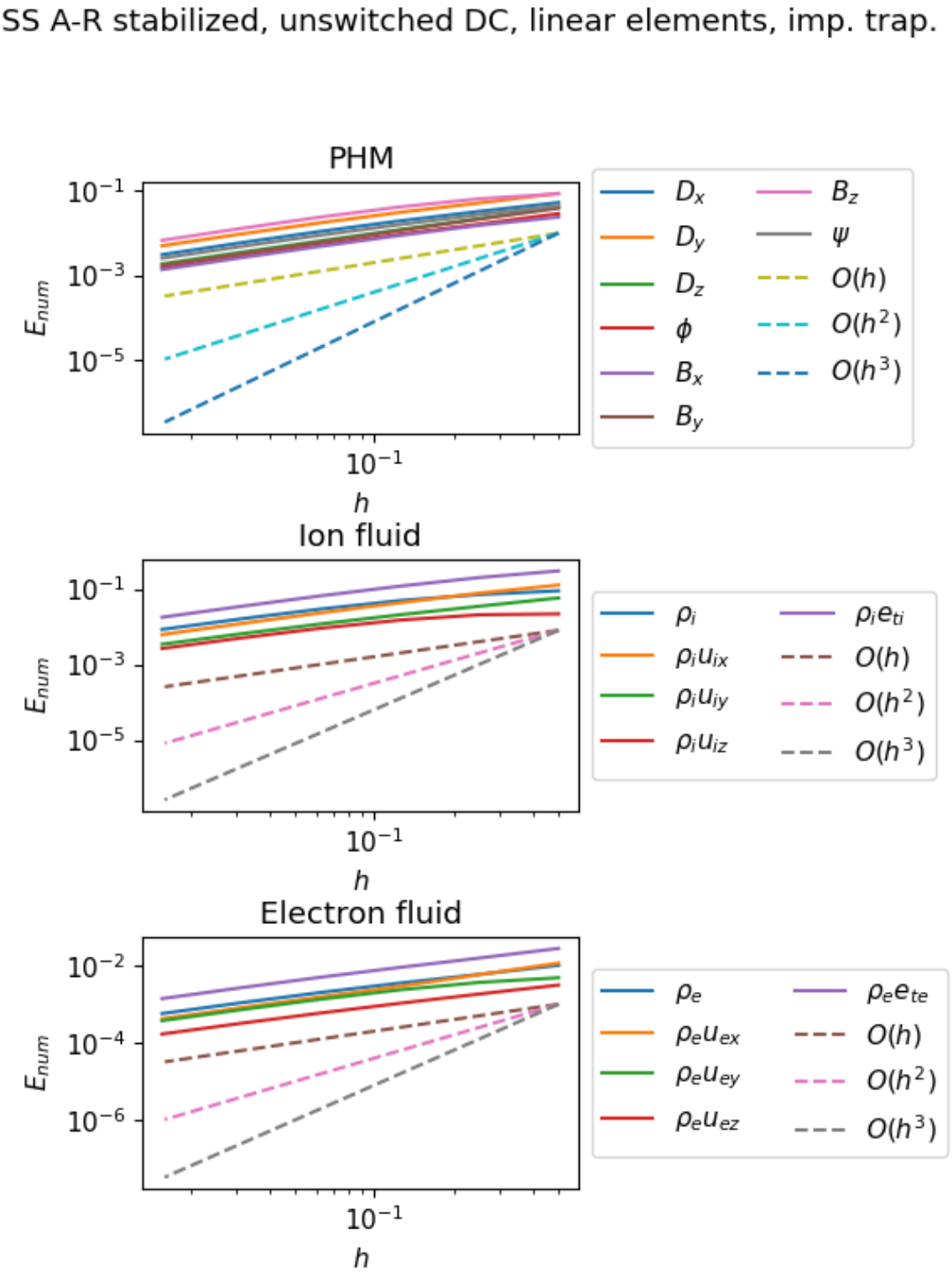


Figure 7.53: Plots of discretization error when using the SS A-R eq. based stabilized method with linear elements, unswitched discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

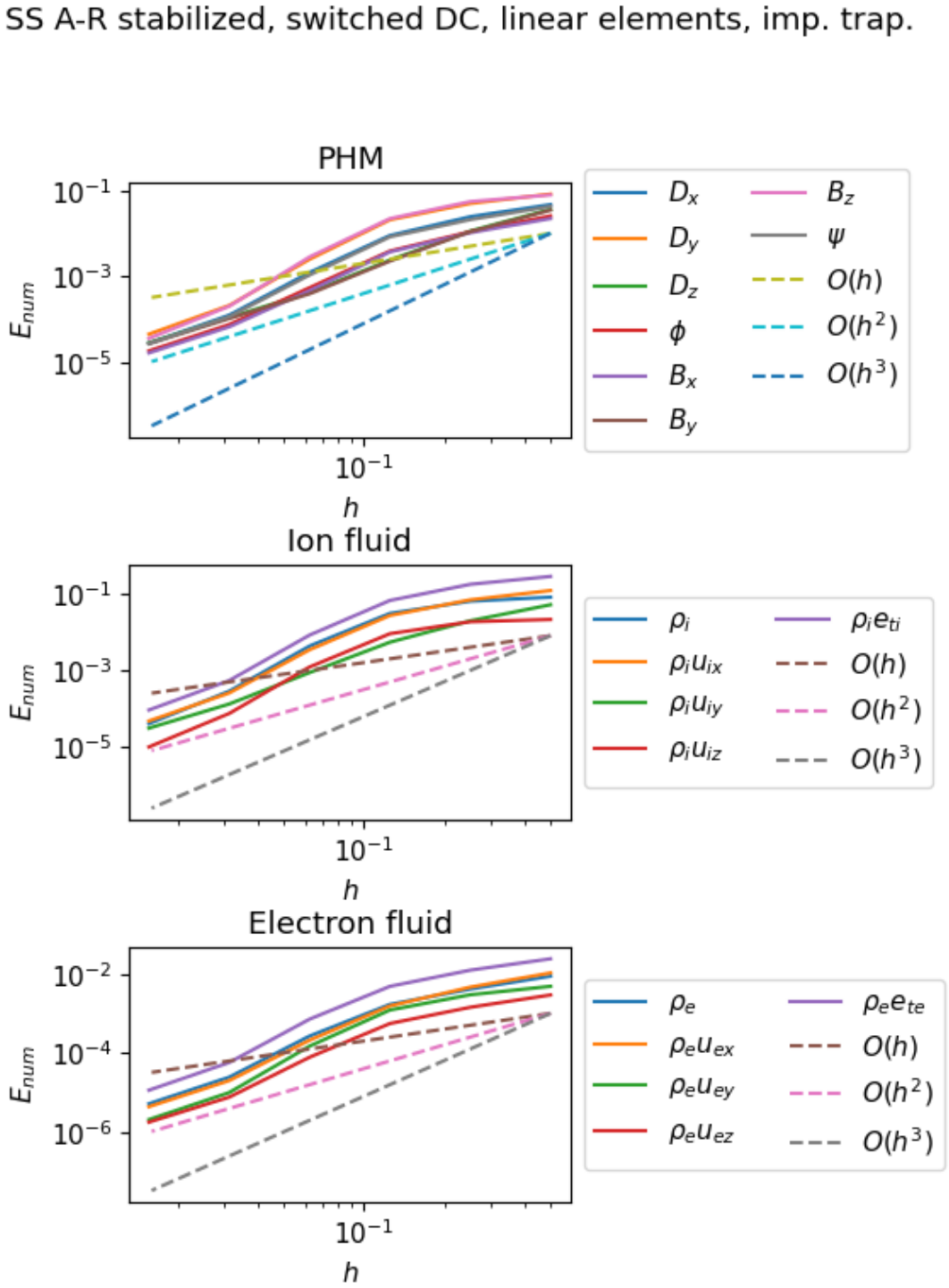


Figure 7.54: Plots of discretization error when using the SS A-R eq. based stabilized method with linear elements, switched discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem. Both axes are logarithmic, $\Delta t = h$ and is refined along with h for these transient simulations.

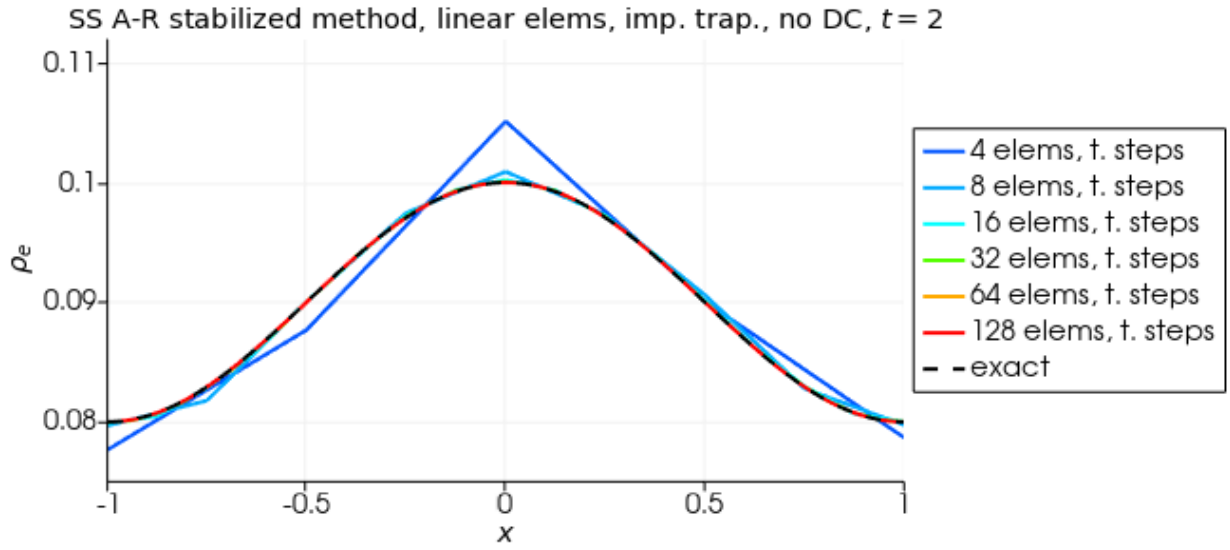


Figure 7.55: Plot of final time electron density fields obtained using the SS A-R eq. based stabilized method with linear elements, no discontinuity capturing term, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem.

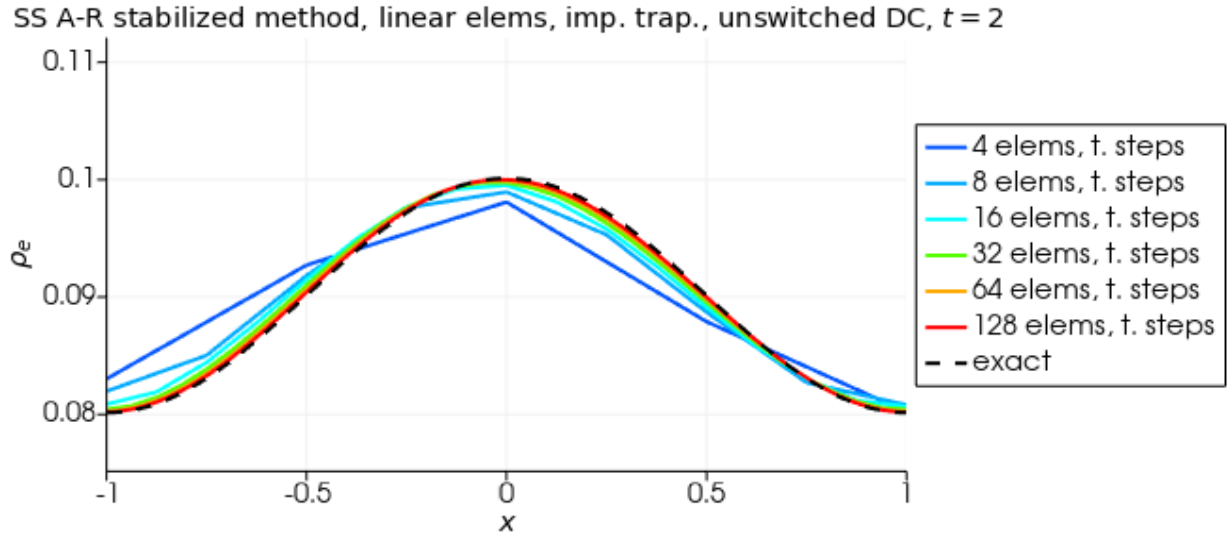


Figure 7.56: Plot of final time electron density fields obtained using the SS A-R eq. based stabilized method with linear elements, unswitched discontinuity capturing terms, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem.

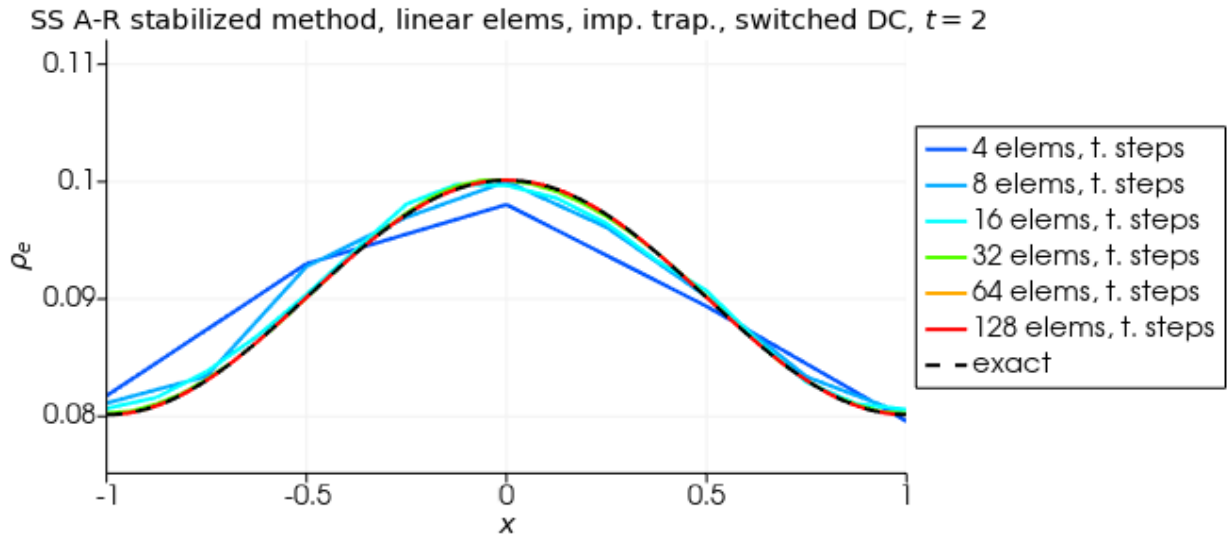


Figure 7.57: Plot of final time electron density fields obtained using the SS A-R eq. based stabilized method with linear elements, switched discontinuity capturing terms, and implicit trapezoid time difference on a one dimensional, transient manufactured solution problem.

comparable to the behavior seen with simple linear models in appendix A. The behavior is better when discontinuity capturing terms are employed.

