Modeling plasma systems using a domain-hybridized physical model

Andrew Ho

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Uri Shumlak, Chair

Anne Greenbaum

Eric Meier

Brian Nelson

Marco Salviato

Program Authorized to Offer Degree:
Department of Aeronautics & Astronautics

University of Washington

Abstract

Modeling plasma systems using a domain-hybridized physical model

Andrew Ho

Chair of the Supervisory Committee:
Uri Shumlak
Department of Aeronautics & Astronautics

Plasma models have regions of validity that depend on local parameters. In some problems a computationally expensive, high-fidelity model is required in a small subset of the domain while lower-cost reduced models can adequately describe the plasma behavior everywhere else. The goal of this research is to investigate methods for spatially coupling the various plasma models such that the simplest plasma model that is locally valid can maintain global physical fidelity while improving computational efficiency. This research has two primary components. The first component is investigating numerical methods which can efficiently couple the various plasma models. Mixed implicit-explicit (ImEx) temporal solution schemes are investigated to produce schemes which are numerically stable when the time scales of interest span several orders of magnitude, and the physics of interest is dominated by long time scale physics. The spatial discretization method of choice is based on the discontinuous Galerkin (DG) method. Traditional DG methods are capable of producing numerical schemes with high spatial accuracy, but produce large and stiff implicit systems. To address this deficiency, the traditional DG scheme is augmented with a hybridized discontinuous Galerkin (HDG) scheme which is specifically designed to handle implicit temporal schemes. The second component of this research is investigating methods for making the plasma model dynamically (re-)decomposed based on local plasma conditions to produce an adaptive scheme. These metrics can be derived by examining the mathematical differences between

the plasma models. This approach allows the determination of relative scale on which non-charge-neutral effects are significant, as well as which metrics are dependent on local gradients or purely on local conditions. Key metrics comparing the MHD and two-fluid models include examining charge neutrality, two-temperature effects, Hall effects, and finite-electron-mass effects.

Contents

Glossary

$A^{(\alpha)}$ Particle mass of species α

$\boldsymbol{B}$ Magnetic field

$\boldsymbol{E}$ Electric field

$e^{(\alpha)}$ Total fluid energy density of species α

$f^{(\alpha)}$ velocity distribution function of species α

$\boldsymbol{h}^{(\alpha)}$ Fluid heat flux vector of species α

$\boldsymbol{j}$ Current density

$n^{(\alpha)}$ Fluid number density of species α

$P^{(\alpha)}$ Fluid pressure of species α

$\overleftrightarrow{\boldsymbol{P}}^{(\alpha)}$ Fluid pressure tensor of species α

$\boldsymbol{p}^{(\alpha)}$ Fluid momentum of species α

$T^{(\alpha)}$ Fluid temperature of species α

$U^{(\alpha)}$ Internal fluid energy density of species α

$\boldsymbol{u}^{(\alpha)}$ Fluid velocity of species α

$\boldsymbol{v}$ phase space velocity coordinate

$Z^{(\alpha)}$ Particle charge state of species α

$\overleftrightarrow{\mathbf{\Pi}}^{(\alpha)}$ Fluid viscous stress tensor of species α

ρ_c Charge density

$\rho^{(\alpha)}$ Fluid density of species α

Chapter 1

Introduction

A large variety of real world plasma systems have plasma parameters which span several different scales. Plasma confinement devices for fusion energy typically have hot dense core plasmas, but near the edges the density and temperature reduces so the confinement vessel itself is not subject to fusion plasma conditions. Example devices include Z-pinches[1, 2, 3, 4], field-reversed configurations[5, 6, 7], and tokamaks[8]. Astrophysical plasmas are also subject to multi-scale behavior. Reddell[9] demonstrated that during geomagnetic reconnection of current sheets the plasma near the edge of the current sheet is highly non-Maxwellian, while elsewhere it is reasonably close to Maxwellian. As a result, several reduced plasma models typically can predict the reconnected magnetic field flux reasonably well[10]. However, Stanier et al.[11] found that while Hall MHD is capable of capturing the reconnection growth rate, it is insufficient for capturing sheet formation and the coupling to larger/longer MHD scales.

There currently exist a wide variety of well developed plasma models of varying physical fidelity and computational expense. In general, models which have high physical fidelity are also computationally expensive. Many of these high fidelity models are still too expensive to use on real world plasma systems, while the cheaper models do not have the required physical fidelity to produce

accurate results. As a result of the wide variety of physics present in many plasma systems, a common approach which attempts to solve this is to globally partition the problem based on the expected behavior of individual species. For example, Stanier et al.[12] developed a hybrid kinetic ion and fluid electron model intended for investigating the interaction of magnetic reconnection at larger and longer scales. Their model treats the ions using a particle-in-cell (PIC) type scheme, while the electrons are treated as a massless fluid.

Related to a global model partition, there has been significant work on adaptive mesh refinement methods for problems with varying scale features where having a uniform resolution would be prohibitively expensive[13]. These methods examine the behavior in a local neighborhood and refine the spatial mesh resolution as needed to achieve sufficient spatial accuracy in regions of the domain while reducing the spatial resolution in quiescent regions to reduce computational cost.

A natural extension to adaptive mesh refinement and global model partitioning is to consider an adaptive model refinement method. The domain could be spatially (re-)decomposed such that the computationally cheapest plasma model required to achieve physically accurate results is used. In order to form a complete unified system, the different plasma models must be coupled together. This can be performed either using intermediate boundary conditions at model-model interfaces, or by introducing a transition domain over which the two models can be mixed progressively together to smooth out any model transition mismatch.

The resulting hybrid plasma model has to contend with the numerical stiffness associated with all constituent models. This is potentially problematic since some plasma models of interest have to contend with speed of light and electron dynamic timescales, while the primary dynamic timescale of interest may be on timescales related to ion sound speeds. To help mitigate these issues, an implicit-explicit temporal discretization is coupled with a discontinuous Galerkin (DG) spatial discretization method. DG methods only define interactions between an element and its immediate

neighboring elements. Additionally, the coupling between each element-element pair at a shared face can be independent of all other element-element or element-boundary interactions while being able to maintain higher order spatial accuracy. This is distinct from other mesh-based higher-order methods where the local coupling neighborhood typically includes a stencil of several elements, some of which may not be direct neighbors of an element in question[14]. There also exist composable modifications to the method via hybridized discontinuous Galerkin which reduce the cost of coupling the discontinuous Galerkin method to an implicit temporal scheme.

The temporal scheme chosen are based on the family of single-step, multi-stage additive Runge-Kutta methods; specifically the class of ARK_2 implicit-explicit methods. Similar to discontinuous Galerkin, single-step Runge-Kutta methods are capable of achieving high order temporal accuracy while maintaining a relatively compact local temporal neighborhood. The choice of single-step methods serves two purposes. First, single-step methods are more amenable to adaptive schemes simply because only the current system state is required in order to advance to the next state. This means that in order to adaptively change the spatial decomposition a single step method only needs to map the current system state onto the new decomposition, while multi-step methods will need to either extrapolate backwards or map multiple temporal slices of the system state onto the new decomposition. Secondly, single-step, multi-stage Runge-Kutta methods are not subject to the Dahlquist barriers[15, 16] on temporal accuracy and time step restrictions placed on multi-step temporal schemes. Specifically, the second Dalhquist barrier states that there are no linear multi-step methods which are both A-stable and achieve an order of convergence greater than 2. Implicit Runge-Kutta methods are capable of achieving high order accuracy and A-stability simultaneously.

High order numerical schemes suffer from high frequency oscillation breakdowns near steep gradients as a result of the Runge and Gibbs phenomena. These oscillations can be particularly problematic for non-linear problems such as fluid-based plasma models, leading to non-positive

density and/or temperature. Not only are these properties non-physical, they also change the fundamental characteristics of the model, and are difficult to treat numerically. For example, the local sound speed $c_s \propto \sqrt{T}$, and for ideal gases $T = \frac{P}{\rho}$. If T is negative, then the local sound speed becomes imaginary, and if $\rho = 0$, then T is undefined entirely.

There are a variety of methods which seek to resolve these issues such as limiters and artificial dissipation. Artificial dissipation methods are attractive since they're built up from the same fundamental operators used to describe the physics model. This reduces implementation complexity, and these methods can easily be defined to conserve mass, momentum, and energy as well as the base discretization method allows. A disadvantage of artificial dissipation models is that they typically have a different per-problem form and tuning parameters. Noh[17] found that for 5-moment fluids, a combination of an artificial body viscosity operator Q and an artificial heat flux $\boldsymbol{H}$ can resolve certain problems encountered by using Q alone. One example of this is problems which can occur at the contact discontinuity. The artificial viscosity operator is typically proportional to $\nabla \cdot \boldsymbol{u}$ or some power of $\nabla \cdot \boldsymbol{u}$, which is 0 across the contact discontinuity. However, the contact discontinuity could have a significant jump in density and/or temperature, leading to Runge and Gibbs phenomenon as it propagates. Artificial heat flux solves this problem by introducing dissipation on ∇T.

The last major consideration for producing an adaptive hybrid plasma model is to define meaningful metrics for what "suitable physical fidelity" means. A useful starting point for this investigation is by examining what mathematical differences are present between the different models. If two models predict different results, then it is assumed that the more physically accurate model is correct. However, this approach is potentially very expensive since it doesn't necessarily solve the problem of having to simulate the most expensive plasma model in the entire domain. By examining the mathematical differences, key physical features can be extracted to construct meaningful metrics.

As a test of the proposed hybrid numerical scheme, the behavior of a simplified plasma opening switch device is investigated in Chapter 10. Plasma opening switches operate as short term inductive energy storage devices[18]. Electrical current from an external generator is pumped through a plasma column discharge, allowing current to build up relatively slowly through the plasma, on the order of microseconds. When the plasma column disconnects, the induction of the system dumps the current into any connected target devices in the span of a few nanoseconds. The characteristics of the plasma column collapse determine the switch opening properties. The collapse characteristics are relatively localized to the collapse front; however this region moves over time as the plasma evolves.

Chapter 2 discusses the constituent plasma models being considered for hybridization in this research. Chapter 3 discusses details of the DG spatial discretization scheme used, including the generalization to non-conservative form PDE's and hybridization for efficient implicit temporal schemes. Chapter 4 discusses the Runge-Kutta temporal schemes, partitioning for mixed implicit-explicit operation, and methods for handling adaptive time stepping for complex plasma systems in a coherent, accurate, and efficient manner. Chapter 5 discusses the artificial dissipation scheme and the regions of applicability and limitations of this scheme for reducing non-physical numerical oscillations. Chapter 6 discusses the various plasma model coupling strategies as well as how these adaptivity metrics could be derived. Chapter 7 discusses the computational framework associated with implementing a model-adaptive HDG method. Chapter 8 presents the results of validating the HDG implementation on various test problems and discusses the choice of divergence cleaning parameters. Chapter 9 presents a magnetized channel flow problem along with analytical solutions for validating the implementation of the anisotropic Braginskii model. Chapter 10 applies the hybrid plasma model to model a plasma erosion opening switch and demonstrate the viability of this model.

1.1 Normalization

The normalization scheme used for this work is based on a representative magnetized proton species. The normalization is designed to be consistent for plasmas on multiple scales, from continuum kinetics to MHD. Define the normalization scheme (the original dimensional quantities are denoted with a tilde)

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}L \qquad \tilde{t} = t\tau \qquad \tilde{\boldsymbol{v}} = \boldsymbol{v}v_A \qquad v_A = \frac{L}{\tau} \qquad \tilde{m}^{(\alpha)} = A^{(\alpha)}m_p$$

$$\tilde{n}^{(\alpha)} = n^{(\alpha)}n_0 \qquad \tilde{q}^{(\alpha)} = Z^{(\alpha)}q_e \qquad \tilde{\nu}^{(\alpha\beta)} = \nu^{(\alpha\beta)}\nu_p \qquad \tilde{\boldsymbol{E}} = \boldsymbol{E}v_A B_0 \qquad \tilde{\boldsymbol{B}} = \boldsymbol{B}B_0$$

$$B_0 = \sqrt{\mu_0 m_p n_0 v_A^2} \qquad \delta_p = \frac{m_p v_A}{q_e B_0} \qquad \omega_p = \sqrt{\frac{q_e^2 n_0}{m_p \epsilon_0}} \qquad \nu_p = \frac{(\omega_p\tau)^4 \ln(\Lambda)}{n_0 L^3 \tau}$$

Assuming that the Coulomb Logarithm is near constant, then all plasma parameters can be specified in terms of the non-dimensional constants $(\omega_p\tau)$, $\left(\frac{\delta_p}{L}\right)$, and $(\nu_p\tau)$ which are the characteristic proton plasma frequency, proton skin depth, and proton collision frequency respectively. These three normalization parameters characterize the primary physical scales of problems seen in plasmas:

- $(\omega_p\tau)$ characterizes how charge neutral the plasma is. Larger values indicate the operating regime is charge neutral.
- $\frac{\delta_p}{L}$ characterizes how tightly bound magnetic fields are to the plasma. Smaller values indicate the magnetic field is tightly bound to plasma motion.
- $(\nu_p\tau)$ characterizes the mean free path and whether a continuum approximation is appropriate or not. Larger values indicate that fluid models may be valid.

While in theory highly accurate kinetic plasma models such as Particle in Cell (PIC) or continuum kinetic models such as the Vlasov and Boltzmann models can be used over a wide variety of

plasma conditions, the feasibility of using these models diminishes dramatically in certain regimes due to the increase in required computational resources. To solve this problem reduced models have been derived which apply simplifying assumptions, limiting the scope of problems for which they can be used accurately but decreasing the computational cost.

Figure 1.1 shows roughly the intersection of a model's computational feasibility and physical fidelity vs. the three primary scales of interest in plasma physics[19].

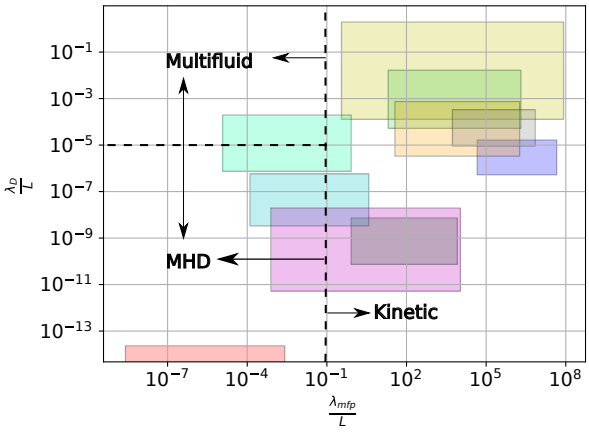


Figure 1.1: Various plasma models and the intersection of their regions of computational feasibility and validity.

Chapter 2

Plasma models

2.1 5N-moment multi-fluid equations

The 5N-moment multi-fluid equations can be derived by taking moments of the Boltzmann equation, shown in Eq. (2.1) in normalized form.

$$\partial_t f^{(\alpha)} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} f^{(\alpha)}) + \nabla_{\boldsymbol{v}} \cdot \left(\frac{\tilde{\boldsymbol{F}}^{(\alpha)}}{A^{(\alpha)}} f^{(\alpha)} \right) = \partial_t f^{(\alpha)}|_{coll} \tag{2.1}$$

$$\tilde{\boldsymbol{F}}^{(\alpha)} = \left(\frac{L}{\delta_p} \right) Z^{(\alpha)} \left(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B} \right) \tag{2.2}$$

Note that here the body force $\tilde{\boldsymbol{F}}^{(\alpha)}$ is taken to be the Lorentz force, though other body forces such as gravity could be included.

Define the density, momentum, and energy as the first five scalar moments of the distribution function.

$$\rho^{(\alpha)} = A^{(\alpha)} n^{(\alpha)} = \int_{-\infty}^{\infty} A^{(\alpha)} f^{(\alpha)} d\boldsymbol{v} \tag{2.3}$$

$$\boldsymbol{p}^{(\alpha)} = \rho^{(\alpha)} \boldsymbol{u}^{(\alpha)} = \int_{-\infty}^{\infty} A^{(\alpha)} \boldsymbol{v} f^{(\alpha)} d\boldsymbol{v} \tag{2.4}$$

$$e^{(\alpha)} = \int_{-\infty}^{\infty} \frac{1}{2} A^{(\alpha)} \boldsymbol{v} \cdot \boldsymbol{v} f^{(\alpha)} d\boldsymbol{v} \tag{2.5}$$

Assuming only non-reacting binary collisions, then the collision term may be re-written as[20]

$$\partial_t f^{(\alpha)}|_{coll} = \sum_{\beta} C^{(\alpha\beta)}(f^{(\alpha)}, f^{(\beta)}) \tag{2.6}$$

Note that the operator $C^{(\alpha\beta)}$ must satisfy conservation of total momentum and energy such that

$$\int A^{(\alpha)}\boldsymbol{v}C^{(\alpha\beta)}d\boldsymbol{v} + \int A^{(\beta)}\boldsymbol{v}C^{(\beta\alpha)}d\boldsymbol{v} = 0 \tag{2.7}$$

$$\int \frac{1}{2}A^{(\alpha)}\boldsymbol{v}\cdot\boldsymbol{v}C^{(\alpha\beta)}d\boldsymbol{v} + \int \frac{1}{2}A^{(\beta)}\boldsymbol{v}\cdot\boldsymbol{v}C^{(\beta\alpha)}d\boldsymbol{v} = 0 \tag{2.8}$$

In order to satisfy the non-reacting property, then

$$\int C^{(\alpha\beta)}d\boldsymbol{v} = 0 \tag{2.9}$$

Define the collisional momentum and energy transfer as

$$\boldsymbol{R}^{(\alpha\beta)} = \int_{-\infty}^{(\infty)} A^{(\alpha)}(\boldsymbol{v}-\boldsymbol{u}^{(\alpha)})\partial_t f^{(\alpha)}|_{coll}d\boldsymbol{v} \tag{2.10}$$

$$Q^{(\alpha\beta)} = \int_{-\infty}^{(\infty)} \frac{1}{2}A^{(\alpha)}(\boldsymbol{v}-\boldsymbol{u}^{(\alpha)})\cdot(\boldsymbol{v}-\boldsymbol{u}^{(\alpha)})\partial_t f^{(\alpha)}|_{coll}d\boldsymbol{v} \tag{2.11}$$

respectively. Notice that since there are no reactions, then

$$\boldsymbol{R}^{(\alpha\beta)} = \int_{-\infty}^{\infty} A^{(\alpha)}\boldsymbol{v}\partial_t f^{(\alpha)}|_{coll}d\boldsymbol{v} \tag{2.12}$$

$$Q^{(\alpha\beta)} = \int_{-\infty}^{(\infty)} \frac{1}{2}A^{(\alpha)}(\boldsymbol{v}\cdot\boldsymbol{v} - 2\boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{v})\partial_t f^{(\alpha)}|_{coll}d\boldsymbol{v}$$

$$Q^{(\alpha\beta)} + \boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{R}^{(\alpha\beta)} = \int_{-\infty}^{(\infty)} \frac{1}{2}A^{(\alpha)}\boldsymbol{v}\cdot\boldsymbol{v}\partial_t f^{(\alpha)}|_{coll}d\boldsymbol{v} \tag{2.13}$$

In order to satisfy conservation of momentum and energy,

$$\boldsymbol{R}^{(\alpha\beta)} + \boldsymbol{R}^{(\beta\alpha)} = 0 \tag{2.14}$$

$$Q^{(\alpha\beta)} + Q^{(\beta\alpha)} = (\boldsymbol{u}^{(\beta)} - \boldsymbol{u}^{(\alpha)})\cdot\boldsymbol{R}^{(\alpha\beta)} \tag{2.15}$$

For intraspecies collisions $\alpha = \beta$, therefore $\boldsymbol{R}^{(\alpha\alpha)} = 0$ and $Q^{(\alpha\alpha)} = 0$. Taking the zeroth moment

of the Boltzmann equation gives the continuity equation.

$$\int_{-\infty}^{\infty} A^{(\alpha)} \left(\partial_t f^{(\alpha)} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} f^{(\alpha)}) + \nabla_{\boldsymbol{v}} \cdot \left(\frac{\tilde{\boldsymbol{F}}^{(\alpha)}}{A^{(\alpha)}} f^{(\alpha)} \right) \right) d\boldsymbol{v} = \int_{-\infty}^{\infty} A^{(\alpha)} \partial_t f^{(\alpha)}|_{coll} d\boldsymbol{v}$$

$$\partial_t \rho^{(\alpha)} + \nabla \cdot \boldsymbol{p}^{(\alpha)} = 0 \tag{2.16}$$

Taking the first moment of the Boltzmann equation gives the momentum equation.

$$\int_{-\infty}^{\infty} A^{(\alpha)} \boldsymbol{v} \left(\partial_t f^{(\alpha)} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} f^{(\alpha)}) + \nabla_{\boldsymbol{v}} \cdot \left(\frac{\tilde{\boldsymbol{F}}^{(\alpha)}}{A^{(\alpha)}} f^{(\alpha)} \right) \right) d\boldsymbol{v} = \int_{-\infty}^{\infty} A^{(\alpha)} \boldsymbol{v} \partial_t f^{(\alpha)}|_{coll} d\boldsymbol{v}$$

$$\partial_t \boldsymbol{p}^{(\alpha)} + \nabla \cdot \left(\boldsymbol{p}^{(\alpha)} \otimes \boldsymbol{u}^{(\alpha)} + P^{(\alpha)} \overleftrightarrow{\boldsymbol{I}} + \overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)} \right) = \boldsymbol{F}^{(\alpha)} + \sum_{\beta \neq \alpha} \boldsymbol{R}^{(\alpha\beta)} \tag{2.17}$$

$$\boldsymbol{F}^{(\alpha)} = \left(\frac{L}{\partial_p} \right) Z^{(\alpha)} n^{(\alpha)} \left(\boldsymbol{E} + \boldsymbol{u}^{(\alpha)} \times \boldsymbol{B} \right) \tag{2.18}$$

$$\overleftrightarrow{\boldsymbol{P}}^{(\alpha)} = P^{(\alpha)} \overleftrightarrow{\boldsymbol{I}} + \overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)} = \int_{-\infty}^{\infty} A^{(\alpha)} (\boldsymbol{v} - \boldsymbol{u}^{(\alpha)}) \otimes (\boldsymbol{v} - \boldsymbol{u}^{(\alpha)}) f^{(\alpha)} d\boldsymbol{v} \tag{2.19}$$

Taking the reduced second moment of the Boltzmann equation gives the energy equation.

$$\int_{-\infty}^{\infty} \frac{1}{2} A^{(\alpha)} \boldsymbol{v} \cdot \boldsymbol{v} \left(\partial_t f^{(\alpha)} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v} f^{(\alpha)}) + \nabla_{\boldsymbol{v}} \cdot \left(\frac{\tilde{\boldsymbol{F}}^{(\alpha)}}{A^{(\alpha)}} f^{(\alpha)} \right) \right) d\boldsymbol{v} = \int_{-\infty}^{\infty} \frac{1}{2} A^{(\alpha)} \boldsymbol{v} \cdot \boldsymbol{v} \partial_t f^{(\alpha)}|_{coll} d\boldsymbol{v}$$

$$\partial_t e^{(\alpha)} + \nabla \cdot \left((e^{(\alpha)} + P^{(\alpha)}) \boldsymbol{u}^{(\alpha)} + \overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)} \cdot \boldsymbol{u}^{(i)} + \boldsymbol{h}^{(\alpha)} \right) = \boldsymbol{F}^{(\alpha)} \cdot \boldsymbol{u}^{(\alpha)} + \sum_{\beta \neq \alpha} \left(Q^{(\alpha\beta)} + \boldsymbol{R}^{(\alpha\beta)} \cdot \boldsymbol{u}^{(\alpha)} \right) \tag{2.20}$$

$$\boldsymbol{h}^{(\alpha)} = \int_{-\infty}^{\infty} \frac{1}{2} A^{(\alpha)} (\boldsymbol{v} - \boldsymbol{u}^{(\alpha)}) \cdot (\boldsymbol{v} - \boldsymbol{u}^{(\alpha)}) (\boldsymbol{v} - \boldsymbol{u}^{(\alpha)}) f^{(\alpha)} d\boldsymbol{v} \tag{2.21}$$

Note that this energy equation contains the time evolution of the internal fluid energy $U^{(\alpha)}$ as well as the fluid's kinetic energy; that is $e^{(\alpha)} = U^{(\alpha)} + \frac{1}{2} \rho^{(\alpha)} \boldsymbol{u}^{(\alpha)} \cdot \boldsymbol{u}^{(\alpha)}$. It is also possible to write the energy equation in terms of only the internal fluid energy.

$$\begin{aligned}
&\partial_t U^{(\alpha)} + \nabla \cdot \left(U^{(\alpha)} \boldsymbol{u}^{(\alpha)} + \boldsymbol{h}^{(\alpha)} \right) \\
&\quad + \partial_t \left(\frac{1}{2} \rho^{(\alpha)} \boldsymbol{u}^{(\alpha)} \cdot \boldsymbol{u}^{(\alpha)} \right) + \nabla \cdot \left(\left(\frac{1}{2} \rho^{(\alpha)} \boldsymbol{u}^{(\alpha)} \cdot \boldsymbol{u}^{(\alpha)} + P^{(\alpha)} \right) \boldsymbol{u}^{(\alpha)} + \overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)} \cdot \boldsymbol{u}^{(i)} \right) \\
&\quad = \underbrace{\boldsymbol{F}^{(\alpha)} \cdot \boldsymbol{u}^{(\alpha)}} + \sum_{\beta \neq \alpha} \left(Q^{(\alpha\beta)} + \underbrace{\boldsymbol{R}^{(\alpha\beta)} \cdot \boldsymbol{u}^{(\alpha)}} \right)
\end{aligned}$$

Using the identities

$$\partial_t \left(\frac{1}{2}\rho^{(\alpha)}\boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{u}^{(\alpha)}\right) = \underbrace{\boldsymbol{u}^{(\alpha)}\cdot\partial_t\boldsymbol{p}^{(\alpha)}}_{} - \underbrace{\frac{1}{2}\boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{u}^{(\alpha)}\partial_t\rho^{(\alpha)}}_{}$$

$$\nabla\cdot\left(\frac{1}{2}\rho^{(\alpha)}\boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{u}^{(\alpha)}\boldsymbol{u}^{(\alpha)}\right) = \boldsymbol{u}^{(\alpha)}\cdot\nabla\cdot(\boldsymbol{p}^{(\alpha)}\otimes\boldsymbol{u}^{(\alpha)}) - \underbrace{\frac{1}{2}\boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{u}^{(\alpha)}\nabla\cdot\boldsymbol{p}^{(\alpha)}}_{}$$

$$\nabla\cdot(P^{(\alpha)}\boldsymbol{u}^{(\alpha)}) = P^{(\alpha)}\nabla\cdot\boldsymbol{u}^{(\alpha)} + \underbrace{\boldsymbol{u}^{(\alpha)}\cdot\nabla\cdot\left(P^{(\alpha)}\overleftrightarrow{\boldsymbol{I}}\right)}_{}$$

$$\nabla\cdot\left(\overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)}\cdot\boldsymbol{u}^{(\alpha)}\right) = \underbrace{\boldsymbol{u}^{(\alpha)}\cdot\nabla\cdot\overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)}}_{} + \overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)}:\nabla\boldsymbol{u}^{(\alpha)}$$

Notice that the red terms combine to give the continuity equation multiplied by $\frac{1}{2}\boldsymbol{u}^{(\alpha)}\cdot\boldsymbol{u}^{(\alpha)}$, while the blue terms are the dot product of the momentum equation and $\boldsymbol{u}^{(\alpha)}$. These cancel to 0, leaving

$$\partial_t U^{(\alpha)} + \nabla\cdot\left(U^{(\alpha)}\boldsymbol{u}^{(\alpha)} + \boldsymbol{h}^{(\alpha)}\right) + P^{(\alpha)}\nabla\cdot\boldsymbol{u}^{(\alpha)} + \overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)}:\nabla\boldsymbol{u}^{(\alpha)} = \sum_{\beta\neq\alpha} Q^{(\alpha\beta)} \tag{2.22}$$

The fluid equations are coupled with Ampere's and Faraday's equations, written in normalized form as

$$\partial_t\boldsymbol{E} - (\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)^2\nabla\times\boldsymbol{B} = -(\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)\sum_\alpha Z^{(\alpha)}n^{(\alpha)}\boldsymbol{u}^{(\alpha)} \tag{2.23}$$

$$\partial_t\boldsymbol{B} + \nabla\times\boldsymbol{E} = 0 \tag{2.24}$$

respectively.

2.1.1 Divergence cleaning and potential formulations

While Ampere's and Faraday's equations provide a complete description for the evolution of the electric and magnetic fields, they provide no guarantees that Gauss's laws are satisfied for either the electric or magnetic fields. These divergence errors can be generated from interactions with boundary conditions, initial conditions, and the numerical approximation of the solution itself. For example, consider the magnetic field profile of a finite size cylindrical pinch with radius a. The

magnetic azimuthal magnetic field is given by

$$B_\theta = \frac{j_0}{4}\left(\frac{L}{\delta_p}\right)\begin{cases} \frac{r(2a^2-r^2)}{a^2} & r \le a \\ \frac{a^2}{r} & r > a \end{cases} \tag{2.25}$$

where j_0 is the current density at $r = 0$. Note that for $r > a$, $B_\theta \propto r^{-1}$, which cannot be expressed as a finite series of polynomials. Thus any discretization based on a finite sequence of polynomials fundamentally cannot represent this profile, introducing a non-zero divergence. Figure 2.1 shows the divergence error caused by using a central difference approximation imposed onto a Cartesian grid.

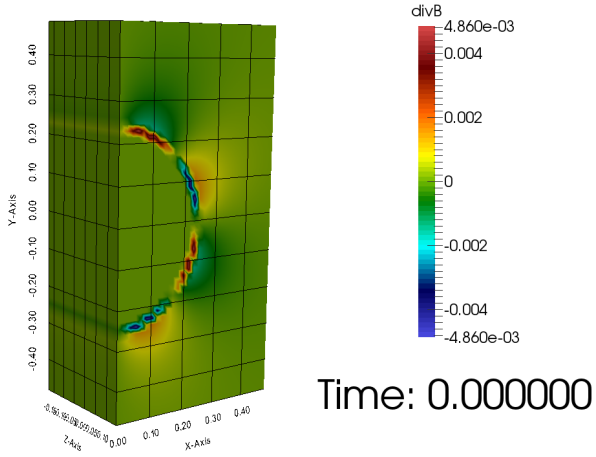


Figure 2.1: $\nabla\cdot\boldsymbol{B}$ error around a cylindrical plasma pinch discretized on a cartesian mesh. The analytical inverse relation of B vs. r around a cylindrical pinch cannot be represented exactly using a finite polynomial series, leading to $\nabla\cdot\boldsymbol{B} \neq 0$ numerically.

The divergence error can be reduced or "cleaned" by introducing auxiliary divergence fields θ and ψ[21]. This modifies Maxwell's equations to be

$$\partial_t \boldsymbol{E} - (\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)^2 \nabla\times\boldsymbol{B} - \nabla\,\theta + (\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)\boldsymbol{j} = 0 \tag{2.26}$$

$$\partial_t \boldsymbol{B} + \nabla\times\boldsymbol{E} - \nabla\,\psi = 0 \tag{2.27}$$

$$\mathcal{F}(\theta) = \nabla\cdot\boldsymbol{E} - (\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)\rho_c \tag{2.28}$$

$$\mathcal{D}(\psi) = \nabla\cdot\boldsymbol{B} \tag{2.29}$$

where $\mathcal{F}$ and $\mathcal{D}$ are linear differential operators. The choice of these operators dictates how divergence errors are reduced. For example, choosing

$$\mathcal{F}(\theta) = \frac{1}{g_h^2}\partial_t\theta + c_p\theta \tag{2.30}$$

will simultaneously hyperbolic and diffuse electric field divergence errors. Choosing

$$\mathcal{F}(\theta) = \frac{1}{c_p}\theta \tag{2.31}$$

results in parabolic cleaning only on the electric field divergence error.

2.1.2 Collisional terms and closure

In order to produce a solvable system, the parameters $\overleftrightarrow{\boldsymbol{\Pi}}^{(\alpha)}$, $P^{(\alpha)}$, $\boldsymbol{h}^{(\alpha)}$, $\boldsymbol{R}^{(\alpha\beta)}$, $\boldsymbol{Q}^{(\alpha\beta)}$, and must be expressed in terms of the lower moment variables $\rho^{(\alpha)}$, $\boldsymbol{p}^{(\alpha)}$, and $e^{(\alpha)}$. The standard procedure for closing the 5N-moment equations is to use the Chapman-Enskog expansion[22]. This is extended by Braginskii[20] to incorporate corrections for magnetized plasmas where these parameters are allowed to become anisotropic with respect to the magnetic field. See the cited literature for the derivation. A summary of their results is presented below with the appropriate normalization. Note that Braginskii assumes a large mass ratio between the two species α and β, a characteristic of a single-species ion/electron plasma. A generalization to multiple ion species for unmagnetized plasmas has been derived by Simakov and Molvig [23], which could with some effort be extended to magnetized plasmas. This generalization is not explored in this research.

$$P^{(\alpha)} = (\gamma^{(\alpha)} - 1)U^{(\alpha)} \tag{2.32}$$

$$\tau^{(\alpha\beta)} = \frac{1}{\nu^{(\alpha\beta)}} = \frac{3\pi^{\frac{3}{2}} \left(\frac{2T^{(\alpha)}}{A^{(\alpha)}} + \frac{2T^{(\beta)}}{A^{(\beta)}}\right)^{\frac{3}{2}}}{\left(1 + \frac{A^{(\alpha)}}{A^{(\beta)}}\right) n^{(\beta)}} \left(\frac{A^{(\alpha)}}{Z^{(\alpha)} Z^{(\beta)}}\right)^2 \tag{2.33}$$

$$\overset{\leftrightarrow}{\boldsymbol{\Pi}}^{(\alpha)} = \frac{P^{(\alpha)}}{(\nu_p \tau)\nu^{(\alpha\alpha)}} \sum_{i=0}^{4} \eta_i^{(\alpha)} \overset{\leftrightarrow}{\boldsymbol{W}}_i^{(\alpha)} \tag{2.34}$$

$$\overset{\leftrightarrow}{\boldsymbol{W}}^{(\alpha)} = \nabla \boldsymbol{u}^{(\alpha)} + (\nabla \boldsymbol{u}^{(\alpha)})^T - \frac{2}{3}(\nabla \cdot \boldsymbol{u}^{(\alpha)}) \overset{\leftrightarrow}{\boldsymbol{I}} \tag{2.35}$$

$$\overset{\leftrightarrow}{\boldsymbol{\delta}}^{\perp} = \overset{\leftrightarrow}{\boldsymbol{I}} - \widehat{\boldsymbol{B}} \otimes \widehat{\boldsymbol{B}} \tag{2.36}$$

$$\overset{\leftrightarrow}{\boldsymbol{W}}_{0,ij}^{(\alpha)} = -\frac{3}{2}\left(\widehat{\boldsymbol{B}}_i \widehat{\boldsymbol{B}}_j - \frac{1}{3} \overset{\leftrightarrow}{\boldsymbol{I}}_{ij}\right)\left(\widehat{\boldsymbol{B}}_k \widehat{\boldsymbol{B}}_l - \frac{1}{3} \overset{\leftrightarrow}{\boldsymbol{I}}_{kl}\right) \overset{\leftrightarrow}{\boldsymbol{W}}_{kl} \tag{2.37}$$

$$\overset{\leftrightarrow}{\boldsymbol{W}}_{1,ij}^{(\alpha)} = -\left(\overset{\leftrightarrow}{\boldsymbol{\delta}}_{ik}^{\perp} \overset{\leftrightarrow}{\boldsymbol{\delta}}_{jl}^{\perp} + \frac{1}{2} \overset{\leftrightarrow}{\boldsymbol{\delta}}_{ij}^{\perp} \widehat{\boldsymbol{B}}_k \widehat{\boldsymbol{B}}_l\right) \overset{\leftrightarrow}{\boldsymbol{W}}_{kl} \tag{2.38}$$

$$\overset{\leftrightarrow}{\boldsymbol{W}}_{2,ij}^{(\alpha)} = -\left(\overset{\leftrightarrow}{\boldsymbol{\delta}}_{ik}^{\perp} \widehat{\boldsymbol{B}}_j \widehat{\boldsymbol{B}}_l + \overset{\leftrightarrow}{\boldsymbol{\delta}}_{jl}^{\perp} \widehat{\boldsymbol{B}}_i \widehat{\boldsymbol{B}}_k\right) \overset{\leftrightarrow}{\boldsymbol{W}}_{kl} \tag{2.39}$$

$$\overset{\leftrightarrow}{\boldsymbol{W}}_{3}^{(\alpha)} = \frac{1}{2}\left(\overset{\leftrightarrow}{\boldsymbol{\delta}}_{ik}^{\perp} \epsilon_{jml} + \overset{\leftrightarrow}{\boldsymbol{\delta}}_{jl}^{\perp} \epsilon_{imk}\right) \widehat{\boldsymbol{B}}_m \overset{\leftrightarrow}{\boldsymbol{W}}_{kl} \tag{2.40}$$

$$\overset{\leftrightarrow}{\boldsymbol{W}}_{4}^{(\alpha)} = \left(\widehat{\boldsymbol{B}}_i \widehat{\boldsymbol{B}}_k \epsilon_{jml} + \widehat{\boldsymbol{B}}_j \widehat{\boldsymbol{B}}_l \epsilon_{imk}\right) \widehat{\boldsymbol{B}}_m \overset{\leftrightarrow}{\boldsymbol{W}}_{kl} \tag{2.41}$$

$$\eta_0^{(\alpha)} = \frac{g_{17}^{(\alpha)}}{g_{15}^{(\alpha)}} \tag{2.42}$$

$$\eta_1^{(\alpha)} = \frac{g_{16}^{(\alpha)} \left(\frac{(\omega_c \tau)}{(\nu_p \tau)}\right)^2 (2\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^2 + g_{17}^{(\alpha)}}{\Delta_3^{(\alpha)}} \tag{2.43}$$

$$\eta_2^{(\alpha)} = \frac{g_{16}^{(\alpha)} \left(\frac{(\omega_c \tau)}{(\nu_p \tau)}\right)^2 (\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^2 + g_{17}^{(\alpha)}}{\Delta_2^{(\alpha)}} \tag{2.44}$$

$$\eta_3^{(\alpha)} = \frac{\frac{(\omega_c \tau)}{(\nu_p \tau)}(2\omega_c^{(\alpha)} \tau^{(\alpha\alpha)}) \left(g_{18}^{(\alpha)} \left(\frac{(\omega_c \tau)}{(\nu_p \tau)}\right)^2 (2\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^2 + g_{19}^{(\alpha)}\right)}{\Delta_3^{(\alpha)}} \tag{2.45}$$

$$\eta_4^{(\alpha)} = \frac{\frac{(\omega_c \tau)}{(\nu_p \tau)}(\omega_c^{(\alpha)} \tau^{(\alpha\alpha)}) \left(g_{18}^{(\alpha)} \left(\frac{(\omega_c \tau)}{(\nu_p \tau)}\right)^2 (\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^2 + g_{19}^{(\alpha)}\right)}{\Delta_2^{(\alpha)}} \tag{2.46}$$

$$\Delta_3^{(\alpha)} = \left(\frac{(\omega_c \tau)}{(\nu_p \tau)}\right)^4 (2\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^4 + g_{14}^{(\alpha)} (2\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^2 + g_{15}^{(\alpha)} \tag{2.47}$$

$$\Delta_2^{(\alpha)} = \left(\frac{(\omega_c \tau)}{(\nu_p \tau)}\right)^4 (\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^4 + g_{14}^{(\alpha)} (\omega_c^{(\alpha)} \tau^{(\alpha\alpha)})^2 + g_{15}^{(\alpha)} \tag{2.48}$$

The Braginskii heat flux and interspecies momentum transfer are defined with respect to a magnetic field coordinate system. These are defined with respect to a coordinate vector triplet

$$\boldsymbol{x}_{\parallel} = (\boldsymbol{x} \cdot \widehat{\boldsymbol{B}})\widehat{\boldsymbol{B}} + (1 - \|\widehat{\boldsymbol{B}}\|)\boldsymbol{x} \tag{2.49}$$

$$\boldsymbol{x}_{\perp} = \widehat{\boldsymbol{B}} \times (\boldsymbol{x} \times \widehat{\boldsymbol{B}}) \tag{2.50}$$

$$\boldsymbol{x}_{\wedge} = \widehat{\boldsymbol{B}} \times \boldsymbol{x} \tag{2.51}$$

Note that this definition collapses to $\boldsymbol{x}_{\parallel} = \boldsymbol{x}$ in the isotropic unmagnetized case. The heat flux and interspecies momentum and energy transfer terms are then defined as

$$\boldsymbol{h}^{(\alpha)} = \boldsymbol{h}_T^{(\alpha)} + \boldsymbol{h}_u^{(\alpha e)} \tag{2.52}$$

$$\boldsymbol{h}_T^{(\alpha)} = -\frac{P^{(\alpha)}\tau^{(\alpha\alpha)}}{A^{(\alpha)}(\nu_p\tau)}\left(C_{3,\parallel}^{(\alpha)}\,\nabla_{\parallel}\,T^{(\alpha)} + C_{3,\perp}^{(\alpha)}\,\nabla_{\perp}\,T^{(\alpha)} + C_{3,\wedge}^{(\alpha)}\,\nabla_{\wedge}\,T^{(\alpha)}\right) \tag{2.53}$$

$$\boldsymbol{h}_u^{(\alpha e)} = P^{(e)}\left(C_{2,\parallel}^{\alpha e}\boldsymbol{u}_{\parallel}^{(\alpha e)} + C_{2,\perp}^{\alpha e}\boldsymbol{u}_{\perp}^{(\alpha e)} + C_{2,\wedge}^{\alpha e}\boldsymbol{u}_{\wedge}^{(\alpha e)}\right) \tag{2.54}$$

$$\boldsymbol{R}^{(\alpha\beta)} = \boldsymbol{R}_u^{(\alpha\beta)} + \boldsymbol{R}_T^{(\alpha e)} \tag{2.55}$$

$$\boldsymbol{R}_u^{(\alpha\beta)} = \frac{\rho^{(e)}(\nu_p\tau)}{\tau^{(e\alpha)}}\left(C_{1,\parallel}\boldsymbol{u}_{\parallel}^{(\beta\alpha)} + C_{1,\perp}\boldsymbol{u}_{\perp}^{(\beta\alpha)} + C_{1,\wedge}\boldsymbol{u}_{\wedge}^{(\beta\alpha)}\right) \tag{2.56}$$

$$\boldsymbol{R}_T^{(\alpha e)} = n^{(e)}\left(C_{2,\parallel}^{(\alpha e)}\,\nabla_{\parallel}\,T^{(e)} + C_{2,\perp}^{(\alpha e)}\,\nabla_{\perp}\,T^{(e)} + C_{2,\wedge}^{(\alpha e)}\,\nabla_{\wedge}\,T^{(e)}\right) \tag{2.57}$$

$$Q^{(\alpha\beta)} = \frac{3\rho^{(\alpha)}(\nu_p\tau)(T^{(\beta)} - T^{(\alpha)})}{\tau^{(\alpha\beta)}(A^{(\alpha)} + A^{(\beta)})} + \frac{A^{(\beta)}}{A^{(\alpha)} + A^{(\beta)}}\boldsymbol{R}^{(\alpha\beta)} \cdot \boldsymbol{u}^{(\beta\alpha)} \tag{2.58}$$

$$\boldsymbol{u}^{(\alpha\beta)} = \boldsymbol{u}^{(\alpha)} - \boldsymbol{u}^{(\beta)} \tag{2.59}$$

where the coefficients are defined as

$$C_{1,\parallel}^{(\alpha e)} = 1 - \frac{g_3}{g_0^{(e)}} \tag{2.60}$$

$$C_{1,\perp}^{(\alpha e)} = 1 - \frac{g_2\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2(\omega_c^{(e)}\tau^{(e\alpha)})^2 + g_3}{\Delta_1^{(e)}} \tag{2.61}$$

$$C_{1,\wedge}^{(\alpha e)} = \frac{\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)(\omega_c^{(e)}\tau^{(e\alpha)})\left(g_4\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2(\omega_c^{(e)}\tau^{(e\alpha)})^2 + g_5\right)}{\Delta_1^{(e)}} \tag{2.62}$$

$$C_{2,\parallel}^{(\alpha e)} = \frac{g_6}{g_0^{(e)}} \tag{2.63}$$

$$C_{2,\perp}^{(\alpha e)} = \frac{g_6 \left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2 (\omega_c^{(e)}\tau^{(e\alpha)})^2 + g_7}{\Delta_1^{(e)}} \tag{2.64}$$

$$C_{2,\wedge}^{(\alpha e)} = \frac{\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)(\omega_c^{(e)}\tau^{(e\alpha)})\left(g_8\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2 (\omega_c^{(e)}\tau^{(e\alpha)})^2 + g_9\right)}{\Delta_1^{(e)}} \tag{2.65}$$

$$C_{3,\parallel}^{(\alpha)} = \frac{g_{11}^{(\alpha)}}{g_0^{(\alpha)}} \tag{2.66}$$

$$C_{3,\perp}^{(\alpha)} = \frac{g_{10}^{(\alpha)} \left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2 (\omega_c^{(\alpha)}\tau^{(\alpha\alpha)})^2 + g_{11}^{(\alpha)}}{\Delta_1^{(\alpha)}} \tag{2.67}$$

$$C_{3,\wedge}^{(\alpha)} = \frac{\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)(\omega_c^{(\alpha)}\tau^{(\alpha\alpha)})\left(g_{12}^{(\alpha)}\left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2 (\omega_c^{(\alpha)}\tau^{(\alpha\alpha)})^2 + g_{13}^{(\alpha)}\right)}{\Delta_1^{(\alpha)}} \tag{2.68}$$

$$\Delta_1^{(\alpha)} = \left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^4 (\omega_c^{(\alpha)}\tau^{(\alpha\alpha)})^4 + \left(\frac{(\omega_c\tau)}{(\nu_p\tau)}\right)^2 g_1^{(\alpha)}(\omega_c^{(\alpha)}\tau^{(\alpha\alpha)})^2 + g_0^{(\alpha)} \tag{2.69}$$

The various constants required are specified in Appendix A.

2.2 Reduction of two-fluid to magnetohydrodynamics (MHD)

Define the bulk fluid properties

$$\rho = \rho^{(i)} + \rho^{(e)}, \qquad p_i = p_i^{(i)} + p_i^{(e)}, \qquad u_i = \frac{p_i}{\rho}$$

$$P = P^{(i)} + P^{(e)}, \qquad \Pi_{ij} = \Pi_{ij}^{(i)} + \Pi_{ij}^{(e)} \qquad P_{ij} = P\delta_{ij} + \Pi_{ij}$$

$$\rho_c = Z^{(i)}n^{(i)} + Z^{(e)}n^{(e)}, \qquad j_i = \frac{Z^{(i)}}{A^{(i)}}p_i^{(i)} + \frac{Z^{(e)}}{A^{(e)}}p_i^{(e)} \qquad e = e^{(i)} + e^{(e)} + \frac{1}{2}\left(B_i^2 + \frac{1}{(\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)^2}E_i^2\right)$$

$$h_i = h_i^{(i)} + h_i^{(e)}$$

Adding the two-fluid continuity equations (2.16) together gives the center of mass (COM) continuity equation.

$$\partial_t\rho + \partial_i p_i = 0 \tag{2.70}$$

Adding the two-fluid momentum equations (2.17) together gives the COM momentum equation.

$$\partial_t p_i + \partial_j \left(p_i^{(i)} u_j^{(i)} + p_i^{(e)} u_j^{(e)} + P_{ij} \right) = \left(\frac{L}{\delta_p} \right) \left(\rho_c E_i + j_k B_j \epsilon_{ijk} \right) \tag{2.71}$$

Applying the standard MHD approximations $A^{(e)} \ll A^{(i)}$ and $\omega_p \tau \to \infty$ (resulting in $\rho_c = 0$) and substituting in from the low frequency Ampere's law $j_i = \frac{\delta_p}{L} \partial_k B_j \epsilon_{ijk}$,

$$\partial_t p_i + \partial_j \left(p_i u_j + \left(P + \frac{1}{2} B_k^2 \right) \delta_{ij} + \Pi_{ij} - B_i B_j \right) + B_i \partial_j B_j = 0 \tag{2.72}$$

Multiplying the ion momentum equation by $\frac{Z^{(i)}}{A^{(i)}}$, the electron momentum equation by $\frac{Z^{(e)}}{A^{(e)}}$, adding these two together, and solving for E_i,

$$E_i = \left(\frac{\delta_p}{L} \right) \frac{\partial_t j_i + \partial_j \left(\frac{Z^{(i)}}{A^{(i)}} p_i^{(i)} u_j^{(i)} + \frac{Z^{(e)}}{A^{(e)}} p_i^{(e)} u_j^{(e)} + \frac{Z^{(i)}}{A^{(i)}} P_{ij}^{(i)} + \frac{Z^{(e)}}{A^{(e)}} P_{ij}^{(e)} \right) + \left(\frac{Z^{(i)}}{A^{(i)}} - \frac{Z^{(e)}}{A^{(e)}} \right) R_i^{(ie)} - G_i}{\left(\frac{(Z^{(i)})^2}{A^{(i)}} n^{(i)} + \frac{(Z^{(e)})^2}{A^{(e)}} n^{(e)} \right)} \tag{2.73}$$

$$G_i = \left(\frac{L}{\delta_p} \right) \left(\left(\frac{Z^{(i)}}{A^{(i)}} + \frac{Z^{(e)}}{A^{(e)}} \right) j_k - \left(\frac{Z^{(i)} Z^{(e)} n^{(e)}}{A^{(i)}} + \frac{Z^{(i)} Z^{(e)} n^{(i)}}{A^{(e)}} \right) u_k \right) B_j \epsilon_{ijk} \tag{2.74}$$

Applying the standard MHD approximations gives generalized Ohm's law.

$$E_i = \frac{1}{Z^{(e)} n^{(e)}} \left(\frac{\delta_p}{L} \right) \left(\partial_j \left(P_{ij}^{(e)} \right) - R_i^{(ie)} - (\partial_m B_l \epsilon_{klm}) B_j \epsilon_{ijk} \right) - u_k B_j \epsilon_{ijk} \tag{2.75}$$

To arrive at a conservation form total energy equation, the ion and electron energy equations are combined with the electromagnetic field energy. The electromagnetic field energy can be found by taking the dot product of Faraday's law with $\boldsymbol{B}$, as well as taking the dot product of $\boldsymbol{E}$ and Ampere's law. This gives

$$B_i \partial_t B_i + B_i \partial_k E_j \epsilon_{ijk} = 0 \tag{2.76}$$

$$(\omega_p \tau)^{-2} \left(\frac{L}{\delta_p} \right)^2 E_i \partial_t E_i - E_i \partial_k B_j \epsilon_{ijk} = - \left(\frac{L}{\delta_p} \right) E_i j_i \tag{2.77}$$

respectively. Adding these together and applying vector identities gives

$$\frac{1}{2} \partial_t \left(B_i^2 + (\omega_p \tau)^{-2} \left(\frac{L}{\delta_p} \right)^2 E_i^2 \right) + \partial_i \left(E_k B_j \epsilon_{ijk} \right) = - \left(\frac{L}{\delta_p} \right) E_i j_i \tag{2.78}$$

This can be combined with the bulk fluid energy to give the conservation of total system energy.

$$\partial_t e + \partial_i \left(\left(e^{(i)} + P^{(i)} \right) u_i^{(i)} + \left(e^{(e)} + P^{(e)} \right) u_i^{(e)} + \Pi_{ij}^{(i)} u_j^{(i)} + h_i + \Pi_{ij}^{(e)} u_j^{(e)} + E_k B_j \epsilon_{ijk} \right) = 0 \quad (2.79)$$

Applying the MHD assumptions $A^{(e)} \ll A^{(i)}$ and $\omega_p \tau \to \infty$,

$$\partial_t e + \partial_i \left(\left(e^{(i)} + e^{(e)} + P \right) u_i + \Pi_{ij} u_j + h_i + E_k B_j \epsilon_{ijk} \right) = 0 \quad (2.80)$$

For a method which keeps track of both fluid temperatures, the individual species energy equations Eq. (2.20) from the multifluid model are used as-is. In ideal MHD, the generalized Ohm's Law reduces to $E_i \approx -u_k B_j \epsilon_{ijk}$. Substituting this in,

$$\partial_t e + \partial_i \left(\left(e + P + \frac{1}{2} B_j^2 - (\omega_p \tau)^{-2} \left(\frac{L}{\delta_p} \right)^2 E_i^2 \right) u_i - u_j B_j B_i \right) \approx 0 \quad (2.81)$$

However, in the asymptotic limit, the electric field energy is assumed to be negligible compared to fluid and magnetic field energies since it is proportional to $\left(\frac{v_A}{c_0} \right)^2 \to 0$, leaving the standard ideal MHD energy equation. This derivation approach provides some advantages for coupling MHD equation systems to multi-fluid models, which will be explored in Chapter 6. Notably, the energy equation's numerical flux can be composed of the flux of ion energy, electron energy, and the Poynting vector for magnetic field energy.

$$(\boldsymbol{q} \cdot \widehat{n})^*_{MHD} = (\boldsymbol{q}^{(i)} \cdot \widehat{n})^* + (\boldsymbol{q}^{(e)} \cdot \widehat{n})^* + (\boldsymbol{S} \cdot \widehat{n})^* \quad (2.82)$$

If MHD energy is instead defined as $e = e_i + e_e + \frac{B^2}{2}$ (neglecting electric field energy), then the MHD flux must be defined in terms of magnetic field energy flux, which cannot be written in conservation form since it must evaluate to $B_i \partial_k E_j \epsilon_{ijk}$. Handling of non-conservation form terms can be done and is necessary in order to implement some Braginskii terms, however it is significantly more difficult to do so. See Section 3.1 for more details on the numerical discretization required.

Chapter 3

Spatial discretization

The equation systems presented in Ch. 2 are nonlinear partial differential equations built from the summation of a few primitive differential operators. These can be classified into one of the forms:

$$\nabla\cdot\left(\overset{\leftrightarrow}{\boldsymbol{F}}\left(\boldsymbol{q},\boldsymbol{r},t\right)\right)=\frac{\partial\,\overset{\leftrightarrow}{\boldsymbol{F}}_{ij}}{\partial\boldsymbol{r}_j} \tag{3.1}$$

$$\overset{\leftrightarrow}{\boldsymbol{A}}\left(\boldsymbol{q},\boldsymbol{r},t\right)\cdot\nabla\,\boldsymbol{q}=\overset{\leftrightarrow}{\boldsymbol{A}}_{ijk}\,\frac{\partial\boldsymbol{q}_k}{\partial\boldsymbol{r}_j} \tag{3.2}$$

$$\nabla\cdot\left(\overset{\leftrightarrow}{\boldsymbol{D}}\left(\boldsymbol{q},\boldsymbol{r},t\right)\cdot\nabla\,\boldsymbol{G}(\boldsymbol{q},\boldsymbol{r},t)\right)=\frac{\partial}{\partial\boldsymbol{r}_j}\left(\overset{\leftrightarrow}{\boldsymbol{D}}_{ijkl}\,\frac{\partial\boldsymbol{G}_k}{\partial\boldsymbol{r}_l}\right) \tag{3.3}$$

$$\boldsymbol{S}(\boldsymbol{q},\boldsymbol{r},t)=\boldsymbol{S}_i \tag{3.4}$$

A total PDE system would contain a summation of one or more of these terms. where the left-hand side shows the vector notation, and the right-hand side shows the index notation. For clarity, the index notation will be used for the remainder of this chapter. Note that $\overset{\leftrightarrow}{\boldsymbol{F}}$, $\overset{\leftrightarrow}{\boldsymbol{A}}$, $\overset{\leftrightarrow}{\boldsymbol{D}}$, $\boldsymbol{G}$, and $\boldsymbol{S}$ are allowed to contain temporal derivatives of $\boldsymbol{q}$, but not spatial derivatives. In general, there is no known analytical solution to these systems for all problems of interest, especially for complex geometries. In order to solve these problems, the equations must be discretized appropriately. One method for discretizing these equations is by taking a method of lines approach[24]: the spatial derivatives are first appropriately discretized, giving a semi-discrete differential-algebraic

equation (DAE) or ordinary differential equation (ODE) system. The temporal derivatives are then discretized separately using classical DAE/ODE solver techniques. This approach has the advantage that it separates the problem of handling spatial discretization with boundary conditions from the temporal initial value problem.

The spatial discretization method used for this research is a mixture of various Discontinuous Galerkin (DG) methods[25]. Notably, the traditional DG method is used to handle terms of the form given by Eqs. (3.1) and (3.2), while terms of the form given by Eq. (3.3) are to be treated using Hybridizable Discontinuous Galerkin (HDG)[26]. Note that it is possible to handle terms of the form (3.3) using traditional DG methods, but HDG has some key advantages discussed in Section 3.2. Source terms of the form (3.4) are handled identically using standard DG or HDG.

3.1 Discontinuous Galerkin

3.1.1 First order derivatives

Consider the PDE

$$\boldsymbol{R}_i + \frac{\partial \overset{\leftrightarrow}{\boldsymbol{F}}_{ij}}{\partial \boldsymbol{r}_j} + \overset{\leftrightarrow}{\boldsymbol{A}}_{ijk} \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{r}_j} = 0 \tag{3.5}$$

where $\boldsymbol{R}_i$ contains all terms not being considered and $\boldsymbol{q}$ is allowed to admit discontinuities. Consider a rotation normal to a discontinuity at $\boldsymbol{r}_d$ such that x_d is located at $\boldsymbol{r}_d$ and $(x_d - \epsilon)\widehat{n}^- = \boldsymbol{r}_d - \epsilon\widehat{n}^-$ where $\widehat{n}^-$ is the normal of the discontinuity.

Maso et al.[27] introduced the concept of a Lipschitz continuous path $\boldsymbol{\xi}$ such that in the neigh-

borhood of a discontinuity x_d such that

$$\boldsymbol{q}(x) = \begin{cases} \boldsymbol{q}^- & x \in (x^-, x_d - \epsilon) \\ \boldsymbol{\xi}\left(\frac{x - x_d + \epsilon}{2\epsilon}\right) & x \in (x_d - \epsilon, x_d + \epsilon) \\ \boldsymbol{q}^+ & x \in (x_d + \epsilon, x^+) \end{cases} \tag{3.6}$$

where $\epsilon > 0$ defines the neighborhood. The function $\boldsymbol{\xi}$ must satisfy the following properties[28]:

1. $\boldsymbol{\xi}(0, \boldsymbol{q}^-, \boldsymbol{q}^+) = \boldsymbol{q}^-$
2. $\boldsymbol{\xi}(1, \boldsymbol{q}^-, \boldsymbol{q}^+) = \boldsymbol{q}^+$
3. $\boldsymbol{\xi}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^-) = \boldsymbol{q}^-$
4. $\left|\frac{\partial \boldsymbol{\xi}}{\partial \tau}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^+)\right| \leq K|\boldsymbol{q}^- - \boldsymbol{q}^+|$, a.e. in $[0, 1]$
5. $\boldsymbol{\xi}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^+) = \boldsymbol{\xi}(1 - \tau, \boldsymbol{q}^+, \boldsymbol{q}^-)$

where property (5) was introduced by Rhebergen in the multi-dimensional case. Define the finite element space on a domain Ω discretized into a mesh $\mathcal{T}$ as

$$\boldsymbol{W} = \left\{\boldsymbol{w} \in (L^2(\Omega))^\alpha : \boldsymbol{w}|_K \in (\mathcal{P}_m(K))^\alpha, \forall K \in \mathcal{T}\right\} \tag{3.7}$$

where α is the number of components of $\boldsymbol{w}$ and $\mathcal{P}_m(K)$ defines a polynomial basis set of order m restricted to the element K. Define the notation x^- and x^+ to denote the value of x on the inside and outside of the face respectively, $\{\{x\}\} = \frac{x^- + x^+}{2}$ as the average of x across a discontinuity, and $[\![x]\!] = x^+ + x^-$ to be the jump of x across the discontinuity.

Integrating Eq. (3.5) over the test functions $\boldsymbol{w} \in \boldsymbol{W}$ and applying the DLM theorem[27] gives

$$\int \boldsymbol{w}_i \left(\boldsymbol{R}_i + \overleftrightarrow{\boldsymbol{B}}_{ijk} \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{r}_j}\right) dV + \oint \boldsymbol{w}_i^* \widehat{n}_j^- \left(\int_0^1 \overleftrightarrow{\boldsymbol{B}}_{ijk} \left(\boldsymbol{\xi}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^+)\right) \frac{\partial \boldsymbol{\xi}_k}{\partial \tau}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^+) d\tau\right) dS = 0, \quad \forall \boldsymbol{w} \in \boldsymbol{W} \tag{3.8}$$

$$B_{ijk} = \frac{\partial \overleftrightarrow{\boldsymbol{F}}_{ij}}{\partial \boldsymbol{q}_k} + \overleftrightarrow{\boldsymbol{A}}_{ijk} \tag{3.9}$$

where $\boldsymbol{w}^*$ is the test function numerical flux. Since $\overleftrightarrow{\boldsymbol{F}}_{ij}$ is in conservation form, the Lipschitz path chosen for $\boldsymbol{\xi}$ doesn't matter. Similarly, if $\overleftrightarrow{\boldsymbol{A}}_{ijk}$ could have been written in conservation form the same property applies, and a modified operator $\overleftrightarrow{\boldsymbol{F}}'$ could be defined where $\overleftrightarrow{\boldsymbol{A}}_{ijk} = \frac{\partial \overleftrightarrow{\boldsymbol{F}}'_{ij}}{\partial \boldsymbol{q}_k}$ and this could be combined into a single conservation form operator with $\overleftrightarrow{\boldsymbol{F}}$. However, for the purposes of this work assume that such an operator $\overleftrightarrow{\boldsymbol{F}}'$ does not exist. It is still possible to separate $\overleftrightarrow{\boldsymbol{B}}$ into two surface integrals: a path-independent conservative numerical flux and a path-dependent, non-conservative product numerical flux. That is,

$$\oint \boldsymbol{w}_i^* \widehat{n}_j^- \left(\int_0^1 \overleftrightarrow{\boldsymbol{B}}_{ijk} \frac{\partial \boldsymbol{\xi}_k}{\partial \tau} d\tau \right) dS = \oint \boldsymbol{w}_i^* \widehat{n}_j^- \left(\int_0^1 \overleftrightarrow{\boldsymbol{A}}_{ijk} \frac{\partial \boldsymbol{\xi}_k}{\partial \tau} d\tau \right) dS - \oint \boldsymbol{w}_i^* [\![\overleftrightarrow{\boldsymbol{F}}_{ij} \, \widehat{n}_j]\!] dS \tag{3.10}$$

Choosing $\boldsymbol{w}^* = \{\!\{\boldsymbol{w}\}\!\}$, the DG weak formulation can be written as find $\boldsymbol{q} \in \boldsymbol{W}$ such that[28, Theorem A.1]

$$\begin{aligned} \int \boldsymbol{w}_i \left(\overleftrightarrow{\boldsymbol{R}}_i + \overleftrightarrow{\boldsymbol{A}}_{ijk} \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{r}_j} \right) - \overleftrightarrow{\boldsymbol{F}}_{ij} \frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{r}_j} dV + \oint \left(\boldsymbol{w}_i^- - \boldsymbol{w}_i^+ \right) (\overleftrightarrow{\boldsymbol{H}}_{ij} \, \widehat{n}_j)^* dS \\ + \oint \{\!\{ \boldsymbol{w}_i \}\!\} \widehat{n}_j^- \left(\int_0^1 \overleftrightarrow{\boldsymbol{A}}_{ijk} \frac{\partial \boldsymbol{\xi}_k}{\partial \tau} d\tau \right) dS = 0, \qquad \forall \boldsymbol{w} \in \boldsymbol{W} \end{aligned} \tag{3.11}$$

where $(\overleftrightarrow{\boldsymbol{H}}_{ij} \, \widehat{n}_j)^*$ is the non-conservative products numerical flux (NCP flux). Note that this numerical flux is influenced by both the conservation form $\overleftrightarrow{\boldsymbol{F}}$ and non-conservation form $\overleftrightarrow{\boldsymbol{A}}$ operators.

3.1.2 Hyperbolic numerical fluxes

A consistent numerical flux should approximately satisfy the generalized Rankine-Hugoniot conditions[29, 30]

$$-\lambda_m \left(\boldsymbol{q}_i^+ - \boldsymbol{q}_i^- \right) - \left(\overleftrightarrow{\boldsymbol{F}}_{ij}^- \widehat{n}_j^- + \overleftrightarrow{\boldsymbol{F}}_{ij}^+ \widehat{n}_j^+ \right) + \widehat{n}_j^- \int_0^1 \overleftrightarrow{\boldsymbol{A}}_{ijk} \frac{\partial \xi_k}{\partial \tau} d\tau = 0, \quad \forall m \tag{3.12}$$

where λ_m are the characteristics describing the propagation of the discontinuity. For linear systems, these can be found by using finding the eigenvalues of the flux Jacobian

$$\overleftrightarrow{\boldsymbol{J}}_{ijk} = \frac{\partial \overleftrightarrow{\boldsymbol{F}}_{ij}}{\partial \boldsymbol{q}_k} + \overleftrightarrow{\boldsymbol{A}}_{ijk} \tag{3.13}$$

This situation is more complicated for nonlinear systems, leading to the development of a wide variety of methods such as Roe averages[31] and locally Lax-Friedrich/Rusanov methods[32]. Two fluxes designed to work for the NCP flux are the HLL-based flux derived by Rhebergen[28], and then a further simplification of this flux to a locally Lax-Friedrich/Rusanov flux.

For an HLL-based flux, consider only the minimum and maximum eigenvalues $\lambda_m \in \{\lambda^-, \lambda^+\}$ where $\lambda^- < \lambda^+$. The wave propagation diagram shown in Fig. 3.1 shows the various regions where the solution is influenced by the various states. In Ω_1, $\boldsymbol{q}$ is entirely the left state. In Ω_4, $\boldsymbol{q}$ is entirely the right state. In $\Omega_2 \cup \Omega_3$, both the left and right states influence the solution.

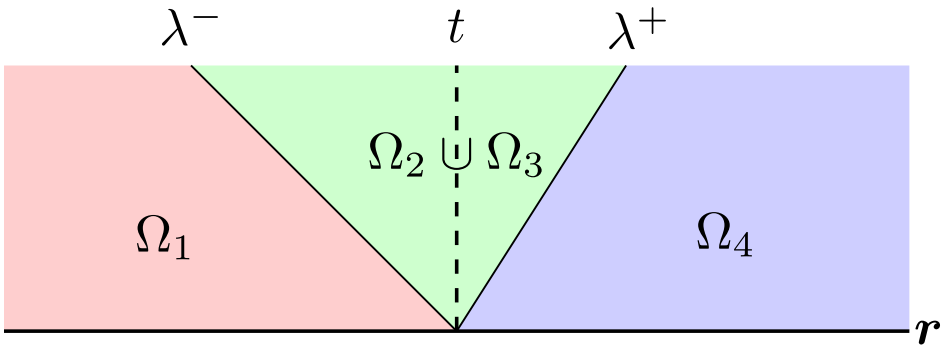


Figure 3.1: Two wave propagation diagram on a static mesh.

Mathematically, this can be stated as in the neighborhood of a jump discontinuity $\boldsymbol{r}_d$,

$$\boldsymbol{q}^* = \begin{cases} \boldsymbol{q}^- & \lambda^- \geq 0 \\ \bar{\boldsymbol{q}}^* & \lambda^- < 0 < \lambda^+ \\ \boldsymbol{q}^+ & \lambda^+ \leq 0 \end{cases} \tag{3.14}$$

where $\bar{\boldsymbol{q}}^*$ is derived in[28] to be

$$\bar{\boldsymbol{q}}_i^* = \frac{\lambda^+ \boldsymbol{q}_i^+ - \lambda^- \boldsymbol{q}_i^- + [\![\overleftrightarrow{\boldsymbol{F}}_{ij}\, \widehat{n}_j]\!]}{\lambda^+ - \lambda^-} - \frac{\widehat{n}_j^-}{\lambda^+ - \lambda^-} \int_0^1 \overleftrightarrow{\boldsymbol{A}}_{ijk} \left(\boldsymbol{\xi}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^+)\right) \frac{\partial \boldsymbol{\xi}_k}{\partial \tau}(\tau, \boldsymbol{q}^-, \boldsymbol{q}^+) d\tau \tag{3.15}$$

The NCP flux $(\overleftrightarrow{\boldsymbol{H}}_{ij}\,\widehat{n}_j)^*$ is then defined as

$$(\overleftrightarrow{\boldsymbol{H}}_{ij}\,\widehat{n}_j)^*_{\text{HLL}} = \begin{cases} \overleftrightarrow{\boldsymbol{F}}^-_{ij}\,\widehat{n}^-_j - \frac{1}{2}\widehat{n}^-_j \int_0^1 \overleftrightarrow{\boldsymbol{A}}_{ijk}\,(\boldsymbol{\xi}(\tau,\boldsymbol{q}^-,\boldsymbol{q}^+))\frac{\partial \boldsymbol{\xi}_k}{\partial \tau}(\tau,\boldsymbol{q}^-,\boldsymbol{q}^+)d\tau & \lambda^- > 0 \\ \left\{\!\!\left\{ \overleftrightarrow{\boldsymbol{F}}_{ij} \right\}\!\!\right\} \widehat{n}^-_j + \frac{1}{2}\left(\lambda^+(\bar{\boldsymbol{q}}^* - \boldsymbol{q}^+) + \lambda^-(\bar{\boldsymbol{q}}^* - \boldsymbol{q}^-)\right) & \lambda^- < 0 < \lambda^+ \\ \overleftrightarrow{\boldsymbol{F}}^+_{ij}\,\widehat{n}^- + \frac{1}{2}\widehat{n}^-_j \int_0^1 \overleftrightarrow{\boldsymbol{A}}_{ijk}\,(\boldsymbol{\xi}(\tau,\boldsymbol{q}^-,\boldsymbol{q}^+))\frac{\partial \boldsymbol{\xi}_k}{\partial \tau}(\tau,\boldsymbol{q}^-,\boldsymbol{q}^+)d\tau & \lambda^+ < 0 \end{cases} \tag{3.16}$$

To simplify this to a Rusanov/locally Lax-Friedrich flux, take $-\lambda^- = \lambda^+ = \lambda_{\max} = \max(|\lambda_m|)$; that is, λ^+ is the largest magnitude characteristic speed, and λ^- is the negative of this speed. Substituting this in,

$$(\overleftrightarrow{\boldsymbol{H}}_{ij}\,\widehat{n}_j)^*_{\text{Rusanov}} = \left\{\!\!\left\{ \overleftrightarrow{\boldsymbol{F}}_{ij} \right\}\!\!\right\} \widehat{n}^-_j + \frac{1}{2}\lambda_{\max}(\boldsymbol{q}^- - \boldsymbol{q}^+) \tag{3.17}$$

Note that the NCP flux appears to no longer be a direct function of $\overleftrightarrow{\boldsymbol{A}}$ at all. However, recall that $\lambda_{\max}$ is still computed using the matrix $\overleftrightarrow{\boldsymbol{J}}$ given in Eq. (3.13) which includes $\overleftrightarrow{\boldsymbol{A}}$.

For the non-conservative products, the choice of a path $\boldsymbol{\xi}$ can theoretically influence the actual scheme, where as for conservative terms any valid path $\boldsymbol{\xi}$ produces identical results. The simplest valid path is to choose a linear parametrization

$$\boldsymbol{\xi} = \boldsymbol{q}^- + \tau\left(\boldsymbol{q}^+ - \boldsymbol{q}^-\right) \tag{3.18}$$

Rhebergen[28, see: Ch. 3.3] investigated several alternative paths and found that in practice the actual choice of path has minimal influence on the numerical solution when used with a higher order DG scheme. For this reason, this research will focus primarily on using just the linear parametrization (3.18).

3.1.3 Discretizing higher-order derivatives

Consider the second order PDE subject to Neumann boundary conditions

$$\boldsymbol{R}_i - \frac{\partial}{\partial \boldsymbol{r}_j}\left(\overleftrightarrow{\boldsymbol{D}}_{ijkl}\,\frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{r}_l}\right) = 0 \tag{3.19}$$

$$\overleftrightarrow{\boldsymbol{D}}_{ijkl} \left. \frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{r}_l} \widehat{n}_j \right|_{\partial\Omega_N} = \boldsymbol{g}_i \tag{3.20}$$

One way to discretize this with DG is to write it as a coupled system of first order PDE's

$$\boldsymbol{R}_i - \frac{\partial}{\partial \boldsymbol{r}_j} \left(\overleftrightarrow{\boldsymbol{D}}_{ijkl} \overleftrightarrow{\boldsymbol{\sigma}}_{kl} \right) = 0 \tag{3.21}$$

$$\overleftrightarrow{\boldsymbol{\sigma}}_{kl} - \frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{r}_l} = 0 \tag{3.22}$$

This system can then be discretized using the standard DG formulation with an appropriate choice of numerical flux. This is known as the flux formulation. One example of a flux formulation method is the local discontinuous Galerkin (LDG) method[33]. The LDG method chooses the numerical fluxes for Eq. (3.21) and (3.22) respectively to be

$$(\overleftrightarrow{\boldsymbol{D}} \cdot \widehat{n})^*_i = \left\{\!\!\left\{ \overleftrightarrow{\boldsymbol{D}}_{ijkl} \overleftrightarrow{\boldsymbol{\sigma}}_{kl} \right\}\!\!\right\} \widehat{n}^-_j + \beta \left[\!\!\left[\overleftrightarrow{\boldsymbol{D}}_{ijkl} \overleftrightarrow{\boldsymbol{\sigma}}_{kl} \, \widehat{n}_j \right]\!\!\right] - \eta_i \widehat{n}^-_j \left[\!\left[\boldsymbol{q}_i \widehat{n}_j \right]\!\right] \tag{3.23}$$

$$(\boldsymbol{G}\widehat{n})^*_{ij} = \left\{\!\left\{ \boldsymbol{G}_i \right\}\!\right\} \widehat{n}^-_j - \beta \left[\!\left[\boldsymbol{G}_i \widehat{n}_j \right]\!\right] \tag{3.24}$$

Flux formulations are well suited for explicit time stepping schemes. The have excellent data locality properties, and are also relatively easy to implement since all existing DG framework for handling hyperbolic PDE's can be reused.

However, this is not the only method which can be used to discretize the higher-order derivative terms. An alternative method is to proceed as you would with standard continuous Galerkin methods to produce a stiffness matrix, and then provide a mechanism for resolving any discontinuities at shared interior faces. This formulation is referred to as the bilinear form. An example method which uses this technique is the Interior Penalty (IP) method[34]. This produces the discretized weak formulation

$$\int_\Omega \boldsymbol{R}_i \boldsymbol{w}_i dV + \int_\Omega \overleftrightarrow{\boldsymbol{D}}_{ijkl} \frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{r}_l} \frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{r}_j} dV + \bigoplus_K \int_{e_K} \eta \left[\!\!\left[\frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{r}_l} \widehat{n}_l \right]\!\!\right] \left[\!\!\left[\frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{r}_l} \widehat{n}_l \right]\!\!\right] dS = \oint_{\partial\Omega_N} \boldsymbol{g}_i \boldsymbol{w}_i dS, \quad \forall \boldsymbol{w} \in \boldsymbol{W} \tag{3.25}$$

An advantage of the bilinear form is that it is more amenable for implicit or direct solvers since the vector of unknowns does not include the auxiliary $\overleftrightarrow{\boldsymbol{\sigma}}$ variable. However, due to the nature of the DG discretization model this formulation still has more degrees of freedom than a CG method since the degrees of freedom of any unknown quantity q at shared element interfaces is necessarily twice as large. The IP method is not strictly inferior to CG methods, though, as the IP method maintains the key advantages of the DG method over CG methods; namely, it has good conservation properties, is well suited for mixed hyperbolic/parabolic systems, and can easily be used with mesh adaptive schemes.

Arnold et al.[35] showed that there is an equivalence between these two formulations; that is it is possible to convert a flux formulation method to the bilinear form and vice-versa. However, this does not necessarily guarantee that there is an equivalent cost in actual implementations for any particular method. For example, it is not advised to formulate an implicit/direct solve method using LDG in either the original flux formulation or the bilinear form. This is due to several factors[36]:

- The flux formulation has significantly more degrees of freedom than the bilinear form since the auxiliary $\overleftrightarrow{\boldsymbol{\sigma}}$ is part of the global vector of unknowns.
- The locality achieved using "bidirectional upwinding fluxes" is lost on implicit schemes; a global matrix system must still be formed and solved. In higher dimensions (greater than 1D), this results in additional coupling parameters between the degrees of freedom of an element and its neighboring elements resulting in more non-zeros in the global system matrix.
- The formulated system has a large condition number, making it difficult to use many efficient large scale linear solver techniques.

LDG and other explicit schemes for second order PDE's also generally have a timestep limit

proportional to Δx^2. For some problems, this may not be an issue. However, in well refined meshes this can result in being restricted to very small timesteps.

3.2 Hybridization of discontinuous Galerkin

In addition to having fewer total degrees of freedom, continuous Galerkin methods can take advantage of static condensation[37] to further reduce the global system size by re-writing true element interior degrees of freedom to be function of element edge degrees of freedom.

From examining the IP method's discretization in Eq. (3.25), it looks identical to the CG discretization with an extra surface integral term on interior faces e_K. Even though the IP method has a greater number of global degrees of freedom, its formulation appears to be almost naively suited for some form of condensation to reduce the global system size. The generalization of static condensation applied to DG methods is known as the Hybridizable Discontinuous Galerkin (HDG) method[26]. The general idea is to consider two finite element spaces: one finite element space defined internal on each element, and a second finite element space which is defined on the facets of the elements. No two elements are directly coupled to another element; instead, all coupling is handled via the interface of a shared facet. Fig. 3.2 shows an example HDG discretized domain.

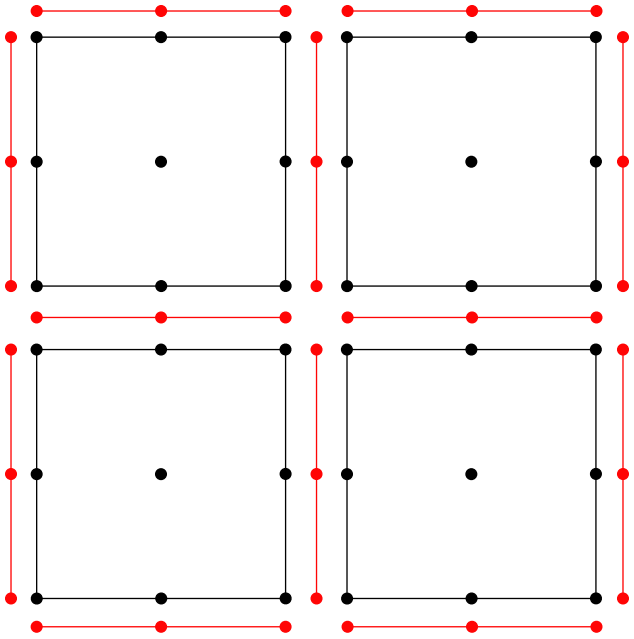


Figure 3.2: HDG discretization model example for 9 node quadrilateral elements. Element "internal" degrees of freedom are shown in black, and the coupling facet degrees of freedom are shown in red. Note that the physical separation between neighboring elements is for illustration purposes.