7.2.2 Two fluid Brio-Wu problem

The two fluid Brio-Wu problem with $\frac{m_e}{m_i} = \frac{1}{100}$, $\frac{u_{rMHD}}{c_0} = \frac{1}{100}$, and $\gamma = 2$ for both fluids was simulated using the advection-reaction equation based stabilized method for ion charge to mass ratios of $\frac{q_i}{m_i} = 1$, 10, and 100. Simulations were performed on a uniform 1000 element per unit length grid on the domain $-9.5 \leq x \leq 10.5$ and a timestep of $\Delta t = 1 \times 10^{-3}$ was used for a CFL number of one based on the speed of light. The implicit trapezoid rule was used for the time difference. Simulations were performed without discontinuity capturing terms and with the full model, switched discontinuity capturing term of eqs. 4.95, 4.96, and 4.99 with the switching function of eq. 4.100 operating based on the sum of eq. 4.101 for both fluids and a threshold of $f_s = 0$ and $f_e - f_s = 1 \times 10^{-3}$.

Simulation without discontinuity capturing terms

Without discontinuity capturing terms, a simulation to $t = 6.2$ was possible with $\frac{q_i}{m_i} = 1$ and a simulation to $t = 4.2$ of a total time interval of 10 was possible with $\frac{q_i}{m_i} = 100$. The simulation with $\frac{q_i}{m_i} = 10$ didn't even make it to $t = 0.5$ before conditions were nonphysical enough to crash the algorithm. With both $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 100$, node to node oscillations were present in the solution as can be seen in figs. 7.58 and 7.59 for $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 100$, respectively. The node to node oscillations are present in all of the fields, but do seem to grow faster in some fields and in some areas of the solution than others, suggesting that they are more/less prone to instability under different conditions in the plasma. The $\frac{q_i}{m_i} = 100$ results are presented at $t = 3.6$ because that is the latest time for which the SUPG solution on the 100000 element per unit length grid was available. It is believed to be a reasonable reference due to its similarity to result shown in ref. [5] for $\hat{r}_g = 10^{-2}$, but does contain some numerical oscillations of its own. The total energy in the domain for the problem is shown in fig. 7.60 for the simulation with $\frac{q_i}{m_i} = 1$ and in fig. 7.60 $\frac{q_i}{m_i} = 100$. It can be seen that the method without discontinuity capturing terms does a very good job of maintaining a constant amount of energy in the domain. A slight increase is seen near the

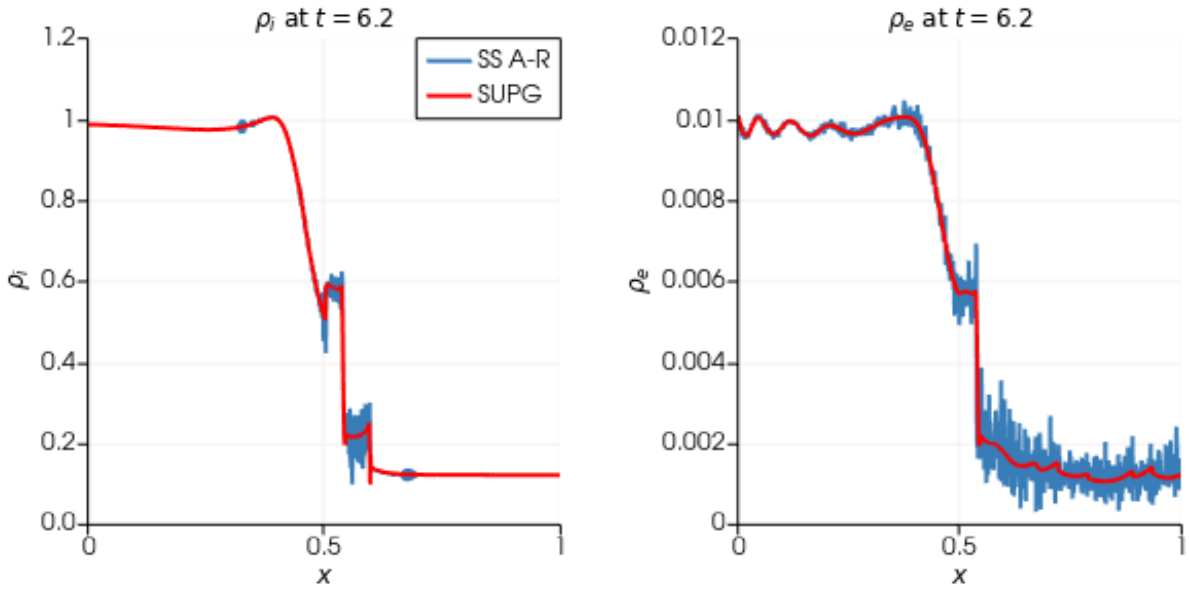


Figure 7.58: Ion and electron densities computed using advection-reaction equation based stabilized FEM without a discontinuity capturing term plotted for $0 \leq x \leq 1$ at $t = 6.2$ for Brio-Wu problem with $\frac{q_i}{m_i} = 1$. Solution obtained using SUPG shown for reference.

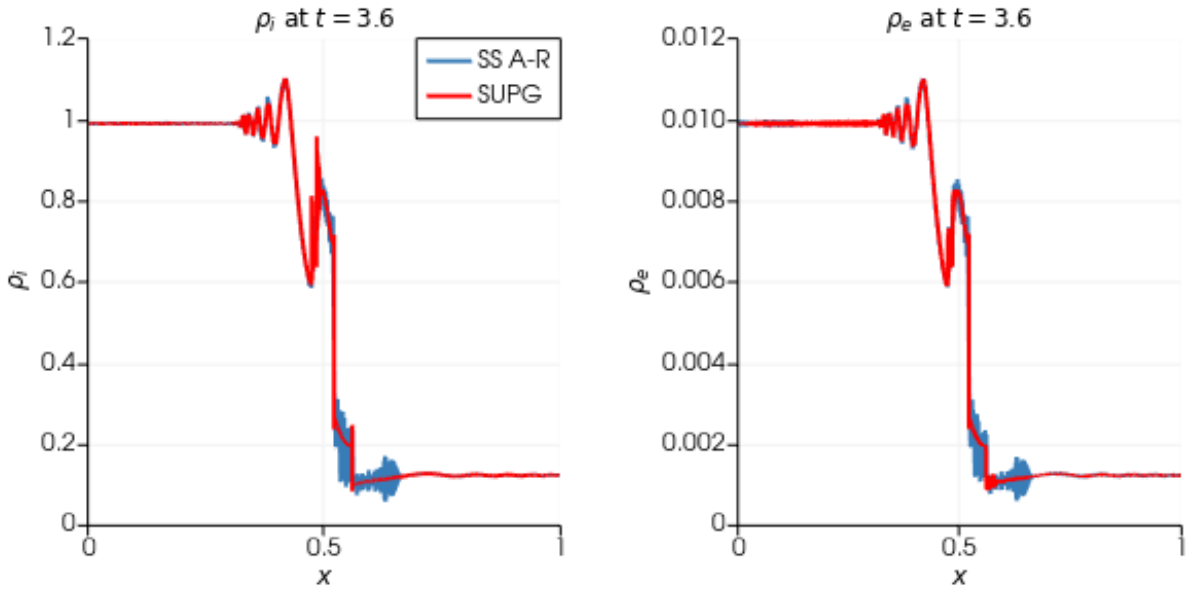


Figure 7.59: Ion and electron densities computed using advection-reaction equation based stabilized FEM without a discontinuity capturing term plotted for $0 \leq x \leq 1$ at $t = 3.6$ for Brio-Wu problem with $\frac{q_i}{m_i} = 100$. Solution obtained using SUPG shown for reference.

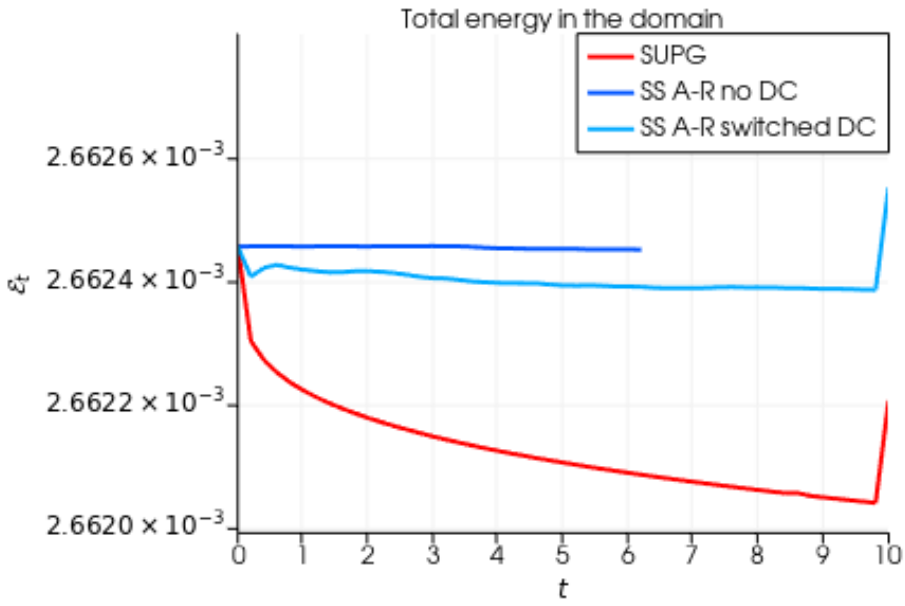


Figure 7.60: Time evolution of total energy in the domain for the Brio-Wu problem with $\frac{q_i}{m_i} = 1$ simulated with the advection-reaction equation based stabilized FEM with and without discontinuity capturing terms. SUPG result included for reference.

failure point of the $\frac{q_i}{m_i} = 100$ simulation, but otherwise, very little change in the amount of energy in the domain is seen.

Simulation with simultaneously switched full model discontinuity capturing term

The simulations that employed the simultaneously switched full model discontinuity capturing term survived the entire simulation time interval with both $\frac{q_i}{m_i} = 1$ and $\frac{q_i}{m_i} = 100$, however for the $\frac{q_i}{m_i} = 100$ simulation the final result is obviously flawed. In fig. 7.61, the $\frac{q_i}{m_i} = 1$ solution can be seen to agree well with the good SUPG solution included for reference. The density fields are only slightly different around $x = 0.5$. In fig. 7.60, the energy conservation is seen to also be quite good in this case, except for the jump at the end, only a slight loss occurs over the course of the simulation. This jump, seen in both the SUPG and advection-reaction equation based stabilized FEM solutions, is believed to be the result of the smeared fast moving waves of the solution reaching the edge of the domain. In hindsight, a slightly longer domain probably should have been used since the numerical methods do tend to smear the solution, allowing changes to occur a little further into the domain than would be suggested by the wave speeds. The results of the simulation with $\frac{q_i}{m_i} = 100$ are not as good. In the ion density plot of fig. 7.62, no severe node to node oscillations are seen, but the solution has picked up some errant features in place of the appropriate oscillations ahead of the rarefaction at about $x = 0.4$. The electron density solution shown in the same figure does show some severe numerical oscillations. In fig. 7.63 it can be seen that substantial nonphysical energy generation occurs when using this approach in the $\frac{q_i}{m_i} = 100$ simulation. In this case, however, the nonphysical energy generation does not appear to coincide with the same sort of large, well resolved oscillations that were seen with the SUPG method. It is associated almost entirely with growth of the electron internal energy occuring throughout the domain, and the oscillations seen are mostly node to node oscillations.

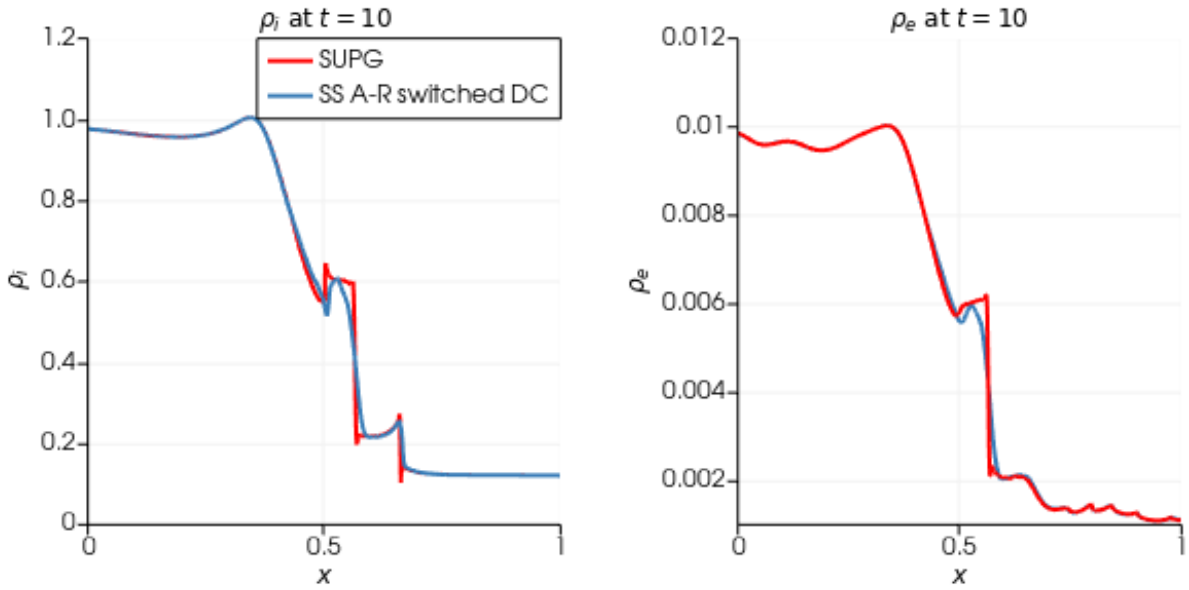


Figure 7.61: Ion and electron densities computed using advection-reaction equation based stabilized FEM with simultaneously switched discontinuity capturing terms plotted for $0 \leq x \leq 1$ at $t = 10$ for Brio-Wu problem with $\frac{q_i}{m_i} = 1$. Solution obtained using SUPG shown for reference.