A given discretized field variable must be continuous and expressible with a finite polynomial series inside an element. Similarly, within a single facet a discretized field variable must be continuous and expressible with a finite polynomial series, and like traditional DG continuity between adjacent facets and elements are not necessarily continuous. This is expressed by augmenting the existing element space $\boldsymbol{W}$ defined in Eq. (3.7) with the face space $\boldsymbol{M}$ defined by

$$\boldsymbol{M} = \left\{ \boldsymbol{\mu} \in (L^2(e))^\alpha : \boldsymbol{\mu}|_\epsilon \in (\mathcal{P}_m(e))^\beta, \forall e \in \mathcal{E} \right\} \tag{3.26}$$

Consider the coupled PDE system

$$\boldsymbol{S}_i + \frac{\partial}{\partial \boldsymbol{r}_j} \left(\overleftrightarrow{\boldsymbol{F}}_{ij}(\boldsymbol{q}) - \overleftrightarrow{\boldsymbol{D}}_{ij}(\boldsymbol{q}, \overleftrightarrow{\boldsymbol{\sigma}}) \right) = 0 \tag{3.27}$$

$$\overleftrightarrow{\boldsymbol{\sigma}}_{ij} - \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{q}_j} = 0 \tag{3.28}$$

Then the local element system must satisfy

$$\int w_i S_i - (F_{ij} - D_{ij})\partial_j w_i dV + \oint ((F_{ij} - D_{ij})n_j)^* \mu_i dS = 0 \qquad \forall \boldsymbol{w}, \boldsymbol{\mu} \in \boldsymbol{W}, \boldsymbol{M} \tag{3.29}$$

$$\int w_i \sigma_{ij} + G_i \partial_j w_i dV - \oint (G_i n_j)^* \mu_i dS = 0 \qquad \forall \boldsymbol{w}, \boldsymbol{\mu} \in \boldsymbol{W}, \boldsymbol{M} \tag{3.30}$$

Similar to standard DG methods the numerical fluxes $((F_{ij} - D_{ij})n_j)^*$ and $(G_i n_j)^*$ must be continuous on a given facet. This is naturally enforced for $(G_i n_j)^*$ by requiring that it is represented only as a function of $\boldsymbol{\lambda} \in \boldsymbol{M}$. Then the numerical flux $((F_{ij} - D_{ij})n_j)^*$ must satisfy

$$\bigoplus_{e \in \mathcal{E}^0} \int_e [\![((F_{ij} - D_{ij})n_j)^*]\!] \mu_i dS + \int_{\mathcal{E}^N} ((F_{ij} - D_{ij})n_j)^* \mu_i dS - \int_{\mathcal{E}^N} (g_N)_i \mu_i dS = 0 \tag{3.31}$$

It is important to note that this approach introduces even more degrees of freedom than traditional DG methods have, as both $\boldsymbol{u} = [\boldsymbol{q}, \overset{\leftrightarrow}{\boldsymbol{\sigma}}]$ and $\boldsymbol{\lambda}$ are unknowns. However, a key advantage of introducing these extra unknowns is that solving for $\boldsymbol{u}$ becomes easily decoupled and parallelized at the element level. The global solve involves only $\boldsymbol{\lambda}$, which for higher order elements has significantly fewer degrees of freedom than $\boldsymbol{u}$. Suppose that a Newton-Raphson type non-linear solver coupled with a line search method is being used with the HDG method. Then the entire coupled system can be written in block matrix form as

$$\begin{bmatrix} \overset{\leftrightarrow}{\boldsymbol{A}}^{(n)} & \overset{\leftrightarrow}{\boldsymbol{B}}^{(n)} \\ \overset{\leftrightarrow}{\boldsymbol{C}}^{(n)} & \overset{\leftrightarrow}{\boldsymbol{D}}^{(n)} \end{bmatrix} \begin{bmatrix} \frac{\Delta \boldsymbol{u}^{(n+1)}}{\alpha^{(n+1)}} \\ \frac{\Delta \boldsymbol{\lambda}^{(n+1)}}{\alpha^{(n+1)}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{f}^{(n)} \\ \boldsymbol{g}^{(n)} \end{bmatrix} = 0 \tag{3.32}$$

where $\boldsymbol{f}^{(n)}$ is the evaluation of the LHS of Eqs. (3.29) and (3.30) and $\boldsymbol{g}^{(n)}$ is the evaluation of the LHS of Eq. (3.31) given a guess $\boldsymbol{u}^{(n)}$ and $\boldsymbol{\lambda}^{(n)}$. $\alpha^{(n+1)}$ is the line search step size. The block matrices are the Jacobian of $\boldsymbol{f}^{(n)}$ and $\boldsymbol{g}^{(n)}$

$$\overset{\leftrightarrow}{\boldsymbol{A}}^{(n)} = \partial_{\boldsymbol{u}^{(n)}} \boldsymbol{f}^{(n)} \tag{3.33}$$

$$\overset{\leftrightarrow}{\boldsymbol{B}}^{(n)} = \partial_{\boldsymbol{\lambda}^{(n)}} \boldsymbol{f}^{(n)} \tag{3.34}$$

$$\overset{\leftrightarrow}{\boldsymbol{C}}^{(n)} = \partial_{\boldsymbol{u}^{(n)}} \boldsymbol{g}^{(n)} \tag{3.35}$$

$$\overset{\leftrightarrow}{\boldsymbol{D}}^{(n)} = \partial_{\boldsymbol{\lambda}^{(n)}} \boldsymbol{g}^{(n)} \tag{3.36}$$

There are two important properties for the different blocks:

1. $\overset{\leftrightarrow}{\boldsymbol{A}}^{(n)}$ is block diagonal, with the blocks corresponding only to the degrees of freedom inside a single element.
2. $\overset{\leftrightarrow}{\boldsymbol{B}}^{(n)}$, $\overset{\leftrightarrow}{\boldsymbol{C}}^{(n)}$, and $\overset{\leftrightarrow}{\boldsymbol{D}}^{(n)}$ are sparse and have well defined sparsity patterns based on element connectivity.

The first property means that $\overset{\leftrightarrow}{\boldsymbol{A}}^{(n)}$ is easily invertible, making the Schur complement relatively easy to compute. The Schur complement system can be computed as

$$\underbrace{\left(\overset{\leftrightarrow}{\boldsymbol{D}}^{(n)} - \overset{\leftrightarrow}{\boldsymbol{C}}^{(n)} (\overset{\leftrightarrow}{\boldsymbol{A}}^{(n)})^{-1} \overset{\leftrightarrow}{\boldsymbol{B}}^{(n)}\right)}_{\text{Schur complement}} \frac{\Delta\boldsymbol{\lambda}^{(n+1)}}{\alpha^{(n+1)}} = \overset{\leftrightarrow}{\boldsymbol{C}}^{(n)} (\overset{\leftrightarrow}{\boldsymbol{A}}^{(n)})^{-1} \boldsymbol{f}^{(n)} - \boldsymbol{g}^{(n)} \tag{3.37}$$

Combined with the second property, in addition to being relatively easy to construct, the Schur complement also has strong guarantees on the sparsity pattern. Typically Eq. (3.37) is smaller than an equivalent global system from traditional DG methods, since $\boldsymbol{\lambda}$ only defined on facets which scales with surface area rather than volume. If the numerical flux is chosen such that given $\boldsymbol{\lambda}$ the element interior solution $\boldsymbol{u}$ can be determined in isolation from all other elements, then solving for the element interior solution becomes trivial to parallelize as

$$\frac{\Delta\boldsymbol{u}_K^{(i+1)}}{\alpha^{(i+1)}} \leftarrow \overset{\leftrightarrow}{\boldsymbol{A}}_K^{-1,(i+1)} \boldsymbol{f}_K^{(i)} - \overset{\leftrightarrow}{\boldsymbol{A}}_K^{-1,(i+1)} \overset{\leftrightarrow}{\boldsymbol{B}}_K^{(i)} \frac{\Delta\boldsymbol{\lambda}_K^{(i+1)}}{\alpha^{(i+1)}} \tag{3.38}$$

The general process for solving the HDG system then is:

function NONLINEAR HDG($\boldsymbol{u}^{(0)}$, $\boldsymbol{\lambda}^{(0)}$)

$i \leftarrow 0$

while solution not converged **do**

for each element K **do**

Evaluate $\overset{\leftrightarrow}{\boldsymbol{A}}_K^{(i)}$, $\overset{\leftrightarrow}{\boldsymbol{B}}_K^{(i)}$, and partial contributions to $\overset{\leftrightarrow}{\boldsymbol{C}}_K^{(i)}$, and $\overset{\leftrightarrow}{\boldsymbol{D}}_K^{(i)}$.

Solve $\overset{\leftrightarrow}{\boldsymbol{A}}_K^{(i)}\overset{\leftrightarrow}{\boldsymbol{\Gamma}}_K^{(i)} = \overset{\leftrightarrow}{\boldsymbol{B}}_K^{(i)}$ and $\overset{\leftrightarrow}{\boldsymbol{A}}_K^{(i)}\,\boldsymbol{\xi}_K^{(i)} = -\boldsymbol{f}_K^{(i)}$ for $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}_K^{(i)}$ and $\boldsymbol{\xi}_K^{(i)}$. Save these for later.

Add contributions of $\overset{\leftrightarrow}{\boldsymbol{C}}_K^{(i)}$ and $\overset{\leftrightarrow}{\boldsymbol{D}}_K^{(i)}$ to the global $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}$ and $\overset{\leftrightarrow}{\boldsymbol{D}}^{(i)}$ respectively.

end for

Construct and solve the Schur system from Eq. (3.37) for $\frac{\Delta\boldsymbol{\lambda}^{(i+1)}}{\alpha^{(i+1)}}$ using $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}$, $\overset{\leftrightarrow}{\boldsymbol{D}}^{(i)}$, $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}$, $\boldsymbol{\xi}^{(i)}$, and $\boldsymbol{g}^{(i)}$.

for each element K **do**

Back-solve for the element interior solution $\frac{\Delta\boldsymbol{u}_K^{(i+1)}}{\alpha^{(i+1)}} \leftarrow \boldsymbol{\xi}_K^{(i)} - \overset{\leftrightarrow}{\boldsymbol{\Gamma}}_K^{(i)}\,\frac{\Delta\boldsymbol{\lambda}_K^{(i+1)}}{\alpha^{(i+1)}}$

end for

Use a line search to determine $\alpha^{(i+1)} \in (0, 1]$ given $\frac{\Delta\boldsymbol{u}^{(i+1)}}{\alpha^{(i+1)}}$ and $\frac{\Delta\boldsymbol{\lambda}^{(i+1)}}{\alpha^{(i+1)}}$.

$\boldsymbol{u}^{(i+1)} \leftarrow \boldsymbol{u}^{(i)} + \Delta\boldsymbol{u}^{(i+1)}$

$\boldsymbol{\lambda}^{(i+1)} \leftarrow \boldsymbol{\lambda}^{(i)} + \Delta\boldsymbol{\lambda}^{(i+1)}$

$i \leftarrow i + 1$

end while

return $\boldsymbol{u}^{(i)}$, $\boldsymbol{\lambda}^{(i)}$

end function

3.2.1 HDG numerical flux

Deriving an appropriate HDG numerical flux proceeds similarly to the process for standard DG numerical fluxes[38]. The fluxes associated with higher order derivative terms are linearly combined

with any hyperbolic term fluxes. A key difference with HDG numerical fluxes is that the flux is no longer a function of the face interior and exterior values; instead, it becomes a function of the face interior and face hybridized variables. Second order terms are handled using a hybridized version of the Interior Penalty numerical flux

$$((F_{ij} - D_{ij})\widehat{n}_j)^* = \left(F_{ij}(\boldsymbol{\lambda}) - D_{ij}(\boldsymbol{\lambda}, \overset{\leftrightarrow}{\boldsymbol{\sigma}}^-)\right)\widehat{n}_j^- + \left(\theta_{ij}(\boldsymbol{\lambda}) + \eta D_{ij}(\boldsymbol{\lambda}, \overset{\leftrightarrow}{\boldsymbol{\sigma}}^-)\right)\left(q_j^- - G_j^{-1}(\boldsymbol{\lambda})\right) \tag{3.39}$$

where η is a user-chosen penalty, $\overset{\leftrightarrow}{\boldsymbol{\theta}}$ is a hyperbolic stabilization term, and $\boldsymbol{G}^{-1}$ denotes the inverse function of $\boldsymbol{G}$ such that

$$\boldsymbol{G}_i^{-1}(\boldsymbol{\lambda}) = \boldsymbol{q}_i \tag{3.40}$$

For example, consider the 1D fluid momentum to fluid velocity mappings

$$\boldsymbol{q}^T = \begin{bmatrix} \rho & p_x \end{bmatrix} \tag{3.41}$$

$$\boldsymbol{\lambda}^T = \begin{bmatrix} \rho & u_x \end{bmatrix} \tag{3.42}$$

The functions $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$ are then defined as

$$\boldsymbol{G}^T = \begin{bmatrix} q_0 & \frac{q_1}{q_0} \end{bmatrix} \tag{3.43}$$

$$\boldsymbol{G}^{-T} = \begin{bmatrix} \lambda_0 & \lambda_0\lambda_1 \end{bmatrix} \tag{3.44}$$

There are a variety of methods for choosing $\overset{\leftrightarrow}{\boldsymbol{\theta}}$; for example, it can be chosen based on an eigendecomposition of the flux Jacobian evaluated using the hybridized variables $\boldsymbol{\lambda}$

$$\overset{\leftrightarrow}{\boldsymbol{A}} = \frac{\partial \overset{\leftrightarrow}{\boldsymbol{F}}}{\partial \boldsymbol{\lambda}} = \overset{\leftrightarrow}{\boldsymbol{X}}\overset{\leftrightarrow}{\boldsymbol{\Lambda}}\overset{\leftrightarrow}{\boldsymbol{X}}^{-1} \tag{3.45}$$

$$\overset{\leftrightarrow}{\boldsymbol{\theta}} = \overset{\leftrightarrow}{\boldsymbol{X}} \mid \overset{\leftrightarrow}{\boldsymbol{\Lambda}} \mid \overset{\leftrightarrow}{\boldsymbol{X}}^{-1} \tag{3.46}$$

Alternatively, a Rusanov-like stabilization matrix can be used where only the fastest characteristic speed is considered.

$$\overset{\leftrightarrow}{\boldsymbol{\theta}} = \max(\mid \overset{\leftrightarrow}{\boldsymbol{\Lambda}} \mid) \overset{\leftrightarrow}{\boldsymbol{I}} \tag{3.47}$$

3.2.2 Implementation details and optimizations for HDG

There are a few interesting implementation details and optimizations which can be found by examining the composition and sparsity pattern of the different block matrices. The first interesting optimization comes from examining Eq. (3.31). If all the integrands are expanded over a common basis, then

$$\left(\bigoplus_{e \in \mathcal{E}^0} [\![((F_{ij} - D_{ijkl}\sigma_{kl})n_j)^*]\!]_{\alpha,e} + ((F_{ij} - D_{ijkl}\sigma_{kl})n_j)^*_{\alpha,\mathcal{E}^N} - g_{N,\alpha} \right) \mathcal{L} = 0 \tag{3.48}$$

$$\mathcal{L} = \int \mu_i \mu_j dS \tag{3.49}$$

where $\mathcal{L}$ is the lifting matrix. Depending on the ordering of the shared facet interface nodes, it is possible to arrange it such that $\overset{\leftrightarrow}{\boldsymbol{D}}^{(i)}$ is block diagonal, with square blocks of width equal to the number of nodes on that facet times the number of components of $\boldsymbol{\lambda}$ at a single node. However, since the RHS should be the zero vector, multiplying by $\mathcal{L}^{-1}$ and it's clear that this must be satisfied on a node by node basis, i.e. $\overset{\leftrightarrow}{\boldsymbol{D}}^{(i)}$ is essentially diagonal; there are still dense blocks of size number of components of $\boldsymbol{\lambda}$ at a single node. This somewhat simplifies the construction of $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}$, $\overset{\leftrightarrow}{\boldsymbol{D}}^{(i)}$, and the Schur complement.

The second observation is that $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}_K$ contributes to unique entries in $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}$, that is if and only if $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}_{K1,\alpha\beta} \neq 0$ then $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}_{K2,\alpha\beta} = 0$. $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}_K$ also generally has a dense slab-like structure, allowing a specialized sparse matrix format to more efficiently represent it than standard CSR/CSC formats. This same sparsity pattern ends up propagating into the Schur complement. Consider a mesh consisting of a 2x2 grid of 3-node triangles arranged as shown in Fig. 3.3.

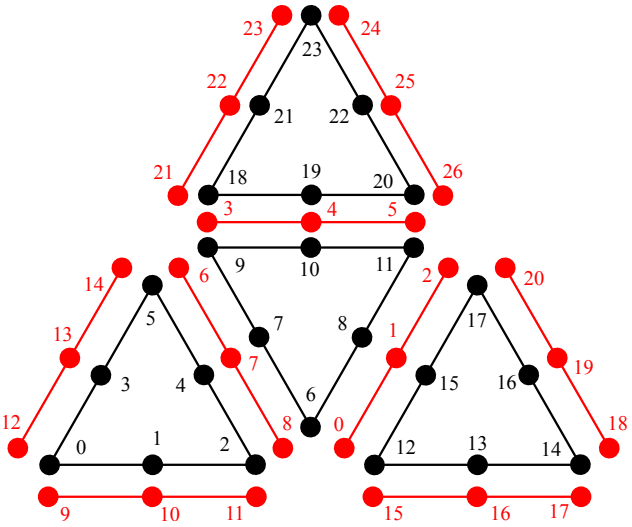
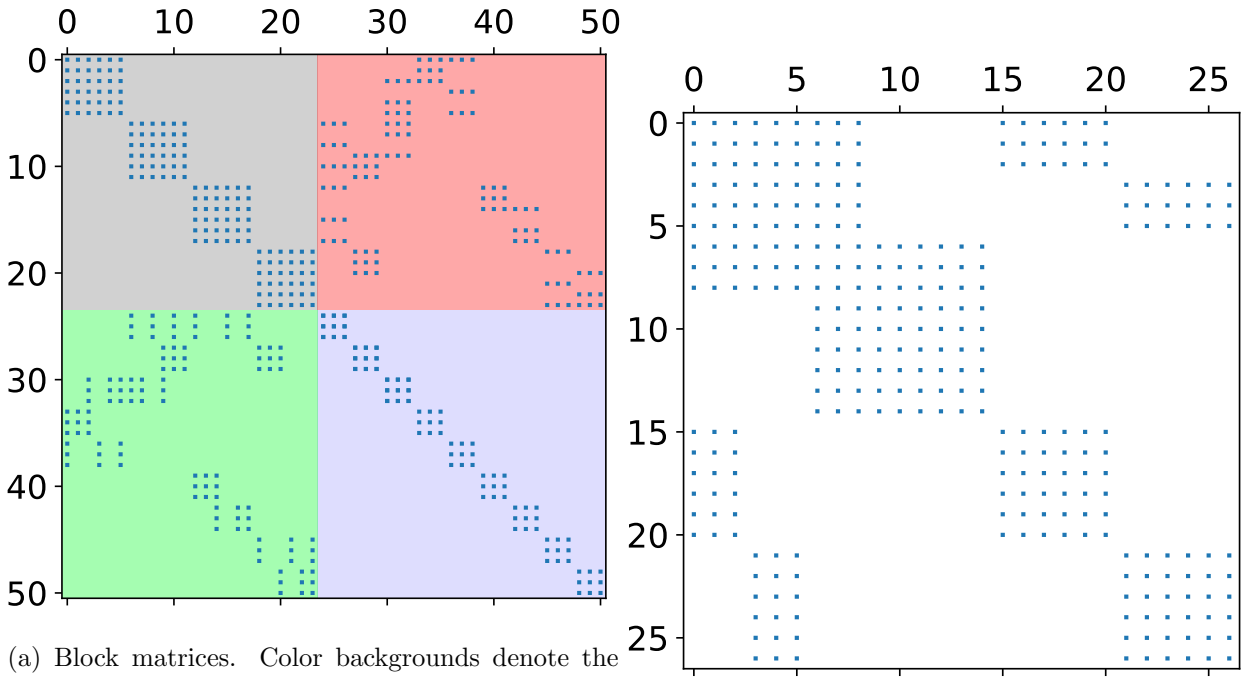


Figure 3.3: Example triangular mesh. Element internal DOF nodes are labeled in black, and facet coupling DOF nodes are labeled in red.

The sparsity pattern of each of the sub-blocks is shown in Fig. 3.4a, and Fig. 3.4b shows the sparsity pattern of the Schur complement. While $\overset{\leftrightarrow(i)}{\boldsymbol{B}}_K$ is more sparse than $\overset{\leftrightarrow(i)}{\boldsymbol{\Gamma}}_K$, storing it in the sparser form is not advantageous as during the solve for $\overset{\leftrightarrow(i)}{\boldsymbol{\Gamma}}_K$ it would have to be copied into a structure at least with the same sparsity pattern of $\overset{\leftrightarrow(i)}{\boldsymbol{\Gamma}}_K$.



(a) Block matrices. Color backgrounds denote the different blocks: gray for $\overset{\leftrightarrow}{\boldsymbol{A}}^{(i)}$, red for $\overset{\leftrightarrow}{\boldsymbol{B}}^{(i)}$, green for $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}$, and blue for $\overset{\leftrightarrow}{\boldsymbol{D}}^{(i)}$.

(b) Schur complement.

Figure 3.4: Sparsity pattern for HDG matrices. The properties of each block produces an easily construct-able Schur complement with strong guarantees on the number of non-zero entries.

Another interesting property is to note the data dependencies which form from the various block matrices. The global $\overset{\leftrightarrow}{\boldsymbol{A}}^{(i)}$ never needs to be formed; instead, only local matrices $\overset{\leftrightarrow}{\boldsymbol{A}}_K^{(i)}$ are needed, and the memory allocated can be re-used. Similarly, forming a global $\overset{\leftrightarrow}{\boldsymbol{B}}^{(i)}$ is not required; these both are exchanged for forming a global $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}$ and $\boldsymbol{\xi}^{(i)}$.

The last major matrices to consider are $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}$ and $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}$. The latter matrix couples all facet nodes which are directly connected to an element a given facet is on. For example, consider the facet $[0, 1, 2]$. It connects the lower right and central element. It's sparsity pattern from the operation $\overset{\leftrightarrow}{\boldsymbol{C}}^{(i)}\overset{\leftrightarrow}{\boldsymbol{\Gamma}}^{(i)}$ includes all facet nodes $[0, 1, 2]$ (itself), $[3, 4, 5]$, $[6, 7, 8]$, $[15, 16, 17]$, and $[18, 19, 20]$. This can be confirmed by inspecting rows $[0, 1, 2]$ in Fig. 3.4b. Similarly, consider the facet $[9, 10, 11]$. This only is connected to the lower left element, thus only includes facet nodes $[6, 7, 8]$, $[9, 10, 11]$ (itself),

and [12, 13, 14]. These facet node coupling blocks are all of size $N_{fp} \times N_{fp}$, where N_{fp} is the number of hybridized degrees of freedom on the facet. In this case, $N_{fp} = 3$. The blocks may have some non-unity striding if the facet nodes within a single facet are not numbered contiguously. Since a facet can only be shared by at most 2 elements, this bounds the number of these $N_{fp} \times N_{fp}$ blocks per row. This number is equal to the number of unique facets between the two elements. Suppose you have an element composed of a simplex tensor product with an orthotope. This general m-orthotope and n-simplex element covers the span of classical quadrilateral/hexahedral/hyperrectangle elements, triangle/tetrahedral elements, and all mixed prisms-type elements. For example, a 2-orthotope is a quadrilateral element, and a 1-orthotope/2-simplex is a triangular prism. The mixed orthotope-simplex elements is useful for discretizations of the continuum kinetic plasma model, where the domain could have a spatially complex geometry but require a relatively uniform phase space. These elements are used by WARPXM for solving the Vlasov-Maxwell system[39].

For a tensor product element of an m-orthotope and n-simplex, the number of facets in an element is given by

$$F = 2m + n + 1 \tag{3.50}$$

The number of unique facets shared between two elements is then

$$N = 2F - 1 \tag{3.51}$$

These values are tabulated for various element types in Table 3.1.

The sparsity pattern of $\overset{\leftrightarrow(i)}{\boldsymbol{C}}\overset{\leftrightarrow(i)}{\boldsymbol{\Gamma}}$ in general is identical with that of the Schur complement, since $\overset{\leftrightarrow(i)}{\boldsymbol{D}}$ only contains coupling of a facet with itself, which is already present in $\overset{\leftrightarrow(i)}{\boldsymbol{C}}\overset{\leftrightarrow(i)}{\boldsymbol{\Gamma}}$. The number of blocks which need to be stored per facet in the Schur complement is either given by Eq. (3.50) for non-periodic boundary facets, or Eq. (3.51) for all other facets.

Now consider the data dependency of the block multiplication of $\overset{\leftrightarrow(i)}{\boldsymbol{C}}$ with $\overset{\leftrightarrow(i)}{\boldsymbol{\Gamma}}$. Even though

for an internal facet the total coupling is shared between all facets of the two adjacent elements, this coupling can be split into the coupling of each element individually. There are no entries in $\overset{\leftrightarrow}{\boldsymbol{C}}_K^{(i)}$ which couple to $\overset{\leftrightarrow}{\boldsymbol{\Gamma}}_J^{(i)}$, for two elements $K \neq J$. Thus, the Schur complement can be built additively by computing and storing $\overset{\leftrightarrow}{\boldsymbol{C}}_K^{(i)}$ for all facets on a single element, then adding it's contribution $C_K^{(i)}\Gamma_K^{(i)}$ to the Schur complement.

Table 3.1: Number of unique facets within a single element, and two joined identical type elements.

Element Type	Facets per element	Unique facets between two elements
1-orthotope	2	3
2-simplex	3	5
2-orthotope	4	7
3-simplex	4	7
3-orthotope	6	11
2-simplex $\otimes$ 2-orthotope	7	13
4-orthotope	8	15
2-simplex $\otimes$ 3-orthotope	9	17
5-orthotope	10	19
3-simplex $\otimes$ 3-orthotope	10	19
6-orthotope	12	23

Eliminating unecessary auxiliary gradient variables

Naively implementing HDG using Eqs. (3.29)-(3.31) might imply that the auxiliary gradient variable $\overset{\leftrightarrow}{\boldsymbol{\sigma}}$ is necessary. However, by examining the total system, notice that if any component of $\overset{\leftrightarrow}{\boldsymbol{\sigma}}$ is not used in Eqs. (3.29) and by extension Eq. (3.31), then Eq. (3.30) for the unused components

of $\overleftrightarrow{\boldsymbol{\sigma}}$ are effectively decoupled. These equations can then be eliminated entirely from the system of equations being solved, reducing the local system size. This can have a significant impact for problems with high dimensionality, such as kinetic models which are primarily concerned with advecting $f^{(\alpha)}$ in phase space. For a full 3D-3V kinetic simulation, there could be up to 36 unused components of $\overleftrightarrow{\boldsymbol{\sigma}}$ per species.

3.2.3 Efficiency of HDG vs. traditional DG

There are certain scenarios where the goals of HDG to reduce the global system size are counteracted since the number of hybridized degrees of freedom may grow larger than the number of regular variables. This may make the local and global system sizes quite substantial, especially when considering continuum kinetic spaces which extend beyond 3D. As the degree of the basis polynomials increases, the global system size balance does tend to shift in the favor of HDG[40]. There are three basic type of elements considered: an N-orthotope, an N-simplex, and a tensor product of the two forming an N_1-orthotope $\otimes$ N_2-simplex.

Consider an N-orthotope tensor product element on which scalar fields span the space

$$W = \left\{ w \in L^2(\Omega) : w|_K \in \bigotimes_{i=1}^{N} \mathcal{P}_{m_i}(K), \forall K \in \mathcal{T} \right\} \tag{3.52}$$

where $\bigotimes$ represents a tensor outer product and $\mathcal{P}_{m_i}$ is a polynomial of degree less than or equal to m_i. The number of volume nodes is given by

$$N_v = \prod_{i=1}^{N} (m_i + 1) \tag{3.53}$$

and the number of facet nodes is given by

$$N_{fp} = 2 \sum_{i=1}^{N} \prod_{j=1, j\neq i}^{N} (m_j + 1) \tag{3.54}$$

Note that this latter figure double counts nodes located at the intersection of two facets; this is because an HDG scheme has separate degrees of freedom per facet even at these intersections. An

N-orthotope domain is then regularly partitioned using N-orthotope elements such that there are M equally sized elements in every direction. Table 3.2 summarizes the various relevant statistics for this type of domain. Assuming that $M \geq N$, then the point at which the HDG global system size is smaller than DG is when $m \geq N$. This can be problematic for kinetic simulations, where the benefits of HDG don't start showing up until you have at least a degree 6 basis for a 3D-3V domain.

Next consider an N-simplex element on which scalar fields span the space

$$W = \left\{ w \in L^2(\Omega) : w|_K \in \bigotimes_{i=1}^{N} \mathcal{P}_{\alpha_i}(K), \sum_{i=1}^{N} \alpha_i \leq m, \forall K \in \mathcal{T} \right\} \tag{3.55}$$

The number of volume nodes is given by

$$N_v = \binom{m+N}{N} \tag{3.56}$$

and the number of facet nodes is given by

$$N_{fp} = (N+1)\binom{m+N-1}{N-1} \tag{3.57}$$

Again, this double-counts the nodes at the intersection of two facets. Take the same domain and partition strategy as used for the N-orthotope comparison. Each N-orthotope element is then further subdivided into N-simplex elements. For example, a 2-orthotope can be partitioned into two 2-simplex, and a 3-orthotope can be partitioned into five 3-simplex[See: 41, Fig. 5(g)]. Unfortunately, it is not trivial to determine the number of simplex elements an N-orthotope element should be divided into. Thus, this comparison only considers the 2-simplex and 3-simplex cases. This is acceptable since higher dimension domains only occur with kinetic simulations; in these cases, the velocity space is expected to be discretized with a structured orthotope mesh. The plain simplex case is a generalization of the mixed orthotope/simplex elements, so the trade-off statistics are combined in Table 3.2. For large M, 2-simplex elements generally require at least a degree 2 basis, while 3-simplex elements generally require at least a degree 4 basis.

Lastly, consider an N_1-orthotope $\otimes$ N_2-simplex element on which scalar fields span the space

$$W = \left\{ w \in L^2(\Omega) : w|_K \in \left(\bigotimes_{i=1}^{N_1} \mathcal{P}_{m_i} \right) \otimes \left(\bigotimes_{i=1}^{N_2} \mathcal{P}_{\alpha_i} \right)(K), \sum_{i=1}^{N_2} \alpha_i \leq n, \forall K \in \mathcal{T} \right\} \tag{3.58}$$

The number of volume nodes is given by

$$N_v = \binom{n+N_2}{N_2} \prod_{i=1}^{N_1} (m_i + 1) \tag{3.59}$$

and the number of facet nodes is given by

$$N_{fp} = (N_2+1) \underbrace{\left(\binom{n+N_2-1}{N_2-1} \prod_{i=1}^{N_1} (m_i+1) \right)}_{\text{Physical space}} + 2 \sum_{i=1}^{N_1} \underbrace{\left(\binom{n+N_2}{N_2} \prod_{j=1, j\neq i}^{N_1} (m_j+1) \right)}_{\text{Velocity space}} \tag{3.60}$$

Assuming simplex elements are being used for physical space and orthotopes are being used in velocity space, then these facets can be classified as having normals aligned with physical space or velocity space. These are denoted in Eq. (3.60). The same N-orthotope domain is used as the previous tests. The partition strategy is to begin with an N_2-orthotope domain and partition it as done previously with simplex elements. The elements are then extruded into the remaining N_1 dimensions to produce the final discretized domain. The trade-off point for the global system size are summarized in Table 3.2.

Table 3.2: Global system coupling DOF comparison for system with p components and p hybridized components. Counts all boundary faces (assuming non-periodic).

Element Type	N-Orthotope
# elements	M^N
# faces	$NM^{N-1}(M+1)$
DG DOF	$pM^N(m+1)^N$
HDG DOF	$NpM^{N-1}(M+1)(m+1)^{N-1}$
$\frac{\text{HDG DOF}}{\text{DG DOF}} < 1$	$m > N - 1 + \frac{N}{M}$
Element Type	N_1-Orthotope $\otimes$ 2-Simplex
# elements	$2M^{2+N_1}$
# faces	$M^{1+N_1}((2N_1+3)M + 2N_1 + 2)$
DG DOF	$2pM^{2+N_1}(m+1)^{N_1}\binom{m+2}{2}$
HDG DOF	$pM^{1+N_1}(m+1)^{N_1}((3M+2)(m+1) + N_1(M+1)(m+2))$
$\frac{\text{HDG DOF}}{\text{DG DOF}} < 1$	$m > \frac{N_1}{2} + \frac{N_1+2+\sqrt{(M+1)((2+N_1)^2+M(4N_1+(2+N_1)^2))}}{2M}$
Element Type	N_1-Orthotope $\otimes$ 3-Simplex
# elements	$5M^{3+N_1}$
# faces	$M^{2+N_1}((5N_1+10)M + 5N_1 + 6)$
DG DOF	$5pM^{3+N_1}(m+1)^{N_1}\binom{m+3}{3}$
HDG DOF	$pM^{2+N_1}(m+2)(m+1)^{N_1}\left((10M+6)\frac{m+1}{2} + \frac{5}{6}N_1(M+1)(m+3)\right)$
$\frac{\text{HDG DOF}}{\text{DG DOF}} < 1$	$m > 1 + \frac{N_1}{2} + \frac{18+5N_1+\sqrt{25(N_1^2+16N_1+16)M^2+10(5N_1^2+58N_1+72)M+25N_1^2+180N_1+324}}{10M}$

Figure 3.5 graphically shows common element types and the point at which HDG could potentially be beneficial. For 1D-3D, HDG reduces the global system solve size with relatively low order elements. The benefits of HDG for 4D-6D elements tapers off at relatively high degree elements,

especially when using mixed simplex/orthotope elements. However, this also highlights another advantage of HDG: since HDG is based on DG, it is relatively easy to mix the two methods. In a kinetic plasma system, the kinetic species can be handled using standard DG while the physical space electromagnetic fields and fluid species can be handled using HDG.

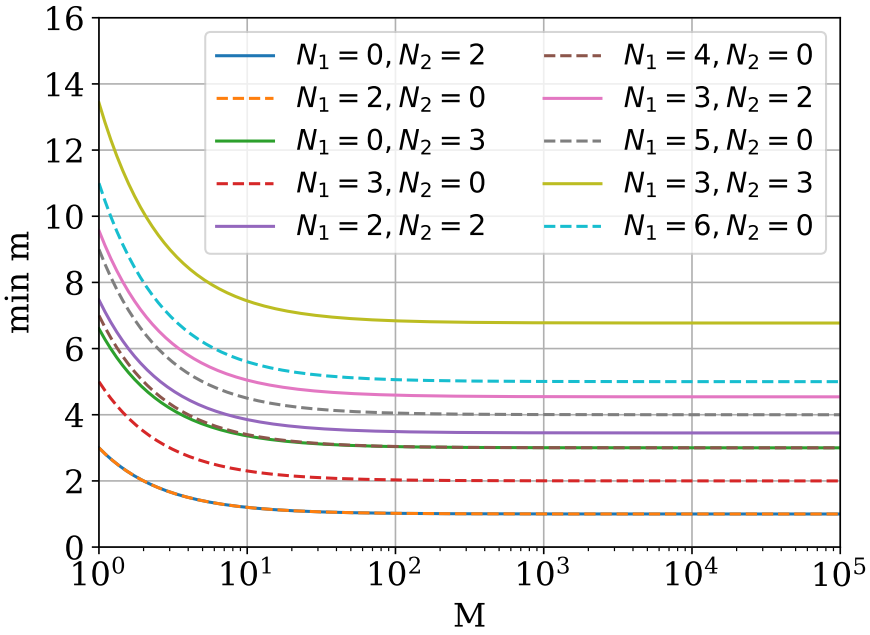


Figure 3.5: Comparison of various element types and what polynomial degree is required for HDG global system size to be equal to DG. 3-orthotope elements can reduce the global system size starting at $m = 2$; however, a 5-orthotope element requires $m \geq 4$ before the global system size is reduced. Use of pure simplex or mixed simplex-orthotope elements generally requires larger m compared to pure orthotope elements of the same dimensionality.

Chapter 4

Temporal discretization

The Discontinuous Galerkin and Hybridized Discontinuous Galerkin methods produce a semi-discrete system of coupled ordinary differential equations (ODE) or differential algebraic equations (DAE), depending on the presence of temporal derivatives in the PDE equations. For example, semi-discrete HDG systems form DAE systems since the introduced auxiliary Eq. (3.28) contains no temporal derivatives. Other common systems such as the Vlasov-Poisson system also produce DAE semi-discrete forms. This method of lines[24] approach allows the code to be flexible and handle a wide variety of system behaviors efficiently. For the purpose of this research, the temporal discretization is restricted to the class of single-step, multi-stage Runge-Kutta style methods. These methods are well suited for spatially adaptive methods since the solution vector $\boldsymbol{q}(t_n)$ is all that is required to advance the solution to $\boldsymbol{q}(t_n + \Delta t)$, while still being able to achieve higher order accuracy. This simplifies the process of adaptively modifying the solution vector since no extra interpolation steps are required to produce estimates of $\boldsymbol{q}$ at other time steps required for multi-step methods[42]. Additionally, single step, multi-stage methods are not subject to the Dahlquist barriers which limit the accuracy and stability bounds of multi-step methods[15, 16].

Consider a coupled system of first order ODE's

$$\partial_t \boldsymbol{q}(t) = \boldsymbol{f}(\boldsymbol{q}, t) \tag{4.1}$$

$$\boldsymbol{q}(t_0) = \boldsymbol{q}_0 \tag{4.2}$$

where $\boldsymbol{f}$ has no temporal derivatives. A Runge-Kutta method defines multiple intermediate stages $\boldsymbol{k}_i$ such that a single timestep for an s-stage method is given by

$$\boldsymbol{k}_i = \boldsymbol{f}\left(\boldsymbol{q}(t_n) + \Delta t \sum_{j=0}^{s-1} a_{ij}\boldsymbol{k}_j, t_n + c_i \Delta t\right) \tag{4.3}$$

$$\boldsymbol{q}(t_n + \Delta t) = \boldsymbol{q}(t_n) + \Delta t \sum_{i=0}^{s-1} a_{si}\boldsymbol{k}_i \tag{4.4}$$

The coefficients a_{ij} and c_i may be summarized compactly using Butcher Tableau's[43], typically written as shown in Table 4.1.

Table 4.1: Example Butcher Tableau

c_0	a_{00}	$\cdots$	$a_{0,s-1}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_{s-1}	$a_{s-1,0}$	$\cdots$	$a_{s-1,s-1}$
	$a_{s,0}$	$\cdots$	$a_{s,s-1}$

Note that the matrix of coefficients $\overleftrightarrow{\boldsymbol{a}}$ has been augmented with an additional row $a_{s,j}$ which are traditionally considered the $\boldsymbol{b}$ coefficients. This is done to make the notation more concise later on. A method is considered to be explicit if and only if $a_{ij} = 0, \forall j \geq i, \forall i \in [0, s-1]$. Otherwise, the method is considered to be implicit. A common modification of the Butcher Tableau is to extend to multiple rows $a_{s,i}^{(\alpha)}$, giving

$$\boldsymbol{q}^{(\alpha)}(t_n + \Delta t) = \boldsymbol{q}(t_n) + \Delta t \sum_{i=0}^{s-1} a_{s,i}^{(\alpha)}\boldsymbol{k}_i \tag{4.5}$$

This provides multiple estimates for the solution vector $\boldsymbol{q}(t_n + \Delta t)$ while re-using as much previous stage results. These are known as embedded Runge-Kutta methods[44]. Examples of such methods include the Runge-Kutta-Fehlberg[45] methods and Dormand Prince[46] methods, though versions for implicit and mixed implicit-explicit Runge-Kutta methods exist[42]. The goal of these methods is to provide an estimate for the time integration error. This allows the time step Δt to be adjusted from one time step to the next such that the time integration error is below an acceptable level. An example embedded Runge-Kutta method Butcher Tableau is shown in Table 4.2. It is uncommon for these methods to require more than two estimated solution vectors.

Table 4.2: Example modified Butcher Tableau for embedded Runge-Kutta methods.

c_0	a_{00}	$\cdots$	$a_{0,s-1}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_{s-1}	$a_{s-1,1}$	$\cdots$	$a_{s-1,s-1}$
	$a^{(1)}_{s,0}$	$\cdots$	$a^{(1)}_{s,s-1}$
	$a^{(2)}_{s,0}$	$\cdots$	$a^{(2)}_{s,s-1}$

In addition to traditional Runge-Kutta methods which handle an entire ODE/DAE system, the operator $\boldsymbol{f}$ can be split into multiple constituent parts such that

$$\partial_t \boldsymbol{q} = \boldsymbol{f}(\boldsymbol{q}, t) = \sum_{v=1}^{N} \boldsymbol{f}^v(\boldsymbol{q}, t) \tag{4.6}$$

Each individual operator $\boldsymbol{f}^v$ can then have a potentially different Runge-Kutta method applied which is tailored to the characteristics of the operator. This general formulation is known as the Additive Runge-Kutta method, or ARK$_N$[47]. If the splitting occurs such that the system can be

re-written as

$$\begin{bmatrix} \partial_t \boldsymbol{q}_1 \\ \partial_t \boldsymbol{q}_2 \\ \vdots \\ \partial_t \boldsymbol{q}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^1(\boldsymbol{q}, t) \\ \boldsymbol{f}^2(\boldsymbol{q}, t) \\ \vdots \\ \boldsymbol{f}^N(\boldsymbol{q}, t) \end{bmatrix} \tag{4.7}$$

then the method is additionally labeled as a partitioned Runge-Kutta (PRK_N) or Runge-Kutta-Nystrom method[47]. This type of splitting is potentially useful for example by handling ion and electron dynamics with different coupled RK methods. Note that PRK_N methods are a subset of the more general ARK_N methods.

4.1 Explicit Runge-Kutta (ERK)

Explicit Runge-Kutta methods are the simplest type of temporal discretization methods to implement. The primary down side of explicit Runge-Kutta methods is that they are subject to time step restrictions in order to maintain numerical stability. For problems where the dominant physical behavior is near the fastest characteristic time scale of the system, these methods can produce some of the fastest high accuracy solvers available since the temporal discretization is reduced into a few vector operations. Since no global direct solve is required, these methods can be trivially parallelizable and thus scale well on modern HPC architectures.

4.1.1 Alternative Representation

The traditional Runge-Kutta and Butcher Tableau formulations arrange the system such that each stage computes an estimate of the RHS $\boldsymbol{k}_i$. The collection of these $\boldsymbol{k}_i$ vectors are then superimposed to produce a new estimate $\boldsymbol{q}_i \approx \boldsymbol{q}(t_n + c_i \Delta t)$. Several explicit Runge-Kutta methods require $\boldsymbol{k}_i$ to be stored for all stages, which for an s-stage method requires storage for $s+2$ solution vectors each

of size $\boldsymbol{q}$ to be stored simultaneously. Additionally, there are extra operations required to compute the estimated $\boldsymbol{q}_i$, since these are needed anyways as inputs to $f(\boldsymbol{q}, t)$. It is possible to re-arrange the Butcher Tableau such that instead of solving for $\boldsymbol{k}_i$ at each stage, it directly solves for $\boldsymbol{q}_i$ at each stage. This reduces the storage requirement to $s + 1$ solution vectors each of size $\boldsymbol{q}$.

Consider a formulation

$$\boldsymbol{q}_i = \begin{cases} \boldsymbol{q}(t_n) & i = 0 \\ D_{i,i}\Delta t \boldsymbol{f}(\boldsymbol{q}_{i-1}, t + c_{i-1}\Delta t) + \sum_{j=0}^{i-1} D_{i,j}\boldsymbol{q}_j & i > 0 \end{cases} \tag{4.8}$$

$$\boldsymbol{q}(t_n + \Delta t) = \boldsymbol{q}_s \tag{4.9}$$

where D_{ij} are the modified coefficients. Assuming $a_{i+1,i} \neq 0, \forall i \in [0, s-1]$, then an explicit formulation for D_{ij} can be written as a function of the traditional Butcher Tableau coefficients.

$$D_{ij} = \begin{cases} 0 & j > i \\ 1 & i = 0, j = 0 \\ a_{i,i-1} & i = j, i > 0 \\ 1 - \sum_{k=0}^{i-2} \frac{a_{i,k}}{a_{k+1,k}} D_{k+1,0} & i > 1, j = 0 \\ \frac{a_{i,j-1}}{a_{j,j-1}} - \sum_{k=j}^{i-2} \frac{a_{i,k}}{a_{k+1,k}} D_{k+1,j} & \text{otherwise} \end{cases} \tag{4.10}$$

This can be arranged in a table as

Table 4.3: Alternative Runge-Kutta formulation table of coefficients.

	1	0	0	0	$\dots$	0
c_0	1	$\boxed{a_{10}}$	0	0	$\dots$	0
c_1	D_{20}	D_{21}	$\boxed{a_{21}}$	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\ddots$	$\vdots$
c_{s-2}	$D_{s-1,0}$	$D_{s-1,1}$	$\dots$	$D_{s-1,s-2}$	$\boxed{a_{s-1,s-2}}$	0
c_{s-1}	$D_{s,0}$	$D_{s,1}$	$\dots$	$D_{s,s-2}$	$D_{s,s-1}$	$\boxed{a_{s,s-1}}$

4.2 Diagonally implicit Runge-Kutta (DIRK)

For the purposes of this research, the only type of implicit Runge-Kutta methods considered are diagonally implicit (DIRK) methods. These methods are defined such that $a_{ij} = 0, \forall j > i, \forall i \in [0, s-1]$ and $a_{ii} \neq 0, \forall i \in [0, s-1]$.

4.2.1 Alternative representation

Similar to the alternative formulation for explicit Runge-Kutta methods, it is possible to re-arrange the Butcher Tableau's of DIRK methods to solve for estimated solution vectors $\boldsymbol{q}_i$ instead of $\boldsymbol{k}_i$. Consider a formulation

$$\boldsymbol{q}'_i = \begin{cases} \boldsymbol{q}(t_n) & i = 0 \\ \boldsymbol{q}_{i-1} + a_{i-1,i-1}\Delta t \boldsymbol{f}(\boldsymbol{q}'_i, t + c_{i-1}\Delta t) & i > 0 \end{cases} \tag{4.11}$$

$$\boldsymbol{q}_i = \boldsymbol{q}_0 + \Delta t \sum_{j=0}^{i-1} a_{i,j} K_j = D_{i,i}\boldsymbol{q}'_i + \sum_{j=0}^{i-1} D_{i,j}\boldsymbol{q}_j \tag{4.12}$$

where $D_{i,j}$ are the modified coefficients. Note the slight difference between this formulation and how the alternative ERK formulation is structured. An appropriate algorithm for implementing

this formulation is

---

function DIRK TIMESTEP($\boldsymbol{q}(t_n)$, t_n, Δt)

$\boldsymbol{q}_0 \leftarrow D_{0,0}\boldsymbol{q}(t_n)$

for $i \in [1, s]$ **do**

Solve $\frac{\boldsymbol{q}'_i - \boldsymbol{q}_{i-1}}{a_{i-1,i-1}\Delta t} = \boldsymbol{f}(\boldsymbol{q}'_i, t_n + c_{i-1}\Delta t)$ for $\boldsymbol{q}'_i$

$\boldsymbol{q}_i \leftarrow D_{i,i}\boldsymbol{q}'_i + \sum_{j=0}^{i-1} D_{i,j}\boldsymbol{q}_j$

end for

return $\boldsymbol{q}_s$

end function

Since $a_{ii} \neq 0, \forall i \in [0, s-1]$, an explicit formulation for D_{ij} can be written as a function of the traditional Butcher Tableau coefficients.

$$D_{ij} = \begin{cases} 0 & j > i \\ 1 & i = 0, j = 0 \\ \frac{a_{i,i-1}}{a_{i-1,i-1}} & i = j, i > 0 \\ 1 - \frac{a_{i,0}}{a_{0,0}} - \sum_{k=j}^{i-2} \frac{a_{i,k} D_{k+1,0}}{a_{k,k} D_{k+1,k+1}} & i > 1, j = 0 \\ \frac{a_{i,j-1}}{a_{j-1,j-1} D_{j,j}} - \frac{a_{i,j}}{a_{j,j}} - \sum_{k=j}^{i-2} \frac{a_{i,k} D_{k+1,j}}{a_{k,k} D_{k+1,k+1}} & \text{otherwise} \end{cases} \tag{4.13}$$

A key optimization comes by noticing that once $\boldsymbol{q}_i$ is computed, $\boldsymbol{q}'_i$ is never used again. This means the buffers of size $\boldsymbol{q}$ which are required for a general DIRK method are s buffers for $\boldsymbol{q}_i$, one buffer for $\boldsymbol{q}'_i$, and one buffer for storing the residual for use with performing the (non)-linear solve for a total of $s + 2$ buffers. A summary of coefficients for various DIRK methods are presented in Appendix B.2. In addition to the coefficients in $\boldsymbol{c}$ and $\overleftrightarrow{\boldsymbol{D}}$, this method also requires diag($\overleftrightarrow{\boldsymbol{A}}$). The format of the modified coefficients follows that shown in Table 4.4.

Table 4.4: Modified DIRK coefficients table template

		$D_{0,0}$	$\dots$	$D_{0,s}$
c_0	$a_{0,0}$	$D_{1,0}$	$\dots$	$D_{1,s}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_{s-1}	$a_{s-1,s-1}$	$D_{s,0}$	$\dots$	$D_{s,s}$

Special sparse methods exists which allow the number of buffers for $\boldsymbol{q}_i$ to be reduced similar to sparse explicit RK methods. For example, consider the IRK3 method[48] shown in Appendix B.2.3. Notice that after computing $\boldsymbol{q}_2$, the buffer being used to store $\boldsymbol{q}_1$ is no longer used (i.e. $D_{i,1} = 0$ for $i > 2$). Rather than allocating a new buffer, $\boldsymbol{q}'_3$ and $\boldsymbol{q}_3$ can use the memory previously used by $\boldsymbol{q}_1$. Using similar reasoning $\boldsymbol{q}'_4$ and $\boldsymbol{q}_4$ can overwrite the memory where $\boldsymbol{q}_2$ was. This reduces the actual memory footprint from needing 6 buffers of size $\boldsymbol{q}$ down to 4 of size $\boldsymbol{q}$.

4.3 Additive Runge-Kutta (ARK)

Both ERK and DIRK methods fall under the class of standard Runge-Kutta methods. Additive Runge-Kutta methods partition the original ODE system as[47]

$$\partial_t \boldsymbol{q} = \boldsymbol{f}(\boldsymbol{q}, t) = \sum_{v=1}^{N} \boldsymbol{f}^v(\boldsymbol{q}, t) \tag{4.14}$$

where the original operator $\boldsymbol{f}$ is partitioned into N sub-operators. Each sub-operator $\boldsymbol{f}^v$ is intended to have some property which can be exploited. A common splitting scheme would be to partition $\boldsymbol{f} = \boldsymbol{f}^I + \boldsymbol{f}^E$, where terms which are numerically stiff are grouped into $\boldsymbol{f}^I$, while the remaining non-numerically stiff terms are captured by $\boldsymbol{f}^E$. Each operator can then have a separate numerical scheme applied to them. For example, $\boldsymbol{f}^I$ can be treated implicitly while $\boldsymbol{f}^E$ has an explicit numerical scheme applied. It is this class of Implicit-Explicit (ImEx) schemes which are of interest

for this research. These type of schemes were originally proposed by Cooper and Sayfy[49, 50]. These methods are broadly labeled as ARK_N methods. For the purposes of this research, only $N = 2$ partitions of $\boldsymbol{f}$ are considered since the number of error coupling terms expands rapidly as more divisions are taken. In order to properly couple the ARK partitioned operators with high order accuracy, it is generally required that $\boldsymbol{c}^I = \boldsymbol{c}^E$. As noted by Ascher et al.[48], it is sometimes possible to augment the implicit method such that this condition is true. For example, if the number of implicit stages is initially one less than the number of explicit stages an extra row and column of zeros can be added to the Butcher Tableau, demonstrated in Table 4.5.

Table 4.5: Augmented Butcher Tableau of an implicit RK scheme to increase the number of stages by one.

0	0	0	$\dots$	0
$\widehat{c}_0^I$	0	$\widehat{a}_{00}^I$	$\dots$	$\widehat{a}_{0,s-1}^I$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\widehat{c}_{s-1}^I$	0	$\widehat{a}_{s-1,0}^I$	$\dots$	$\widehat{a}_{s-1,s-1}^I$
	0	$\widehat{a}_{s,0}^I$	$\dots$	$\widehat{a}_{s,s-1}^I$

Note that because $c_0^I = c_0^E = 0$, the first row $a_{0,j}^I = a_{0,j}^E = 0$; that is, $\boldsymbol{k}_0^I = \boldsymbol{f}^I(\boldsymbol{q}_0, t_n)$, which can be evaluated explicitly. The process of handling the ARK_2 schemes is then:

function ARK$_2$ TIMESTEP($\boldsymbol{q}(t_n)$, t_n, Δt)

$\boldsymbol{q}_0 \leftarrow \boldsymbol{q}(t_n)$

for each stage $i \in [0, s-1]$ **do**

Solve $\boldsymbol{k}_i^I = \boldsymbol{f}^I \left(\boldsymbol{q}_0 + \Delta t \left(\sum_{j=0}^{i} a_{ij}^I \boldsymbol{k}_j^I + \sum_{j=0}^{i-1} a_{ij}^E \boldsymbol{k}_j^E \right), t_n + c_i \Delta t \right)$ for $\boldsymbol{k}_i^I$

$\boldsymbol{k}_i^E \leftarrow \boldsymbol{f}^E \left(\boldsymbol{q}_0 + \Delta t \left(\sum_{j=0}^{i} a_{ij}^I \boldsymbol{k}_j^I + \sum_{j=0}^{i-1} a_{ij}^E \boldsymbol{k}_j^E \right), t_n + c_i \Delta t \right)$

end for

return $\boldsymbol{q}_0 + \Delta t \sum_{i=0}^{s-1} \left(a_{s,i}^I \boldsymbol{k}_i^I + a_{s,i}^E \boldsymbol{k}_i^E \right)$

end function

Similar to ERK and DIRK methods, this formulation can be re-cast to solve for $\boldsymbol{q}_i^I$ and $\boldsymbol{q}_i$ instead of $\boldsymbol{k}_i^I$ and $\boldsymbol{k}_i^E$. Note that if $a_{i,0}^I \neq 0 \ \forall i \in [1, s]$, then $\boldsymbol{k}_0^I$ must be stored in order to avoid a division by zero. This formulation results in the algorithm

function ARK$_2$ ALTERNATIVE FORMULATION TIMESTEP($\boldsymbol{q}(t_n)$, t_n, Δt)

$\boldsymbol{q}_0 \leftarrow \boldsymbol{q}(t_n)$

$\boldsymbol{k}_0^I \leftarrow f^I(\boldsymbol{q}_0, t_n + c_0 \Delta t)$

$\boldsymbol{q}_1 \leftarrow \boldsymbol{q}_0 + \Delta t \left(g_0 \boldsymbol{k}_0^I + a_{10}^E \boldsymbol{f}^E \left(\boldsymbol{q}_0, t_n + c_0 \Delta t \right) \right)$

for each stage $i \in [1, s-1]$ **do**

Solve $\boldsymbol{f}^I(\boldsymbol{q}_i^I, t_n + c_i \Delta t) = \frac{\boldsymbol{q}_i^I - \boldsymbol{q}_i}{a_{i,i}^I \Delta t}$ for $\boldsymbol{q}_i^I$

$\boldsymbol{q}_{i+1} \leftarrow \Delta t \left(g_i \boldsymbol{k}_0^I + a_{i+1,i}^E \boldsymbol{f}^E(\boldsymbol{q}_i^I, t_n + c_i \Delta t) \right) + \sum_{j=0}^{i-1} D_{i,j}^I \boldsymbol{q}_{j+1}^I + \sum_{j=0}^{i-1} D_{i,j}^E \boldsymbol{q}_j$

end for

return $\boldsymbol{q}_s$

end function

where the coefficients are defined as

$$g_i = a^I_{i+1,0} - \sum_{j=0}^{i-1} \frac{a^E_{i+1,j}}{D^E_{j+1,j+1}} g_j \tag{4.15}$$

$$D^I_{ij} = \begin{cases} 0 & j \geq i \\ \frac{a^I_{i+1,j+1}}{a^I_{j+1,j+1}} - \sum_{k=j}^{i-2} \frac{a^E_{i+1,k+1}}{a^E_{k+2,k+1}} D^I_{k+1,j} & \text{otherwise} \end{cases} \tag{4.16}$$

$$D^E_{ij} = \begin{cases} 0 & j > i \\ 1 & i = j = 0 \\ a^E_{i,i-1} & i = j, j > 0 \\ 1 - \sum_{k=1}^{i-1} \frac{a^E_{i,k-1}}{a^E_{k,k-1}} D^E_{k,0} & i > 0, j = 0 \\ \frac{a^E_{i,j-1}}{D^E_{j,j}} - \frac{a^I_{i,j}}{a^I_{j,j}} - \sum_{k=j}^{i-2} \frac{a^E_{i,k}}{a^E_{k+1,k}} D^E_{k+1,j} & \text{otherwise} \end{cases} \tag{4.17}$$

These can be summarized in tabular form as

Table 4.6: Alternative ARK formulation table of coefficients

(a) Explicit

	1	0	0	0	$\dots$	0
c_0	1	$\boxed{a^E_{10}}$	0	0	$\dots$	0
c_1	D^E_{20}	D^E_{21}	$\boxed{a^E_{21}}$	0	$\ddots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\ddots$	$\vdots$
c_{s-2}	$D^E_{s-1,0}$	$D^E_{s-1,1}$	$\dots$	$D^E_{s-1,s-2}$	$\boxed{a_{s-1,s-2}}$	0
c_{s-1}	$D^E_{s,0}$	$D^E_{s,1}$	$\dots$	$D^E_{s,s-2}$	$D^E_{s,s-1}$	$\boxed{a_{s,s-1}}$

(b) Implicit

c_1	$\boxed{a^I_{11}}$	g_0	0	0	$\dots$	0
c_2	$\boxed{a^I_{22}}$	g_1	D^I_{10}	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$
c_{s-1}	$\boxed{a^I_{s-1,s-1}}$	g_{s-2}	$D^I_{s-2,0}$	$D^I_{s-2,1}$	$\dots$	0
		g_{s-1}	$D^I_{s-1,0}$	$D^I_{s-1,1}$	$\dots$	$D^I_{s-1,s-2}$

4.4 Adaptive time stepping with digital filters

While well designed implicit methods are typically not limited by numerical stability for choosing an appropriate time step, they are still subject to discretization accuracy considerations for choosing a timestep. Choosing a constant timestep which guarantees sufficient accuracy for the entire simulation could be overly restrictive and drastically increase simulation time. Instead it desirable to adaptively determine what timestep is required to maintain a near-constant accuracy level per timestep. Since rapidly adjusting the timestep can lead to instabilities, an appropriate controller must be used

in order to maintain suitable accuracy and stability. There are a variety of controllers available[51], but for an ImEx scheme the most popular kind are PID-controllers[42]. Consider an embedded RK scheme where the local error estimate is given by $\tau_i^{(n+1)} = q_i^{(1)}(t_n + \Delta t_n) - q_i^{(2)}(t_n + \Delta t_n)$. A configurable error metric from Shampine et al.[52] can be defined as

$$\epsilon_{\text{tol},n+1} = \max\left(\frac{|\tau_i^{(n+1)}|}{\tau_{\text{abs},i} + \tau_{\text{rel},i}|q_{n+1,i}|}\right) \tag{4.18}$$

where $\boldsymbol{\tau}_{\mathbf{abs}}$ is a user-chosen absolute tolerance and $\boldsymbol{\tau}_{\text{rel}}$ is a user-chosen relative tolerance. Note that the absolute and relative tolerances can either be chosen uniformly for all components of $\boldsymbol{q}$, or vary per component. This is potentially useful for devising a time stepping scheme where the electron dynamics may be weighted less heavily compared to ion dynamics or vice-versa.

A PID-controller can then choose a new timestep to be

$$\Delta t_{n+1} = \kappa \Delta t_n \left(\epsilon_{\text{tol},n+1}\right)^{\alpha} \left(\epsilon_{\text{tol},n}\right)^{\beta} \left(\epsilon_{\text{tol},n-1}\right)^{\gamma} \tag{4.19}$$

$$-p\alpha = k_I + k_P + \left(\frac{2\omega_n}{1+\omega_n}\right) k_D \tag{4.20}$$

$$p\beta = k_P + 2\omega_n k_D \tag{4.21}$$

$$-p\gamma = \left(\frac{2\omega_n^2}{1+\omega_n}\right) k_D \tag{4.22}$$

where k_P, k_I, k_D are the PID-controller gains, $\kappa = 0.9$, $\omega_n = \frac{\Delta t_n}{\Delta t_{n-1}}$ is the relative timestep size change, and p is the order of accuracy of the error estimate. The characteristic equation for this controller is given by

$$\zeta^3 + (p\alpha - 1)\zeta^2 - p\beta\zeta + p\gamma = 0 \tag{4.23}$$

For the ARK3(2)4L[2]SA and ARK4(3)6L[2]SA methods listed in Appendix B.3.5 and B.3.6 respectively, choosing $k_I = 0.25$, $k_P = 0.14$, and $k_D = 0.1$ are suitably stable[42]. For the Dormand-Prince method[46], it is typically sufficient to use an I-controller where $k_I = \frac{p}{p+1}$ and $k_D = k_P = 0$. In addition to controller stability, the explicit operator must also be kept stable.

Note that this formulation for the controller is not limited to only considering absolute or error metrics; the timestep controller is responsible for seeking new timesteps such that $\epsilon_{\text{tol}} \to 1$.

It is conceivable that for some problems a CFL-type stability condition may be appropriate to include; in this case, the controller tolerance could be defined as

$$\epsilon_{\text{tol},n+1} = \frac{\lambda_{\max,n} \Delta t_n C_{\text{CFL}}}{\Delta x} \tag{4.24}$$

where $\lambda_{\max,n}$ is the fastest wave speed being considered, and C_{CFL} is a user-chosen limit for how close to the CFL condition the controller should seek to approach. This can be combined with the error-based metric to produce a hybrid controller

$$\epsilon_{\text{tol},n+1} = \frac{w_{\text{error}} \epsilon_{\text{error},n+1} + w_{\text{CFL}} \epsilon_{\text{CFL},n+1}}{w_{\text{error}} + w_{\text{CFL}}} \tag{4.25}$$

where $w_{\text{error}} \geq 0$ and $w_{\text{CFL}} \geq 0$ denote the user-chosen weights of the error and CFL metrics respectively.

The full adaptive timestep controller algorithm is given by

function ADAPTIVE TIMESTEP CONTROLLER($\boldsymbol{q}(t_n)$, Δt_{n-1}, Δt_n, $\epsilon_{\text{tol},n-1}$, $\epsilon_{\text{tol},n}$)

Try to advance $\boldsymbol{q}(t_n) \to \boldsymbol{q}(t_{n+1})$

if advance failed **then**

go to `step failure`

end if

Compute $\epsilon_{tol,n+1}$

if $\epsilon_{\text{tol},n+1} \geq \epsilon_{\max}$ **then**

go to `step failure`

end if

$\Delta t_{n+1} \leftarrow \min\left(\Delta t_n \min\left(\kappa(\epsilon_{\text{tol},n+1})^{\alpha}(\epsilon_{\text{tol},n})^{\beta}(\epsilon_{\text{tol},n-1})^{\gamma}, r_{\max}\right), \Delta t_{\max}\right)$

return step success, Δt_{n+1}, $\epsilon_{\text{tol},n+1}$, $\boldsymbol{q}(t_{n+1})$

`step failure:`

if $r_{\text{fail}}\Delta t_n < \Delta t_{\min}$ **then**

Terminate simulation (Δt too small)

end if

return step failure, $r_{\text{fail}}\Delta t_n$

end function

The parameters $r_{\max}$ and r_{fail} denote the maximum timestep growth and timestep retry reduction rates respectively. Based on preliminary testing, choosing $r_{\max} = 10$ and $r_{\text{fail}} = 0.5$ seem to give reasonable growth and retry limits.

Chapter 5

An artificial dissipation model for 5-moment fluids

High order discretization schemes seek to reduce the amount of dissipation in the solution. This is usually fine and leads to more accurate fits of the solution. However, in areas with steep gradients this can lead to excessive oscillations which can cause the simulation to produce nonsensical results. Artificial dissipation can be added to the PDE system to selectively re-introduce the necessary dissipation to reduce the oscillations. Other options include limiting the solution slope or limiting the numerical flux. Artificial dissipation has the advantage of being relatively easy and efficient to implement vs. limiter schemes since it only uses the existing discretization primitives. Additionally, since artificial dissipation schemes re-use existing discretization primitives, the conservative properties of the PDE can be easily maintained as well as the base spatial and temporal discretization schemes would allow, while limiter schemes must take additional care to maintain the conservation properties. However, an artificial dissipation model must be crafted to fit each particular PDE, and potentially may even require problem-specific tuning.

There has been extensive research done on artificial dissipation models for the 5-Moment Fluid

system which can be leveraged to produce a scheme appropriate for DG. The model presented in this chapter introduces an artificial volume viscosity and artificial heat flux, similar to work by Noh[17].

Consider the augmented 5-moment Euler equations

$$\partial_t \rho + \nabla \cdot \boldsymbol{p} = 0 \tag{5.1}$$

$$\partial_t \boldsymbol{p} + \nabla \cdot \left(\boldsymbol{p} \otimes \boldsymbol{u} + (P + Q) \overleftrightarrow{\boldsymbol{I}} \right) = 0 \tag{5.2}$$

$$\partial_t e + \nabla \cdot \left((e + P + Q)\boldsymbol{u} + \boldsymbol{h} \right) = 0 \tag{5.3}$$

where Q and $\boldsymbol{h}$ are the artificial viscosity and artificial heat flux operators respectively. Noh defines these two operators as

$$Q = - \left(C_0 l_c^2 |\nabla \cdot \boldsymbol{u}| + C_1 l_c c_s \right) \rho \nabla \cdot \boldsymbol{u} \tag{5.4}$$

$$\boldsymbol{h} = - \left(C_2 l_c^2 |\nabla \cdot \boldsymbol{u}| + C_3 l_c c_s \right) \rho \frac{\nabla T}{A} \tag{5.5}$$

where C_0, C_1, C_2, and C_3 are user-chosen coefficients which must be non-negative, c_s is the local sound speed, and l_c is the characteristic length scale, typically chosen as $l_c = \sqrt[N]{V}$, where N is the number of dimensions of the element (e.g. $N = 1$ for lines, $N = 2$ for triangles/quadrilaterals, etc.) and V is the element volume. These coefficients need not be constants, and it is often advantageous to tune these coefficients differently based on the local compressibility; that is, these coefficients can be split

$$C_i = \begin{cases} C_i^- & \nabla \cdot \boldsymbol{u} < 0 \\ 0 & \nabla \cdot \boldsymbol{u} = 0 \\ C_i^+ & \nabla \cdot \boldsymbol{u} > 0 \end{cases} \tag{5.6}$$

These coefficients can be tuned manually on a per-problem basis, however Morgan[53] noted that there is a physical phenomena involving the relationship between the equation of states and entropy

of the system which would lead to an appropriate amount of artificial viscosity, and Landshoff[54] derived this relationship for gamma-law gases. Unfortunately, no similar relation has been derived for artificial heat flux, however from initial testing it appears that the same form for the coefficients produced from the artificial viscosity analysis work quite well for artificial heat flux. Tests were performed to determine what combination of artificial viscosity and artificial heat flux should be applied based on the sign of $\nabla\cdot\boldsymbol{u}$. These results are tabulated in Appendix C, and are summarized below. It was noted that artificial viscosity and artificial heat flux were both equally capable of providing the necessary dissipation for the shock discontinuity; however, artificial viscosity cannot reduce oscillations in the vicinity of the contact discontinuity since $\nabla\cdot\boldsymbol{u}\approx 0$. Applying artificial heat flux when $\nabla\cdot\boldsymbol{u}>0$ produces anomalous unphysical oscillations in the rarefaction wave, while artificial viscosity was capable of reducing non-monotonicity in the rarefaction wave. This results in choosing

$$C_0 = \begin{cases} 0 & \nabla\cdot\boldsymbol{u}\leq 0 \\ \frac{\gamma+1}{4} & \nabla\cdot\boldsymbol{u}>0 \end{cases} \tag{5.7}$$

$$C_1 = \begin{cases} 0 & \nabla\cdot\boldsymbol{u}\leq 0 \\ \sqrt{\left(\frac{\gamma+1}{4}l_c\,\nabla\cdot\boldsymbol{u}\right)^2+c_s^2}-c_s & \nabla\cdot\boldsymbol{u}>0 \end{cases} \tag{5.8}$$

$$C_2 = \begin{cases} \frac{\gamma+1}{4} & \nabla\cdot\boldsymbol{u}<0 \\ 0 & \nabla\cdot\boldsymbol{u}\geq 0 \end{cases} \tag{5.9}$$

$$C_3 = \begin{cases} \sqrt{\left(\frac{\gamma+1}{4}l_c\,\nabla\cdot\boldsymbol{u}\right)^2+c_s^2}-c_s & \nabla\cdot\boldsymbol{u}<0 \\ 0 & \nabla\cdot\boldsymbol{u}\geq 0 \end{cases} \tag{5.10}$$

This formulation performs well for minor to moderate jump discontinuities, however it appears to be insufficient for large jump discontinuities. For large jump discontinuities, using only artificial

viscosity produces better results (i.e. larger timesteps can be taken for similar solution error estimates, or producing a simulation which runs at all). It's unclear if this is due to not properly tuning the artificial heat flux coefficients, or if there is some fundamental breakdown in the model.

Additional method performance can be achieved by introducing artificial floors for density and internal fluid energy; that is, require

$$\rho \geq \rho_{min} \tag{5.11}$$

$$e \geq U_{min} + \frac{\boldsymbol{p} \cdot \boldsymbol{p}}{2\rho} \tag{5.12}$$

This simplistic "limiter" normally is insufficient for producing a suitable numerical discretization scheme since by itself it doesn't provide any damping or reduction of high frequency numerical oscillations. However, when coupled with an artificial dissipation model, it prevents numerical errors such as division by zero or taking the square root of negative numbers from crashing the simulation long enough for the artificial dissipation model to damp the numerical oscillations and produce meaningful results. Note that the use of a floor on density and internal energy will artificially introduce mass and energy into the system. It is assumed that for problems of interest the amount of mass or energy introduced is insignificant, however this assumption should be carefully evaluated on a per-problem basis.

Chapter 6

Coupling plasma models

There are two primary strategies for coupling different plasma models. The first strategy is to partition the domain with non-overlapping subdomains where in each subdomain contains only a single plasma model. The coupling can then be handled by either producing a consistent mapping of the plasma properties for both models at the interface, or by producing a consistent numerical flux. This approach has the advantage of being able to guarantee conservation of important quantities across the boundary. However, a potential problem is that a sharp interface can introduce spurious oscillations.

The second strategy is to introduce an overlapping transition region where the different models are weighted to produce an appropriate new plasma state over time. The goal is that a smooth weighting function can be used which allows any intermediate dynamics to smoothly transition between the two models. However, this method requires simultaneously computing multiple plasma models in the transition region, which may be expensive. Additionally, it's not clear what a suitable transition region when more than two plasma models meet should be and it is very difficult to enforce conservation properties using this approach.

6.1 Single Temperature MHD to two-fluid

The derivation strategy for the single temperature MHD model in Section 2.2 was carefully chosen to demonstrate how a center of mass model relates to the two-fluid model's individual fluid species properties. Namely, these relations are

$$\rho = \rho^{(i)} + \rho^{(e)} \tag{6.1}$$

$$\boldsymbol{p}_i = \boldsymbol{p}_i^{(i)} + \boldsymbol{p}_i^{(e)} \tag{6.2}$$

$$\boldsymbol{u}_i = \frac{\boldsymbol{p}_i}{\rho} \tag{6.3}$$

$$e = e^{(i)} + e^{(e)} + \frac{1}{2}\left(\boldsymbol{B}_i^2 + \frac{1}{(\omega_p \tau)^2 \left(\frac{\delta_p}{L}\right)^2}\boldsymbol{E}_i^2\right) \tag{6.4}$$

where e is defined as the total system energy, not just the total fluid energy. Traditionally under the MHD assumptions, these reduce to

$$\rho \approx \rho^{(i)} \tag{6.5}$$

$$\boldsymbol{p}_i \approx \boldsymbol{p}_i^{(i)} \tag{6.6}$$

$$e \approx U^{(i)} + U^{(e)} + \frac{1}{2}\left(\boldsymbol{p}_i \boldsymbol{u}_i + \boldsymbol{B}_i^2\right) \tag{6.7}$$

where notably the total system energy completely neglects any electric field energy. In order for any coupling scheme to be successful, these assumptions must be valid in the vicinity of the coupling region. This is because the MHD model must be valid in the vicinity of the coupling region; otherwise, it makes no sense to be using the MHD model in the first place.

Coupling via model boundary conditions

Consider the first coupling strategy where the two plasma models are coupled by introducing appropriate boundary intermediate conditions. The MHD properties can be mapped to two-fluid

properties in one of two ways. Using Eqs. (6.1)-(6.4) and

$$Z^{(i)}n^{(i)} + Z^{(e)}n^{(e)} = 0 \tag{6.8}$$

$$\boldsymbol{j}_i = \frac{Z^{(i)}}{A^{(i)}}\boldsymbol{p}_i^{(i)} + \frac{Z^{(e)}}{A^{(e)}}\boldsymbol{p}_i^{(e)} \tag{6.9}$$

$$T^{(\beta)} = \theta^{(\beta\alpha)}T^{(\alpha)} \tag{6.10}$$

the individual fluid species properties can be derived as

$$\rho^{(\alpha)} = \frac{\rho}{1 - \frac{Z^{(\alpha)}A^{(\beta)}}{Z^{(\beta)}A^{(\alpha)}}} \tag{6.11}$$

$$\boldsymbol{p}_i^{(\alpha)} = \frac{A^{(\alpha)}(Z^{(\beta)}\boldsymbol{p}_i - A^{(\beta)}\boldsymbol{j}_i)}{A^{(\alpha)}Z^{(\beta)} - A^{(\beta)}Z^{(\alpha)}} \tag{6.12}$$

$$\boldsymbol{u}_i^{(\alpha)} = \boldsymbol{u}_i - \frac{A^{(\beta)}\boldsymbol{j}_i}{Z^{(\beta)}\rho} \tag{6.13}$$

$$e^{(\alpha)} = \frac{1}{1 - \theta^{(\beta\alpha)}\frac{Z^{(\alpha)}}{Z^{(\beta)}}}\left(e - \frac{1}{2}\left(\boldsymbol{p}_i^{(\beta)}\boldsymbol{u}_i^{(\beta)} + \frac{Z^{(\alpha)}}{Z^{(\beta)}}\theta^{(\beta\alpha)}\boldsymbol{p}_i^{(\alpha)}\boldsymbol{u}_i^{(\alpha)} + \boldsymbol{B}_i^2 + \frac{1}{(\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)^2}\boldsymbol{E}_i^2\right)\right) \tag{6.14}$$

If instead Eqs. (6.5)-(6.7) were used, then

$$\rho^{(i)} \approx \rho \tag{6.15}$$

$$\rho^{(e)} \approx -\frac{A^{(e)}Z^{(i)}}{A^{(i)}Z^{(e)}}\rho \tag{6.16}$$

$$\boldsymbol{p}_i^{(i)} \approx \boldsymbol{p}_i \tag{6.17}$$

$$\boldsymbol{p}_i^{(e)} \approx \frac{A^{(e)}}{A^{(i)}Z^{(e)}}(A^{(i)}\boldsymbol{j}_i - Z^{(i)}\boldsymbol{p}_i) \tag{6.18}$$

$$T_i = T_e \approx \frac{1}{n^{(i)} + n^{(e)}}\left(e - \frac{1}{2}\left(\boldsymbol{p}_i\boldsymbol{u}_i + \boldsymbol{B}_i^2\right)\right) \tag{6.19}$$

Either set of relations can be used with similar success to provide coupling boundary conditions since full two-fluid and MHD variable states are known on both sides of the boundary[55].

This can be done non-conservatively by naively using the individual model numerical fluxes, or by choosing one model's numerical fluxes and deriving appropriate conservative numerical fluxes for the other model. For example, take the two-fluid numerical fluxes $(\boldsymbol{p}^{(\alpha)}\cdot\widehat{n})^*$, $(\overleftrightarrow{\boldsymbol{H}}^{(\alpha)}\cdot\widehat{n})^*$, $(\boldsymbol{q}^{(\alpha)}\cdot\widehat{n})^*$,

$(\overleftrightarrow{\boldsymbol{B}} \cdot \widehat{n})^*$, and $(\boldsymbol{S} \cdot \widehat{n})^*$ to be calculated from the two-fluid variable state given by Eqs. (6.11)-(6.14) on both sides of the coupling boundary and $(\boldsymbol{S} \cdot \widehat{n})^*$ is an appropriate numerical flux for Eq. (2.78). Then an appropriate set of MHD numerical fluxes which guarantee conservation properties are

$$(\boldsymbol{p} \cdot \widehat{n})^*_{\text{MHD}} = (\boldsymbol{p}^{(i)} \cdot \widehat{n})^*_{\text{two-fluid}} + (\boldsymbol{p}^{(e)} \cdot \widehat{n})^*_{\text{two-fluid}} \tag{6.20}$$

$$(\boldsymbol{H} \cdot \widehat{n})^*_{\text{MHD}} = (\boldsymbol{H}^{(i)} \cdot \widehat{n})^*_{\text{two-fluid}} + (\boldsymbol{H}^{(e)} \cdot \widehat{n})^*_{\text{two-fluid}} \tag{6.21}$$

$$(\boldsymbol{q} \cdot \widehat{n})^*_{\text{MHD}} = (\boldsymbol{q}^{(i)} \cdot \widehat{n})^*_{\text{two-fluid}} + (\boldsymbol{q}^{(e)} \cdot \widehat{n})^*_{\text{two-fluid}} + (\boldsymbol{S} \cdot \widehat{n})^*_{\text{two-fluid}} \tag{6.22}$$

$$(\overleftrightarrow{\boldsymbol{B}} \cdot \widehat{n})^*_{\text{MHD}} = (\overleftrightarrow{\boldsymbol{B}} \cdot \widehat{n})^*_{\text{two-fluid}} \tag{6.23}$$

Coupling via a transition region

Consider the second coupling strategy where an intermediate transition region of thickness τ is introduced between the two models. Define a mixing parameter $\zeta(\tau) \in [0, 1]$. Then the fluid properties can be written as

$$q = \zeta \tilde{q}_{\text{two-fluid}} + (1 - \zeta)\tilde{q}_{\text{MHD}} \tag{6.24}$$

where the tilde quantities are the plasma properties predicted by the respective models. For example, the ion density can be written as

$$\rho^{(i)} = \zeta \tilde{\rho}^{(i)}_{\text{two-fluid}} + (1 - \zeta)\frac{\tilde{\rho}_{\text{MHD}}}{1 - \frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}} \tag{6.25}$$

This method lacks conservation guarantees even though each individual model is conservative[55]. To prove this, define the initial mass in the transition region at time t_n to be

$$\int \tilde{\rho}^{(i)}_{\text{two-fluid}}(t_n, \tau)d\tau = \int \frac{\tilde{\rho}_{\text{MHD}}(t_n, \tau)}{1 - \frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}}d\tau = \int \rho^{(i)}(t_n, \tau)d\tau \tag{6.26}$$

If each constituent model has mass conservation guarantees, then

$$\int \tilde{\rho}^{(i)}_{\text{two-fluid}}(t_{n+1}, \tau)d\tau = \int \frac{\tilde{\rho}_{\text{MHD}}(t_{n+1}, \tau)}{1 - \frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}}d\tau = \int \rho^{(i)}(t_n, \tau)d\tau \tag{6.27}$$

Applying the mixing function,

$$\int \rho^{(i)}(t_{n+1},\tau)d\tau = \int \zeta(\tau)\left(\tilde{\rho}^{(i)}_{\text{two-fluid}}(t_{n+1},\tau) - \frac{\tilde{\rho}_{\text{MHD}}(t_{n+1},\tau)}{1-\frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}}\right) + \frac{\tilde{\rho}_{\text{MHD}}(t_{n+1},\tau)}{1-\frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}}d\tau \tag{6.28}$$

$$= \int \zeta(\tau)\left(\tilde{\rho}^{(i)}_{\text{two-fluid}}(t_{n+1},\tau) - \frac{\tilde{\rho}_{\text{MHD}}(t_{n+1},\tau)}{1-\frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}}\right)d\tau + \int \rho^{(i)}(t_n,\tau)d\tau \tag{6.29}$$

The method conserves ion mass conservation when

$$\int \rho^{(i)}(t_{n+1},\tau)d\tau - \int \rho^{(i)}(t_n,\tau)d\tau = 0 \tag{6.30}$$

This is true if and only if

$$\int \zeta(\tau)\left(\tilde{\rho}^{(i)}_{\text{two-fluid}}(t_{n+1},\tau) - \frac{\tilde{\rho}_{\text{MHD}}(t_{n+1},\tau)}{1-\frac{Z^{(i)}A^{(e)}}{Z^{(e)}A^{(i)}}}\right)d\tau = 0 \tag{6.31}$$

This leads to two primary conditions for conservation guarantees:

1. If $\zeta(\tau)$ is a constant, then the method is guaranteed to conserve mass.

2. If the two-fluid and MHD models predict the same flow of ion density then the method conserves ion mass density for any mixing functions ζ.

If neither conditions 1 or 2 apply, then there is a loss of mass conservation roughly proportional to the difference in the predicted ion density flow of the two models. Similar conclusions can be reached for all other quantities such as fluid momentum, energy, etc. While this lack of conservation guarantees appears to be problematic, the transition region must be located in a region where both constituent models must be physically valid; that is, they should predict similar flows. Investigating how detrimental this is for actual implementations compared to the benefits of smoothing out the model transition is an area of future work.