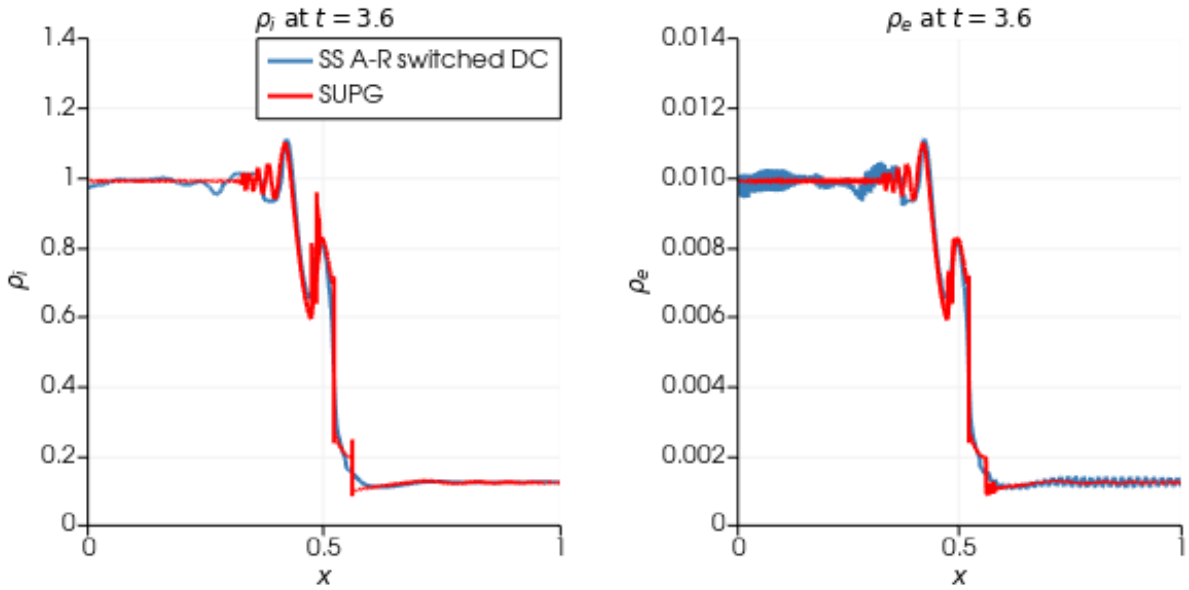


Figure 7.62: Ion and electron densities computed using advection-reaction equation based stabilized FEM with simultaneously switched discontinuity capturing terms plotted for $0 \leq x \leq 1$ at $t = 3.6$ for Brio-Wu problem with $\frac{q_i}{m_i} = 100$. Solution obtained using SUPG shown for reference.

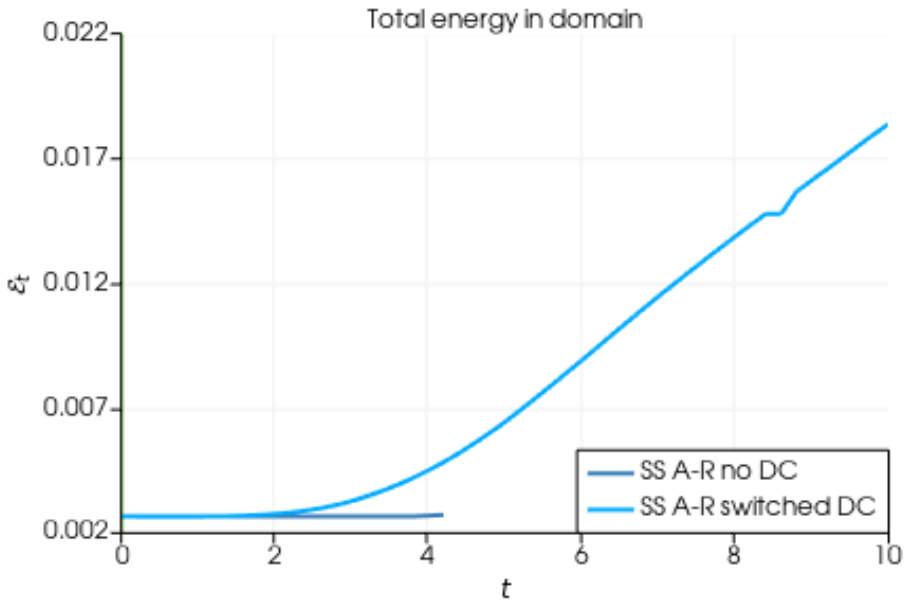


Figure 7.63: Time evolution of total energy in the domain for the Brio-Wu problem with $\frac{q_i}{m_i} = 100$ simulated with the advection-reaction equation based stabilized FEM with and without discontinuity capturing terms.

Chapter 8

Conclusion

In conclusion, the approach of using the SUPG FEM to numerically model the ideal two fluid plasma model was partially successful. The results of tests with the method of manufactured solutions, the two fluid Brio-Wu problem, and the GEM challenge problem indicated that a correct implementation was likely achieved, but also indicate that further development related to the method and/or model may be necessary for complete success. At lower charge to mass ratios, the method works well for the two fluid Brio-Wu problem, but as the charge to mass ratio increases and the source terms become stronger, issues with long wavelength oscillations in the solution fields growing in what should be quiescent plasma far from the initial discontinuity arise. It does appear that sufficient grid resolution would prevent this, but the amount of grid resolution that it takes seems excessive. It would not be required to resolve the near constant conditions that should occur there, nor the oscillations themselves, as they often span enough elements to appear smooth in the solution. Interestingly, this effect was also found to occur in results produced by a separately implemented finite volume code, and, very surprisingly, it appears that more diffusive spatial discretizations, both SUPG with discontinuity capturing terms and the finite volume method with first order reconstruction, make the oscillations worse. The effect is more acute in two dimensions since the need for spatial resolution increases the element and degree of freedom counts quadratically rather than linearly. The smoothly switched discontinuity capturing term strategies employed were effective at reducing the discontinuity capturing viscosity away from steep solution gradients, enabling the additional viscosity to be localized where it was needed, but there is room for improvement in this area since the discontinuity capturing term was apparently not reduced

enough in smooth areas to fully eliminate the growth of nonphysical oscillations in the Brio-Wu problem solutions on coarser grids.

It is believed that this effort to use the SUPG method for numerical simulations of the ideal two fluid plasma model was the first attempt to do so. As described in sec. 4.3.2 this method has properties that seemed to have the potential to reduce the computational cost of this very computationally expensive model. Unfortunately, the results described in sec. 7.1.2 do not indicate that an advantage was obtained through the use of the SUPG FEM as implemented in this work. It does, however, appear that the method's performance is, at least, comparable to that of the more commonly employed finite volume method.

There are many areas for further work on this approach to numerical simulation of the multi-fluid plasma model, most with the objective of reducing the grid resolution necessary to preclude the development of the long wavelength oscillations that led to the failure of many of the simulations attempted in this work. More work on the discontinuity sensor function may lead to one capable of more aggressively removing the discontinuity capturing viscosity in regions where it is not needed. If the large scale oscillations arising in the quiescent plasma are being excited from within the areas of more solution complexity, perhaps adaptive grid refinement could be used to prevent them from forming without the added computation cost associated with uniform refinement. Perhaps replacement of the ion fluid equations with a total fluid continuity and momentum equation and a total energy equation including electromagnetic energy would prevent the total energy in the domain from being able to nonphysically grow or decay and there by prevent the oscillations from being able to grow so large.

The development of the advection-reaction equation based stabilized finite element method and its application to the ideal two fluid plasma model is also believed to be an original effort. As for that method, it does not appear to be worth pursuing further since its solutions on the Brio-Wu problem were full of node to node oscillations. The method had shown some promise for the smooth manufactured solution problem, but perhaps the strength of the fictitious source term aided it in that case. This result is ultimately consistent with the result seen in preliminary analysis for a single scalar equation with a square wave initial condition. It appears that the linear problem solution structure that the method was derived around is too simple compared to what is encountered with

a system of nonlinear equations, or more general initial conditions. Further work in this direction would likely involve attempting to derive a weight modification based on a more complex model.

Bibliography

[1] MIT Plasma Science and Fusion Center. *What is Plasma?*, (accessed September 18, 2020). `https://www.psfc.mit.edu/vision/what_is_plasma`.

[2] E. H. Holt and R. E. Haskell. *Foundations of Plasma Dynamics*. Macmillan, New York, 1 edition, 1965.

[3] U. Shumlak. Algorithm development for the multi-fluid plasma model. Technical report, University of Washington, May 2011.

[4] J. Loverich. *A discontinuous Galerkin method for the two-fluid plasma system and its application to the Z-pinch*. dissertation, The University of Washington, 2005.

[5] A. Alvarez Laguna, N. Ozak, A. Lani, Deconinck H., and S. Poedts. Fully-implicit finite volume method for the ideal two-fluid plasma model. *Computer Physics Communications*, 231:31–44, 2018.

[6] K. A. Croft and T. M. Moeller. Two fluid plasma dynamics simulation using the SUPG FEM with simultaneously switched scalar artificial diffusion based discontinuity capturing terms. In *AIAA AVIATION 2021 FORUM*, 2021.

[7] J. Derouillat, A. Beck, F. Pérez, T. Vinci, M. Chiaramello, A. Grassi, M. Flé, G. Bouchard, I. Plotnikov, N. Aunai, J. Dargent, C. Riconda, and M. Grech. SMILEI: A collaborative, open-source, multi-purpose particle-in-cell code for plasma simulation. *Computer Physics Communications*, 222:351–373, 2018.

[8] J. Loverich. *A finite volume algorithm for the two-fluid plasma system in one dimension*. thesis, The University of Washington, 2003.

[9] A. H. Hakim. *High resolution wave propagation schemes for two-fluid plasma simulations.* dissertation, The University of Washington, 2006.

[10] U. Shumlak and J. Loverich. Approximate Riemann solver for the two-fluid plasma model. *Journal of Computational Physics*, 187:620–638, 2003.

[11] M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *Journal of computational physics*, 75:400–422, 1988.

[12] J. Birn, J. F. Drake, M. A. Shay, B. N. Rogers, R. E. Denton, M. Hesse, M. Kuznetsova, Z. W. Ma, A. Bhattacharjee, A. Otto, and P. L. Pritchett. Geospace environmental modeling (GEM) magnetic reconnection challenge. *Journal of geophysical research*, 106(A3):3715–3719, 2001.

[13] T. W. Swafford. Notes for ENCM 5100 computational fluid dynamics I, 2013.

[14] D. Kincaid and W. Cheney. *Numerical analysis: mathematics of scientific computing.* American Mathematical Society, Providence, RI, 3rd edition edition, 2009.

[15] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics.* Springer-Verlag, Berlin, 3rd edition edition, 2009.

[16] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt. *Concepts and applications of finite element analysis.* John Wiley and Sons, Inc., Hoboken, NJ, 4th edition edition, 2001.

[17] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32:199, 259, 1982.

[18] T. E. Tezduyar and T. J. R. Hughes. Development of time accurate finite element techniques for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Technical report, Stanford University, 1982.

[19] T. J. R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. symmetric forms of the compressible euler and Navier-Stokes equations and the second law of thermodynamics. *Computer methods in applied mechanics and engineering*, 54:223, 234, 1986.

[20] T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: III. the generalized streamline operator for multidimensional advective-diffusive systems. *Computer methods in applied mechanics and engineering*, 58:305, 328, 1986.

[21] T. J. R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. beyond SUPG. *Computer methods in applied mechanics and engineering*, 54:341, 355, 1986.

[22] T. J. R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: IV. a discontinuity-capturing operator for multidimensional advective-diffusive systems. *Computer methods in applied mechanics and engineering*, 58:329–336, 1986.

[23] G. J. Le Beau, S. E. Ray, S. K. Aliabadi, and T. E. Tezduyar. SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. *Computer methods in applied mechanics and engineering*, 104:397–422, 1993.

[24] J. T. Erwin. *Stabilized finite elements for compressible turbulent Navier-Stokes.* dissertation, The University of Tennessee at Chattanooga, 2013.

[25] W. K. Anderson, J. C. Newman, and S. L. Karman. Stabilized finite elements in FUN3D. In *55th AIAA Aerospace Sciences Meeting*, 2017.

[26] T. E. Tezduyar. Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows. In *Proceedings of the european congress on computational methods in applied sciences and engineering*, ECCOMAS 2004, Jyvaskayla, Finland, 2004.

[27] S. Rajamohan. *Streamline upwind/ Petrov-Galerkin FEM based time-accurate solution of 3D time-domain Maxwell's equations for dispersive materials.* dissertation, The University of Tennessee at Chattanooga, 2014.

[28] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective-diffusive equations. *Computer methods in applied mechanics and engineering*, 73:173,189, 1989.

[29] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kroner, M. Ohlberger, and C. Rohde, editors, *An introduction to recent developments in theory*

and numerics for conservation laws; proceedings of the international school on theory and numerics for conservation laws, Freiburg/Littenweiler, October 20-24, 1997, Lecture notes in computational science and engineering, pages 195–285. Springer, Berlin, Heidelberg, 1999.

[30] B. Jiang and L. A. Povinelli. Least-squares finite element method for fluid dynamics. *Computer methods in applied mechanics and engineering*, 81:13,37, 1990.

[31] S. Idelsohn, N. Nigro, M. Storti, and G. Buscaglia. A Petrov-Galerkin formulation for advection-reaction-diffusion problems. *Computer methods in applied mechanics and engineering*, 136:27–46, 1996.

[32] T. E. Tezduyar and Y. J. Park. Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Computer methods in applied mechanics and engineering*, 59:307–325, 1986.

[33] S Yan and J. Jian-Ming. A continuity-preserving and divergence-cleaning algorithm based on purely and damped hyperbolic Maxwell equations in inhomogeneous media. *Journal of computational physics*, 334:392–418, 2017.

[34] C. D. Munz, P. Ommes, and R. Schneider. A three-dimensional finite-volume solver for the Maxwell equations with divergence cleaning on unstructured meshes. *Computer physics communications*, 130:83–117, 2000.

[35] G. I. Ogilvie. Astrophysical fluid dynamics. *Journal of Plasma Physics*, 82(3), May 2016.

[36] G. B. Thomas, R. L. Finney, M. D. Weir, and F. R. Giordano. *Thomas' calculus: early transcendentals*. Addison Wesley, Boston, MA, 10 edition, 2000.

[37] R. J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge university press, Cambridge, 2005.

[38] A. Harten, P. Lax, and B. Van Leer. On upstream differencing and Godunov-type schemes for conservation laws. *SIAM Review*, 25(1):35–51, 1983.