6.2 Adaptivity metrics

A crucial component of being able to couple multiple plasma models is that each model must be valid in the subdomains they are applied to. These metrics can be derived by mathematically

considering what the difference in predicted flows are. For coupling the two-fluid and MHD models, this comparison can be made by writing the two-fluid model as a center of mass model without applying any additional assumptions. Subtracting the MHD model from the full center of mass model gives the requirements

$$\frac{A^{(i)}A^{(e)}}{Z^{(i)}Z^{(e)}}\partial_j\left(\frac{j_i j_j}{\rho^2}\right)+\left(\frac{L}{\delta_p}\right)\rho_c E_i+\left(\left(\frac{L}{\delta_p}\right)j_k-\partial_m B_l\epsilon_{klm}\right)B_j\epsilon_{ijk}\to 0 \tag{6.32}$$

$$\partial_i\left(\frac{1}{2}(\omega_p\tau)^{-2}\left(\frac{L}{\delta_p}\right)^2 E_j^2 u_i+\frac{j_j}{\rho}\left(\frac{A^{(e)}}{Z^{(e)}}\left(e^{(i)}+P_{ij}^{(i)}\right)+\frac{A^{(i)}}{Z^{(i)}}\left(e^{(e)}+P_{ij}^{(e)}\right)\right)\right)\to 0 \tag{6.33}$$

$$(\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)\rho_c\to 0 \tag{6.34}$$

$$\frac{1}{(\omega_p\tau)^2\left(\frac{\delta_p}{L}\right)^2}\partial_t E_i\to 0 \tag{6.35}$$

$$E_i-\left(\frac{1}{Z^{(e)}n^{(e)}}\left(\frac{\delta_p}{L}\right)\left(\partial_j\left(P_{ij}^{(e)}\right)-R_i^{(ie)}-(\partial_m B_l\epsilon_{klm})B_j\epsilon_{ijk}\right)-u_k B_j\epsilon_{ijk}\right)\to 0 \tag{6.36}$$

where $\boldsymbol{E}$ in Eq. (6.36) is given by Eq. (2.73). While this form is mathematically exact, it is computationally expensive. Additionally, some terms are redundant. These relations are useful for understanding which local parameters are important, though. For example, local charge imbalance becomes a problem on the scale of

$$\left(\frac{L}{\delta_p}\right)\rho_c E_i\to 0 \tag{6.37}$$

This demonstrates that a useful charge neutrality metric is not dependent on any local gradients. On the other hand, the single temperature assumption is dependent on gradients. Namely,

$$\partial_i\left(\frac{j_j}{\rho}\left(\frac{A^{(e)}}{Z^{(e)}}\left(e^{(i)}+P_{ij}^{(i)}\right)+\frac{A^{(i)}}{Z^{(i)}}\left(e^{(e)}+P_{ij}^{(e)}\right)\right)\right)\to 0 \tag{6.38}$$

This can be further reduced to

$$\frac{\gamma}{\gamma-1}\partial_i\left(\frac{j_i}{\rho}\left(\frac{A^{(e)}}{Z^{(e)}}P^{(i)}+\frac{A^{(i)}}{Z^{(i)}}P^{(e)}\right)\right)\to 0 \tag{6.39}$$

It is unclear what reduced metrics are useful for identifying if finite electron momentum effects are significant. One possibility comes from Eq. (6.32):

$$\frac{A^{(i)}A^{(e)}}{Z^{(i)}Z^{(e)}}\partial_j\left(\frac{j_i j_j}{\rho^2}\right) \to 0 \tag{6.40}$$

Another possible metric comes from examining Eq. (6.36). In the raw form it doesn't provide any significantly useful insight. However, consider what differences exist if we only apply the charge quasi-neutrality assumption to Eq. (6.8). Equation (6.36) then reduces to

$$\begin{aligned}\frac{1}{Z^{(e)}n^{(e)}}\left(\frac{\delta_p}{L}\right)\Bigg(&\frac{\partial_t j_i + \partial_j\left(\frac{Z^{(i)}}{A^{(i)}}p_i^{(i)}u_j^{(i)} + \frac{Z^{(e)}}{A^{(e)}}p_i^{(e)}u_j^{(e)}\right)}{\frac{Z^{(e)}}{A^{(e)}} - \frac{Z^{(i)}}{A^{(i)}}} + \frac{A^{(e)}Z^{(i)}}{A^{(i)}Z^{(e)} - A^{(e)}Z^{(i)}}\partial_j\left(P_{ij}^{(i)} + P_{ij}^{(e)}\right)\Bigg) \\ &+ \frac{B_j\epsilon_{ijk}}{Z^{(e)}n^{(e)}}\left(\frac{A^{(e)}Z^{(i)} + A^{(i)}Z^{(e)}}{A^{(e)}Z^{(i)} - A^{(i)}Z^{(e)}}j_k + \left(\frac{\delta_p}{L}\right)\partial_m B_l\epsilon_{klm}\right) \to 0\end{aligned} \tag{6.41}$$

Since the Generalized Ohm's Law equation used already takes into account the Hall effect, taking finite electron mass effects into account for this term is effectively a constant per species:

$$\frac{A^{(e)}Z^{(i)} + A^{(i)}Z^{(e)}}{A^{(e)}Z^{(i)} - A^{(i)}Z^{(e)}} \to -1 \tag{6.42}$$

Unless the plasma is expected to contain muons or positrons this metric is expected to evaluate to -0.99 or closer to -1 when the low-frequency Ampere's law is a reasonable approximation. Similarly, the diamagnetic electron pressure term is typically also modeled with Hall MHD (diamagnetic ion pressure is typically neglected). If the divergence of the total pressure tensor is significant, finite electron mass effects may influence the diamagnetic pressure term. This produces a metric

$$\frac{1}{Z^{(e)}n^{(e)}}\left(\frac{\delta_p}{L}\right)\frac{A^{(e)}Z^{(i)}}{A^{(i)}Z^{(e)} - A^{(e)}Z^{(i)}}\partial_j\left(P_{ij}^{(i)} + P_{ij}^{(e)}\right) \to 0 \tag{6.43}$$

The remaining terms are entirely correlated to finite electron mass effects which are neglected by MHD-type models; that is

$$\frac{A^{(e)}A^{(i)}}{Z^{(e)}n^{(e)}(A^{(i)}Z^{(e)} - A^{(e)}Z^{(i)})}\left(\frac{\delta_p}{L}\right)\left(\partial_t j_i + \partial_j\left(\frac{Z^{(i)}}{A^{(i)}}p_i^{(i)}u_j^{(i)} + \frac{Z^{(e)}}{A^{(e)}}p_i^{(e)}u_j^{(e)}\right)\right) \to 0 \tag{6.44}$$

Metrics based on Eqs. (6.43) or (6.44) both are relatively difficult to evaluate, and it's not immediately clear if they provide any advantages over Eq. (6.40) for evaluating if finite electron mass effects are significant or not.

A minimum set of metrics based on Eqs. (6.37), (6.39), and (6.40) may be sufficient since they capture the assumptions of charge neutrality, equal species temperature, and finite electron mass effects. Other interesting metrics might also be to examine the relationship between various plasma timescales; for example, the Alfvén velocity describes the propagation of a wave coupling the ions to the magnetic field. Comparing this to the speed of light,

$$\frac{v_A}{c_0} = \frac{1}{c_0}\sqrt{\frac{B^2}{\rho}} \to 0 \tag{6.45}$$

in the MHD limit. When this ratio approaches or exceeds 1, then it is likely that the dynamics of Alfvén wave may begin to interact strongly with two-fluid dynamics such as the lower and upper hybrid plasma frequencies. Other similar characteristic dynamics of MHD systems are the magnetosonic waves. Examining the ratio of these waves to the lower and upper hybrid waves may provide similar insight between the region of validity of MHD and two-fluid models.

As a generalization, examining the collision frequency vs. dynamics on the Alfvén/magnetosonic timescales can give insight into whether the plasma is behaving collisionless (collision timescales significantly slower), collisionally (collision timescales with a similar magnitide), or inviscid (collision timescales significantly larger). This could give insight into the importance of including collision terms, or even indicate that a 5-moment fluid model may be inadequate for describing the dominant dynamics.

Chapter 7

Computational framework

The WARPXM (Washington Approximate Riemann Plasma eXtended modeling platform-Many Core version) was used for testing of the anisotropic Braginskii model and preliminary work investigating methods for coupling single temperature MHD and the ideal two-fluid system. For details on WARPXM's structure and framework, see Sean Miller's dissertation[56].

The HDG and adaptive model were implemented in a separate C++ demonstration code with MPI-based CPU parallelization, along with supporting Python tools used for generating C++ code. The details of this implementation are described in the sections below.

7.1 Meshing and partitioning

A custom HDF5-based mesh file format for representing unstructured hexahedral meshes was developed with various additional annotation information which made implementing the multi-model HDG method easier. These mesh state files also double as state files for checkpointing and restarting the simulation. A tool capable of generating a hexahedral mesh was also developed with uniform spacing in each dimension. Periodic boundaries are considered topologically.

The primary motivation for developing the new mesh format is to include necessary annotation

information such as element to face, element to element, and unique face and element ID's required for HDG that would be difficult to produce otherwise.

The mesh format also utilizes two distinct coordinate systems and includes necessary orientation information to easily transform between the two systems when necessary. The first coordinate system is the general global XYZ Cartesian coordinate system. All element and face geometry vertex positions are stored in reference to this coordinate system. The second coordinate system is a face-local normal-binormal-tangent NBT coordinate system which is unique to each face.

Every face has a unique NBT coordinate system, and neighboring elements both utilize the face's NBT coordinate system for the purposes of computing numerical fluxes. Note that a natural consequence of this is that the outwards normal for one element will be $\widehat{n} = (1, 0, 0)$ while for the other element it is $\widehat{n} = (-1, 0, 0)$ in face NBT space.

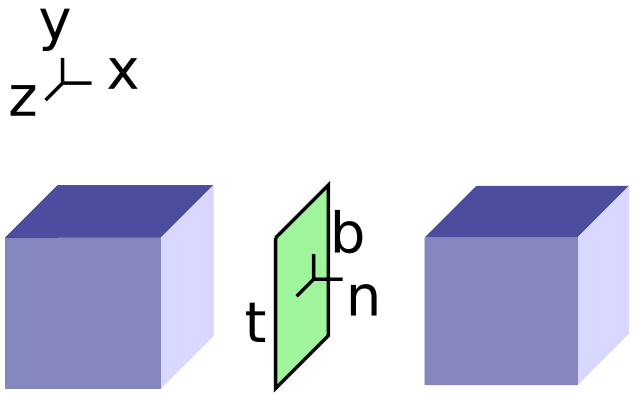


Figure 7.1: Global and face coordinate systems for two elements and an intermediate x-oriented face.

The reason for utilizing the NBT coordinate system for numerical fluxes can be demonstrated by examining the numerical flux associated with Laplace's equation:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{7.1}$$

$$\boldsymbol{u} = \nabla q \tag{7.2}$$

$$(F\widehat{n})^* = \begin{bmatrix} \boldsymbol{u}_\lambda \cdot \widehat{n}^- + \tau(q^- - q_\lambda) \\ -q_\lambda \overleftrightarrow{\boldsymbol{I}}\, \widehat{n}^- + \tau(\boldsymbol{u}^- - \boldsymbol{u}_\lambda) \end{bmatrix} \tag{7.3}$$

In the global coordinates $\widehat{n}^-$ generally has 3 non-zero components, so $\boldsymbol{u}_\lambda \cdot \widehat{n}^-$ requires knowing all 3 components of $\boldsymbol{u}_\lambda$. however, in NBT space $\widehat{n}^-$ is guaranteed to have the last 2 components be zeroes, so the numerical flux reduces to

$$(F\widehat{n})' = \begin{bmatrix} \boldsymbol{u}_{\lambda,n} n^- + \tau(q^- - q_\lambda) \\ \begin{bmatrix} -qn^- + \tau(u_n^- - u_{n,\lambda}) \\ \tau(u_b^- - u_{b,\lambda}) \\ \tau(u_t^- - u_{t,\lambda}) \end{bmatrix} \end{bmatrix} \tag{7.4}$$

Notice that $u_{b,\lambda}$ and $u_{t,\lambda}$ are effectively just superfluous degrees of freedom, with the last 2 numerical flux terms consisting only of a penalty term. It's not necessary to keep these penalties, though, and these two variables can be dropped entirely from the global system:

$$(F\widehat{n})'' = \begin{bmatrix} \boldsymbol{u}_{\lambda,n} n^- + \tau(q^- - q_\lambda) \\ \begin{bmatrix} -qn^- + \tau(u_n^- - u_{n,\lambda}) \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} \tag{7.5}$$

Partitioning of the mesh is performed in parallel using ParMETIS[57], and is treated as a single global partition with hexahedral elements connected to faces. This has a few key consequences:

- It is possible for an MPI rank to end up owning 0 elements, 0 faces, or 0 elements and 0 faces.
- There is no guaranteed sub-grouping continuity of the system being solved in elements/faces. For example, it is perfectly feasible for the elements of a rank to be [MHD, TF, MHD, TF, TF, MHD, MHD], or any other permutation.

Note that any additional degrees of freedom (for example an external resistor voltage drop) are assigned to the last rank.

7.2 Adaptive model and solver infrastructure

To handle an adaptive multi-model system what model is being solved in each element or face is tracked using a per-element ID and per-face ID. This ID can be changed at any time, with a minimal amount of overhead and no re-partitioning. Every element/face is allocated space capable of storing the number of degrees of freedom (DOFs) required to handle the most expensive model. If the model utilized in that element/face requires fewer DOFs, the remaining space is left unutilized. Note that this is *not* true for the global Schur system solve; the global Schur matrix and associated solution/residual vector are always compacted to remove unused DOFs. This is done as a limitation of the PETSc library used for global system solves, since it is not possible to ignore these DOFs without introducing some extra trivial equation setting these unused DOFs to 0. Minimizing the size of the global system solve has a significant impact on the global solve speed, so this is considered an acceptable trade off. Note that because the per-element/face DOFs are already compacted contiguously inside of the element/face, local solves also do not have to introduce any extra trivial equations to handle the memory padding.

The general procedure for implementing the MHD/two-fluid adaptive remapping is:

```
function ADAPT MODEL
    for each element do
        if element needs to change model (dictated by appropriate metrics) then
            Remap element variables to the new model
            Change element model ID
        end if
    end for
    Synchronize element model ID's to all processes
    for each non-boundary face do
        if either neighboring element is two-fluid then
            Remap face variables to two-fluid
            Change face model ID to two-fluid
        else
            Remap face variables to MHD
            Change face model ID to MHD
        end if
    end for
    if Any face model IDs were changed then
        Update datastructures associated with the global system solve
    end if
end function
```

7.3 Code generation

One major drawback with utilizing an HDG scheme is the construction of the Schur complement and global system solve. Unfortunately, the novelty of the HDG method makes it difficult to utilize existing tools designed for implementing implicit finite element methods, while at the same time properly building the Schur complement for the two-fluid system requires generating 4 large distinct matrices for the Schur complement, with block sizes of 51×51, 51×41, 41×51, and 41×41 even before considering the variations required to couple with MHD and boundary conditions. While these matrices are somewhat sparse (for example $\overset{\leftrightarrow}{\boldsymbol{A}}$ has 51×51 blocks, but only about 277 nonzeros per mode, or about 10% non-zero), constructing this matrix is a large endeavor and very resistant to identifying bugs when partially correct. Luckily there exist a wide variety of computer algebra tools capable of analytically computing these matrices given relatively simple inputs such as the residual expressions.

For the purposes of this research a Python-based HDG framework utilizing SymPy[58] was used to generate the code required for implementing an implicit non-linear HDG solver. This combination of tools was chosen due to the degree of customizability and flexibility required to produce a near fully functional C++ program, with some additional general numerical method library code to handle meshing, partitioning, numerical quadrature, and other useful general tools. The capabilities of the implemented Python code generation tool are:

- Represent sources, internal flux terms, and numerical flux terms analytically on multiple element model ID's and multiple face model ID's.
- Computing HDG numerical fluxes given the internal flux tensor.
- Implement tools for rotating tensors from the global coordinate system to the face NBT coordinate system and vice-versa.

- Identify what degrees of freedom are used, and eliminate any unused DOFs.
- Identify any equations which are unused/zero and eliminate them from the generated source.
- Compute the required HDG Jacobian matrices analytically.
- Generate sources for a complete C++ class capable of computing specified initial conditions, compute the system residual, constructing the Schur complement, performing the non-linear HDG solve procedure, and producing desired post-processing outputs to ParaView Catalyst.
- Perform some mathematical simplifications to reduce the computational effort required to compile and execute.

While the tool is quite capable, for large systems it can take a significant amount of time to run, potentially near indefinitely as it tries to simplify very long and complex expressions. One example of this is that it does not seem to be possible to generate the coupled two-fluid/mhd system with variable viscosity, heat flux, and resistivity. The generated expressions also suffer from some of the code bloat associated with using computer algebra systems on complex expressions since nearly all terms are included inline. The latter problem can potentially be solved by utilizing common subexpression elimination and manually pre-computing some values outside of the computer algebra system, though that remains an area of future investigation.

7.4 Simplified dissipation models

To assist the HDG code generation tool, simplified dissipation models using quasi-constant coefficients for viscosity, heat flux, interspecies momentum and energy transfer, and resistivity can be used. Attempting to use a properly scaled model causes an excessive number of terms to be generated, leading to either the symbolic simplification code to stall. It is unclear how much of this is a

limitation of the SymPy library, and how much of it is inherent to trying to simplify such complex expressions.

This simplified model uses

$$\overleftrightarrow{\boldsymbol{\Pi}}_\alpha = -\mu_c \sqrt{A_\alpha}\, \overleftrightarrow{\boldsymbol{W}}_\alpha \tag{7.6}$$

$$\boldsymbol{h}_\alpha = -\frac{\kappa_c}{\sqrt{A_\alpha}} \nabla T_\alpha \tag{7.7}$$

$$\boldsymbol{R}_{\alpha\beta} = R_c \rho_e (u_\beta - u_\alpha) \tag{7.8}$$

$$Q_{\alpha\beta} = Q_c (T_\beta - T_\alpha) \tag{7.9}$$

$$\boldsymbol{E} = \eta_c \boldsymbol{j} - \boldsymbol{u} \times \boldsymbol{B} \tag{7.10}$$

where μ_c, κ_c, R_c, Q_c, and η_c are constants.

Chapter 8

HDG method numerical performance

Three linear test problems were chosen to test the convergence rates of the HDG method. These problems were chosen to excercise the different numerical terms used. A 1D Maxwell's plane wave propagation is then used to study the spectrum of numerical accuracy over various spatial and temporal resolutions. To validate the non-linear plasma models, a quasi-1D magnetized Brio-Wu shock tube along with a shallow unmagnetized Langmuir wave propagation problem are used. These problems are also used to study the behavior of the various divergence cleaning methods.

8.1 Linear advection

To validate the basic HDG solver infrastructure, a 3D linear advection problem on the periodic domain $\boldsymbol{r} \in [0,1]^3$ is tested.

$$\partial_t q + \nabla\cdot(\boldsymbol{a}q) = 0 \tag{8.1}$$

$$\boldsymbol{a} = \begin{bmatrix} 1 & \frac{1}{2} & 2 \end{bmatrix} \tag{8.2}$$

$$q_a = \cos(2\pi(x - a_x t))\cos(2\pi(y - a_y t))\cos(2\pi(z - a_z t)) \tag{8.3}$$

This was solved using the DIRK3 implicit temporal solver with $\Delta t = 5 \cdot 10^{-3}$ and $t \in [0, 0.1]$. Eight Gauss-Lobatto quadrature points per dimension were used to handle the numerical quadrature and L-2 error integral

$$\epsilon = \sqrt{\int (q - q_a)^2 dV} \tag{8.4}$$

Figure 8.1 demonstrates that the method does converge at optimal rates, with $N = 4$ rapidly reaching the error floor resulting from time step integration errors.

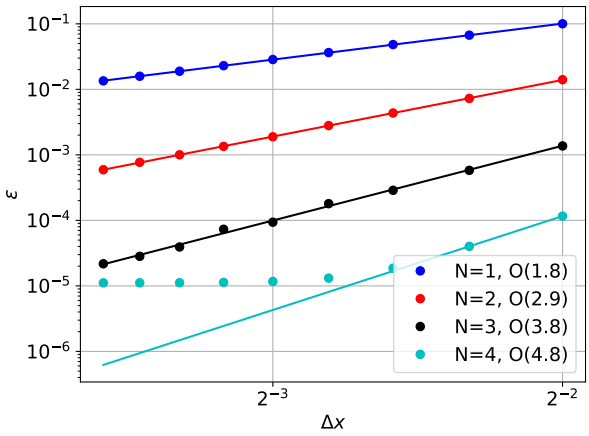


Figure 8.1: L-2 error for q converges at optimal $O(N + 1)$ rates for degree N basis.

8.2 Linear diffusion

To validate that gradient terms and mixed differential algebraic systems can be solved correctly, a 3d linear diffusion equation problem on the periodic domain $\boldsymbol{r} \in [0, 1]^3$ is tested.

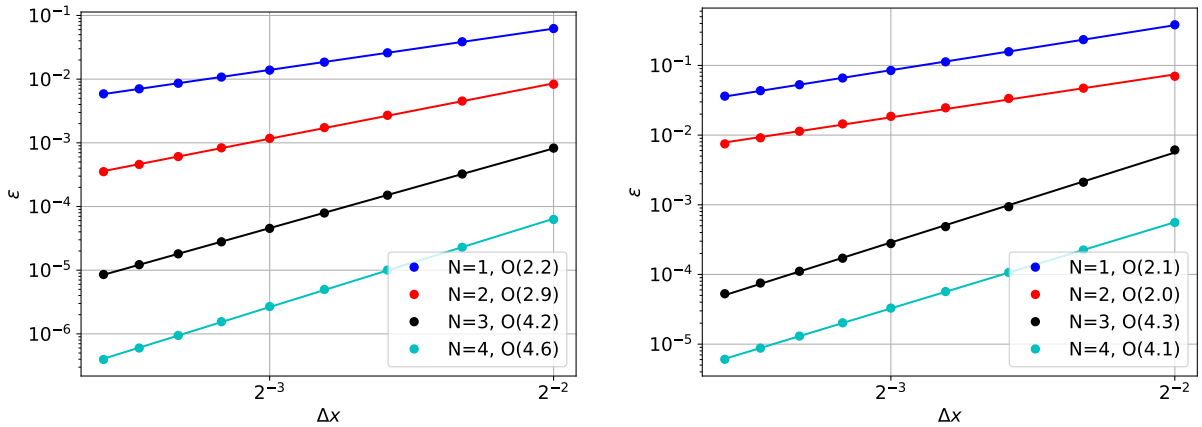
$$\partial_t q - \nabla \cdot (k\boldsymbol{\sigma}) = 0 \tag{8.5}$$

$$\boldsymbol{\sigma} - \nabla q = 0 \tag{8.6}$$

$$k = 10^{-2} \tag{8.7}$$

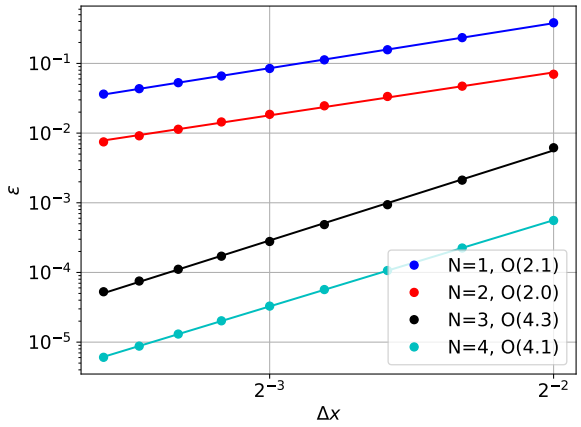
$$q_a = \exp(-3(2\pi)^2 kt) \sin(2\pi x) \sin(2\pi y) \sin(2\pi z) \tag{8.8}$$

This was solved using the DIRK3 implicit temporal solver with $\Delta t = 5 \cdot 10^{-3}$ and $t \in [0, 0.1]$. Eight Gauss-Lobatto quadrature points per dimension were used to handle the numerical quadrature and L-2 error integration. Figure 8.2 demonstrates that the method does converge at optimal rates for q, however the gradients $\boldsymbol{\sigma}$ has an oscillatory odd-even convergence rate between $O(N+1)$ and $O(N)$. It is suspected that this behavior is a result of the numerical flux form chosen for HDG, resembling similar convergence rates comparable to a central flux for DG methods[25]. While not optimal for all degrees N, the simplicity of the numerical flux form has significant advantages for implicitly solved systems by making computing the analytical Jacobian easier and have better numerical properties with respect to the linear/nonlinear solver.

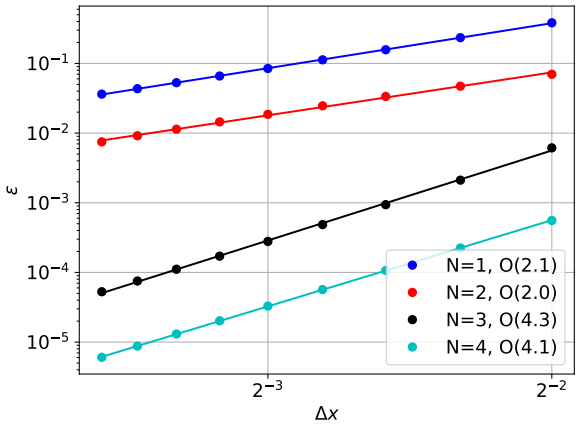


(a) q

(b) σ_x



(c) σ_y



(d) σ_z

Figure 8.2: L-2 error for q converges at $O(N+1)$, while $\boldsymbol{\sigma}$ has an odd-even oscillation between $O(N+1)$ and $O(N)$ convergence rates.

8.3 Wave equation

There are several ways the wave equation could be reduced from a second order PDE to a coupled system of first order PDE's. Two such methods are tested:

1. The first approach is to naively define the derivatives in space and time using auxiliary

variables, then re-write the wave equation in terms of these auxiliary variables.

$$\partial_t q - v = 0 \tag{8.9}$$

$$\boldsymbol{\sigma} - \nabla q = 0 \tag{8.10}$$

$$\partial_t v - \nabla\cdot(c^2\boldsymbol{\sigma}) = 0 \tag{8.11}$$

Note that a downside of this approach is that it requires additional equations to solve for λ_q since there are no spatial derivatives in Eq. (8.9). One option is to solve

$$\partial_t \lambda_q - \lambda_v = 0 \tag{8.12}$$

on the faces. Another option is to require that

$$\int_{\partial\Omega} \tau(q^- - \lambda_q)\phi dS = 0 \tag{8.13}$$

For the purposes exercising a numerically distinct method, the first approach is used.

2. The second approach builds off of the previous definition, then notice that by taking solving for $\boldsymbol{\sigma}$ a temporal differential equation allows λ_q to be removed from the degrees of freedom entirely, though at the cost of requiring $\boldsymbol{\sigma}$ to be solved as a temporal differential variable instead of as an algebraic variable.

$$\partial_t q - v = 0 \tag{8.14}$$

$$\partial_t \boldsymbol{\sigma} - \nabla v = 0 \tag{8.15}$$

$$\partial_t v - \nabla\cdot(c^2\boldsymbol{\sigma}) = 0 \tag{8.16}$$

This approach also does not require any new numerical infrastructure to handle how the global system solve is constructed beyond what's required for the linear advection equation.

This approach can be further re-arranged to eliminate solving for v entirely as

$$\partial_t \boldsymbol{q} - \nabla\cdot(c^2\boldsymbol{\sigma}) = 0 \tag{8.17}$$

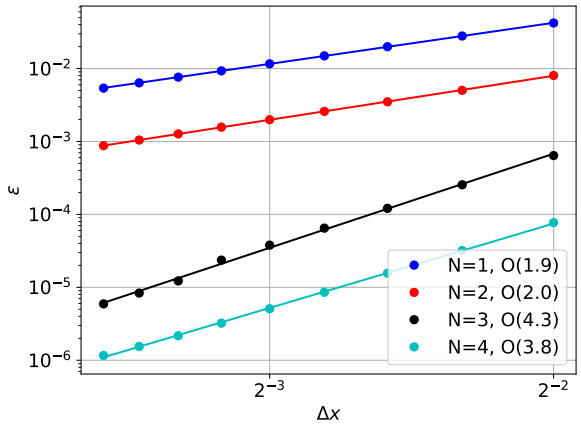
$$\partial_t \boldsymbol{\sigma} - \nabla\, q = 0 \tag{8.18}$$

which has a convenient symmetric form. Note that this latter form was not tested for convergence rates as it exercises a subset of the numerical tools required for the form which solves for v as well.

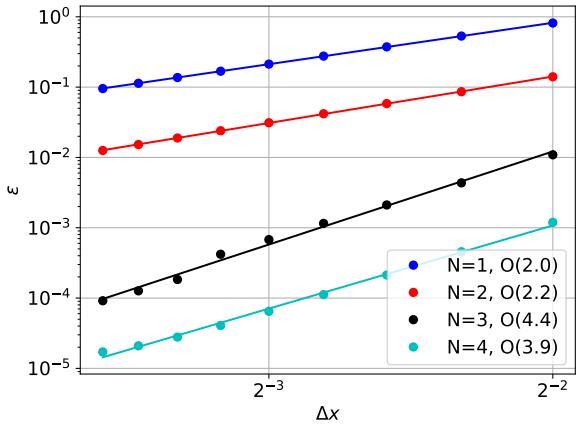
Both methods were tested using a DIRK3 temporal solver with $\Delta t = 5\cdot 10^{-3}$ and $t \in [0, 0.1]$ on a periodic domain with $\boldsymbol{r} \in [0,1]^3$. The expected solution is constructed to be

$$q_a = \cos\left(2\pi t\sqrt{3c^2}\right)\sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \tag{8.19}$$

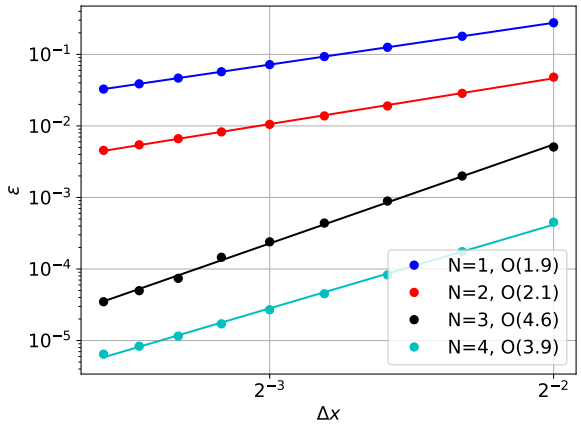
Method 1's convergence rates for all variables produces an odd-even oscillation between $O(N+1)$ and $O(N)$. It is suspected that the numerical flux is the source of this behavior in $\boldsymbol{\sigma}$, then the ODE integration for q results in a propagation of this convergence rate to all the other variables.



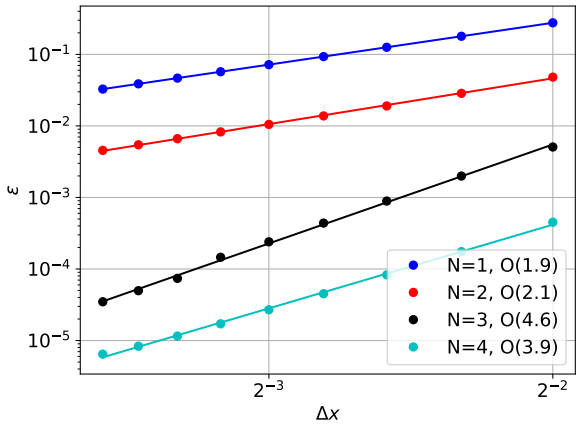
(a) q



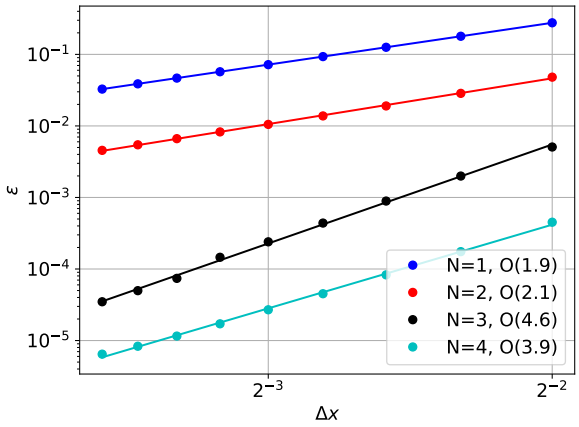
(b) $dqdt$



(c) σ_x



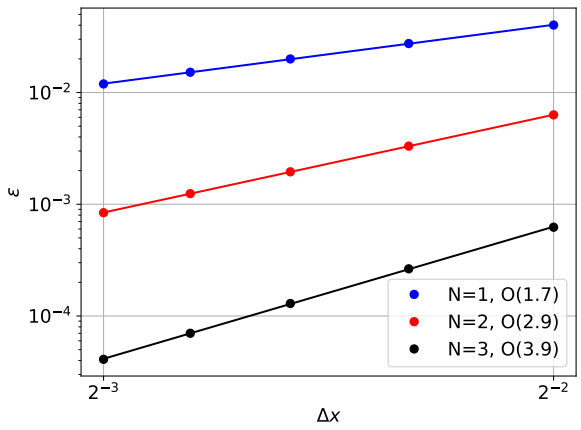
(d) σ_y



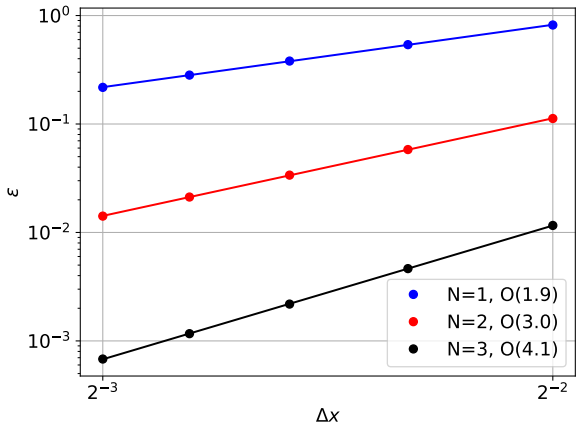
(e) σ_z

Figure 8.3: L-2 error for method 1. All variables converge with an odd-even $O(N+1)$ and $O(N)$ behavior.

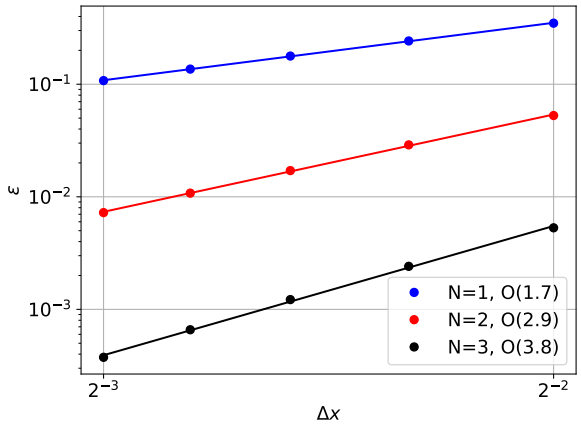
Method 2 converges at $O(N+1)$ for all variables, since it is effectively just a coupled system of linear advection equations with an ODE integration to independently solve for q.



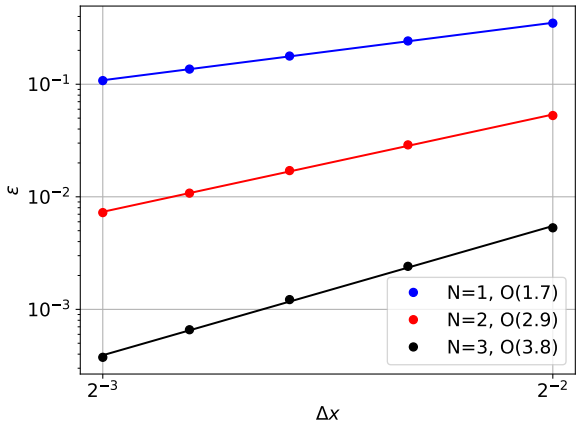
(a) q



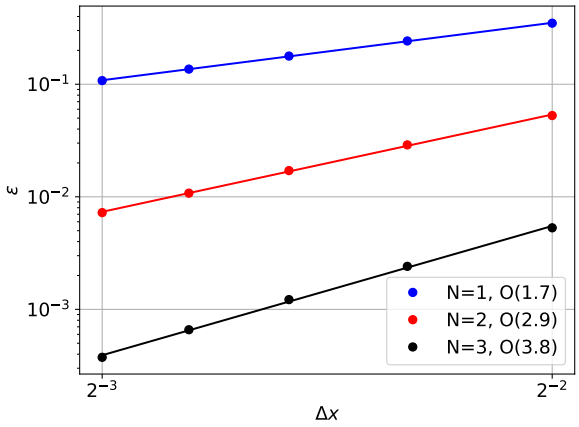
(b) $dqdt$



(c) σ_x



(d) σ_y



(e) σ_z

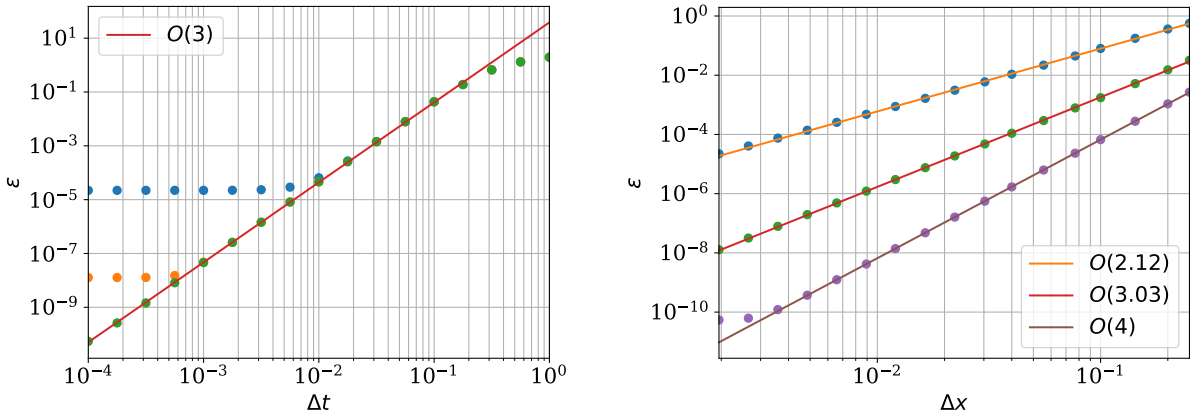
Figure 8.4: L-2 error for method 2. All variables converge at the optimal rate of $O(N+1)$.

8.4 Plane wave propagation

A 1D plane wave is initialized in a periodic domain with $x \in [0, 1]$ such that

$$E_y = E_z = B_y = B_z = \sin(2\pi x), \qquad c_0 = 1$$

This is coupled to a DIRK temporal solver with $t \in [0, 1]$. Figure 8.5 demonstrates that keeping either Δx or Δt fixed while changing the other can be used to verify that both the temporal solver and HDG method converges at optimal rates.



(a) $\Delta x = \frac{1}{512}$ (b) $\Delta t = 10^{-4}$

Figure 8.5: Ideal convergence rates are achieve for 1D Maxwell's equations with a DIRK3 temporal solver, keeping either Δx or Δt fixed.

Notice in Fig. 8.5a that although all methods converge at the DIRK3 $O(3)$ rate, different error floors are reached depending on the spatial order of accuracy. To explore this further, a full scan over various combinations of Δx and Δt with a degree 3 basis is shown in Fig. 8.6. This demonstrates that the overall numerical error is a effectively a maximum between the spatial error and temporal error. Additionally, due to the L-stable temporal solver there are no CFL timestep stability limits allowing numerical truncation error to be the only factor to consider for choosing what spatial and temporal resolution is used.

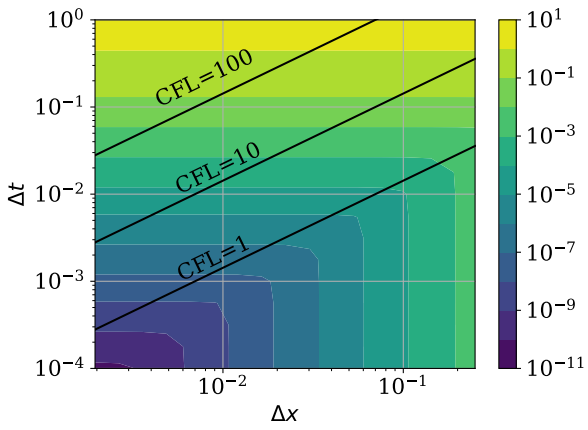


Figure 8.6: Scan of numerical error over various Δt and Δx demonstrates that the error is either dominated by spatial errors or temporal errors. Black lines of constant CFL show that the L-stable temporal solver has no CFL timestep stability dependence.