[39] B. Cockburn, G. E. Karniadakis, and C. Shu. The development of discontinuous Galerkin methods. Technical report, The University of Minnesota Digital Conservancy, https://hdl.handle.net/11299/3384, 1999.

[40] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Computer methods in applied mechanics and engineering*, 188(1):61–82, 2000.

[41] K. W. Anderson, L. Wang, S. Kapadia, C. Tanis, and B. Hilbert. Petrov-Galerkin and discontinuous Galerkin methods for time-domain and frequency domain electromagnetic simulations. *Journal of computational physics*, 230:8360–8385, 2011.

[42] F. Shakib, T. J. R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. the compressible Euler and Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 89:141–219, 1991.

[43] E. W. Weisstein. *Moore-Penrose Matrix Inverse*, accessed November 29, 2023. https://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html.

[44] N. K. Madsen. The method of lines for the numerical solution of partial differential equations (abstract). *SIGNUM Newsl.*, 10(4):5–7, dec 1975.

[45] T. H. Pulliam. Time accuracy and the uses of implicit methods. Technical report, NASA Ames Research Center, 2014. https://overflow.larc.nasa.gov/wp-content/uploads/sites/54/2014/06/Time_Accurate_Methods_V8.pdf.

[46] K. A. Croft and T. M. Moeller. Validation of a SUPG finite element solver for the two-fluid plasma model using the Brio-Wu MHD shock tube problem. In *AIAA AVIATION 2019 FORUM*. AIAA, 2019.

[47] J. Larsson, R. Vicquelin, and I. Bermejo-Moreno. Large eddy simulations of the hyshot II scramjet. Technical report, Center for Turbulence Research, 2011.

[48] P. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. Technical report, Massachusetts Institute of Technology, 2006.

[49] G. F. Naterer and J. A. Camberos. Entropy and the second law fluid flow and heat transfer simulation. *Journal of Thermophysics and Heat Transfer*, 17(3):360–371, 2003.

[50] Y. Lv and M. Ihme. Entropy residual shock detector for shock-capturing using high-order discontinuous Galerkin scheme. Technical report, Center for Turbulence Research, 2016.

[51] A. Guardone. *Non-ideal compressible flow in a supersonic nozzle*, 2020. `https://su2code.github.io/tutorials/NICFD_nozzle/`.

[52] E. Gabriel et al. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

[53] C. Geuzaine and J. F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[54] L. S. Avila et al. *The VTK user's guide*, chapter 19. Kitware, New York, 11 edition, 2010.

[55] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[56] P. Knupp and K. Salari. *Verification of computer codes in computational science and engineering*. Chapman and Hall/CRC, New York, 1st edition, 2002.

[57] P. L. Pritchett. Geospace environmental modeling magnetic reconnection challenge: simulations with a full particle electromagnetic code. *Journal of geophysical research*, 106(A3):3783–3798, 2001.

[58] K. A. Croft and T. M. Moeller. Implementation of a SUPG stabilized finite element method for a two fluid plasma model. In *AIAA AVIATION 2018 FORUM*. AIAA, 2018.

[59] Kenneth G. Powell, Philip L. Roe, Timur J. Linde, Tamas I. Gombosi, and Darren L. De Zeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *Journal of Computational Physics*, 154(2):284–309, 1999.

[60] J. W. Slater. *NPARC alliance validation archive: shock tube*, 2012. `https://www.grc.nasa.gov/WWW/wind/valid/stube/stube.html`.

[61] J. Ahrens, B. Geveci, and C. Law. *Paraview: an end-user tool for large data visualization, visualization handbook*. Elsevier, 2005.

[62] M-S Liou and C. J. Jr. Steffen. A new flux splitting scheme. *Journal of computational physics*, 107:23–39, 1993.

[63] G. J. Barclay, D. F. Griffiths, and D. J. Higham. Theta method dynamics. *LMS Journal of Computation and Mathematics*, 3:27–43, 2000.

[64] K A. Croft and T. M. Moeller. Simulating the GEM challenge problem using a SUPG finite element method for the two fluid plasma model. In *AIAA AVIATION 2020 FORUM*. AIAA, 2020.

[65] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. SymPy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017.

Appendices

A Preliminary numerical evaluation of stabilized FEM based on advection-reaction equation

Prior to utilizing the stabilized FEM based on the advection-reaction equation on the ideal two fluid model, numerical simulations were performed using the method for simpler models including steady state and transient advection-reaction/ advection-oscillation, and a two equation system of these equations. Since the method is derived based on these simpler models, it would be expected to perform best for them and, as one or two governing equation models with linear advection and reaction terms, they result in much less computational expense than the eighteen equation, nonlinear ideal two fluid model. Due to this, simulating these simpler models allows a more rapid evaluation of the merits and limitations of the method. The results of five of these tests are presented here, starting with steady state scalar advection-reaction and advection oscillation simulations, then proceeding to transient scalar equations, before finally testing some two equation systems.

A.1 Steady state advection-reaction

The first test undertaken is simulation of a steady state advection-reaction equation like eq. 4.43, repeated here:

$$a\frac{\mathrm{d}q}{\mathrm{d}x} = sq. \tag{A.1}$$

Advection and source coefficients of $a = 1$ and $s = -20$ were chosen to produce a very steep solution gradient at the left end of the domain, $0 \leq x \leq 1$. The Dirichlet boundary condition $q(0) = 1$ and

the Neumann boundary condition $\boldsymbol{a}q \cdot \hat{\boldsymbol{n}}|_{x=0} = -1$ result in the same exact solution,

$$q(x) = \mathrm{e}^{-20x}. \tag{A.2}$$

Since the exact solution only contains one problem dependent constant, only one boundary condition should be enforced and since $a > 0$ the left boundary condition is enforced while the right boundary value is extrapolated from the numerical solution, consistent with the characteristic solution of the transient equation. The results of simulations with various numbers of linear elements are shown in figs. A.1 and A.2. From the figures it can be seen that the method performs as intended when the strongly enforced Dirichlet boundary condition is used, but use of the weakly enforced Neumann boundary condition immediately results in the loss of the method's ability to produce a nodally exact solution. The Neumann boundary condition solution does not develop node to node oscillations, however, and is of reasonable quality at the 16 element grid resolution.

A.2 Steady state advection-oscillation

The steady state advection-oscillation equation can also be expressed as eqn. A.1, but with s being imaginary valued and q now being potentially complex valued. This can be handled numerically by breaking the equation into two equations, one for the real part of q and one for the imaginary part of q as discussed in sec. 4.3.2, or by simply proceeding with complex variables and math functions. To test the performance of the derived stabilized scheme for situations where the source term causes oscillations, the particular advection-oscillation scenario with $a = 1$ and $s = \mathrm{i}8\pi$ was simulated on $0 \leq x \leq 1$ using the scalar scheme with complex math function/variables. Again, the Dirichlet boundary condition, $q(0) = 1$, and Neumann boundary condition, $\boldsymbol{a}q \cdot \hat{\boldsymbol{n}}|_{x=0} = -1$ are considered, leading to the exact solution,

$$q(x) = \cos(8\pi x) + \mathrm{i}\sin(8\pi x) \tag{A.3}$$

The results of simulations with the Dirichlet left boundary condition and various numbers of elements are presented in fig. A.3, while those for simulations employing a Neumann left boundary condition are presented in fig. A.4. It can be seen that neither results in a nodally exact solution.

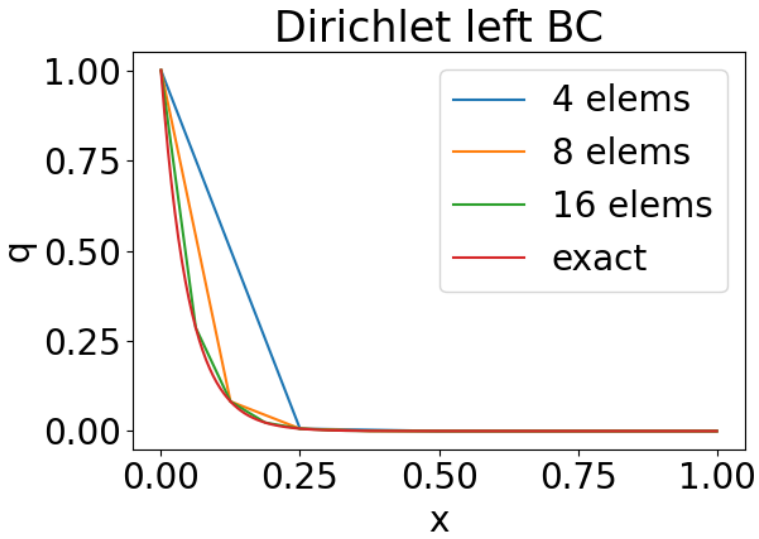


Figure A.1: Solution of the test problem with $a = 1$, $s = -20$ with the strongly enforced Dirichlet boundary condition, $q(0) = 1$.

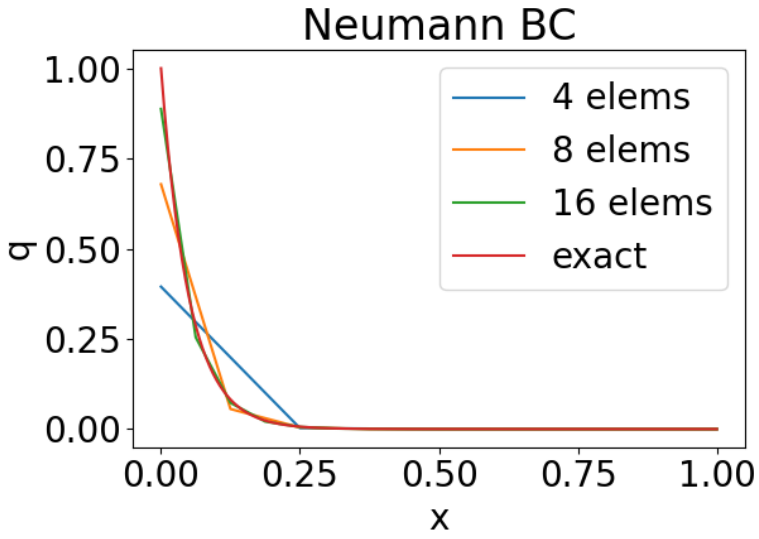


Figure A.2: Solution of the test problem with $a = 1$, $s = -20$ with the weakly enforced Neumann boundary condition, $\boldsymbol{a}q \cdot \hat{\boldsymbol{n}}|_{x=0} = -1$.

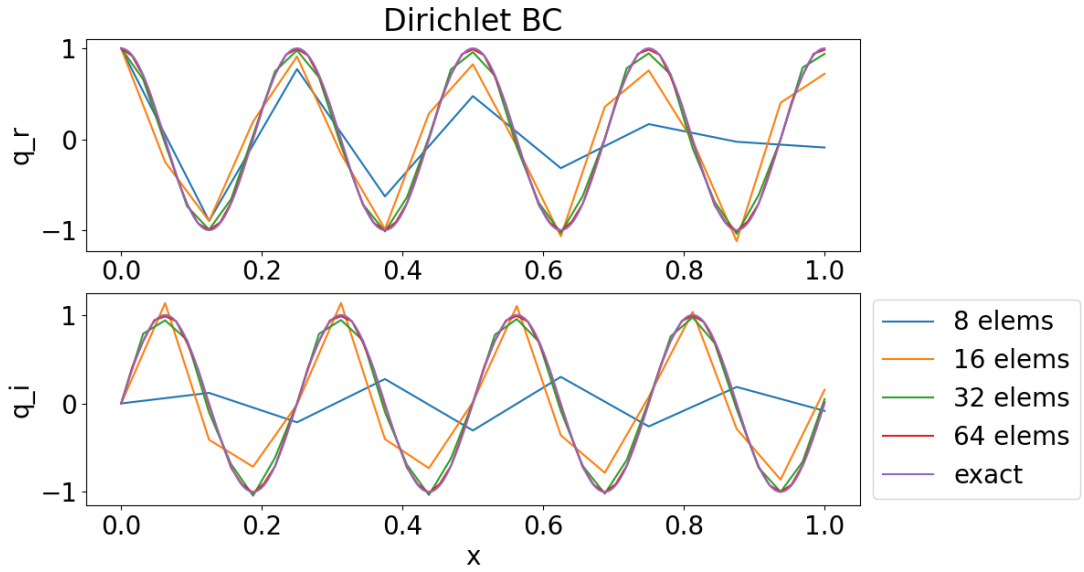


Figure A.3: Solution of the test problem with $a = 1$, $s = \mathrm{i}8\pi$ with the strongly enforced Dirichlet left boundary condition, $q(0) = 1$ and extrapolated right boundary condition. The plot with axis "q_r" is a plot of the real component of the field while the plot with axis "q_i" is the imaginary component.

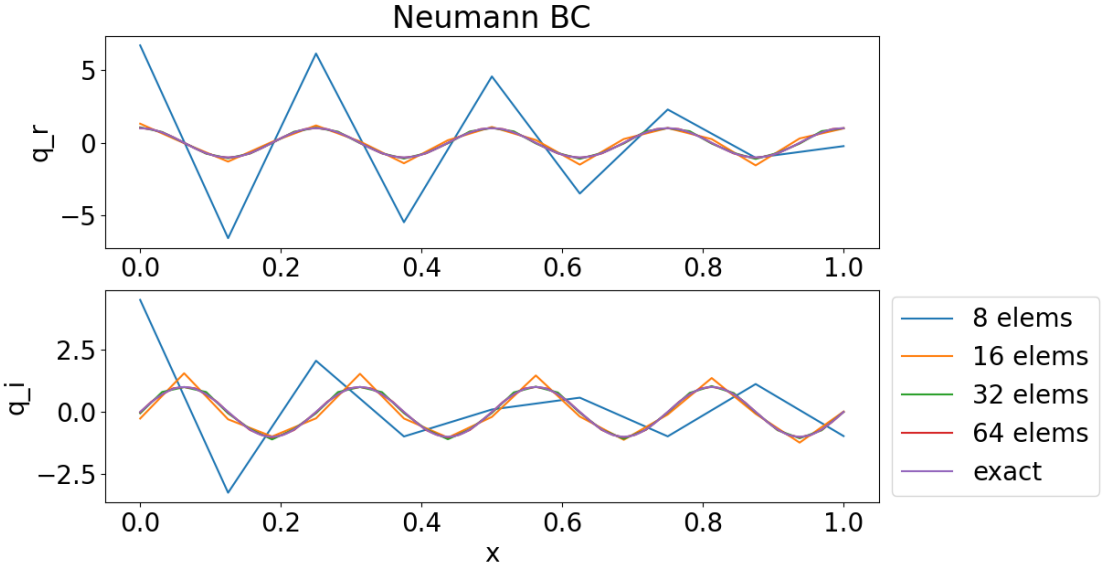


Figure A.4: Solution of the test problem with $a = 1$, $s = \mathrm{i}8\pi$ with the weakly enforced Neumann boundary condition, $\boldsymbol{a}q \cdot \hat{\boldsymbol{n}}|_{x=0} = -1$ and extrapolated right boundary condition. The plot with axis "q_r" is a plot of the real component of the field while the plot with axis "q_i" is the imaginary component.