8.5 Magnetized shocktube

A magnetized Brio-Wu shocktube is setup and solved using MHD and two-fluid utilizing mixed parabolic/hyperbolic cleaning. Conditions are chosen to match Loverich et al.[59] These are:

$$n_i = n_e = \frac{0.125 - 1}{2}(\tanh(1000x) + 1) + 1$$

$$U_e = U_i = \frac{7.5 \cdot 10^{-6} - 7.5 \cdot 10^{-5}}{2}(\tanh(1000x) + 1) + 7.5 \cdot 10^{-5}$$

$$B_x = 7.5 \cdot 10^{-3}$$

$$B_y = -10^{-2} \tanh(1000x)$$

$$\frac{\delta_p}{L} = 1$$

$$\gamma = \frac{5}{3}$$

$$A_i = 1836 A_e = 1$$

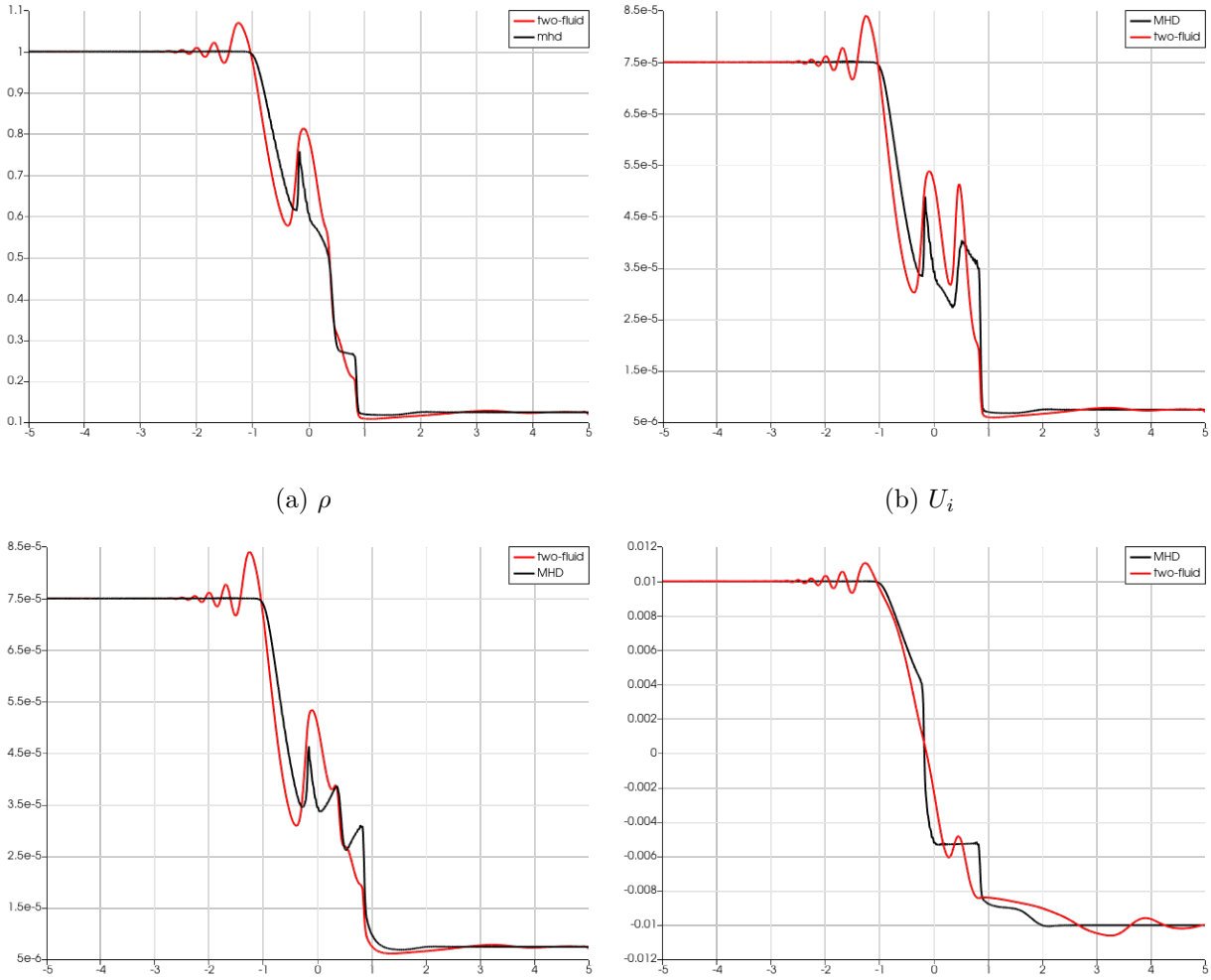
$$c_0 = c_h = c_p = 1$$

$$Z_i = -Z_e = 10$$

In addition, a small amount of dissipation is added with

$$\mu_c = Q_c = \kappa_c = R_c = 10^{-5}$$

$256 \times 1 \times 1$ parabolic elements are used with 4 quadrature points per dimension, and the simulation is run from $t \in [0, 100]$.



(a) ρ (b) U_i

(c) U_e (d) B_y

Figure 8.7: Brio-Wu plasma properties at $t = 100$. Similar features to those found in Loverich et al.[59] are present for both two-fluid and MHD. Notable differences include some smoothing near the sharp MHD discontinuities resulting from the dissipation applied, along with a discrepancy between T_i and T_e since Loverich used a single temperature MHD model.

Both MHD and the two-fluid model developed significantly different ion and electron temperatures, due to the relatively small Q_c forcing a single temperature.

8.6 Comparing divergence cleaning methods

To compare how various divergence cleaning techniques work with plasma models, a mild unmagnetized two-fluid shocktube is initialized to propagate Langmuir waves.

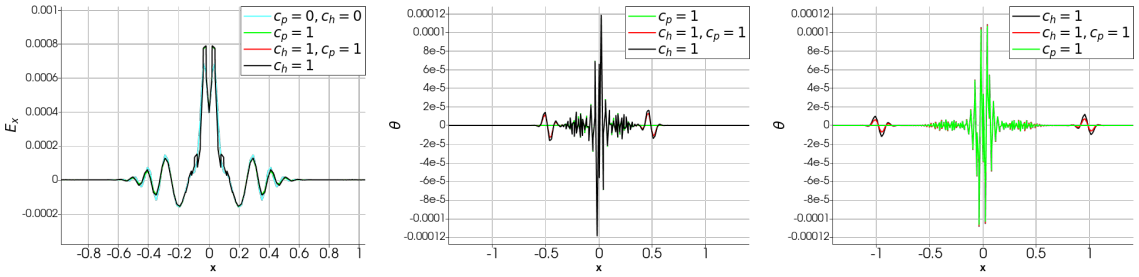
This is initialized with

$$n_L = 1.00005 \quad n_R = 0.99995 \quad U_L = 0.7500375 \quad U_R = 0.7499625 \quad \gamma = \frac{5}{3}$$

$$A_i = 1836 \quad A_e = 1 \quad c_0 = 100 \quad \frac{\delta_p}{L} = 1 \quad Z_i = 1 \quad Z_e = -1$$

A DIRK2 temporal solver with $\Delta t = 10^{-2}$ is coupled to the degree 1 Legendre HDG basis, with $x \in [-5, 5]$ and $\Delta x = \frac{10}{512}$.

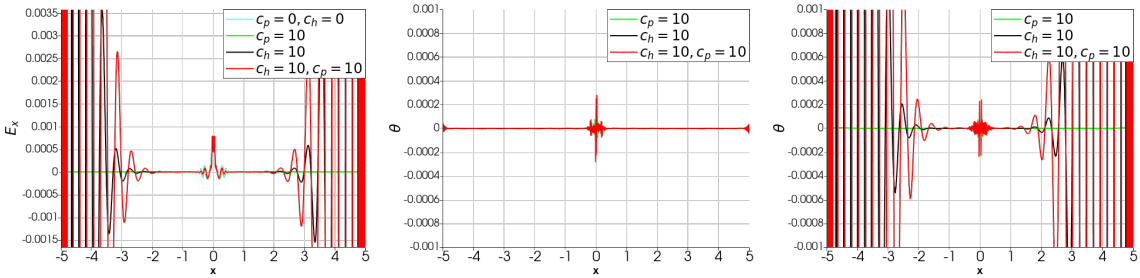
Choosing relatively modest coefficients for c_h and c_p to be 1, all the divergence cleaning methods perform similarly. Note that in these cases c_h was chosen to be significantly slower than c_0, but still faster than the Langmuir wave's propagation speed, which in this case is the practical fastest wave speed over which the fields changes. Another interesting observation is that there remains a relatively identical region of divergence error in the vicinity of the primary dynamics which doesn't directly propagate outwards when $c_h \neq 0$. This suggests that c_h isn't providing a significant divergence cleaning advantage over just parabolic cleaning. For explicit temporal solvers this is likely an acceptable or even desirable property since the CFL condition scales linearly with Δx, rather than quadratically with Δx^2 for parabolic cleaning. However, for an implicit temporal scheme it is only providing additional packets of divergence error which may interact with the plasma.



(a) E_x at $t = 1$ (b) θ at $t = 0.5$ (c) θ at $t = 1$

Figure 8.8: Modest divergence cleaning coefficients produce reasonably well behaved solutions while slowly removing numerical divergence errors.

To investigate if stronger divergence cleaning can improve the solution further, c_h and c_p are increased to be 10. Surprisingly increasing c_h or c_p significantly leads to spurious errors caused by divergence error reflecting off the boundaries, though the exact mechanism for why this leads to an amplification of errors at the boundaries is not well understood.



(a) E_x at $t = 1$ (b) θ at $t = 0.5$ (c) θ at $t = 1$

Figure 8.9: Strong divergence cleaning coefficients produce spurious numerical artifacts.

Further increasing c_h and c_p to 100 simply increases the associated problem since at this point divergence error is transiting the domain every few timesteps. These tests suggests that a relatively small amount of divergence cleaning may be desirable over stronger divergence cleaning. This has several advantages even for implicit time stepping, since even though stability is not a major

concern, the Runge-Kutta truncation error will be smaller with less divergence cleaning. Another interesting result is that for implicit time stepping schemes choosing $c_h \geq c_0$ does not seem to be a strict requirement or even always advantageous.

Chapter 9

Magnetized channel flow

Magnetized channel flow is a useful problem for validating the implementation of anisotropic viscosity present in Braginskii's model as well as higher moment fluid models such as the 13N-moment model[60]. By tuning the effective magnetization, the anisotropicity of the viscosity operator can be used to tune the shape of the boundary layer which forms. Significant deviations away from the isotropic solution are observed at moderate to high effective magnetization levels. The anisotropic magnetized channel flow solution may also be useful for comparing the Braginskii model's accuracy and validity by comparing against appropriately designed experiments.

9.1 Problem description

Consider the generalized channel flow problem shown in Fig. 9.1. There are two counter-sliding plates in the y direction along with a constant pressure gradient $\partial_y P^{(\alpha)}$ and $\partial_z P^{(\alpha)}$. A perpendicular magnetic field B_0 is applied, which induces a reaction current j_z. This produces a B_y and B_z, which in turn drive a current j_y. This setup is a generalization of the Couette flow ($\partial_y P^{(\alpha)} = \partial_z P^{(\alpha)} = 0$) and Poiseuille flow ($\boldsymbol{u}_0 = 0$). For $B_0 \neq 0$, the problem is also known as the Hartmann flow problem.

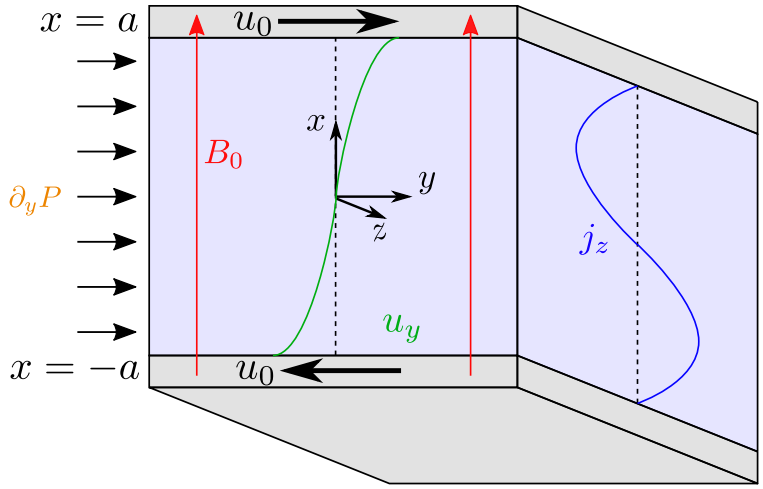


Figure 9.1: General 1D channel flow with optional transverse magnetic fields B_0, differential wall shear u_0, and inlet constant channel pressure gradient $\partial_y P$.

Take the normalized two-fluid equations in conservative form for species α, assuming the problem has reached an incompressible steady state with uniform temperature and variation of all quantities only in the x direction (except for a constant $\partial_y P^{(\alpha)}$ and $\partial_z P^{(\alpha)}$).

$$\partial_x B_x = \partial_x E_y = \partial_x E_z = \partial_x p_x^{(\alpha)} = \partial_x u_x^{(\alpha)} = \partial_x n^{(\alpha)} = 0 \tag{9.1}$$

$$\partial_x \left(P^{(\alpha)} + \Pi_{xx}^{(\alpha)} \right) = F_x^{(\alpha)} - \sum_{\beta \neq \alpha} R_x^{(\alpha\beta)} \tag{9.2}$$

$$\partial_x \left(\Pi_{xy}^{(\alpha)} \right) + \partial_y P^{(\alpha)} = F_y^{(\alpha)} - \sum_{\beta \neq \alpha} R_y^{(\alpha\beta)} \tag{9.3}$$

$$\partial_x \left(\Pi_{xz}^{(\alpha)} \right) + \partial_z P^{(\alpha)} = F_z^{(\alpha)} - \sum_{\beta \neq \alpha} R_z^{(\alpha\beta)} \tag{9.4}$$

$$F_i^{(\alpha)} = \left(\frac{L}{\delta_p} \right) Z^{(\alpha)} n^{(\alpha)} \left(E_i + u_k^{(\alpha)} B_j \epsilon_{ijk} \right) \tag{9.5}$$

$$\partial_x B_y = \left(\frac{L}{\delta_p} \right) \sum_\alpha Z^{(\alpha)} n^{(\alpha)} u_z^{(\alpha)} \tag{9.6}$$

$$-\partial_x B_z = \left(\frac{L}{\delta_p} \right) \sum_\alpha Z^{(\alpha)} n^{(\alpha)} u_y^{(\alpha)} \tag{9.7}$$

Applying conducting wall boundary conditions, then $u_x^{(\alpha)} = E_y = E_z = 0$ inside the entire domain.

The general solution follows Miller et al.[60] with a few modifications to allow constant pressure gradients and anisotropic transport properties.

9.2 Isotropic solution

In the isotropic case[61],

$$\Pi_{ij}^{(\alpha)} = \left(\frac{n^{(\alpha)}T^{(\alpha)}}{\nu^{(\alpha\alpha)}(\nu_p\tau)}\right)\left(\partial_i u_j + \partial_j u_i - \frac{2}{3}\partial_k u_k \delta_{ij}\right) = \eta^{(\alpha)}\left(\partial_i u_j + \partial_j u_i - \frac{2}{3}\partial_k u_k \delta_{ij}\right) \tag{9.8}$$

$$R_i^{(\alpha\beta)} = \rho^{(\alpha)}\nu^{(\alpha\beta)}(\nu_p\tau)(u_i^{(\beta)} - u_i^{(\alpha)}) = \sigma^{(\alpha\beta)}(u_i^{(\beta)} - u_i^{(\alpha)}) \tag{9.9}$$

Substituting into the PDE system,

$$0 = \left(\frac{L}{\delta_p}\right) Z^{(\alpha)} n^{(\alpha)}\left(u_y^{(\alpha)}B_z - u_z^{(\alpha)}B_y\right) \tag{9.10}$$

$$\eta^{(\alpha)}\partial_x\left(\partial_x u_y^{(\alpha)}\right) + \partial_y P^{(\alpha)} = \left(\frac{L}{\delta_p}\right) Z^{(\alpha)} n^{(\alpha)} u_z^{(\alpha)} B_x - \sum_{\beta\neq\alpha}\sigma^{(\alpha\beta)}(u_y^{(\beta)} - u_y^{(\alpha)}) \tag{9.11}$$

$$\eta^{(\alpha)}\partial_x\left(\partial_x u_z^{(\alpha)}\right) + \partial_z P^{(\alpha)} = -\left(\frac{L}{\delta_p}\right) Z^{(\alpha)} n^{(\alpha)} u_y^{(\alpha)} B_x - \sum_{\beta\neq\alpha}\sigma^{(\alpha\beta)}(u_z^{(\beta)} - u_z^{(\alpha)}) \tag{9.12}$$

Since Eqs. (9.11) and (9.12) only vary with $u_y^{(\alpha)}$ and $u_z^{(\alpha)}$, these equations can be solved as a couple system of ODE's. Re-writing in matrix form,

$$\partial_x^2 \boldsymbol{q} = \overleftrightarrow{\boldsymbol{A}}\,\boldsymbol{q} - \boldsymbol{P} \tag{9.13}$$

$$\boldsymbol{q}=\begin{bmatrix} u_y^{(1)} \\ u_z^{(1)} \\ u_y^{(2)} \\ u_z^{(2)} \\ \vdots \end{bmatrix}, \quad \boldsymbol{P}=\begin{bmatrix} \partial_y P^{(1)} \\ \partial_z P^{(1)} \\ \partial_y P^{(2)} \\ \partial_z P^{(2)} \\ \vdots \end{bmatrix}, \quad A_{ij}=\frac{1}{\eta^{(\alpha)}}\begin{cases} \sum_{\beta\neq\alpha}\sigma^{(\alpha\beta)} & i=j=2\alpha-1 \\ \left(\frac{L}{\delta_p}\right)Z^{(\alpha)}n^{(\alpha)}B_x & i=j-1=2\alpha-1 \\ \sum_{\beta\neq\alpha}\sigma^{(\alpha\beta)} & i=j=2\alpha \\ -\left(\frac{L}{\delta_p}\right)Z^{(\alpha)}n^{(\alpha)}B_x & i=j+1=2\alpha \\ -\sigma^{(\alpha\beta)} & i=2\alpha-1, j=2\beta-1, \beta\neq\alpha \\ -\sigma^{(\alpha\beta)} & i=2\alpha, j=2\beta, \beta\neq\alpha \\ 0 & \text{otherwise} \end{cases}$$

where $\overleftrightarrow{\boldsymbol{A}}$ is constant. Assuming that $\overleftrightarrow{\boldsymbol{A}}$ is diagonalizable as $\overleftrightarrow{\boldsymbol{A}}=\overleftrightarrow{\boldsymbol{X}}\overleftrightarrow{\boldsymbol{\Lambda}}\overleftrightarrow{\boldsymbol{X}}^{-1}$, then

$$\partial_x^2\boldsymbol{w}=\overleftrightarrow{\boldsymbol{\Lambda}}\,\boldsymbol{w}-\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{P} \tag{9.14}$$

$$\boldsymbol{w}=\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{q} \tag{9.15}$$

This process produces a set of decoupled ODE's to solve for, and has a general solution

$$w_i=\begin{cases} c_{1,i}e^{x\sqrt{\lambda_i}}+c_{2,i}e^{-x\sqrt{\lambda_i}}+\left(\overleftrightarrow{\boldsymbol{\Lambda}}^{-1}\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{P}\right)_i & \lambda_i\neq 0 \\ c_{1,i}+c_{2,i}x-\left(\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{P}\right)_i\frac{x^2}{2} & \lambda_i=0 \end{cases} \tag{9.16}$$

where the constants $c_{1,i}$ and $c_{2,i}$ can be found by applying the boundary conditions $w_i(-a)=(\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{q})_i|_{-a}$ and $w_i(a)=(\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{q})_i|_a$. The magnetic field components B_y and B_z can be found by integrating Eqs. (9.6) and (9.7) respectively. Since q_i is just a linear combination of various w_i, this integration can be performed analytically. This gives

$$\int w_i dx=\begin{cases} \frac{1}{\sqrt{\lambda_i}}\left(c_{1,i}e^{x\sqrt{\lambda_i}}-c_{2,i}e^{-x\sqrt{\lambda_i}}\right)+\left(\overleftrightarrow{\boldsymbol{\Lambda}}^{-1}\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{P}\right)_i x & \lambda_i\neq 0 \\ c_{1,i}x+c_{2,i}\frac{x^2}{2}+\left(\overleftrightarrow{\boldsymbol{X}}^{-1}\,\boldsymbol{P}\right)_i\frac{x^3}{6} & \lambda_i=0 \end{cases} \tag{9.17}$$

$$B_y=\left(\frac{L}{\delta_p}\right)\sum_\alpha Z^{(\alpha)}n^{(\alpha)}X_{2\alpha,i}\int w_i dx+C_y \tag{9.18}$$

$$B_z = -\left(\frac{L}{\delta_p}\right)\sum_\alpha Z^{(\alpha)} n^{(\alpha)} X_{2\alpha-1,i}\int w_i dx + C_z \tag{9.19}$$

In order to conserve magnetic flux as required by the conducting wall boundary conditions,

$$\int_{-a}^{a} B_y dx = 0 \tag{9.20}$$

$$\int_{-a}^{a} B_z dx = 0 \tag{9.21}$$

Solving for the integration constants C_y and C_z,

$$\int_{-a}^{a}\int w_i dx dx = \begin{cases} \frac{2(c_{1,i}-c_{2,i})\sinh\left(a\sqrt{\lambda_i}\right)}{\lambda_i} & \lambda_i \neq 0 \\ c_{2,i}\frac{a^3}{3} & \lambda_i = 0 \end{cases} \tag{9.22}$$

$$C_y = -\frac{1}{2a}\left(\frac{L}{\delta_p}\right)\sum_\alpha Z^{(\alpha)} n^{(\alpha)} X_{2\alpha,i}\int_{-a}^{a}\int w_i dx dx \tag{9.23}$$

$$C_z = \frac{1}{2a}\left(\frac{L}{\delta_p}\right)\sum_\alpha Z^{(\alpha)} n^{(\alpha)} X_{2\alpha-1,i}\int_{-a}^{a}\int w_i dx dx \tag{9.24}$$

9.3 Anisotropic Braginskii solution

In order to arrive at an analytical solution, assume that $\boldsymbol{B} \approx B_0\widehat{x}$ for the purposes of computing transport coefficients. This means that the Braginskii transport is entirely in the transverse directions. The stress tensor $\overleftrightarrow{\mathbf{\Pi}}$ simplifies to

$$\overleftrightarrow{\mathbf{\Pi}}^{(\alpha)} = \sum_{k=0}^{4} n_k^{(\alpha)}\ \overleftrightarrow{\boldsymbol{W}}_k^{(\alpha)} \tag{9.25}$$

$$\overleftrightarrow{\boldsymbol{W}}_0^{(\alpha)} = \overleftrightarrow{\boldsymbol{W}}_1^{(\alpha)} = \overleftrightarrow{\boldsymbol{W}}_3^{(\alpha)} = 0 \tag{9.26}$$

$$\overleftrightarrow{\boldsymbol{W}}_2^{(\alpha)} = -\begin{bmatrix} 0 & \partial_x u_y^{(\alpha)} & \partial_x u_z^{(\alpha)} \\ \partial_x u_y^{(\alpha)} & 0 & 0 \\ \partial_x u_z^{(\alpha)} & 0 & 0 \end{bmatrix},\quad \overleftrightarrow{\boldsymbol{W}}_4^{(\alpha)} = \begin{bmatrix} 0 & -\partial_x u_z^{(\alpha)} & \partial_x u_y^{(\alpha)} \\ -\partial_x u_z^{(\alpha)} & 0 & 0 \\ \partial_x u_y^{(\alpha)} & 0 & 0 \end{bmatrix} \tag{9.27}$$

Similarly, the interspecies momentum transfer term reduces to

$$\boldsymbol{R}^{(\alpha\beta)} = -\left(C_{1,\perp}\boldsymbol{u}_{\perp}^{(\alpha\beta)} - C_{1,\wedge}\boldsymbol{u}_{\wedge}^{(\alpha\beta)}\right) \tag{9.28}$$

$$\boldsymbol{u}_{\perp}^{(\alpha\beta)} = \begin{bmatrix} 0 \\ u_y^{(\alpha)} - u_y^{(\beta)} \\ u_z^{(\alpha)} - u_z^{(\beta)} \end{bmatrix}, \quad \boldsymbol{u}_{\wedge}^{(\alpha\beta)} = \begin{bmatrix} 0 \\ u_z^{(\beta)} - u_z^{(\alpha)} \\ u_y^{(\alpha)} - u_y^{(\beta)} \end{bmatrix} \tag{9.29}$$

Thus, the PDE system can be re-written as

$$\overleftrightarrow{\boldsymbol{B}}\,\partial_x^2\boldsymbol{q} = (\overleftrightarrow{\boldsymbol{D}} + \overleftrightarrow{\boldsymbol{F}})\boldsymbol{q} - \tilde{\boldsymbol{P}} \tag{9.30}$$

where

$$\boldsymbol{q} = \begin{bmatrix} u_y^{(i)} \\ u_z^{(i)} \\ u_y^{(e)} \\ u_z^{(i)} \end{bmatrix}, \quad \tilde{\boldsymbol{P}} = \begin{bmatrix} \partial_y P^{(i)} \\ \partial_z P^{(i)} \\ \partial_y P^{(e)} \\ \partial_z P^{(e)} \end{bmatrix}, \qquad \overleftrightarrow{\boldsymbol{B}} = \begin{bmatrix} -\eta_2^{(i)} & -\eta_4^{(i)} & 0 & 0 \\ \eta_4^{(i)} & -\eta_2^{(i)} & 0 & 0 \\ 0 & 0 & -\eta_2^{(e)} & -\eta_4^{(e)} \\ 0 & 0 & \eta_4^{(e)} & -\eta_2^{(e)} \end{bmatrix}$$

$$\overleftrightarrow{\boldsymbol{D}} = \begin{bmatrix} -C_{1,\perp} & -C_{1,\wedge} & C_{1,\perp} & C_{1,\wedge} \\ C_{1,\wedge} & -C_{1,\perp} & -C_{1,\wedge} & C_{1,\perp} \\ C_{1,\perp} & C_{1,\wedge} & -C_{1,\perp} & -C_{1,\wedge} \\ -C_{1,\wedge} & C_{1,\perp} & C_{1,\wedge} & -C_{1,\perp} \end{bmatrix}, \quad \overleftrightarrow{\boldsymbol{F}} = \left(\frac{L}{\delta_p}\right) B_0 \begin{bmatrix} 0 & Z^{(i)}n^{(i)} & 0 & 0 \\ -Z^{(i)}n^{(i)} & 0 & 0 & 0 \\ 0 & 0 & 0 & Z^{(e)}n^{(e)} \\ 0 & 0 & -Z^{(e)}n^{(e)} & 0 \end{bmatrix}$$

Multiplying by $\overleftrightarrow{\boldsymbol{B}}^{-1}$ and defining $\overleftrightarrow{\boldsymbol{A}}=\overleftrightarrow{\boldsymbol{B}}^{-1}(\overleftrightarrow{\boldsymbol{D}} + \overleftrightarrow{\boldsymbol{F}})$ and $\boldsymbol{P} =\overleftrightarrow{\boldsymbol{B}}^{-1}\tilde{\boldsymbol{P}}$, Eq. (9.30) can be re-written in a similar form as the isotropic case

$$\partial_x^2\boldsymbol{q} =\overleftrightarrow{\boldsymbol{A}}\,\boldsymbol{q} - \boldsymbol{P}$$

This can be solved in a similar fashion as the isotropic solution was solved by eigendecomposing $\overleftrightarrow{\boldsymbol{A}}$.

9.4 Comparing the isotropic and anisotropic Braginskii solutions

Before the isotropic solution derived in Section 9.2 can be compared to the Braginskii solution derived in Section 9.3, the coefficients must be appropriately normalized with each other. This is done by taking the Braginskii coefficients in the limit $\|\boldsymbol{B}\| \to 0$. This gives

$$\bar{\eta}^{(i)} = 0.957081545\eta^{(i)}, \qquad \bar{\sigma}^{(ie)} = 0.51277087\sigma^{(ie)}$$

$$\bar{\eta}^{(e)} = 0.73275862\eta^{(e)}, \qquad \bar{\sigma}^{(ei)} = 0.51277087\sigma^{(ei)}$$

The isotropic transport properties are then replaced with $\bar{\eta}^{(i)}$, $\bar{\eta}^{(e)}$, $\bar{\sigma}^{(ei)}$, and $\bar{\sigma}^{(ie)}$.

9.4.1 Differences between analytical solutions

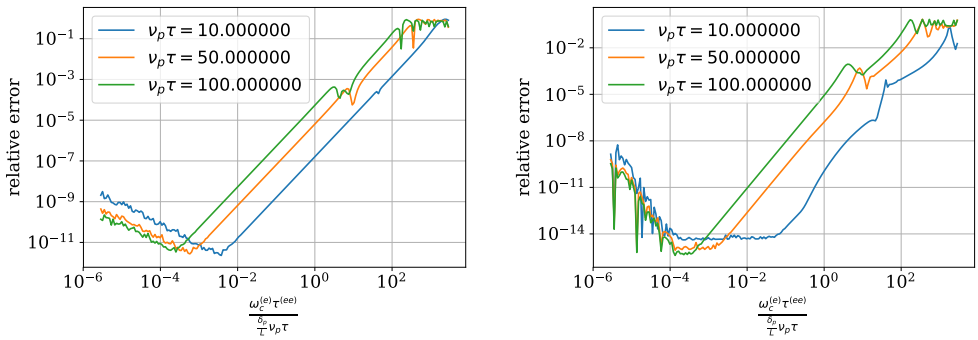
As a simple test comparison of the two analytical solutions a hydrogen plasma is placed between the plates with shear flow driven in a purely Couette fashion. The effective magnetization $\frac{\omega_c^{(e)}\tau^{(ee)}}{\frac{\delta_p}{L}\nu_p\tau}$, which compares the gyrofrequency to the collision frequency, is varied over a wide range. The test parameters are given in Eq. (9.31) and boundary conditions given in Eq. (9.32).

$$\boldsymbol{P} = 0, \quad A_i = 1, \quad A_e = \frac{1}{1836}, \quad a = \frac{1}{2}, \quad n_0 = T_0 = B_0 = 1 \tag{9.31}$$

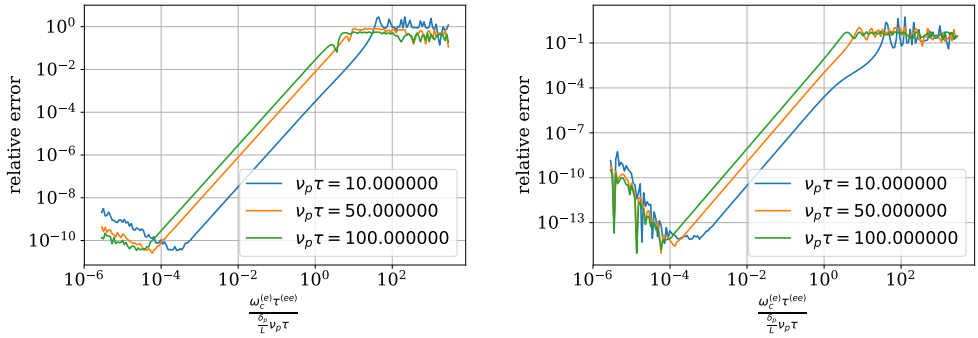
$$\boldsymbol{q}_R = -\boldsymbol{q}_L = \begin{bmatrix} 10^{-3} & 0 & 10^{-3} & 0 \end{bmatrix}^T \tag{9.32}$$

Figure 9.2 shows the relative errors, normalized vs. the maximum velocity or magnetic field

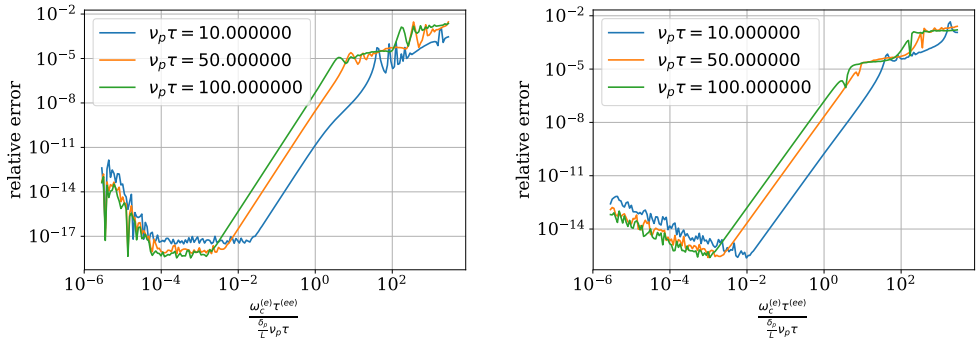
$$\epsilon_u = \sqrt{\int_{-a}^{a} \left(\frac{u_B(x) - u_I(x)}{\max(\|\boldsymbol{u}_B\|, \|\boldsymbol{u}_I\|)} \right)^2 dx}, \qquad \epsilon_B = \sqrt{\int_{-a}^{a} \left(\frac{B_B(x) - B_I(x)}{\max(\|\boldsymbol{B}_B\|, \|\boldsymbol{B}_I\|)} \right)^2 dx} \tag{9.33}$$



(a) $u_y^{(i)}$ (b) $u_z^{(i)}$



(c) $u_y^{(e)}$ (d) $u_z^{(e)}$



(e) B_y (f) B_z

Figure 9.2: Plot of the relative L-2 error measured using Eq. (9.33) vs. effective magnetization. At effective magnetization $\geq 10^{-3}$, the anisotropic Braginskii and isotropic solutions start to deviate. At very low effective magnetization the increasing system condition number produces numerical anomalies which dominate the error metric.

There are a few interesting features of note:

- At effective magnetization levels $\frac{\omega_c^{(e)}\tau^{(ee)}}{\frac{\delta_p}{L}\nu_p\tau} \geq 10^{-3}$ the isotropic solution starts to deviate from the Braginskii anisotropic solution.

- There is some variation in error just by varying $\nu_p\tau$ at the same effective magnetization.

- At very low magnetization, there is significant noise in the error. This behavior can be explained by examining the condition number of the matrix $\overleftrightarrow{\boldsymbol{A}}$. At very low magnetization, $u_z \to 0$ while u_y is still $O(u_0)$, producing increasing condition numbers. As a result at very low magnetization, the large condition numbers become a challenge to handle appropriately using finite precision arithmetic. Note that there does not appear to be any variation of the condition number vs. $\nu_p\tau$, only on effective magnetization. Thus Fig. 9.3 only shows the condition numbers for $\nu_p\tau = 100$.

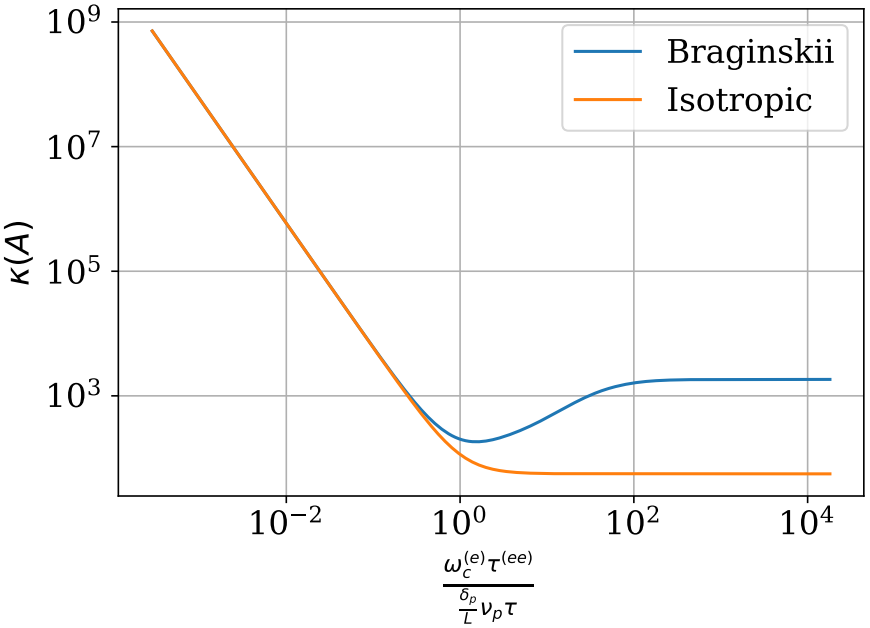


Figure 9.3: Condition number increases rapidly with decreasing effective magnetization

9.4.2 Comparing analytical solutions to simulations

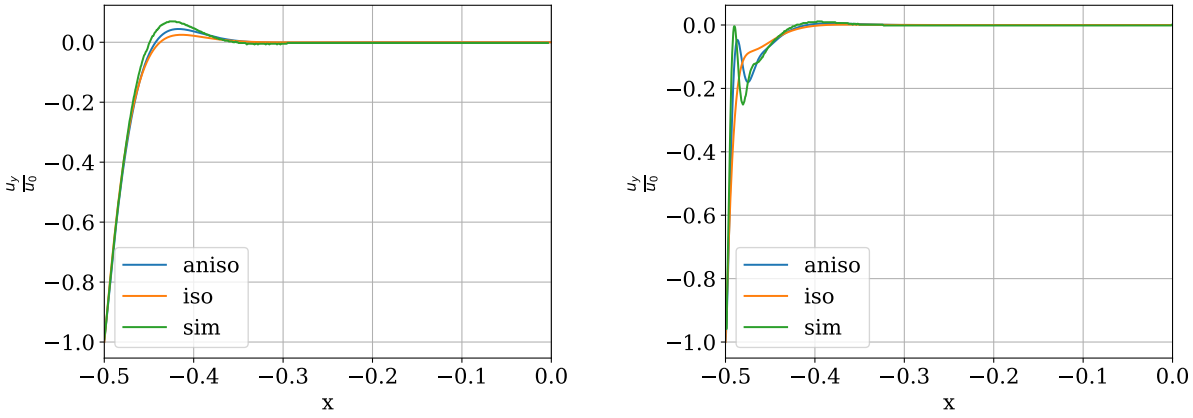
Initial simulation results from a high order K-exact finite volume implementation[56] are compared against the derived analytical solution. Simulations were run using 800 elements and a degree 3 reconstruction with a 4th order Runge-Kutta method for pseudo-timestepping[43]. The simulation was run in a pure Couette-type flow regime using the parameters given by Eq. (9.34) and boundary conditions given in Eq. (9.35).

$$A_i = 1, \quad A_e = 10^{-2}, \quad \frac{\delta_p}{L} = \frac{1}{20}, \quad \nu_p \tau = 1200 \cdot \pi^{\frac{3}{2}}, \quad a = \frac{1}{2}, \quad \gamma = \frac{5}{3}, \quad n_0 = T_0 = B_0 = 1 \tag{9.34}$$

$$\boldsymbol{q}_R = -\boldsymbol{q}_L = \begin{bmatrix} 10^{-3} & 0 & 10^{-3} & 0 \end{bmatrix}^T \tag{9.35}$$

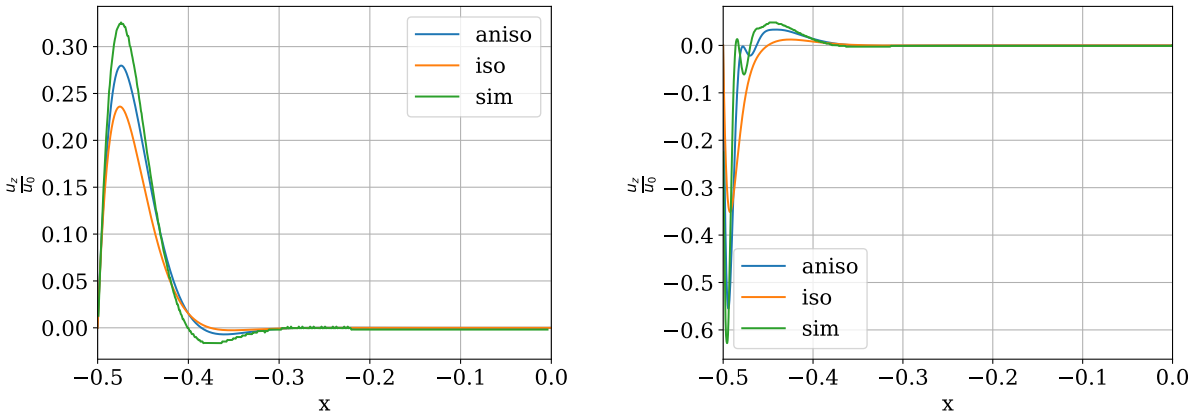
In this regime, the effective magnetization $\frac{\omega_c^{(e)} \tau^{(ee)}}{\frac{\delta_p}{L} \nu_p \tau} \approx 2$. Solutions are plotted only on the left half of the domain ($x \in [-a, 0]$) since all quantities are are either odd or even functions about $x = 0$. Note that it was verified that the simulation results satisfy this property since this property is not automatically satisfied; the analytical solutions are trivially verifiable to satisfy this property.

Figures 9.4-9.6 demonstrate that there are some features present in the simulation results which are present in the analytical anisotropic solutions, however there are also significant differences. For example, the ion flow velocity $u_y^{(i)}$ peaks to a higher value than either anisotropic solutions at $x \approx -0.42$. Similarly, the boundary layer oscillations of electron flow velocity $u_y^{(e)}$ has a similar but larger magnitude oscillation. What is causing these exaggerated simulation results compared to the anisotropic solution is still a subject of investigation. One possible theory could be due to the way boundary conditions are implemented for finite volume schemes, where an extrapolated virtual ghost cell value is imposed rather than directly setting the value at the boundary as DG schemes do. Comparisons with a DG scheme could resolve this issue.