With the Dirichlet boundary condition, the solution quality is still reasonable, but oscillations seem to be heavily damped at the coarsest grid resolution. The use of the weakly enforced Neumann boundary condition results in a poor solution that greatly overshoots/undershoots the true solution at the coarsest grid resolution, but the solution quality improves greatly once the spatial oscillation is better resolved. Interestingly, as seen in fig. A.5, use of Dirichlet boundary conditions at both boundaries, technically an ill-posed problem setup, does result in the scheme producing nodally exact solutions even on the coarsest grid resolution considered.

A.3 Transient advection-reaction/advection-oscillation

The transient advection-reaction/advection-oscillation equation adds a time derivative to eq. A.1, resulting in

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = sq. \tag{A.4}$$

This is more similar to the transient scenarios that the ideal two fluid algorithm is to be tested with and will ultimately be used to simulate if it performs adequately. As mentioned in chapter 4, this equation is satisfied by

$$q\left(x,t\right) = c_2 \mathrm{e}^{c_1\left(t-\frac{x}{a}\right)+\frac{s}{a}x} \tag{A.5}$$

with constants c_1 and c_2 and by

$$q\left(x,t\right) = q_0\left(x-at\right)\mathrm{e}^{st} \tag{A.6}$$

with initial condition $q\left(x,0\right) = q_0\left(x\right)$. Solutions of the form of eq. A.5 will contain sinusoidal spatial oscillation when s and/or c_1 has an imaginary component, and they will be moving waves when c_1 has an imaginary component. The real part of c_1 and s causes exponential growth/decay across the spatial domain of a problem, and c_1 with a real part will also cause exponential growth/decay in time. This exponential growth/decay can lead to sharp layers, while the oscillations can be of short wavelength but of such small magnitude that resolving them adds little accuracy to the numerical solution. These are both situations where a numerical solver that can hit correct nodal values without having to have the grid resolution necessary to resolve these features may be desirable.

The particular case with $a = 1$, $s = \mathrm{i}10\pi$, $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$ is an interesting test case. This scenario results in non-decaying moving waves and is periodic on $0 \leq x \leq 1$. Since c_1 is

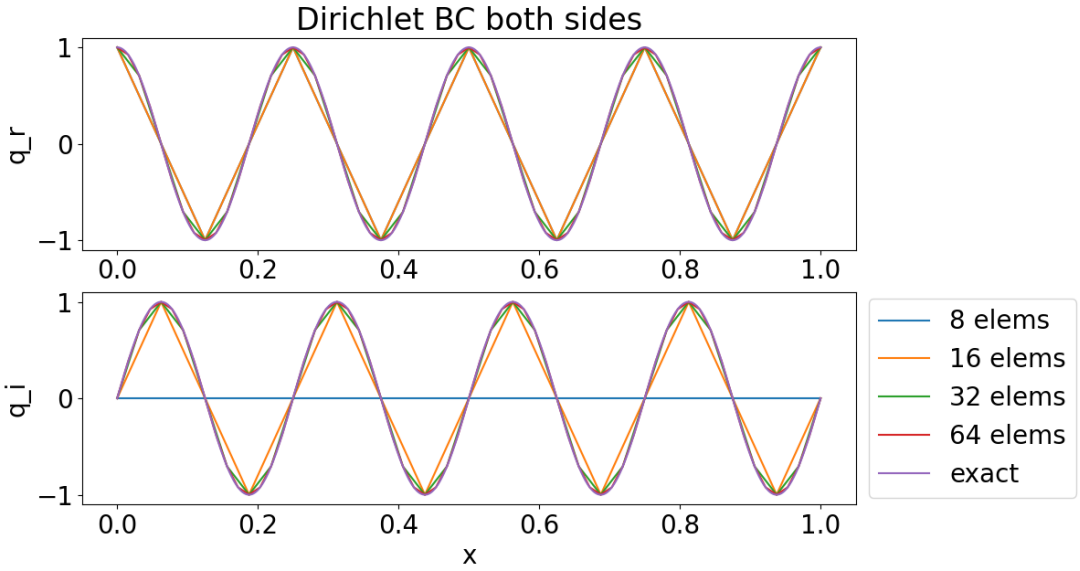


Figure A.5: Solution of the test problem with $a = 1$, $s = \mathrm{i}8\pi$ with the strongly enforced Dirichlet left and right boundary conditions, $q(0) = q(1) = 1$. The plot with axis "q_r" is a plot of the real component of the field while the plot with axis "q_i" is the imaginary component. The 8 element solution and 16 element solution for q_r coincide, also, the 8 element solution for q_i is indeed nodally exact, the solution is zero at all of the nodes in that grid.

known, the stabilized method derived based on the transient equation solution of eq. A.5 can be used. (See section 4.3.2 for the details.) With periodic boundary conditions, or Dirichlet boundary conditions on both boundaries, this method can maintain a nodally exact solution even when the waves are poorly resolved as shown in fig. A.6. (Periodic BC solution shown, Dirichlet BC solution is essentially the same.) The stabilized method based on the steady state equation does not produce nodally exact solutions, but performs reasonably well with periodic boundary conditions and sufficient grid resolution. It can be seen in fig. A.7, that simulation with 32 elements, 8 elements per wavelength, produces a solution with only a small phase shift after 4 units of time. The phase shift seen with periodic boundary conditions increases dramatically when the grid is coarsened to 16 elements, but the wave amplitude appears to be consistent. The solution is of similar quality with strongly enforced Dirichlet boundary conditions at 32 elements, but the quality decreases dramatically at 16 elements. Simulation with Neumann left/ extrapolated right boundary conditions also results in similar solution quality at 32 elements, though some overshooting of the exact solution is noted at wave peaks/troughs by the final time, as shown in fig. A.8. Comparing that solution to the 16 element solution shown in fig. A.9 gives the impression that overshooting/undershooting could be an issue for underresolved oscillation in areas influenced by Neumann boundary conditions. As in the steady state case, even the transient solution based stabilization can't maintain a nodally exact solution with the Neumann left/ extrapolated right boundary condition setup. The final time solution obtained using that method can be seen in fig. A.10, and does not appear to perform much better than the steady state solution based stabilization for this case.

Adding a large negative component to s results in a rapid decay of the solution magnitude resulting in a sharp layer near the left boundary. The final time solution of a simulation of the transient advection-reaction equation with $s = -20 + \mathrm{i}10\pi$ and the other parameters as before, using the steady state solution based stabilization on a grid too coarse to resolve the sharp layer, or oscillations, is shown in fig. A.11 while simulations employing the SUPG FEM and Galerkin FEM are shown in figs. A.12 and A.13, respectively. These simulations all employed the Neumann left boundary condition and extrapolated right boundary condition. The behavior of the stabilized method based on the steady state advection-reaction equation is comparable to that seen in the steady state simulations with Neumann left boundary conditions. It is arguably an improvement

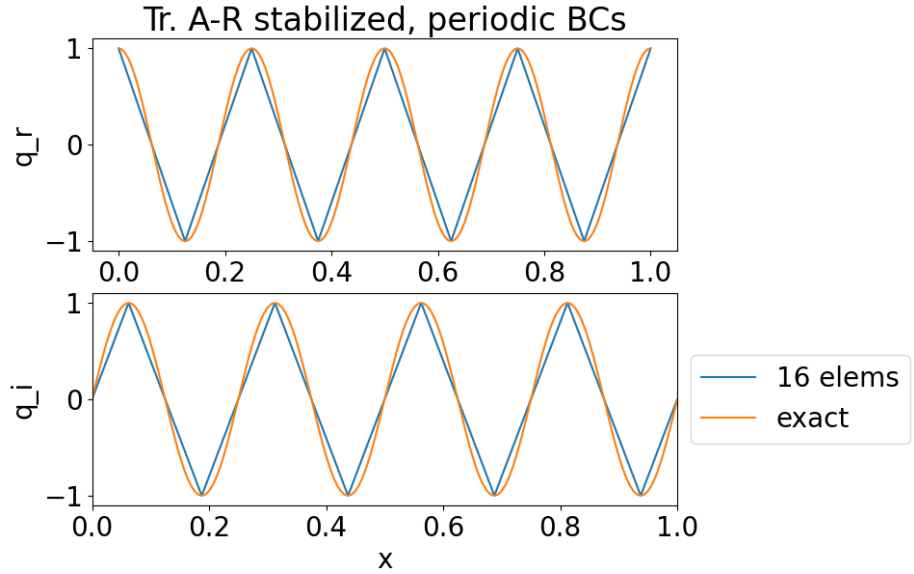


Figure A.6: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the transient solution based stabilization for linear elements with implicit trapezoid time scheme on 16 elements with periodic boundary conditions.

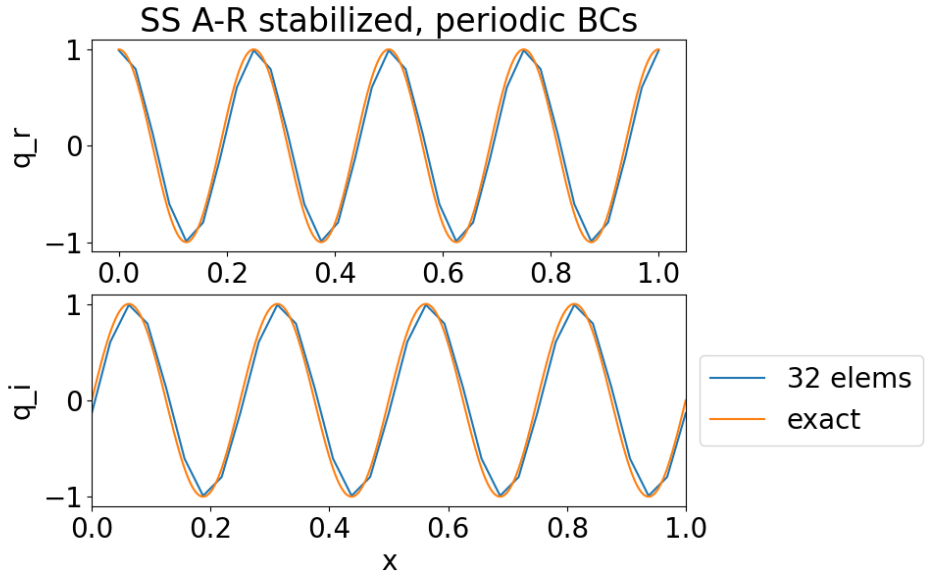


Figure A.7: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the steady state solution based stabilization for linear elements with implicit trapezoid time scheme on 32 elements with periodic boundary conditions.

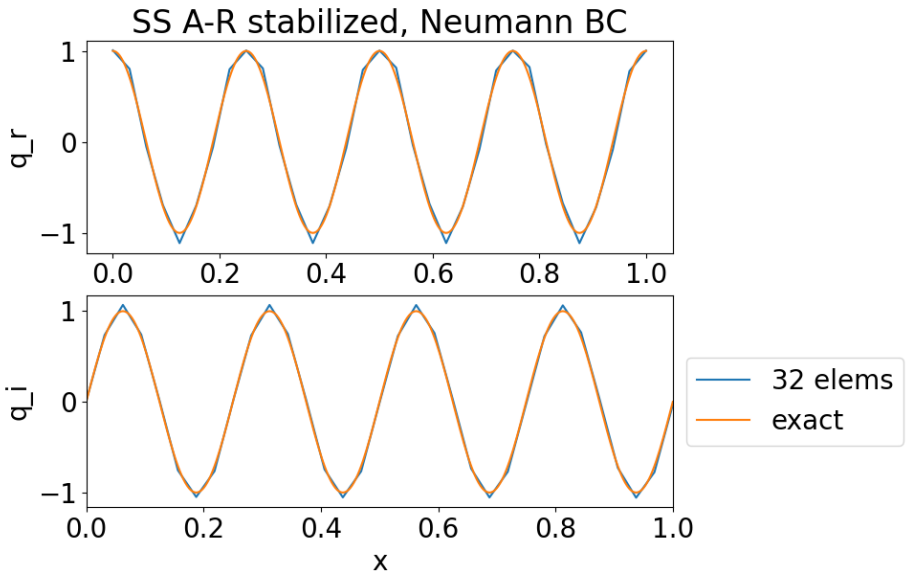


Figure A.8: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the steady state solution based stabilization for linear elements with implicit trapezoid time scheme on 32 elements with Neumann left boundary condition and extrapolated right boundary condition.

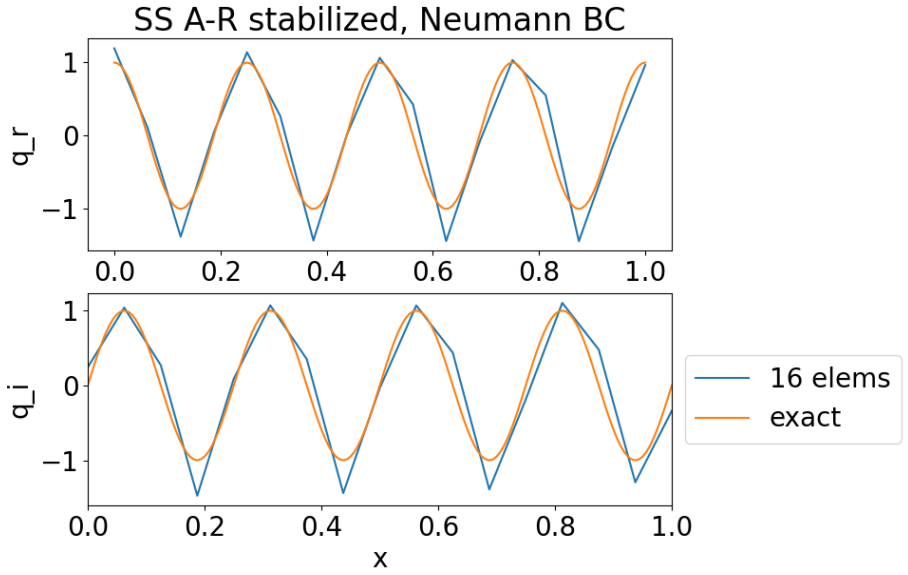


Figure A.9: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the steady state solution based stabilization for linear elements with implicit trapezoid time scheme on 16 elements with Neumann left boundary condition and extrapolated right boundary condition.

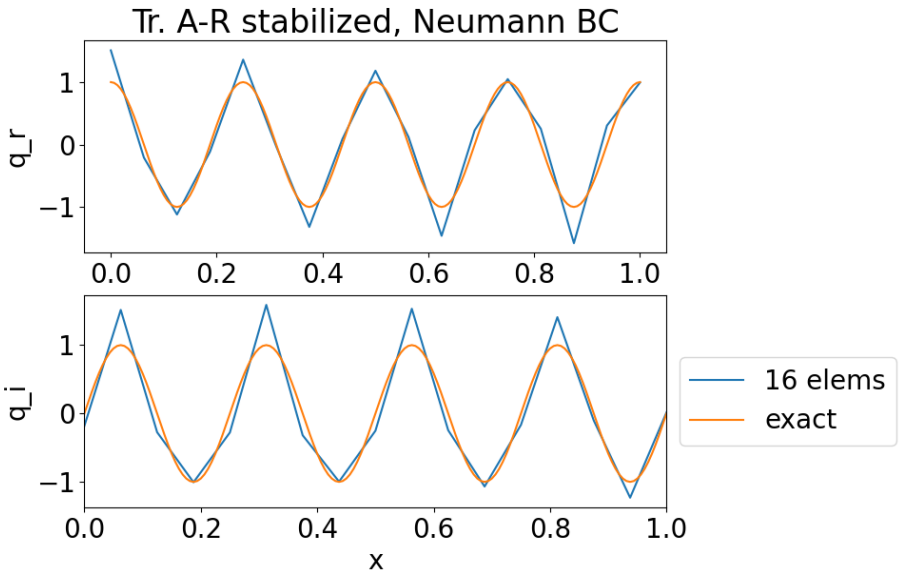


Figure A.10: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the transient solution based stabilization for linear elements with implicit trapezoid time scheme on 16 elements with Neumann left boundary condition and extrapolated right boundary condition.

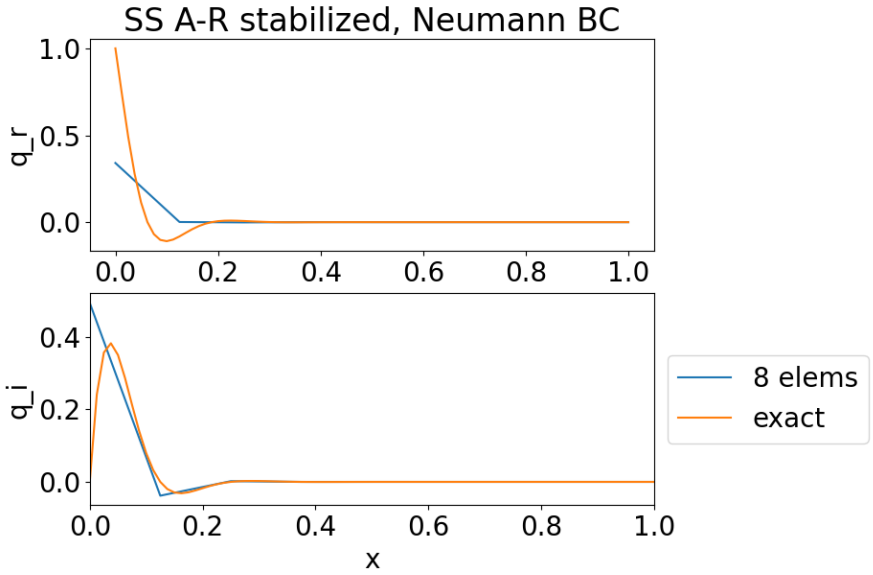


Figure A.11: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = -20 + \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the steady state solution based stabilization for linear elements with implicit trapezoid time scheme on 8 elements with Neumann left boundary condition and extrapolated right boundary condition.

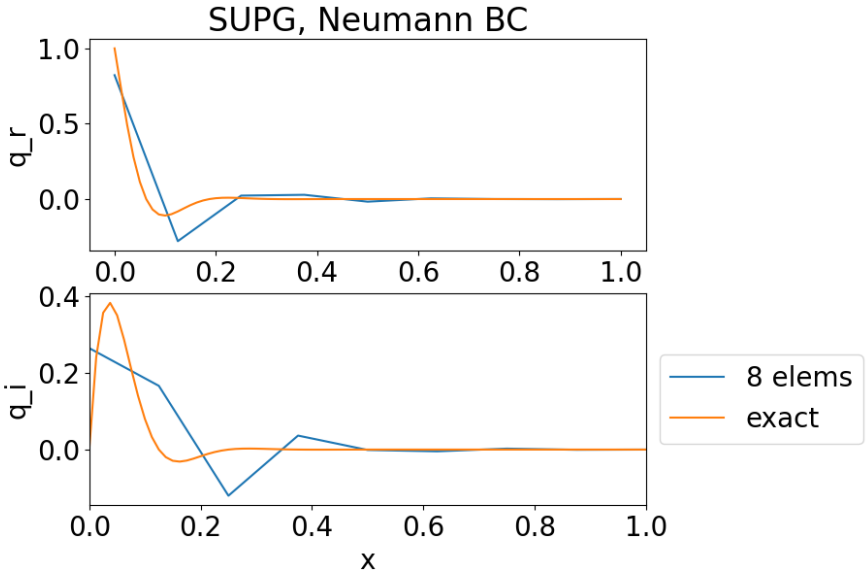


Figure A.12: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = -20 + \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the SUPG FEM on 8 elements with Neumann left boundary condition and extrapolated right boundary condition.

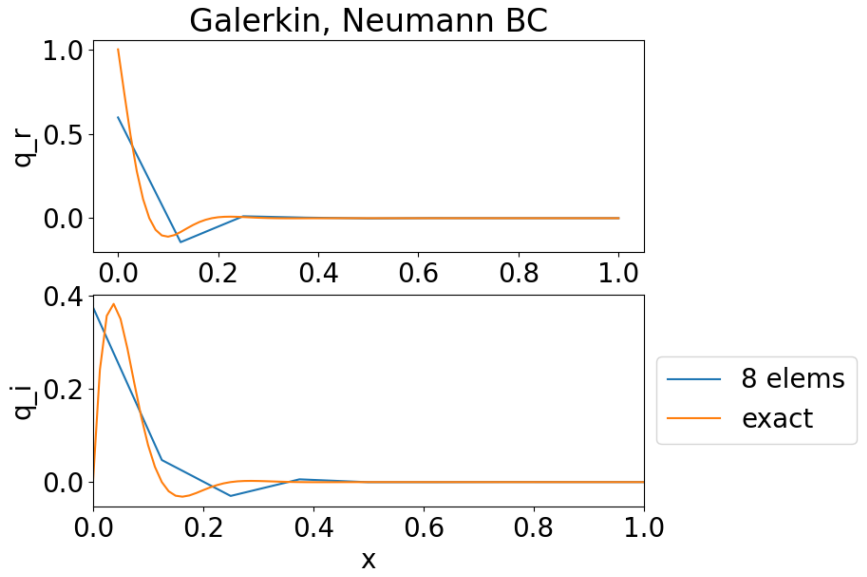


Figure A.13: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = -20 + \mathrm{i}10\pi$, and initial/boundary conditions resulting in the solution given by eq. A.5 with coefficients $c_1 = \mathrm{i}2\pi$, and $c_2 = 1$. Simulation with the Galerkin FEM on 8 elements with Neumann left boundary condition and extrapolated right boundary condition.

over the SUPG solution, which develops some spurious solution oscillation. Interestingly, the Galerkin method's solution appears to be quite good in this case, it is the only one that maintains the numerical solution between the extrema of the exact solution, at least at the presented instant in time. It is also unable to do that at other times, but that is difficult in this case since the left boundary value is oscillating over time and there is such rapid decay due to the large, negative real component of s.

The preceding cases both had solutions that have the basic form of eq. A.5, scenarios with the more general solution of eq. A.6 will now be considered. Since these will not have the basic solution form, the problem specific constant used in the transient advection-reaction equation based stabilization calculation is not available. Some informal testing consisting of simulation of problems with the basic solution structure with a wrong value for c_1 used in the stabilization calculations, and problems with the more general characteristic solution structure with ad-hoc values for c_1 made it apparent that, without significant work towards determining an optimized value for that parameter, the approach was unlikely to outperform the steady state solution based stabilization for real problems where c_1 is unknown or the solution does not have the basic form. Thus, that has not been pursued at this time and only the steady state solution based stabilization has been pursued further.

An example scenario that illustrates the performance of the method with the steady state advection-reaction equation based stabilization on problems with a solution of the form of eq. A.6 is the case with $a = 1$ and $s = 0.2$ on $0 \leq x \leq 1$ with initial condition,

$$q(x, t = 0) = \begin{cases} -1 & \text{if } x \leq 0.25 \\ 1 & \text{if } 0.25 < x \leq 0.75 \\ -1 & \text{if } x > 0.75 \end{cases}, \tag{A.7}$$

and periodic boundary conditions. The exact solution is simply a growing square wave cycling across the domain. Solutions obtained on a 64 linear element mesh using the Galerkin method, SUPG method, and steady state advection-reaction equation derived stabilized method with the implicit trapezoid rule time term are presented in fig. A.14. It can be seen that the steady state advection-reaction equation derived method, in the plot labelled "SS A-R stabilized", is very

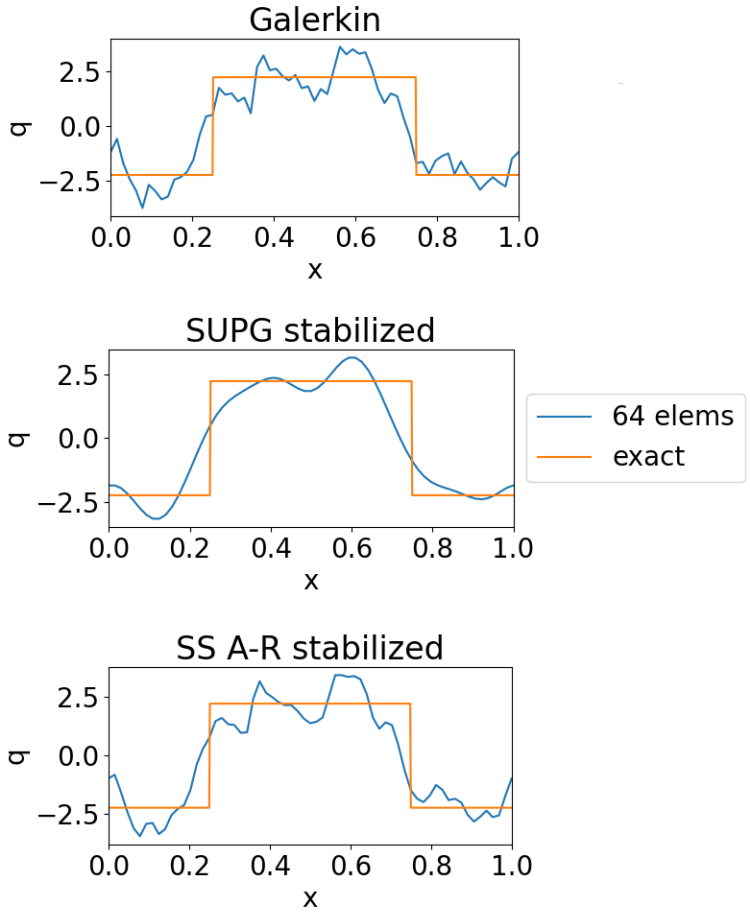


Figure A.14: Solution at $t = 4$ for the transient advection-oscillation problem with $a = 1$, $s = 0.2$, with a square wave initial condition and periodic boundary conditions using various finite element methods with 64 linear elements and the implicit trapezoid rule time term.

similar to the Galerkin method's solution, with only a little bit of the node-to-node oscillation smoothed out. It should be noted that the τ of this method behaves similarly to the τ of the original advection-diffusion equation based SUPG method but in the parameter, $\theta = \frac{sh}{a}$, instead of the element length scale Peclet number, $P = \frac{ah}{2k}$. It smoothly asymtotes to values of $-\frac{1}{3}$ and $\frac{1}{3}$ as $\theta \to -\infty$ and $\theta \to \infty$, respectively, passing through 0 at $\theta = 0$ With the values of s and a chosen and the small h due to the 64 element grid, θ is quite small for this case. Near $\theta = 0$, the τ indicated by eq. 4.48 is nearly zero, so it is not surprising that the method behaves similarly to the Galerkin method in this case. It was somewhat surprising that that is the prescribed behavior, however. This behavior is particularly ill suited to the transient equation when $s = 0$, resulting in simply a transient advection equation, for which something approaching the pure advection SUPG τ would be expected to work well. In that case, the steady state advection-reaction equation based stabilized method ends up approaching the Galerkin method, instead. The cause of this is likely that the steady state equation's basic solution, upon which the stabilized method was based, is just too simple. For it, when $s = 0$ the solution is just a constant value of q across the domain and the Galerkin method does, indeed, provide a nodally exact solution, while it is only with nonzero s that solution curvature arises and stabilization is needed to maintain a nodally exact solution. The transient equation's solutions are not like that, though, they can be non-trivial when $s = 0$. When s is positive, the method effectively shifts weight upwind as θ becomes larger much like SUPG did with a rising element Peclet number, however, when s is negative it, rather peculiarly, shifts weight downwind. While that apparently works for steady state simulations, it is another behavior that seems ill suited to use in transient simulations, given that the transient equation is known to have a characteristic solution.

All of the simulations discussed here were performed on uniform grids with the smallest number of timesteps that results in a CFL number less than one.