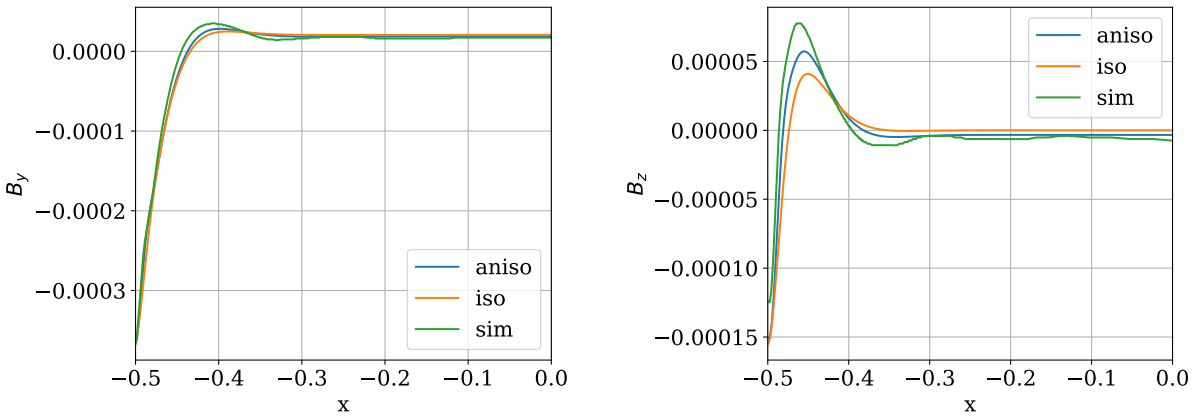
(a) Ions (b) Electrons

Figure 9.4: Comparison of u_y between HOFVM simulations, isotropic analytical solution, and anisotropic analytical solution. The simulation results have similar features to the anisotropic solution, but with larger magnitudes.



(a) Ions (b) Electrons

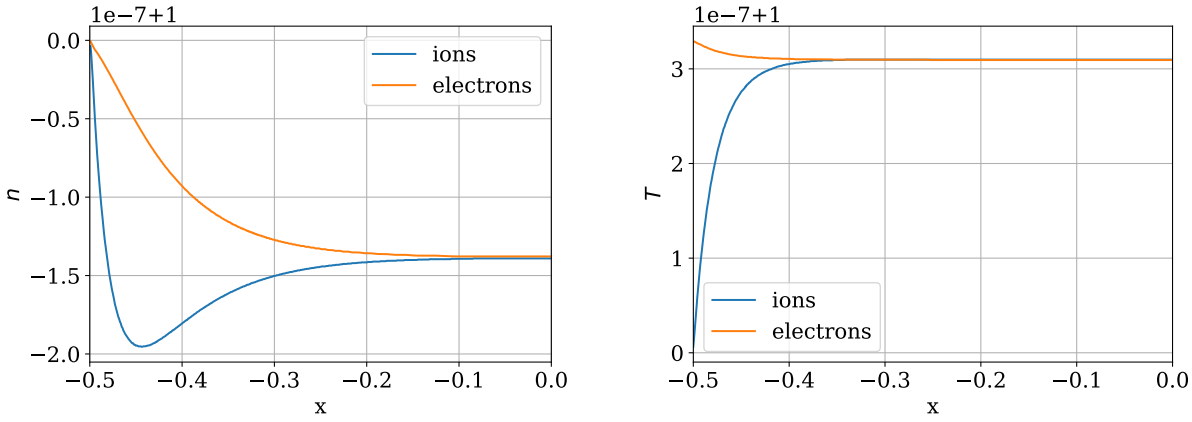
Figure 9.5: Comparison of u_z between HOFVM simulations, isotropic analytical solution, and anisotropic analytical solution. The simulation results have similar features to the anisotropic solution, but with larger magnitudes.



(a) B_y (b) B_z

Figure 9.6: Comparison of B between HOFVM simulations, isotropic analytical solution, and anisotropic analytical solution. The simulation results have similar features to the anisotropic solution, but with larger magnitudes.

Examining the number density and temperature profiles of the simulation demonstrates that the assumption these are constants are likely reasonable, however the number density does reveal some charge density build-up in the boundary layer regions. There is also a drop in the total number density. The exact cause for this drop is unknown, however it's suspected that the boundary conditions implementation may be playing some role in this.



(a) Number density

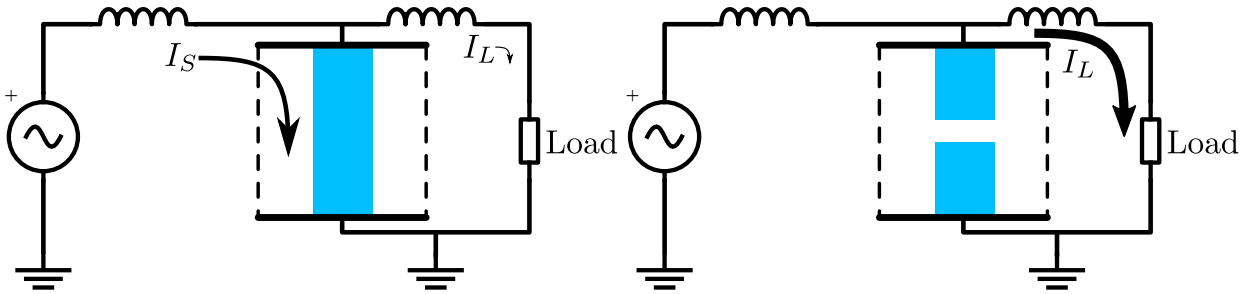
(b) Temperature

Figure 9.7: Constant number density and temperature assumptions appear to be reasonable, however there is a troubling drop in total number density.

Chapter 10

2D planar plasma opening switch

A plasma opening switch or plasma erosion opening switch (PEOS) is a device used to temporarily store inductive energy to be dumped into a target load[18]. It is intended to be used as the last stage compressor for generating a short but large pulse of energy into a target load. Figure 10.1 shows an example configuration of such a device in use. The PEOS is connected in parallel with the target load, and must have a relatively low resistance compared to the target load. This produces a configuration where it effectively short circuits the load. Over the period of approximately $1\mu s$, an external generator is used to drive a current through the PEOS. Once a desired current for the load has been reached, the PEOS is opened rapidly, forcing any current flowing through it into the load. This process is intended to occur over just a few nanoseconds, providing a large power amplification for the target load.



(a) "Closed"/charging phase (b) "Open"/discharge phase

Figure 10.1: Plasma erosion opening switch system schematic diagram

The switch opening process is dictated by the physics of the collapse/erosion of the plasma current sheet. The faster this occurs, the shorter the pulse width and conversely greater the power amplification is achieved. Similarly, spurious reconnection or partial opening of the plasma current sheet can negatively impact the performance. The plasma column must also be sufficiently stable during the charging phase to prevent premature opening.

These devices can have a co-axial or cylindrical configuration, however planar configurations have also been considered both experimentally[62] and with simulations[63, 64]. For the purposes of this research, the primary focus will be on the planar configurations to simplify the implementation. One area where previous simulations have not been able to resolve properly is the rapid penetration of magnetic fields into the plasma via plasma instability development faster than classical resistivity alone allows. Some groups[65, 66] have noted that the inclusion of Hall effects does appear to enhance the penetration rates to be potentially consistent with some experiments, but there are discrepancies with some other experiments[67, 62].

10.1 Initial simulation setup

The simulation setup of the planar plasma opening switch is shown in Figure 10.2. The domain initially contains an isobaric plasma which extends infinitely in y. There are periodic perturbations in the plasma density, which are added to induce a break-off point. At time $t = 0$, a strong magnetic field $B_z = 1$ is induced on the left boundary. The purpose of the imposed magnetic field is to induce a current density j_y in the plasma.

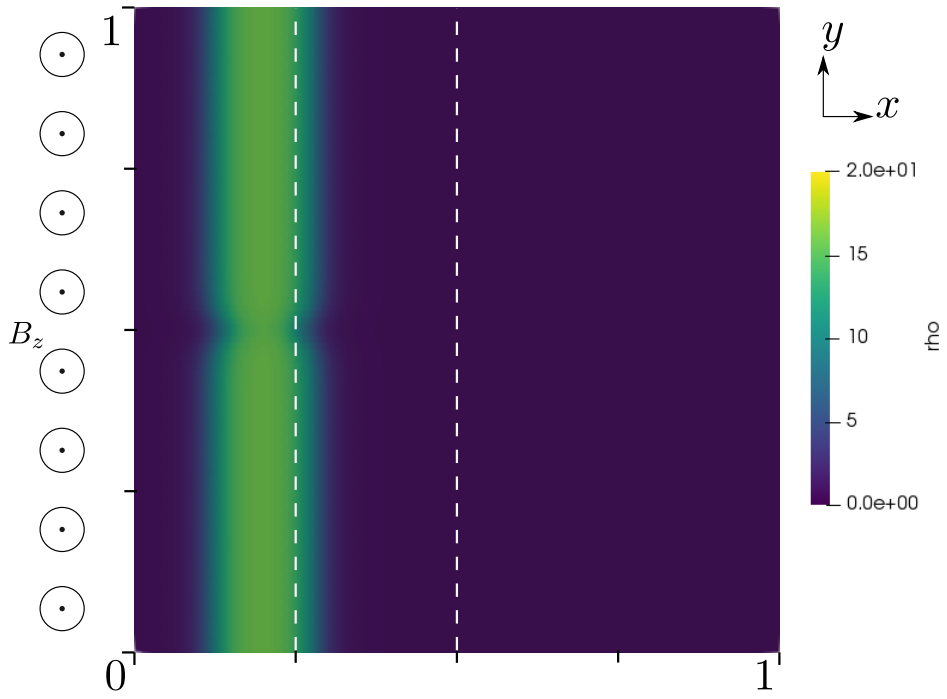


Figure 10.2: Planar plasma opening initial conditions and problem setup.

Two division marks are denoted with white dashed lines at $x = 0.25$ and $x = 0.5$. These divisions are used to delineate where each plasma model is used in any particular portion of the domain. The purpose of this initial setup is to demonstrate that the feasibility of model coupling methods described in Section 6.1. Additionally, this setup also allows the temporal re-mapping of plasma models. In the proposed work, these pre-determined division marks will eventually be removed entirely as metrics defined in Section 6.2 will be used on an element-by-element basis for

determining the plasma model partitioning. Memory for the entire simulation can be allocated during simulation initialization as one large contiguous block, which is then partitioned between the various models as needed. The parameter regime tested is

$$c_0 = 20, \quad \frac{\delta_p}{L} = 10^{-2}, \quad Z_i = 1, \quad Z_e = -1, \quad A_i = 1, \quad A_e = 10^{-2}, \quad \gamma = \frac{5}{3}, \quad P_i = P_e = \frac{1}{2}$$

The density is specified by the function

$$\rho(x,y) = \frac{b-a}{2}\left(\tanh\left(\frac{2\pi}{t_w}(x - x_c - \phi(y))\right) - \tanh\left(\frac{2\pi}{t_w}(x - x_c + \phi(y))\right)\right) + a \tag{10.1}$$

$$\phi(y) = \frac{(w_1 - w_0)\cosh\left(\frac{2\pi\left(y-\frac{1}{2}\right)}{w_y}\right)^{-2} - w_1}{2} \tag{10.2}$$

a is the background density and b is the peak density. w_1, w_0, x_c, t_w, and w_y tuning the specific shape of the pulse. For the initial simulation results these are set to

$$a = 1 \qquad b = 16 \qquad t_w = 0.2 \qquad w_y = 0.1875 \qquad w_0 = 0.1125 \qquad w_1 = 0.15 \qquad x_c = 0.2$$

10.2 Initial simulation results

Five domain configurations are tested to compare the compatibility of the two models. These are labeled as:

(a) Uniform MHD: The entire domain uses the MHD plasma model.

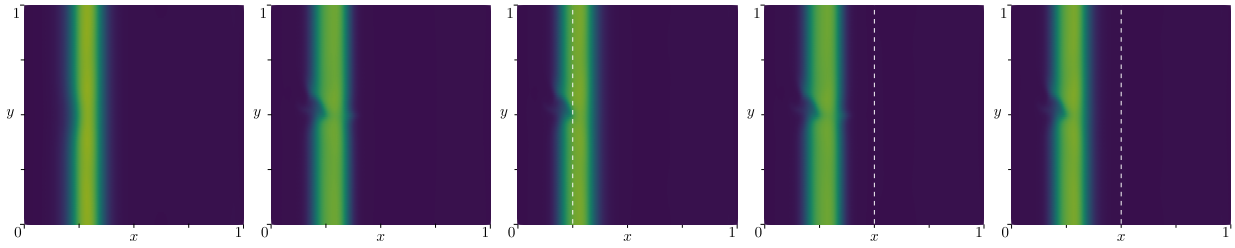
(b) Uniform two-fluid: The entire domain uses the two-fluid plasma model.

(c) Mixed.25: A two-fluid model is used in the region $x \in [0, 0.25]$ and an MHD model is used in the region $x \in [0.25, 1]$.

(d) Mixed.5: A two-fluid model is used in the region $x \in [0, 0.5]$ and an MHD model is used in the region $x \in [0.5, 1]$.

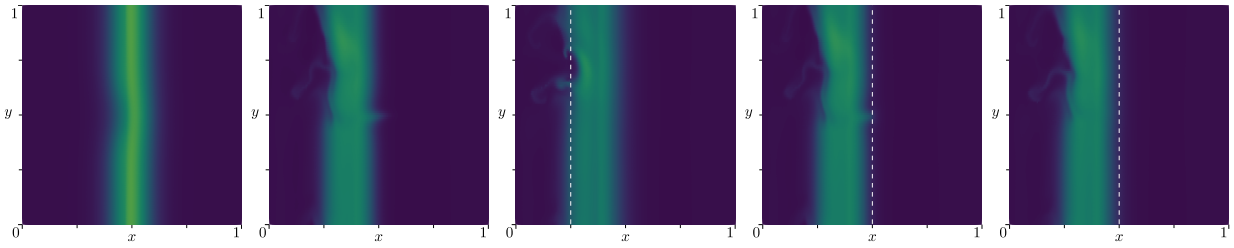
(e) Adaptive: The domain is configured with the same subdivision as the Mixed.25 case for $t \in [0, 0.8]$. At the transition time $t_c = 0.8$ the current solution is re-mapped onto the subdivision used with the Mixed.5 case and is used where $t \geq t_c$.

Figure 10.3 shows the plasma density at $t = 1$, shortly after the adaptive method has remapped the domain. An instability begins developing around where the initial perturbation was in all cases except for the uniform MHD case. At $t = 2$, Fig. 10.4 shows that the adaptive case is capable of matching the fine-scale instability structure of the Uniform two-fluid and Mixed.5 cases. However, the instabilities which developed on the two-fluid side of the Mixed.25 case balloons on the MHD side and loses the fine-scale structure. This is especially prevalent at $t = 2.5$ in Fig. 10.5. At this point the adaptive case is still capable of maintaining fine-scale instability structures on the trailing edge of the bulk plasma; however, the leading edge does not have similar instability structures as the Uniform two-fluid and Mixed.5 cases. It is suspected that this is due to the leading edge being on the MHD side initially before the adaptive method moves the model interface ahead of the leading edge.



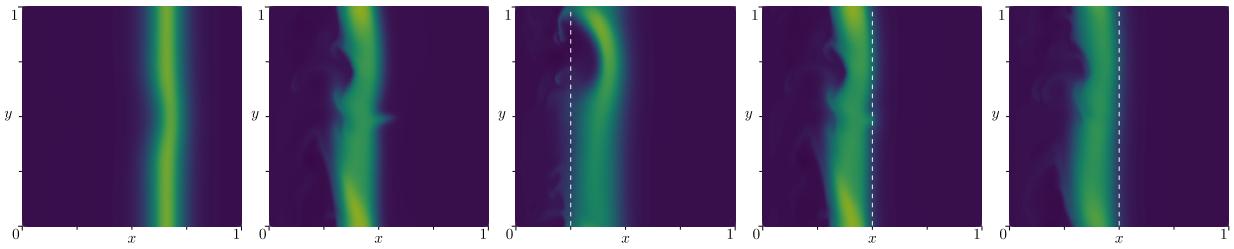
(a) MHD (b) Two-fluid (c) Mixed.25 (d) Mixed.5 (e) Adaptive

Figure 10.3: Plasma density at t=1. The plasma sheet has partially passed passed $x = 0.25$. The re-mapping of MHD to two-fluid in case (e) has not produced any noticeable anomalies.



(a) MHD (b) Two-fluid (c) Mixed.25 (d) Mixed.5 (e) Adaptive

Figure 10.4: Plasma density at t=2. The plasma sheet has fully passed $x = 0.25$. Instability growth on the trailing edge of the sheet develops in simulations which have two-fluid on the trailing edge, but on in the MHD only case.



(a) MHD (b) Two-fluid (c) Mixed.25 (d) Mixed.5 (e) Adaptive

Figure 10.5: Plasma density at t=2.5. The leading edge of the plasma sheet is reaching $x = 0.5$ for the two-fluid and mixed model simulations. Some of the energy appears to have been dissipated in the turbulent structure which develops in the two-fluid regions, while the MHD-only case has accelerated almost to $x = 0.75$ with no significant turbulent structure formation.

10.3 Adaptive setup

For the adaptive simulations, b was changed to 4. Additionally, an initial magnetic field was initialized such that

$$B_z = \begin{cases} \frac{B_0(x_c^4 - 4x_c x^3 + 3x^4)}{x_c^4} & x < x_c \\ 0 & x \geq x_c \end{cases} \tag{10.3}$$

$$\tag{10.4}$$

Note that for two-fluid the electric field and initial current density are initialized to be consistent with the MHD model; that is

$$\boldsymbol{j} = \frac{\delta_p}{L} \nabla \times \boldsymbol{B} \tag{10.5}$$

$$\boldsymbol{E} = \eta \boldsymbol{j} \tag{10.6}$$

$$\boldsymbol{p}_\alpha = \frac{A_\alpha A_\beta \boldsymbol{j}}{A_\beta Z_\alpha - A_\alpha Z_\beta} \tag{10.7}$$

This was done since it appears that the preliminary simulations had an excessive amount of artificial magnetic field penetration, likely as a result of the degree 0 basis used. Using a higher order basis significantly reduces the amount of artificial magnetic field penetration without a large resistivity or skin depth.

This is coupled with a reduced collisionality model with parabolic divergence cleaning. Artificial viscosity is also added in the form

$$Q_\alpha = -\mu_a \rho_\alpha |\nabla \cdot \boldsymbol{u}_\alpha| \nabla \cdot \boldsymbol{u}_\alpha \tag{10.8}$$

The dissipation coefficients are chosen as

$$\mu_c = \eta_c = R_c = \kappa_c = 0, \qquad Q_c = 1, \qquad c_p = 10^{-3}, \qquad \mu_a = 10^{-5}$$

This is solved using a DIRK3 HDG method with a degree 1 normalized Legendre basis, using restart GMRES for the global Schur solve. Max Δt was set to 10^{-2}, however Δt was allowed to change if the solver failed to find a solution.

10.4 Alfvén velocity metric

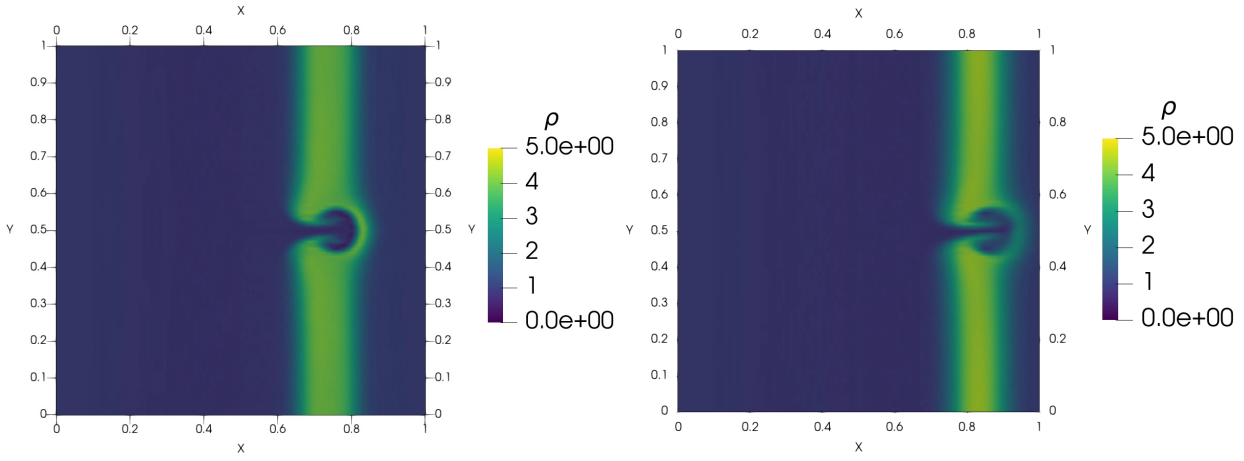
This metric chooses the element model based on

$$\text{model} = \begin{cases} \text{two-fluid} & \sqrt{\frac{B^2}{\rho}} \geq \tau_A c_0 \\ \text{MHD} & \sqrt{\frac{B^2}{\rho}} < \tau_A c_0 \end{cases} \tag{10.9}$$

where τ_A is a user-specified tolerance for the ratio between the Alfvén velocity and the speed of light at which the switching occurs. Choosing $\tau_A = 0$ forces the entire domain to be solved using two-fluid while $\tau_A \to \infty$ forces the entire domain to be solved using MHD. The relevant fields are evaluated using the element average values. The domain remapping is performed approximately when the current simulation time exceeds $t_{sim} = 10^{-2}N$, where N is the write out frame.

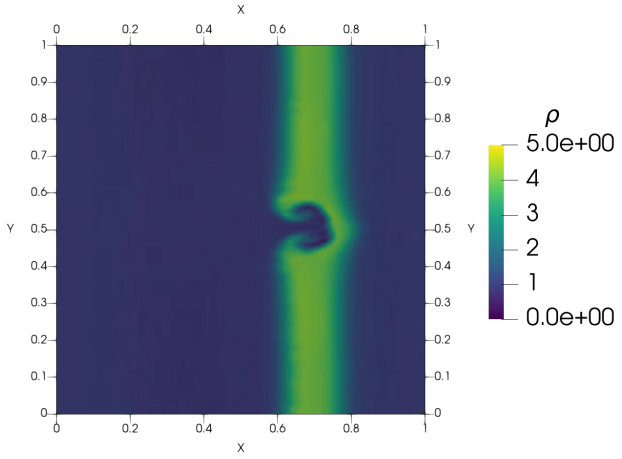
10.4.1 Small Hall parameter

Choosing $\frac{\delta_p}{L} = 10^{-2}$, the MHD approximation should be relatively accurate. Figure 10.6 supports this expectation, with MHD and two-fluid developing a near symmetric magnetic Raleigh-Taylor instability. The adaptive model also produces similar overall results, validating that in a parameter regime where the two models are well matched the variable remapping and coupling numerical flux performs well, however it does lead to some slight asymmetry in the Raleigh-Taylor instability.



(a) MHD

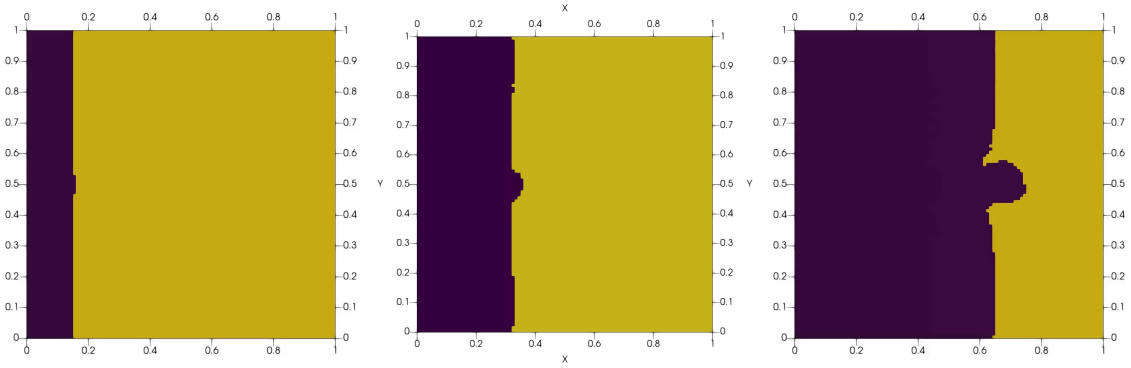
(b) two-fluid



(c) adaptive $\tau_A = 10^{-2}$

Figure 10.6: Plasma density at $t = 1.8$. All 3 solutions are relatively well matched, though there is some discrepancy in the overall plasma column bulk x velocity.

Figure 10.7 shows which model is used in the adaptive method as the simulation progresses. This tracks the propagation of the plasma column over time.



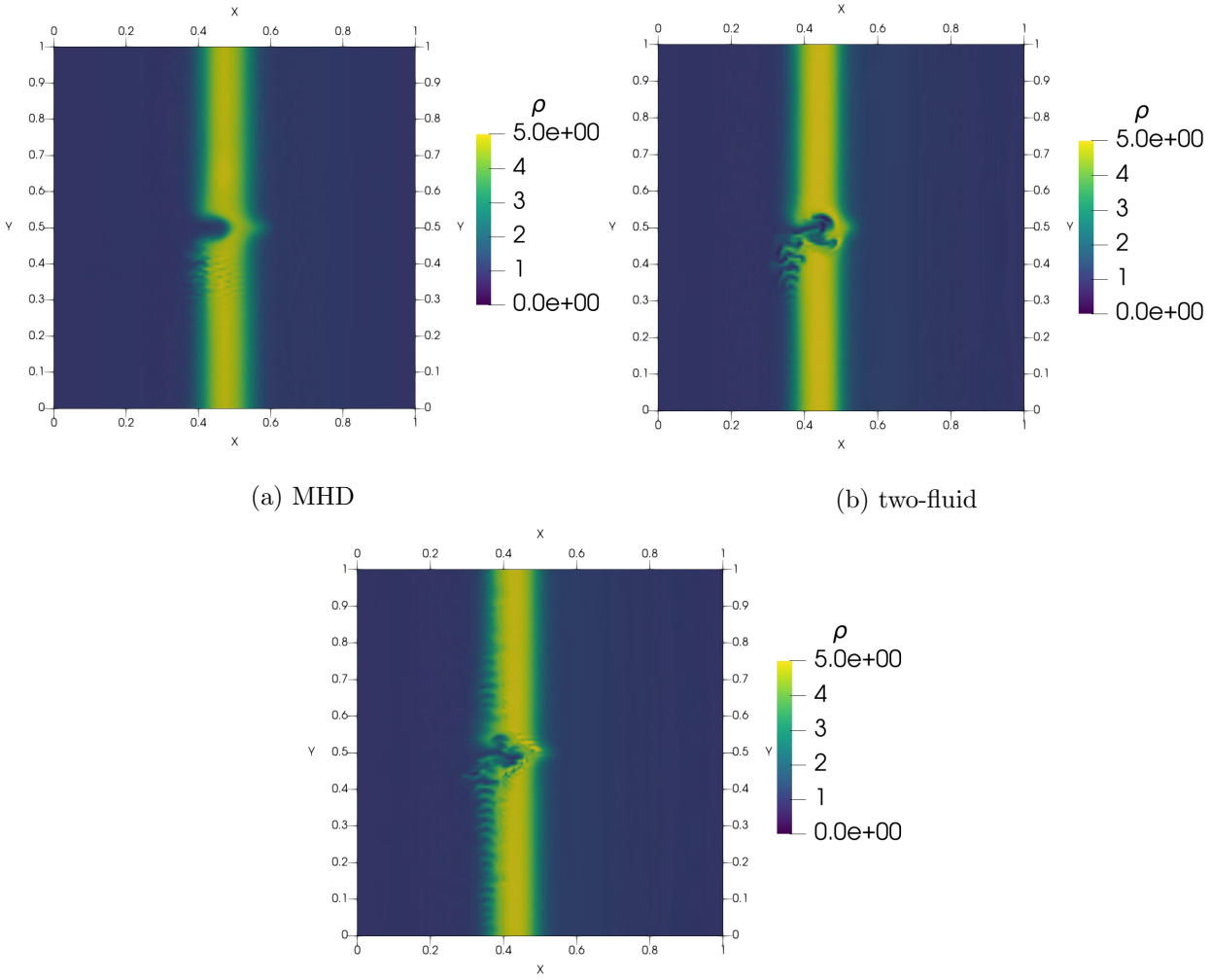
(a) $t = 0$ (b) $t = 0.9$ (c) $t = 1.8$

Figure 10.7: Plasma model used for $\tau_A = 10^{-2}$. Purple is two-fluid and yellow is MHD.

Monitoring Δt for the MHD only and two-fluid only sims, the majority of the time was spent at the max Δt. For reference, this is roughly 500x the allowed two-fluid explicit timestep limit, and is 60x the speed of light CFL condition. The mixed model simulation oscillated around $\Delta t \approx 5 \cdot 10^{-3}$, which while still significantly larger than the CFL limit, is smaller than solving the entire domain with just two-fluid. The MHD only simulation operated much closer to the CFL limit, though this is not unexpected as these timescales are close to the timescales of the bulk plasma dynamics.

10.4.2 Moderate Hall parameter

Choosing $\frac{\delta_p}{L} = 10^{-1}$, there should be a small but significant Hall effect. Figure 10.8 shows that at this point a secondary asymmetric instability develops in the plasma column, accelerating the column erosion process. Without Hall physics the primary Rayleigh-Taylor instability in the MHD model remains symmetric, while the two-fluid model develops an asymmetric shape. The mismatch error between MHD and two-fluid is also noticeable.



(a) MHD

(b) two-fluid

(c) adaptive $\tau_A = 10^{-2}$

Figure 10.8: Plasma density at $t = 0.9$. A secondary instability develops. The primary magnetic Rayleigh-Taylor instability becomes asymmetric when solved using two-fluid as a result of Hall physics. There are noticeable numerical artifacts at the transition boundary of the mixed model.

Chapter 11

Conclusion

The wide span of time scales found in many plasma systems makes many types of problems difficult to simulate effectively. This research aimed to reduce the computational cost of doing so while maintaining an appropriate level of physical fidelity.

The primary methods investigated to accomplish this goal are implementing a spatially adaptive hybrid plasma model, and identifying effective metrics for evaluating the physical fidelity in-situ. The methods are coupled with a diagonally implicit Runge-Kutta temporal solver coupled with hybridizable Discontinuous Galerkin. A summary of the primary results from this research are provided below.

11.1 Plasma model

This research implemented a 3D 5-moment two-fluid model coupled to a two-temperature MHD model. The model implements a simplified collisionality model including artificial viscosity and mixed hyperbolic/parabolic divergence cleaning. The two-fluid model is capable of accurately capturing Hall effects and finite electron inertia that the MHD model does not; however, the two-fluid model is also more computationally expensive to solve. The higher cost of the two-fluid model

can be mitigated by utilizing a hybrid model which adaptively chooses which plasma model to use such that the appropriate physics can be captured without paying for the higher cost of the two-fluid model in the entire domain. Methods were developed to allow the two-fluid and MHD models to be remapped to each other, along with producing numerical fluxes which can accurately model transport between the two models. The hybrid model was coupled with a metric comparing the local Alfvén velocity with the speed of light to allow the utilized model to change on an element by element basis. For the plasma erosion opening switch this metric is adequate, despite the slight conflict that this metric effectively forces MHD to be used in regions which are weakly magnetized. The boundary between models should be located somewhere that both models are valid; otherwise, the coupling techniques can produce significant numerical artifacts.

11.2 Numerical methods

In order to solve the timestep stability limitations of the two-fluid model, a diagonally implicit Runge-Kutta temporal solver was used. L-stable DIRK temporal solvers allows a timestep to be chosen based on the desired numerical truncation accuracy instead of being limited by the CFL condition. In practice it was found that this timestep is not completely unrestricted for a non-linear model such as the two-fluid model since it is difficult to produce sufficiently accurate initial guesses for the nonlinear solver to converge. Despite this limitation, timesteps in excess of 60x the CFL limit were observed to be possible while maintaining sufficiently small truncation errors.

Attempts to implement an implicit traditional discontinuous Galerkin method proved to be fraught with issues including poor solver convergence for even linear problems as well as having an excessive global system size, resulting in poor computational performance. Both of these issues were mitigated by utilizing hybridizable discontinuous Galerkin, allowing the hybrid model with over 3 million degrees of freedom to be solved in a fully implicit fashion. It was noted that the most

expensive step of solving the HDG system was constructing the HDG Schur complement, which is required for each iteration of the non-linear solve.

11.3 Code framework

In addition to utilizing various smaller test codes and WARPXM, this research developed a tool for generating HDG codes. The complexity of the HDG system renders the use of code generation tools practically required for correctly constructing the HDG Schur system for large models. Additionally due to the novel structure of HDG, using existing numerical tools seemed to be intractable. This tool was developed based on the SymPy symbolic math package and multiple matrix libraries including Eigen3 and PETSc. SymPy was used to generate and simplify the local and global block matrices required for constructing the HDG Schur complement, as well as eliminating any unnecessary degrees of freedom to both produce a well posed system and reduce computational costs. For the hybrid plasma model utilized in this research this code generated 1898 distinct non-zero entries for the Jacobian along with over 5k lines of code to construct the Schur complement. This tool supports generating HDG solvers for 3D unstructured hexahedral meshes. A major limitation of the generated code is the lack of common sub expression elimination, along with limitations to the expression complexity as SymPy is unable to handle expressions above a certain size.

11.4 Publications

This research is to be published in 4 papers. The first[55] has been published already and describes the methods for coupling two-fluid and MHD into a hybrid model. The second paper to be published presents the results of applying HDG to the two-fluid plasma model. The third paper to be published will present the effectiveness of various divergence cleaning methods such as purely

hyperbolic, purely parabolic, or mixed hyperbolic/parabolic for the two-fluid plasma model. This would include a parameter scan to identify what cleaning parameters work best as it was found that stronger divergence cleaning did not necessarily produce strictly superior results. The last paper would present the results of using the fully adaptive hybrid plasma model used to model the planar plasma erosion opening switch.

11.5 Future work

Due to the size of systems being solved with millions of degrees of freedom, an iterative linear solver is practically required. Investigating methods for improving the speed of this step such as which preconditioners and iterative method work best for plasma HDG systems can significantly improve the HDG solve speed. It was noted that many of the block matrices for the hybrid plasma model can have relative well defined sparsity patterns in addition to those already guaranteed by HDG, leaving many of the blocks with only about 10% non-zero entries. Investigating using sparse matrices during the Schur construction process could also provide meaningful speedups. In addition to improving the HDG solve speed directly, not all parts of the hybrid plasma model have stiff explicit timestep restrictions. Identifying how to best partition the plasma model between implicit and explicit timestepping could reduce the number of implicit degrees of freedom while still allowing timesteps on the scale of the desired dynamics.

Bibliography

[1] Haines, M. G., Lebedev, S. V., Chittenden, J. P., Beg, F. N., Bland, S. N., and Dangor, A. E., "The past, present, and future of Z pinches," *Physics of Plasmas*, Vol. 7, No. 5, 2000, pp. 1672–1680.

[2] Shumlak, U., Nelson, B. A., Claveau, E. L., Forbes, E. G., Golingo, R. P., Hughes, M. C., Oberto, R. J., Ross, M. P., and Weber, T. R., "Increasing plasma parameters using sheared flow stabilization of a Z-pinch," *Physics of Plasmas*, Vol. 24, No. 055702, 2017, pp. 1–10.

[3] Tummel, K., Higginson, D. P., Link, A. J., Schmidt, A. E. W., Offermann, D. T., Welch, D. R., Clark, R. E., Shumlak, U., Nelson, B. A., Golingo, R. P., and McLean, H. S., "Kinetic simulations of sheard flow stabilization in high-temperature Z-pinch plasmas," *Physics of Plasmas*, Vol. 26, 2019.

[4] Shumlak, U., "Z-pinch fusion," *Journal of Applied Physics*, Vol. 127, 2020.

[5] Wang, M. and Miley, G., "Particle orbits in field-reversed mirrors," *Nuclear Fusion*, Vol. 19, No. 1, 1979, pp. 39–49.

[6] Tuszewski, M., "Field reversed configurations," *Nuclear Fusion*, Vol. 28, No. 11, 1988, pp. 2033.

[7] Steinhauer, L. C., "Review of field-reversed configurations," *Physics of Plasmas*, Vol. 18, No. 070501, 2011, pp. 1–38.

[8] Tskhakaya, D., "On Recent Massively Parallelized PIC Simulations of the SOL," *Contributions to Plasma Physics*, Vol. 52, No. 5-6, 2012, pp. 490–499.

[9] Reddell, N., *A Kinetic Vlasov Model for Plasma Simulation Using Discontinuous Galerkin Method on Many-Core Architectures*, Ph.D. thesis, University of Washington, Seattle, WA, 2016.

[10] Birn, J., Drake, J. F., Shay, M. A., Rogers, B. N., Denton, R. E., Hesse, M., Kuznetsova, M., Ma, Z. W., Bhattacharjee, A., Otto, A., and Pritchett, P. L., "Geospace Environmental Modeling (GEM) Magnetic Reconnection Challenge," *Journal of Geophysical Research*, Vol. 106, No. 2, 2001, pp. 3715.

[11] Stanier, A., Daughton, W., Chacon, L., Karimabadi, H., Huang, J. N. Y.-M., Hakim, A., and Bhattacharjee, A., "Role of Ion Kinetic Physics in the Interaction of Magnetic Flux Ropes," *Physical Review Letters*, Vol. 115, October 2015.

[12] Stanier, A., Chacon, L., and Chen, G., "A fully implicit, conservative, non-linear, electromagnetic hybrid particle-ion/fluid-electron algorithm," *Journal of Computational Physics*, Vol. 376, 2019, pp. 597–616.

[13] Plewa, T., Linde, T., and Weirs, V. G., editors, *Adaptive Mesh Refinement - Theory and Applications*, No. 41 in Lecture Notes in Computational Science and Engineering, Springer.

[14] Ollivier-Gooch, C. and Altena, M. V., "A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation," *Journal of Computational Physics*, Vol. 181, No. 2, 2002, pp. 729–752.

[15] Dahlquist, G., "Convergence and stability in the numerical integration of ordinary differential equations," *Mathematica Scandinavica*, Vol. 4, 1956, pp. 33–53.

[16] Dahlquist, G., "A special stability problem for linear multistep methods," *BIT Numerical Mathematics*, Vol. 3, No. 1, 1963, pp. 27–43.

[17] Noh, W. F., "Errors for Calculations of Strong Shocks Using an Artificial Viscosity and an Artificial Heat Flux," *Journal of Computational Physics*, Vol. 72, 1978, pp. 78–120.

[18] Weber, B. V., Commisso, R. J., Cooperstein, G., Grossmann, J. M., Hinshelwood, D. D., Mosher, D., Neri, J. M., Ottinger, P. F., and Stephanakis, S. J., "Plasma Erosion Opening Switch Research at NRL," *IEEE Transactions on Plasma Science*, Vol. PS-15, No. 6, 1987, pp. 635–648.

[19] Goldston, R. J. and Rutherford, P. H., *Introduction to Plasma Physics*, Taylor and Francis Group, 1995.

[20] Braginskii, S. I., "Transport Processes in a Plasma," *Review of Plasma Physics*, Vol. 1, 1965, pp. 205–311.

[21] Dedner, A., Kemm, F., Kroner, D., Munz, C.-D., Schnitzer, T., and Wesenberg, M., "Hyperbolic Divergence Cleaning for the MHD Equations," *Journal of Computational Physics*, Vol. 175, 2002, pp. 645–673.

[22] Chapman, S. and Cowling, T. G., *The mathematical theory of non-uniform gases*, Cambridge University Press, 1970.

[23] Simakov, A. N. and Molvig, K., "Hydrodynamic description of an unmagnetized plasma with multiple ion species. I. General formulation," *Physics of Plasmas*, Vol. 23, No. 032115, 2016.

[24] Schiesser, W., *The Numerical Method of Lines*, Academic Press, 1st ed., 1991.

[25] Hesthaven, J. S. and Warburton, T., *Nodal Discontinuous Galerkin Methods*, Springer, 2008.

[26] Cockburn, B., Dong, B., Guzman, J., Restelli, M., and Sacco, R., "A Hybridizable Discontinuous Galerkin Method for Steady-State Convection-Diffusion-Reaction Problems," *SIAM Journal on Scientific Computing*, Vol. 31, No. 5, 2009, pp. 3827–3846.

[27] Maso, G. D., LeFloch, P. G., and Murat, F., "Definition and weak stability of nonconservative products," *Journal de Mathematiques Pures et Appliquees*, Vol. 74, No. 483, 1995.

[28] Rhebergen, S., *Discontinuous Galerkin Finite Element Methods for (Non)Conservative Partial Differential Equations*, Ph.D. thesis, University of Twente, 2010.

[29] LeFloch, P. G., "Shock waves for nonlinear hyperbolic systems in nonconservative form," Tech. rep., Institute for Mathematics and its Applications, Minneapolis, MN, 1989, REport 593.

[30] LeFloch, P. G. and Tzavaras, A. E., "Representation of Weak Limits and Definition of Nonconservative Products," *SIAM Journal on Mathematical Analysis*, Vol. 30, No. 6, 1999.