A.4 steady state advection-reaction two equation system

Having some insight into the performance of the method for scalar equations, the next step is to see how the extension to systems of equations performs. The steady state systems are evaluated first since the method is expected to perform well for them, if properly implemented. As seen with the scalar scheme, the method works properly with full Dirichlet boundary conditions, but loses

its ability to produce nodally exact solutions with the upwinded Neumann/extrapolated boundary conditions that are more appropriate and more likely to be used in real computations. To illustrate the performance of the method, simulations of the system of advection-reaction equations,

$$\boldsymbol{A}\frac{\mathrm{d}}{\mathrm{d}x}\boldsymbol{q} = \boldsymbol{S}\boldsymbol{q}, \tag{A.8}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.5 \\ 0.25 & -0.3 \end{bmatrix} \tag{A.9}$$

and

$$\boldsymbol{S} = \begin{bmatrix} -1.888 & 2.334 \\ 3.5116 & 4.6788 \end{bmatrix} \tag{A.10}$$

are considered. The values of $\boldsymbol{S}$ are not rounded, but are constructed to result in $\boldsymbol{A}^{-1}\boldsymbol{S}$ having the eigenvalues -0.1 and -15 with the arbitrarily chosen $\boldsymbol{A}$ and an arbitrarily chosen set of eigenvectors for $\boldsymbol{A}^{-1}\boldsymbol{S}$. These eigenvalues result in a solution with one characteristic field having a relatively sharp layer at the left boundary, while the other is nearly linear across the domain, $0 \leq x \leq 1$. The chosen $\boldsymbol{A}$ also has one positive and one negative eigenvalue such that one characteristic field will have a Neumann boundary condition and the other will be extrapolated at each boundary. The solution coefficient for each characteristic field is chosen to be 1 and the boundary conditions are set based on the exact solution for the scenario. The solution with Dirichlet boundary conditions is shown in fig. A.15 and can be seen to be nodally exact, both in terms of the conserved variables and characteristic variables. Interestingly, the solution quality with Neumann/extrapolated boundary conditions is significantly affected by the approach taken to recover the Galerkin weight for the integration by parts used to obtain the Neumann boundary condition terms. First, fig. A.16 presents the solution obtained with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.11}$$

and

$$\boldsymbol{S} = \begin{bmatrix} -0.1 & 0 \\ 0 & -15 \end{bmatrix}, \tag{A.12}$$

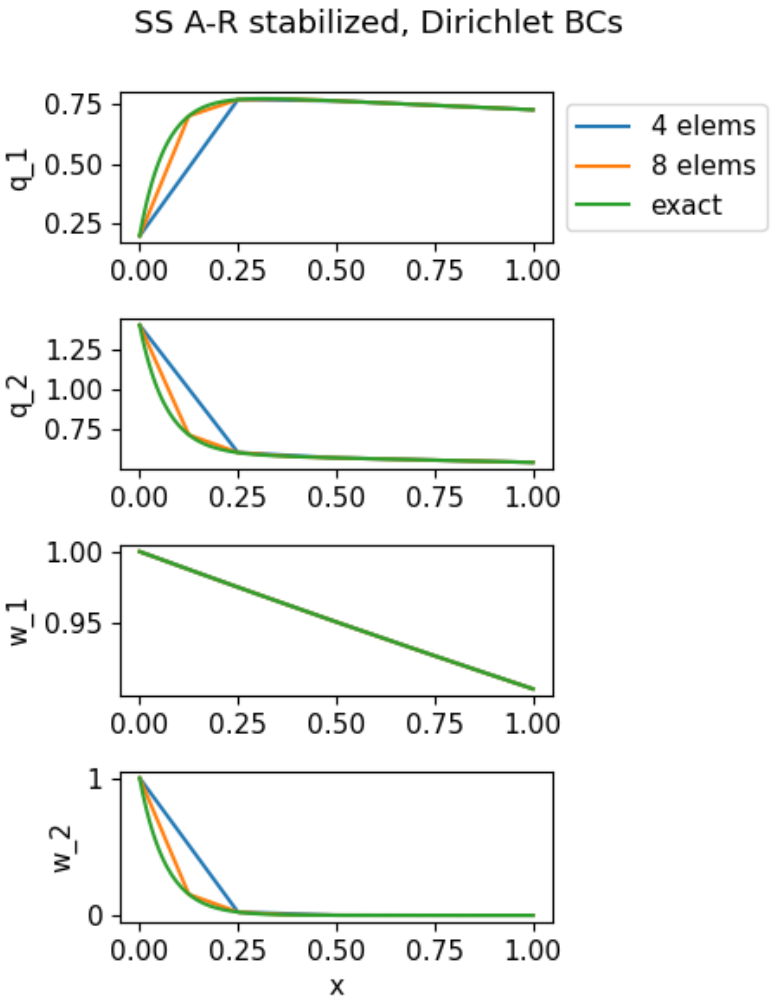


Figure A.15: Results of simulation of the system of steady state advection-reaction equations defined by eqs. A.9 and A.10 using the stabilized method for systems of equations based on the advection-reaction equation with the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.74 and 4.75, respectively. The conserved variables are denoted "q_1" and "q_2" while the characteristic variables are denoted "w_1" and "w_2".

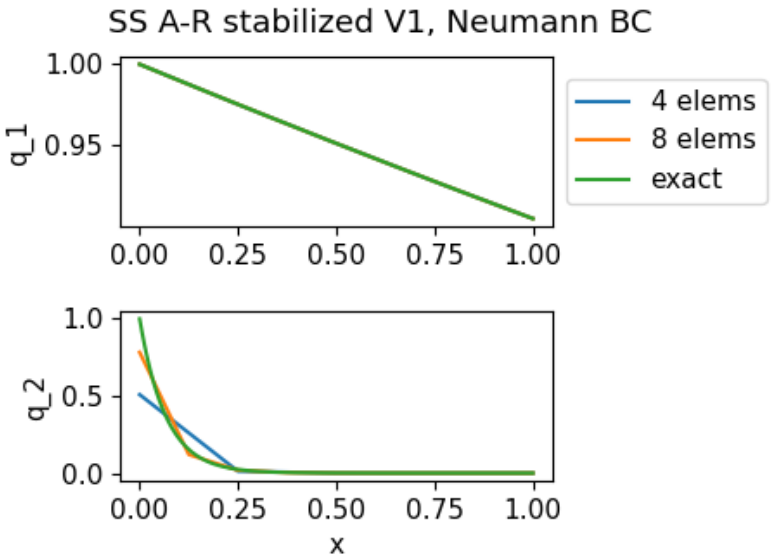


Figure A.16: Results of simulation of the system of steady state advection-reaction equations defined by eqs. A.11 and A.12 (The equivalent characteristic variable system) using the stabilized method for systems of equations based on the advection-reaction equation with the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.68 and 4.69, respectively. For this case (using the characteristic variables directly) the use of eqs. 4.68 and 4.69, or eqs. 4.74 and 4.75 yield the same result since $\boldsymbol{A}$ and the resulting $\boldsymbol{R}$ are identity matrices.

and appropriately transformed boundary conditions so that this amounts to solving the same problem expressed directly in the characteristic variables, where it is just two uncoupled scalar equations. The performance with upwinded Neumann/extapolated boundary conditions is comparable to that seen with the scalar scheme in this case, but when the conserved variables are utilized, node-to-node oscillations are seen even with substantially finer grids. Additionally, when the approach of eqs. 4.68 and 4.69 is used, there is an offset between the numerical solution and the exact solution. This is shown in fig. A.17 while the result with the stabilizing parameters of eqs. 4.74 and 4.75 is shown in fig. A.18. It can be seen that the offset and oscillations do shrink dramatically with grid refinement in both cases.

A.5 Transient advection-oscillation as a two equation system

A transient avection-oscillation equation can be written as a two equation system with real coefficients and real conserved variables in the same way as described for the steady state equation in sec. 4.3.2. The problem to be considered to illustrate the method's performance for this scenario is the same as the transient advection-oscillation scenario considered in sec. A.3 as a single equation with complex valued coefficients and fields. That complex valued single field/equation problem with coefficients, $a = 1$, $s = \mathrm{i}10\pi$, $c_1 = \mathrm{i}2\pi$, $c_2 = 1$ is analogous to the real valued two field/equation problem with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{A.13}$$

$$\boldsymbol{S} = \begin{bmatrix} 0 & -10\pi \\ 10\pi & 0 \end{bmatrix}. \tag{A.14}$$

and exact solution,

$$\boldsymbol{q} = \begin{bmatrix} \cos\left(2\pi t + 8\pi x\right) \\ \sin\left(2\pi t + 8\pi x\right) \end{bmatrix}. \tag{A.15}$$

Note that the solution fields are both real valued and all of the coefficients defining the problem are real when expressed as a two equation system. This would suggest that it should be possible to employ real math functions to implement a numerical solver for this model, but it was actually found to be necessary to use complex math functions when the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.68

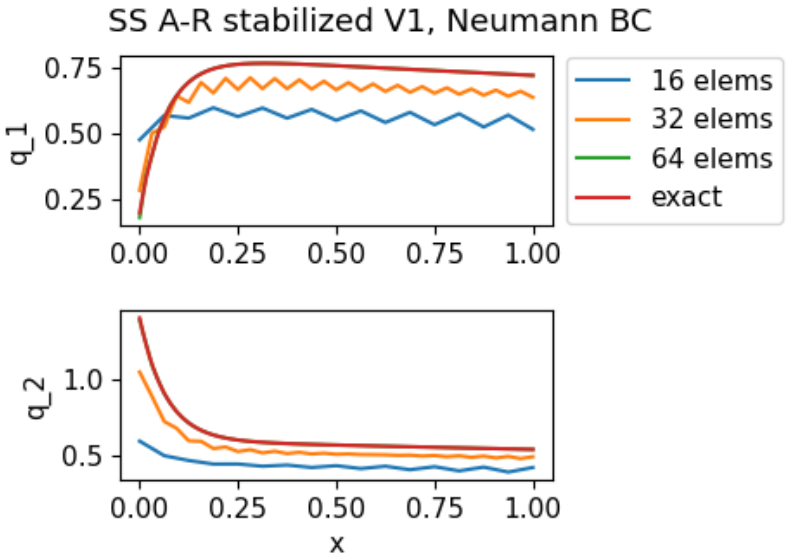


Figure A.17: Results of simulation of the system of steady state advection-reaction equations defined by eqs. A.9 and A.10 using the stabilized method for systems of equations based on the advection-reaction equation with the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.68 and 4.69, respectively. The 64 element solution is overlapped by the exact solution.

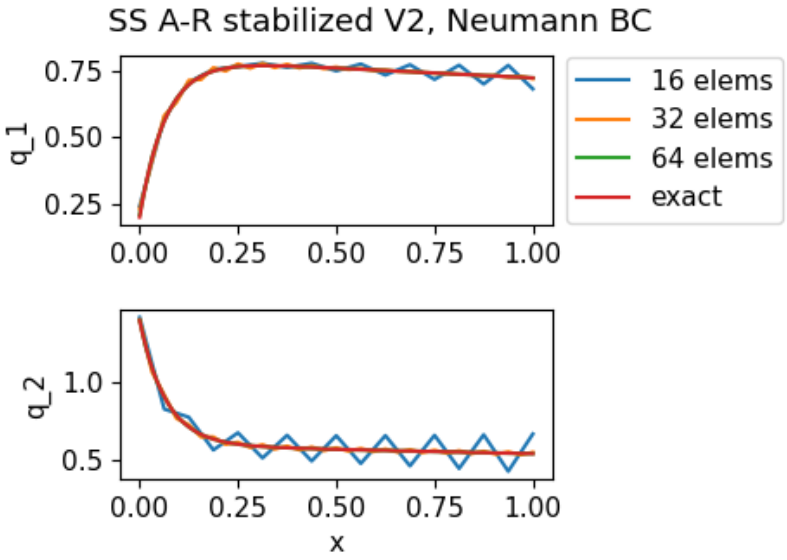


Figure A.18: Results of simulation of the system of steady state advection-reaction equations defined by eqs. A.9 and A.10 using the stabilized method for systems of equations based on the advection-reaction equation with the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.74 and 4.75, respectively. The 32 and 64 element solutions are significantly overlapped with the exact solution, node-to-node oscillation in the 32 element solution is most evident in q_1 near $x = 0.1$ where there is the most solution curvature.

and 4.69 were used because they often result in complex valued weight matrices. Non-negligible imaginary weight components have not been observed when employing the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eq. 4.74 and 4.75, however. This advantages the latter approach of recovering the Galerkin weight for systems of equations that may have complex eigenvalues, but have real valued coefficients and should have real valued solutions. The $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ matrices resulting from eqs. 4.68 and 4.69 for this problem on the domain $0 \leq x \leq 1$, discretized with 32 linear elements, are approximately

$$\boldsymbol{\tau}_1 \approx \begin{bmatrix} -1 - \mathrm{i}0.707 & 0.707 \\ \mathrm{i}0.707 & -0.293 \end{bmatrix} \tag{A.16}$$

and

$$\boldsymbol{\tau}_2 \approx \begin{bmatrix} 1.47 \times 10^{-3} - \mathrm{i}6.62 \times 10^{-17} & 6.62 \times 10^{-17} + \mathrm{i}1.47 \times 10^{-3} \\ 1.47 \times 10^{-3} + \mathrm{i}6.62 \times 10^{-17} & 6.62 \times 10^{-17} - \mathrm{i}1.47 \times 10^{-3} \end{bmatrix}. \tag{A.17}$$

Whereas when eqs. 4.74 and 4.75 are used, $\boldsymbol{\tau}_1$ is (trivially) a zero matrix and

$$\boldsymbol{\tau}_2 \approx \begin{bmatrix} 9.36 \times 10^{-17} & -2.08 \times 10^{-3} \\ 2.08 \times 10^{-3} & 9.36 \times 10^{-17} \end{bmatrix}. \tag{A.18}$$

Clearly, the imaginary components of some of the entries in both $\boldsymbol{\tau}$'s obtained using eqs. 4.68 and 4.69 are not just round-off error that can be dropped, whereas the imaginary components of the entries in the $\boldsymbol{\tau}$'s obtained using eqs. 4.74 and 4.75 do evaluate exactly to zeros.

With periodic boundary conditions on a 32 linear element mesh and using the implicit trapezoid rule for the time derivative with 129 timesteps to $t = 4$, the solution does remain real to round-off error when eqs. 4.68 and 4.69 are used as can be seen in the final time fields shown in fig. A.19, but this is not the case when equivalent Neumann/extrapolated boundary conditions are employed. When the problem is setup with Neumann/extrapolated boundary conditions, the method appears unstable for reasonable time step sizes when the implicit trapezoid rule is used, but a solution can be obtained with a backward difference time derivative and the same timestep size that was used in the simulation that employed periodic boundary conditions. It can be seen that the solution fields pickup an imaginary component that is small, but greater than roundoff error in the final time solution presented in fig. A.20. Neither this development of an erroneous imaginary component in

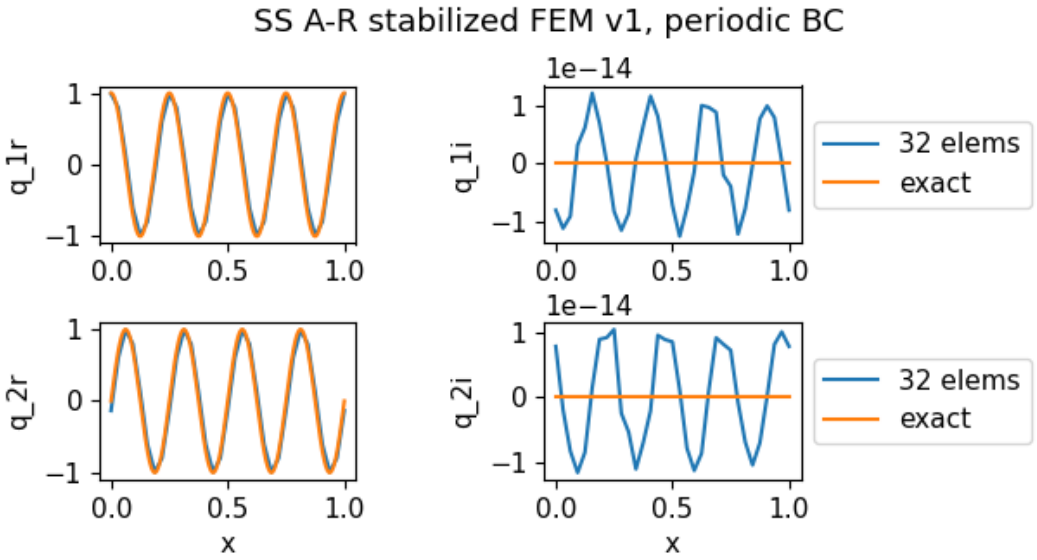


Figure A.19: Results of simulation of the system of transient advection-reaction equations defined by eqs. A.13 and A.14 using the stabilized method for systems of equations based on the advection-reaction equation with the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.68 and 4.69, respectively. Simulation using periodic boundary conditions, implicit trapezoid time difference and 129 timesteps, results labelled "q_nr" and "q_ni" are the real and imaginary components of field "q_n," respectively.

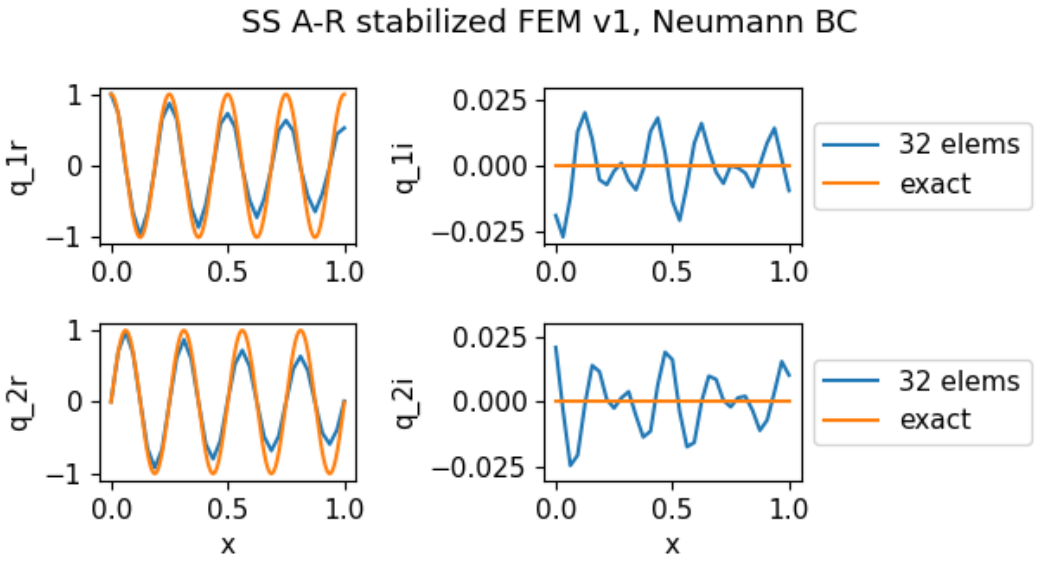


Figure A.20: Results of simulation of the system of transient advection-reaction equations defined by eqs. A.13 and A.14 using the stabilized method for systems of equations based on the advection-reaction equation with the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.68 and 4.69, respectively. Simulation using Neumann/extrapolated boundary conditions, implicit Euler time difference and 129 timesteps.

the solution fields nor the significantly degraded stability with Neumann/extrapolated boundary conditions are experienced when using the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ definitions of eqs. 4.74 and 4.75. The results obtained using that approach with the same discretizations, except that the implicit trapezoid time difference is used in both cases, are shown in fig. A.21 for periodic boundary conditions and fig. A.22 for Neumann/extrapolated boundary conditions. In the results discussed for this scenario, the real part of field "q_1" is comparable to the real part of the solution to the scalar advection-oscillation equation described in sec. A.3 while the real part of field "q_2" is comparable to the complex part of that solution. It can be seen that the extension to system of equations resulting in the $\boldsymbol{\tau}$ matrices of eqs. 4.74 and 4.75 results in very similar results to the complex valued single field/equation version. Regardless of which $\boldsymbol{\tau}$ definitions are used, it is desirable to be able to return to real math functions as quickly as possible once any complex numbers from the eigensystem of $\boldsymbol{A}^{-1}\boldsymbol{S}$ have been handled. When using the $\boldsymbol{\tau}$ matrices of eqs. 4.74 and 4.75 it appears that any imaginary components of the $\boldsymbol{\tau}_2$ matrix are likely erroneous and can simply be dropped. It was noted that for this problem, the $\boldsymbol{\tau}$ matrices obtained via those equations truly had no entries with nonzero imaginary components. It is possible to establish that the $\boldsymbol{\tau}_2$ defined by eq. 4.75 will be real for up to five equation, five unknown systems of advection-reaction equations with real $\boldsymbol{A}$ and $\boldsymbol{S}$, for which $\boldsymbol{A}^{-1}\boldsymbol{S}$ is diagonalizable but has any possible number of complex eigenvalues using the symbolic mathematics program, SymPy [65], and the facts that:

- For real $\boldsymbol{A}$ and $\boldsymbol{S}$, if $\boldsymbol{A}^{-1}$ exists, it will also be real and thus $\boldsymbol{A}^{-1}\boldsymbol{S}$ will be real.
- If a real matrix has complex eigenvalues, they occur in complex conjugate pairs and the corresponding components of their associated eigenvectors are also complex conjugates.
- The $\tau_i(\lambda_i)$ function defined by eq. 4.73 has the property that $\tau_i(\lambda_i^*) = (\tau_i(\lambda_i))^*$ where the superscript $*$ denotes complex conjugation, and is real for real λ_i.

For larger sets of equations, the algebra necessary to prove that the $\boldsymbol{\tau}_2$ of eq. 4.75 is always real using only the aforementioned properties is overwhelming for the computers at hand. However, the results of numerical computations involving larger systems of equations suggest that that may be the case for them as well. Clearly, this is not the case when the $\boldsymbol{\tau}$ definitions of eqs. 4.68 and 4.69 are used as the considered two equation, two unknown system resulted in $\boldsymbol{\tau}$'s containing complex

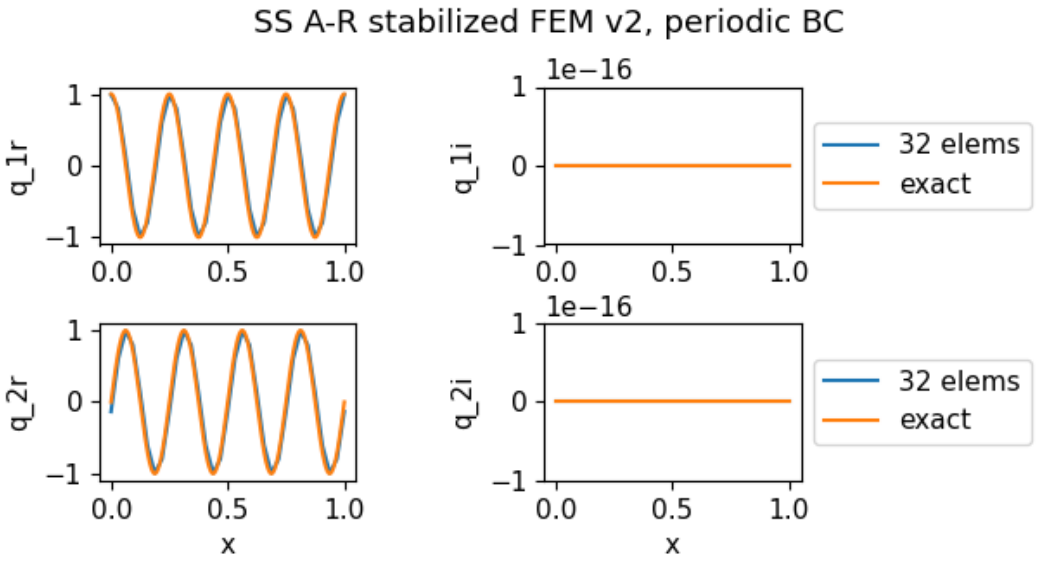


Figure A.21: Results of simulation of the system of transient advection-reaction equations defined by eqs. A.13 and A.14 using the stabilized method for systems of equations based on the advection-reaction equation with the τ_1 and τ_2 definitions of eqs. 4.74 and 4.75, respectively. Simulation using periodic boundary conditions, implicit trapezoid time difference and 129 timesteps.

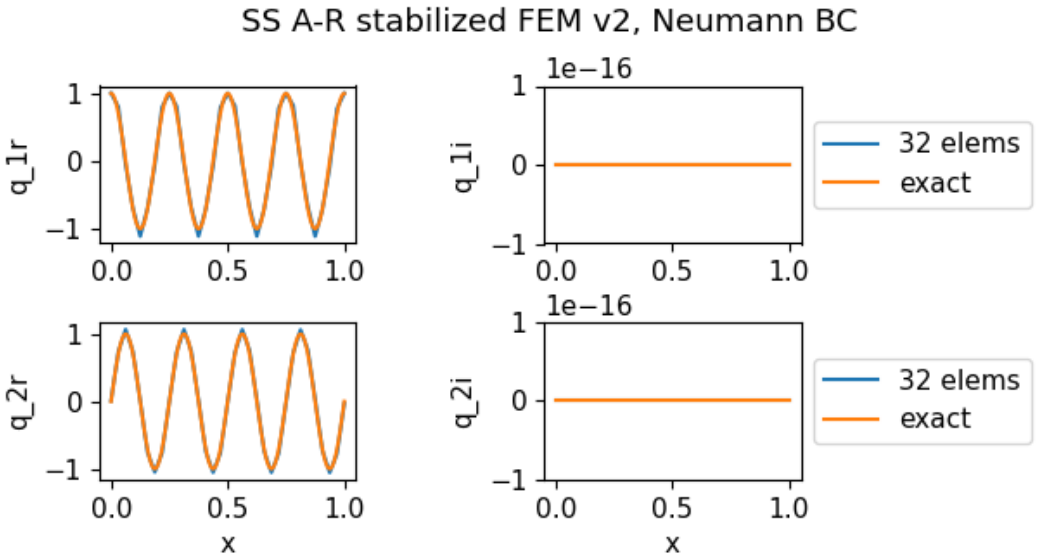


Figure A.22: Results of simulation of the system of transient advection-reaction equations defined by eqs. A.13 and A.14 using the stabilized method for systems of equations based on the advection-reaction equation with the τ_1 and τ_2 definitions of eqs. 4.74 and 4.75, respectively. Simulation using Neumann/extrapolated boundary conditions, implicit Euler time difference and 129 timesteps.

components. If complex weights are used, but the solution or solution update is assumed to be real, the complex global system, which can be expressed as

$$\left(\mathbf{A}_r + \mathrm{i}\mathbf{A}_i\right) \boldsymbol{\Delta q}_r = \left(\mathbf{b}_r + \mathrm{i}\mathbf{b}_i\right), \tag{A.19}$$

where the r and i subscripts denote matrices of the real and imaginary components of the global matrices/vectors, respectively, can be broken into two decoupled real global systems:

$$\mathbf{A}_r \boldsymbol{\Delta q}_r = \mathbf{b}_r \tag{A.20}$$

and

$$\mathbf{A}_i \boldsymbol{\Delta q}_r = \mathbf{b}_i. \tag{A.21}$$

If these systems are not satisfied by the same $\boldsymbol{\Delta q}_r$, then obviously the assumption that $\boldsymbol{\Delta q}$ must be real can't be made and the global system and fields of the problem must be treated as complex numbers, but if they are satisfied by the same $\boldsymbol{\Delta q}_r$, then solving either global system should work. These real global systems can be constructed by simply dropping the real or imaginary component of the $\boldsymbol{\tau}$ matrices computed using eqs. 4.68 and 4.69. Other real global systems which should be valid under the real fields assumption can be constructed by simply adding or differencing the coefficients of the real and imaginary components of the $\boldsymbol{\tau}$ matrices. For the case considered, returning to real number arithmetic by simply dropping the imaginary components of the $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ obtained using eqs. 4.68 and 4.69 results in a singular left hand side matrix when Neumann/extrapolated boundary conditions and a backward Euler time difference are used with a 32 linear element mesh and 129 timesteps. (This is the same discretization that worked when complex variables and arithmetic were maintained throughout the solver.) Keeping the imaginary components and discarding the real components does not work much better, resulting in an unstable solution for this discretization. Summing the coefficients of the real and imaginary components of the $\boldsymbol{\tau}$ entries and treating the result as a real valued $\boldsymbol{\tau}$ results in a scheme that does work with the backward Euler time difference for this problem, and results in a solution similar to that obtained using complex variables and arithmetic. Interestingly, differencing the coefficients of the real and imaginary components of the $\boldsymbol{\tau}$ works quite well, resulting in a scheme that is stable with the implicit trapezoid time difference

and produces a solution comparable to that obtained using the $\boldsymbol{\tau}$ matrix definitions of eqs. 4.74 and 4.75. This approach does not appear to work in general, though. For different $\boldsymbol{A}$ and $\boldsymbol{S}$, one of the approaches discussed for constructing real valued $\boldsymbol{\tau}$ matrices from the complex valued $\boldsymbol{\tau}$ matrices produced by eqs. 4.68 and 4.69 usually works, but it does not appear that any one of them will work every time. The more sophisticated approach of satisfying the two global systems of eqs. A.20 and A.21 in a least squares sense seems promising but does not appear to be achievable through simple operations on the element $\boldsymbol{\tau}$ or weight matrices. Given that the alternative $\boldsymbol{\tau}$ definitions of eqs. 4.74 and 4.75 do appear to provide the real $\boldsymbol{\tau}$'s sought, that has not been pursued further in this work.

Vita

Kenneth A. Croft studied as a graduate student at the University of Tennessee Space Institute.