[31] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Jounral of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[32] Rusanov, V. V., "The calculation of the interaction of non-stationary shock waves with barriers," *USSR Computational Mathematics and Mathematical Physics*, Vol. 1, No. 2, 1962, pp. 304–320.

[33] Cockburn, B. and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1997, pp. 2440–2463.

[34] Douglas, Jr., J. and Dupont, T., "Interior Penalty Procedures for Elliptic and Parabolic Galerkin Methods," *Computing Methods in Applied Sciences*, edited by R. Glowinski and

J. L. Lions, Vol. 58 of *Lecture Notes in Physics*, Institut de Recherche d'Informatique et d'Automatique, Springer-Verlag, 1975, pp. 207–216.

[35] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems," *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.

[36] Peraire, J. and Persson, P.-O., "The Compact Discontinuous Galerkin (CDG) Method for Elliptic Problems," *SIAM Journal on Scientific Computing*, Vol. 30, No. 4, 2008, pp. 1806–1824.

[37] Kirby, R. M., Sherwin, S. J., and Cockburn, B., "To CG or to HDG: A Comparative Study," *Journal of Scientific Computing*, Vol. 51, 2012, pp. 183–212.

[38] Nguyen, N. C. and Peraire, J., "Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics," *Journal of Computational Physics*, Vol. 231, 2012, pp. 5955–5988.

[39] Datta, I., *A Domain-Hybridized Plasma Model using Discontinuous Galerkin Finite Elements*, Ph.D. thesis, University of Washington, 2021.

[40] Bui-Thanh, T., "Construction and Analysis of HDG Methods for Linearized Shallow Water Equations," *SIAM Journal on Scientific Computing*, Vol. 38, No. 6, 2016, pp. A3696–A3719.

[41] Apel, T. and Duvelmeyer, N., "Transformation of Hexahedral Finite Element Meshes into Tetrahedral Meshes According to Quality Criteria," *Computing*, Vol. 71, 2003, pp. 293–304.

[42] Kennedy, C. A. and Carpenter, M. H., "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," *Applied Numerical Mathematics*, Vol. 44, 2003, pp. 139–181.

[43] Butcher, J. C., "Coefficients for the Study of Runge-Kutta Integration Processes," *Journal of the Australian Mathematical Society*, Vol. 3, No. 2, 1963, pp. 185–201.

[44] Hairer, E. and Wanner, G., *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer Berlin Heidelberg, 3rd ed., 2008.

[45] Behlberg, E., "Low-Order Classical Runge-Kutta Formulas with stepsize control and their application to some heat transfer problems," Tech. rep., George C. Marshall Space Flight Center, National Aeronatics and Space Administration, Washingon, D. C., July 1969.

[46] Dormand, J. R. and Prince, P. J., "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, Vol. 6, No. 1, 1980.

[47] Murua, A., "Formal series and numerical integrators, Part I: Systems of ODEs and symplectic integrators," *Applied Numerical Mathematics*, Vol. 29, 1999, pp. 221–251.

[48] Ascher, U. M., Ruuth, S. J., and Spiteri, R. J., "Implicit-Explicit Runge-Kutta Methods for Time-Dependent Partial Differential Equations," *Applied Numerical Mathematics*, Vol. 25, No. 2-3, 1997, pp. 151–167.

[49] Cooper, G. J. and Sayfy, A., "Additive Methods for the Numerical Solution of Ordinary Differential Equations," *Mathematics of Computation*, Vol. 35, No. 152, 1980, pp. 1159–1172.

[50] Cooper, G. J. and Sayfy, A., "Additive Runge-Kutta Methods for Stiff Ordinary Differential Equations," *Mathematics of Computation*, Vol. 40, No. 161, 1983, pp. 207–218.

[51] Soderlind, G., "Digital filters in adaptive time-stepping," *ACM Transactions on Mathematical Software*, Vol. 29, No. 1, 2003, pp. 1–26.

[52] Shampine, L. F. and Reichelt, M. W., "The Matlab ODE Suite," *SIAM Journal on Scientific Computing*, Vol. 18, 1997, pp. 1–22.

[53] Morgan, N. R., "A dissipation model for staggered grid Lagrangian hydrodynamics," *Computers & Fluids*, , No. 83, 2013, pp. 48–57.

[54] Landshoff, R. A., "A numerical method for treating fluid flow in presence of shocks." Tech. rep., Los Alamos National Laboratory, 1955.

[55] Ho, A., Datta, I., and Shumlak, U., "Physics-Based-Adaptive Plasma Model for High-Fidelity Numerical Simulations," *Frontiers in Physics*, 2018.

[56] Miller, S., *Modeling Collisional processes in plasmas using discontinuous numerical methods*, Ph.D. thesis, University of Washington, 2016.

[57] Karypis, G. and Kumar, V., "MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0," `http://www.cs.umn.edu/~metis`, 2009.

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

[59] Loverich, J., Hakim, A., and Shumlak, U., "A Discontinuous Galerkin Method for Ideal Two-Fluid Plasma Equations," *Communications in Computational Physics*, Vol. 9, No. 2, 2011, pp. 240–268.

[60] Miller, S. T. and Shumlak, U., "A multi-species 13-moment model for moderately collisional plasmas," *Physics of Plasmas*, Vol. 23, 2016.

[61] Zhdanov, V. M., *Transport processes in multicomponent plasma*, CRC Press, 2002.

[62] Osin, D., Arad, R., Tsigutkin, K., Doron, R., Starobinits, A., Bernshtam, V., and Maron, Y., "The role of the plasma composition in the magnetic field evolution in plasma opening switches," *Digest of Technical Papers, PPC-2003, 14th IEEE International Pulsed Power Conference (IEEE Cat. No.03CH37472*, Vol. 1, 2003, pp. 577–582.

[63] Mason, R. J., Jones, M. E., Grossmann, J. M., and Ottinger, P. F., "Three-fluid simulation of the plasma-erosion-opening switch," *Journal of Applied Physics*, Vol. 64, 1988, pp. 4208–4211.

[64] Strauss, H. R., Doron, R., Arad, R., Rubinstein, B., Maron, Y., and Fruchtman, A., "Magnetic field propagation in a two ion species planar plasma opening switch," *Physics of Plasmas*, Vol. 14, 2007.

[65] Sarfaty, M., Maron, Y., Krasik, Y. E., Weingarten, A., Arad, R., Shpitalnik, R., Fruchtman, A., and Alexiou, S., "Spectroscopic investigations of the plasma behavior in a plasma opening switch experiment," *Physics of Plasmas*, Vol. 2, No. 6, 1995, pp. 2122–2137.

[66] Shpitalnik, R., Weingarten, A., Gomberoff, K., Krasik, Y., and Maron, Y., "Observations of two-dimensional magnetic field evolution in a plasma opening switch," *Physics of Plasmas*, Vol. 5, No. 3, 1998, pp. 792–198.

[67] Arad, R., Tsigutkin, K., Maron, Y., Fruchtman, A., and Huba, J. D., "Observation of faster-than-diffusion magnetic field penetration into a plasma," *Physics of Plasmas*, Vol. 10, No. 1, 2003, pp. 112–125.

[68] Gottlieb, S., "On High Order Strong Stability Preserving Runge-Kutta and Multi Step Time Discretizations," *Journal on Scientific Computing*, Vol. 25, No. 1, 2005.

[69] Hairer, E. and Wanner, G., *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer Berlin Heidelberg, Berlin, 1991.

[70] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag Berlin Heidelberg, 2009.

Appendix A

Braginskii coefficient constants

This appendix tabulates the various constants used in Braginskii's two-fluid model[20]. These are listed using the normalization and conventions presented in Ch. 2.1.2. Note that Braginskii only computes a complete set of coefficients for $Z^{(i)} = 1$ under the assumption $A^{(e)} \ll A^{(i)}$.

Table A.1: Braginskii coefficients for $Z^{(i)} = 1$

g_{0e}	3.7703
g_{0i}	0.677
g_{1e}	14.79
g_{1i}	2.70
g_2	6.416
g_3	1.837
g_4	1.704
g_5	0.7796
g_6	5.101
g_7	2.681
g_8	$\frac{3}{2}$
g_9	3.053
g_{10e}	4.664
g_{10i}	2
g_{11e}	11.92
g_{11i}	2.645
g_{12e}	$\frac{5}{2}$
g_{12i}	$-\frac{5}{2}$
g_{13e}	21.67
g_{13i}	-4.65
g_{14e}	13.8
g_{14i}	4.03
g_{15e}	11.6
g_{15i}	2.33
g_{16e}	-2.05
g_{16i}	$-\frac{6}{5}$
g_{17e}	-8.50
g_{17i}	-2.23
g_{18e}	-1
g_{18i}	1
g_{19e}	-7.91
g_{19i}	2.38

Appendix B

Tables of coefficients for various Runge-Kutta schemes

This appendix summarizes the table of coefficients for various Runge-Kutta style methods. It contains both the traditional Butcher Tableau forms and the alternative formulation presented in Ch. 4.

B.1 Explicit Runge-Kutta

B.1.1 ERK1 (Forward Euler)

Table B.1: Butcher Tableau

0	0
	1

Table B.2: Alternative Formulation

	1	0
0	1	$\boxed{1}$

B.1.2 ERK2

This formulation is general for a large class of 2-stage ERK methods[43], including the traditional midpoint method and Heun's method. The midpoint method chooses $\alpha = \frac{1}{2}$, Heun's method chooses $\alpha = 1$, and Ralston's method chooses $\alpha = \frac{2}{3}$.

Table B.3: Butcher Tableau

0	0	0
α	α	0
	$1 - \frac{1}{2\alpha}$	$\frac{1}{2\alpha}$

Table B.4: Alternative Formulation

	1	0	0
0	1	$\boxed{\alpha}$	0
α	$1 + \frac{1}{2\alpha^2} - \frac{1}{\alpha}$	$\frac{1}{2\alpha^2}$	$\boxed{1 - \frac{1}{2\alpha}}$

B.1.3 ERK3 (Traditional)

This is Kutta's formulation of the 3-stage, 3rd-order ERK method[43].

Table B.5: Butcher Tableau

0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0
1	-1	2	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table B.6: Alternative Formulation

	1	0	0	0
0	1	$\boxed{\frac{1}{2}}$	0	0
$\frac{1}{2}$	3	-2	$\boxed{2}$	0
1	$-\frac{1}{3}$	1	$\frac{1}{3}$	$\boxed{\frac{1}{6}}$

B.1.4 Strong Stability Preserving ERK3

This is a 3-stage, 3rd order accurate ERK method with the strong stability preserving property[68].

It is also known as the Shu-Osher method.

Table B.7: Butcher Tableau

0	0	0	0
1	1	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0
	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$

Table B.8: Alternative Formulation

	1	0	0	0
0	1	$\boxed{1}$	0	0
1	$\frac{3}{4}$	$\frac{1}{4}$	$\boxed{\frac{1}{4}}$	0
$\frac{1}{2}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\boxed{\frac{2}{3}}$

B.1.5 ERK4

This is the "3-8ths" rule formulation of the 4-stage, 4th order ERK method[43].

Table B.9: Butcher Tableau

0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0	0
$\frac{2}{3}$	$-\frac{1}{3}$	1	0	0
1	1	-1	1	0
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

Table B.10: Alternative Formulation

	1	0	0	0	0
0	1	$\boxed{\frac{1}{3}}$	0	0	0
$\frac{1}{3}$	2	-1	$\boxed{1}$	0	0
$\frac{2}{3}$	0	2	-1	$\boxed{1}$	0
1	$-\frac{1}{8}$	0	$\frac{3}{4}$	$\frac{3}{8}$	$\boxed{\frac{1}{8}}$

B.1.6 ERK5(4) Method

This is a 5th-order scheme with an embedded 4th order error estimator proposed by Dormand and Prince[46]. The 5th-order accurate solution is given as the first row below the horizontal line, and the second row below the horizontal line gives the approximate solution. An estimate of the 4th order error is given by

$$\boldsymbol{\epsilon} \approx \boldsymbol{q}^{(1)} - \boldsymbol{q}^{(2)} \tag{B.1}$$

Table B.11: Butcher Tableau

0	0	0	0	0	0	0	0
$\frac{1}{5}$	$\frac{1}{5}$	0	0	0	0	0	0
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	0	0	0	0	0
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$	0	0	0	0
$\frac{8}{9}$	$\frac{19372}{6561}$	$\frac{-25360}{2187}$	$\frac{64448}{6561}$	$\frac{-212}{729}$	0	0	0
1	$\frac{9017}{3168}$	$\frac{-355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	0	0
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	0
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	0
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Table B.12: Alternative Formulation

	1	0	0	0	0	0	0	0
0	1	$\boxed{\frac{1}{5}}$	0	0	0	0	0	0
$\frac{1}{5}$	$\frac{5}{8}$	$\frac{3}{8}$	$\boxed{\frac{9}{40}}$	0	0	0	0	0
$\frac{3}{10}$	$\frac{175}{27}$	$\frac{100}{9}$	$-\frac{448}{27}$	$\boxed{\frac{32}{9}}$	0	0	0	0
$\frac{4}{5}$	$\frac{3551}{6561}$	$\frac{7420}{2187}$	$-\frac{37376}{6561}$	$\frac{2014}{729}$	$\boxed{-\frac{212}{729}}$	0	0	0
$\frac{8}{9}$	$\frac{313397}{335808}$	$\frac{424025}{55968}$	$-\frac{61400}{5247}$	$\frac{96075}{18656}$	$-\frac{35721}{37312}$	$\boxed{-\frac{5103}{18656}}$	0	0
1	$-\frac{563}{3456}$	$-\frac{575}{252}$	$\frac{31400}{10017}$	$\frac{325}{1344}$	$-\frac{7533}{6784}$	$\frac{33}{28}$	$\boxed{\frac{11}{84}}$	0
1	0	0	0	0	0	0	1	0
1	$\frac{8813}{172800}$	$\frac{41}{180}$	$-\frac{4294}{10017}$	$\frac{263}{384}$	$-\frac{137781}{339200}$	$\frac{803}{4200}$	$\frac{17}{25}$	$\boxed{\frac{1}{40}}$

B.2 Implicit Runge-Kutta

B.2.1 IRK1 (Backwards Euler)

Table B.13: Butcher Tableau

1	1
	1

Table B.14: Alternative Formulation

		1	0
1	1	0	1

B.2.2 IRK2

This is an L-stable two stage, second order diagonally implicit RK method[69].

Table B.15: Butcher Tableau

γ	γ	0
1	$1-\gamma$	γ
	$1-\gamma$	γ

$$\gamma = \frac{2-\sqrt{2}}{2} \tag{B.2}$$

Table B.16: Alternative Formulation

		1	0	0
γ	γ	$-\sqrt{2}$	$1+\sqrt{2}$	0
1	γ	0	0	1

B.2.3 IRK3

This is an L-stable four stage, third order diagonally implicit RK method[48].

Table B.17: Butcher Tableau

$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{2}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	0	0
$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
1	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

Table B.18: Alternative Formulation

		1	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{3}$	0	0	0
$\frac{2}{3}$	$\frac{1}{2}$	4	-4	1	0	0
$\frac{1}{2}$	$\frac{1}{2}$	4	0	-4	1	0
1	$\frac{1}{2}$	0	0	0	0	1

B.3 Additive Runge-Kutta ImEx schemes (ARK_2)

B.3.1 ARS(1,1,1)

This is the simplest first order accurate ImEx scheme proposed by Ascher et al.[48].

Table B.19: Butcher Tableaus

(a) Explicit

0	0	0
1	1	0
	1	0

(b) Implicit

0	0	0
1	0	1
	0	1

Table B.20: Alternative Formulation

(a) Explicit

	1	0	0
0	1	$\boxed{1}$	0
1	0	0	$\boxed{0}$

(b) Implicit

1	$\boxed{1}$	0	0
		0	1

B.3.2 ARS(1,2,2)

This is a mixed midpoint method which has reasonably good performance when $f^I = 0$ proposed by Ascher et al. with good symmetry properties. [48]

Table B.21: Butcher Tableaus

(a) Explicit

0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1

(b) Implicit

0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$
	0	1

Table B.22: Alternative Formulation

(a) Explicit

	1	0	0
0	1	$\boxed{\frac{1}{2}}$	0
$\frac{1}{2}$	1	-2	$\boxed{\frac{1}{2}}$

(b) Implicit

$\frac{1}{2}$	$\boxed{\frac{1}{2}}$	0	0
		0	2

B.3.3 ARS(2,2,2)

This is an L-stable second order stiffly-accurate method proposed by Ascher et al. [48]

Table B.23: Butcher Tableaus

(a) Explicit

0	0	0	0
γ	γ	0	0
1	δ	$1-\delta$	0
	δ	$1-\delta$	0

(b) Implicit

0	0	0	0
γ	0	γ	0
1	0	$1-\gamma$	γ
	0	$1-\gamma$	γ

$$\gamma = \frac{2-\sqrt{2}}{2} \tag{B.3}$$

$$\delta = 1 - \frac{1}{2\gamma} \tag{B.4}$$

Table B.24: Alternative Formulation

(a) Explicit

	1	0	0	0
0	1	$\boxed{1-\frac{1}{\sqrt{2}}}$	0	0
γ	$2+\sqrt{2}$	$-2(1+\sqrt{2})$	$\boxed{1+\frac{1}{\sqrt{2}}}$	0
1	0	0	0	$\boxed{0}$

(b) Implicit

γ	$\boxed{\gamma}$	0	0	0
1	$\boxed{\gamma}$	0	$1+\sqrt{2}$	0
		0	0	1

B.3.4 ARS(4,4,3)

This is an L-stable third order stiffly-accurate method proposed by Ascher et al. [48]

Table B.25: Butcher Tableaus

(a) Explicit

0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
$\frac{2}{3}$	$\frac{11}{18}$	$\frac{1}{18}$	0	0	0
$\frac{1}{2}$	$\frac{5}{6}$	$-\frac{5}{6}$	$\frac{1}{2}$	0	0
1	$\frac{1}{4}$	$\frac{7}{4}$	$\frac{3}{4}$	$-\frac{7}{4}$	0
	$\frac{1}{4}$	$\frac{7}{4}$	$\frac{3}{4}$	$-\frac{7}{4}$	0

(b) Implicit

0	0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0
$\frac{2}{3}$	0	$\frac{1}{6}$	$\frac{1}{2}$	0	0
$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
1	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

Table B.26: Alternative Formulation

(a) Explicit

	1	0	0	0	0	0
0	1	$\boxed{\frac{1}{2}}$	0	0	0	0
$\frac{1}{2}$	$-\frac{2}{9}$	$\frac{8}{9}$	$\boxed{\frac{1}{18}}$	0	0	0
$\frac{2}{3}$	-4	16	-16	$\boxed{\frac{1}{2}}$	0	0
$\frac{1}{2}$	$\frac{27}{2}$	$-\frac{109}{2}$	$\frac{117}{2}$	$\frac{1}{2}$	$\boxed{-\frac{7}{4}}$	0
1	0	0	0	0	0	$\boxed{0}$

(b) Implicit

$\frac{1}{2}$	$\boxed{\frac{1}{2}}$	0	0	0	0	0
$\frac{2}{3}$	$\boxed{\frac{1}{2}}$	0	$\frac{1}{3}$	0	0	0
$\frac{1}{2}$	$\boxed{\frac{1}{2}}$	0	4	1	0	0
1	$\boxed{\frac{1}{2}}$	0	$-\frac{27}{2}$	$-\frac{9}{2}$	1	0
		0	0	0	0	1

B.3.5 ARK3(2)4L[2]SA

This is a third-order accurate (with an embedded second order error estimator) which is L-stable and stiffly-accurate proposed by Kennedy et al.[42]. The third order accurate solution is given by the first row below the horizontal line, and the second row below the horizontal line gives the

embedded second order solution. An estimate of the second order error is given by

$$\boldsymbol{\tau} \approx \boldsymbol{q}^{(1)} - \boldsymbol{q}^{(2)} \tag{B.5}$$

Table B.27: Butcher Tableaus

(a) Explicit

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{2027836641118}$	0	0	0
$\frac{3}{5}$	$\frac{5535828885825}{10492691773637}$	$\frac{788022342437}{10882634858940}$	0	0
1	$\frac{6485989280629}{16251701735622}$	$-\frac{4246266847089}{9704473918619}$	$\frac{10755448449292}{10357097424841}$	0
	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
	$\frac{2756255671327}{12835298489170}$	$-\frac{10771552573575}{22201958757719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

(b) Implicit

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
$\frac{3}{5}$	$\frac{2746238789719}{10658868560708}$	$-\frac{640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
1	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
	$\frac{2756255671327}{12835298489170}$	$-\frac{10771552573575}{22201958757719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

Table B.28: Alternative Formulation. Note: these are truncated to 25 digits accuracy.

(a) Explicit

	1	0	0	0	0
0	1	$\frac{1767732205903}{2027836641118}$	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{242428203753}{614082060734}$	$\frac{2195376142252}{2678046816373}$	$\frac{214076796647}{2956413142394}$	0	0
$\frac{3}{5}$	$\frac{16281260659748}{5561061074679}$	$\frac{26987923559575}{3982160831642}$	$-\frac{19588248685893}{2367945327308}$	$\frac{10755448449292}{10357097424841}$	0
1	$\frac{1590393888182}{1232361101833}$	$\frac{1369492187417}{692249032422}$	$-\frac{4407221830767}{1626584384428}$	$-\frac{237370831847}{3697227368284}$	$\frac{1767732205903}{4055673282236}$
1	$\frac{5226797055357}{5504458260211}$	$\frac{15212079363994}{12864276586381}$	$-\frac{5764012919707}{3250877926109}$	$-\frac{766096664479}{9007013911515}$	$\frac{2193209047091}{5459859503100}$

(b) Implicit

$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0	0
$\frac{3}{5}$	$\frac{1767732205903}{4055673282236}$	$-\frac{3110497978}{506079797733}$	$-\frac{482097480709}{2247026404427}$	0	0
1	$\frac{1767732205903}{4055673282236}$	$-\frac{120596014718}{2458786465077}$	$-\frac{10644448623847}{3998313746575}$	$\frac{6112610909662}{2741623861831}$	0
		$\frac{373820436274}{4191302137549}$	$-\frac{1574166267727}{2466205923007}$	$\frac{348578165441}{2435174606918}$	1
		$\frac{435465575127}{4061432407214}$	$-\frac{1355267739033}{4189250483900}$	$\frac{301127990792}{2353266570375}$	$\frac{2493600503814}{2705713622183}$

B.3.6 ARK4(3)6L[2]SA

This is a 4th-order accurate (with an embedded third order error estimator) which is L-stable and stiffly-accurate proposed by Kennedy et al.[42]. The 4th order accurate solution is given by the first row below the horizontal line, and the second row below the horizontal line gives the embedded third order solution. An estimate of the third order error is given by

$$\boldsymbol{\tau} \approx \boldsymbol{q}^{(1)} - \boldsymbol{q}^{(2)} \tag{B.6}$$

Table B.29: Butcher Tableaus

(a) Explicit

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0
$\frac{83}{250}$	$\frac{13861}{62500}$	$\frac{6889}{62500}$	0	0	0	0
$\frac{31}{50}$	$-\frac{116923316275}{2393684061468}$	$-\frac{2731218467317}{15368042101831}$	$\frac{9408046702089}{11113171139209}$	0	0	0
$\frac{17}{20}$	$-\frac{451086348788}{2902428689909}$	$-\frac{2682348792572}{7519795681897}$	$\frac{12662868775082}{11960479115383}$	$\frac{3355817975965}{11060851509271}$	0	0
1	$\frac{647845179188}{3216320057751}$	$\frac{73281519250}{8382639484533}$	$\frac{552539513391}{3454668386233}$	$\frac{3354512671639}{8306763924573}$	$\frac{4040}{17871}$	0
	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$-\frac{3700637}{11593932}$	$\frac{61727}{225920}$

(b) Implicit

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$\frac{1}{4}$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
$\frac{17}{20}$	$\frac{15267082809}{155376265600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$-\frac{3700637}{11593932}$	$\frac{61727}{225920}$

Table B.30: Alternative Formulation. Note: these are truncated to 25 digits accuracy.

(a) Explicit

	1	0	0	0	0	0	0
0	1	$\boxed{\frac{1}{2}}$	0	0	0	0	0
$\frac{1}{2}$	$\frac{17389}{31250}$	$\frac{20833}{31250}$	$\boxed{\frac{6889}{62500}}$	0	0	0	0
$\frac{83}{250}$	$\frac{2172616401434}{1089092410795}$	$\frac{1474136555735}{787077949561}$	$-\frac{4480841270700}{1314207323833}$	$\boxed{\frac{4604707481543}{5439269585792}}$	0	0	0
$\frac{31}{50}$	$\frac{1564645482298}{2536819695767}$	$\frac{10900161525709}{5827509590361}$	$-\frac{7589599382477}{3430076857982}$	$\frac{602716744621}{5109810053954}$	$\boxed{\frac{622363004273}{2051322456829}}$	0	0
$\frac{17}{20}$	$-\frac{1937345649199}{3004494093495}$	$-\frac{2384315802340}{956205443333}$	$\frac{6295701810836}{2154918075961}$	$-\frac{7175217609219}{2667037677430}$	$\frac{12190902877584}{5012723701201}$	$\boxed{\frac{4040}{17871}}$	0
1	$-\frac{5763249743158}{2994634597905}$	$-\frac{19568102568517}{2669942320838}$	$\frac{21245116554521}{2491931721674}$	$-\frac{21456746391197}{3551381529759}$	$\frac{26206866847341}{4156425716924}$	$-\frac{7615}{3434}$	$\boxed{\frac{1}{4}}$
1	$-\frac{8832954428148}{4220438822279}$	$-\frac{49548435204254}{6226224538257}$	$\frac{33026428432727}{3570335679444}$	$-\frac{37637982774485}{5654588805373}$	$\frac{19584124824613}{2787970406406}$	$-\frac{60727207}{24244040}$	$\frac{61727}{225920}$

(b) Implicit

$\frac{1}{2}$	$\boxed{\frac{1}{4}}$	$\frac{1}{4}$	0	0	0	0	0
$\frac{83}{250}$	$\boxed{\frac{1}{4}}$	$\frac{3361}{125000}$	$-\frac{3486}{15625}$	0	0	0	0
$\frac{31}{50}$	$\boxed{\frac{1}{4}}$	$\frac{2141662815316}{10082528300329}$	$-\frac{6206647758309}{4943758040947}$	$\frac{174375}{97027}$	0	0	0
$\frac{17}{20}$	$\boxed{\frac{1}{4}}$	$-\frac{7980156053}{2994913410991}$	$-\frac{2285198751501}{1505300220742}$	$\frac{1972820689480}{1986904169153}$	$\frac{2285395}{2017728}$	0	0
1	$\boxed{\frac{1}{4}}$	$\frac{96412531921}{5215020909889}$	$\frac{3947416374149}{1734729607757}$	$-\frac{11582646403955}{12671143517077}$	$\frac{2298037186111}{1891917610862}$	$-\frac{9040}{8211}$	0
		$\frac{61632802203}{1017305945290}$	$\frac{24332810895752}{3770878562619}$	$-\frac{18192934761799}{6085337818580}$	$\frac{3030382875373}{1825073796316}$	$-\frac{34419800}{14098287}$	1
		$\frac{304769643797}{4903704507668}$	$\frac{14137738549662}{2017863446453}$	$-\frac{10240352150505}{3167407125772}$	$\frac{2418363185273}{1271300101867}$	$-\frac{28180350755}{9953390622}$	$\frac{61727}{56480}$

Appendix C

5-Moment artificial dissipation numerical performance on a 1D shocktube

This appendix presents initial numerical testing of the artificial dissipation model described by Eqs. (5.1)-(5.5) on a single neutral fluid. Consider a 1D shock tube where

$$x \in [0, 10] \qquad x_s = 5 \qquad \frac{\rho_R}{\rho_L} = \frac{P_R}{P_L}$$

The numerical discretization for higher-order derivatives is handled using Local Discontinuous Galerkin (LDG)[33] configured with a pure bi-directional upwinding and choosing the jump penalty $\eta = 0$. A basic Rusanov[32] numerical flux is used for the Euler equations. The temporal discretization is handled using the embedded Dormand-Prince[46] ERK5(4) method, coupled with an I-controller described in Section 4.4, with $\Delta t_0 = 10^{-9}$, $\epsilon_{tol} = 10^{-8}$, and the additional restriction that $\Delta t \leq 10^{-5}$. Test problems are not allowed to run long enough such that the boundary conditions will effect the results.

C.1 Choice of artificial dissipation coefficients based on local compressibility

Consider a test case where

$$\frac{\rho_R}{\rho_L} = 10 \qquad P = \frac{\rho}{\gamma} \qquad \gamma = \frac{5}{3} \qquad A = 1$$

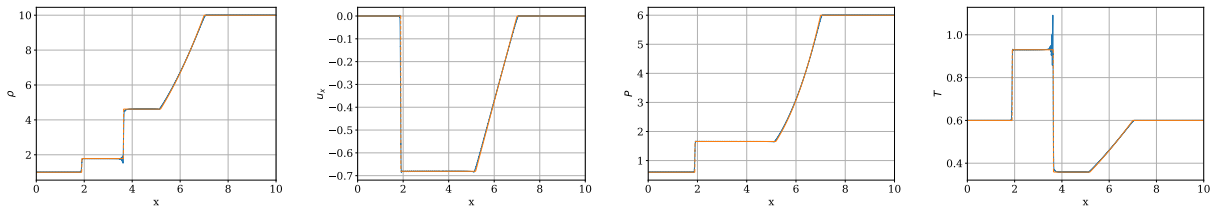
The domain is discretized with 400 equally sized elements, using a degree 3 polynomial basis.

Define

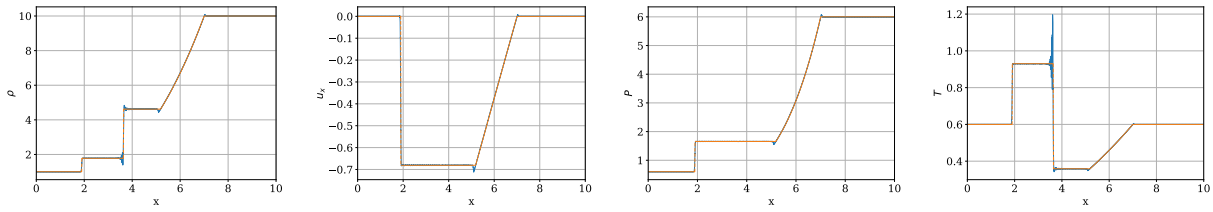
$$D = \sqrt{\left(\frac{\gamma+1}{4} l_c \nabla\cdot\boldsymbol{u}\right)^2 + c_s^2} - c_s \tag{C.1}$$

The effects of artificial viscosity and artificial heat flux are selectively enabled depending on the sign of $\nabla\cdot\boldsymbol{u}$. The tested cases are enumerated below along with the non-zero coefficients. Simulation results for density, velocity, pressure, and temperature are plotted in a solid blue line, and the analytical shock tube solution is plotted as an orange dashed line. The simulation solution are plotted at time $t = 2$. See Toro[70] Chapter 4 for a derivation of the analytical solution. Note that two tested cases failed to run for any meaningful amount of time. this is noted in the specific cases, and no accompanying data plots are provided since there's little meaningful simulation output.

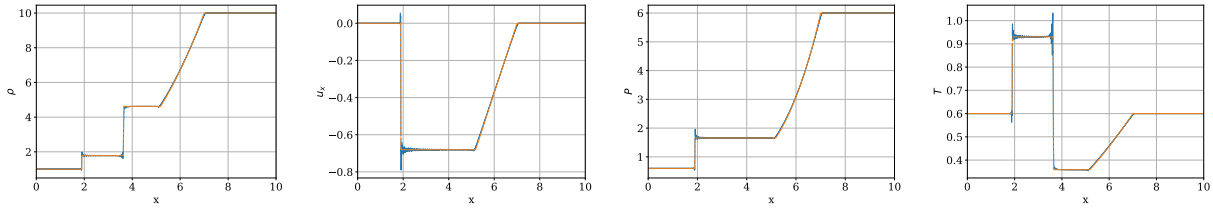
1. $C_0 = \frac{\gamma+1}{4}$, $C_1 = D$ (viscosity only in the entire domain)



2. $C_0^- = \frac{\gamma+1}{4}$, $C_1^- = D$ (viscosity only when $\nabla\cdot\boldsymbol{u} < 0$)

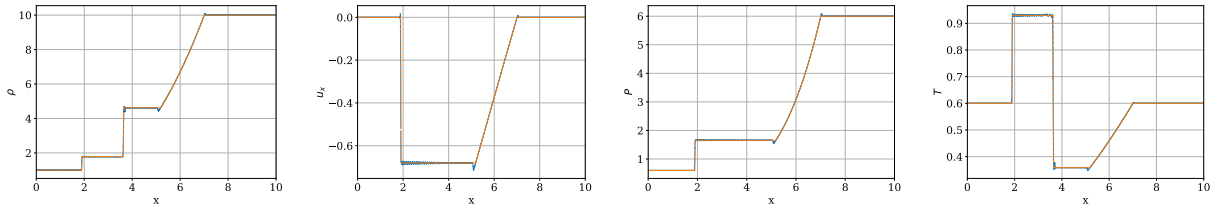


3. $C_0^+ = \frac{\gamma+1}{4}$, $C_1^+ = D$ (viscosity only when $\nabla \cdot \boldsymbol{u} > 0$)



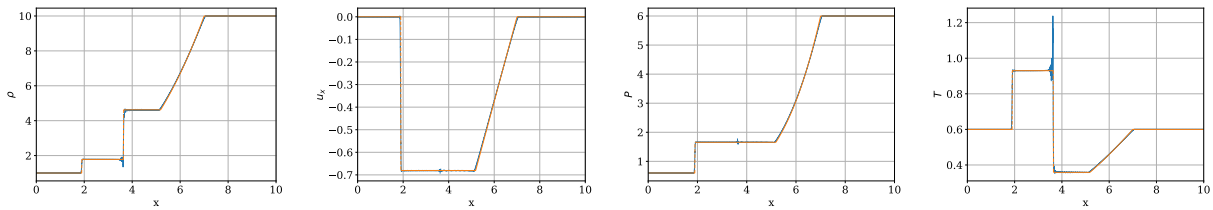
4. $C_2 = \frac{\gamma+1}{4}$, $C_3 = D$ (heat flux only in the entire domain) - doesn't run

5. $C_2^- = \frac{\gamma+1}{4}$, $C_3^- = D$ (heat flux only when $\nabla \cdot \boldsymbol{u} < 0$)

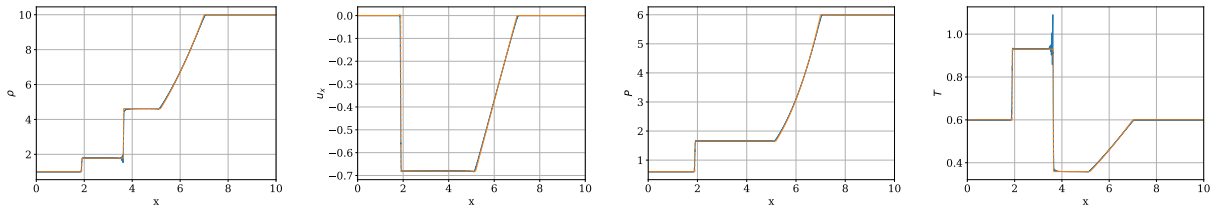


6. $C_2^+ = \frac{\gamma+1}{4}$, $C_3^+ = D$ (heat flux only when $\nabla \cdot \boldsymbol{u} > 0$) - doesn't run

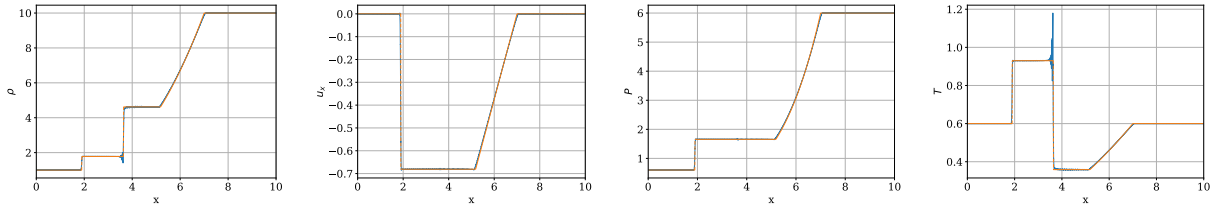
7. $C_0 = C_2 = \frac{\gamma+1}{4}$, $C_1 = C_3 = D$ (viscosity and heat flux in the entire domain)



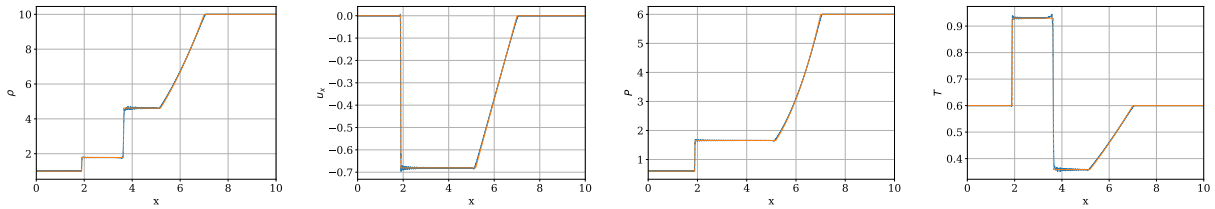
8. $C_0 = C_2^- = \frac{\gamma+1}{4}$, $C_1 = C_3^- = D$ (viscosity in the entire domain, and heat flux when $\nabla \cdot \boldsymbol{u} < 0$)



9. $C_0 = C_2^+ = \frac{\gamma+1}{4}$, $C_1 = C_3^+ = D$ (viscosity in the entire domain, and heat flux when $\nabla \cdot \boldsymbol{u} > 0$)

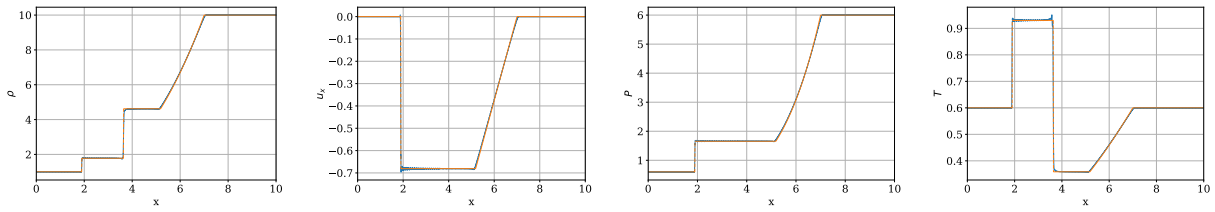


10. $C_0^+ = C_2 = \frac{\gamma+1}{4}$, $C_1^+ = C_3 = D$ (viscosity when $\nabla \cdot \boldsymbol{u} > 0$, and heat flux in the entire domain)

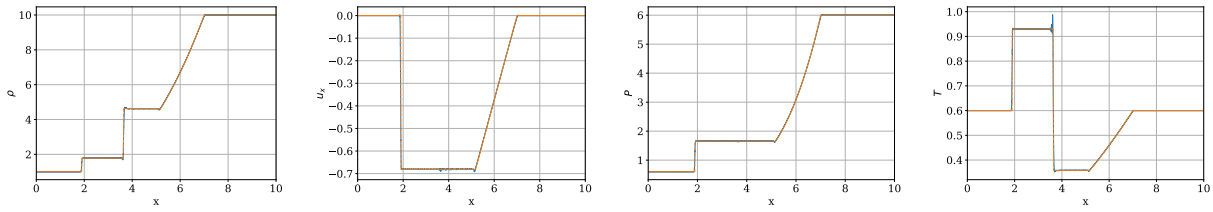


11. $C_0^+ = C_2^+ = \frac{\gamma+1}{4}$, $C_1^+ = C_3^+ = D$ (viscosity when $\nabla \cdot \boldsymbol{u} > 0$, and heat flux when $\nabla \cdot \boldsymbol{u} > 0$) - doesn't run

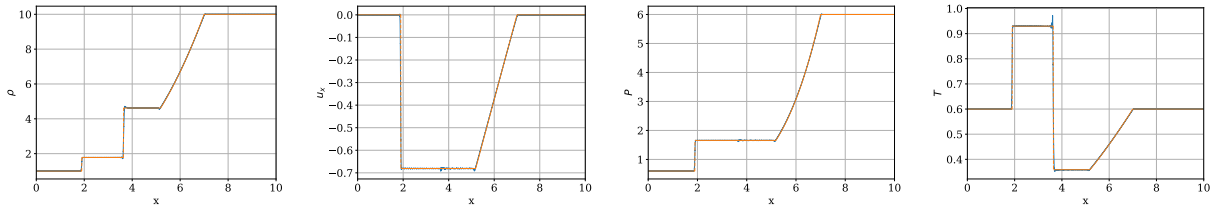
12. $C_0^+ = C_2^- = \frac{\gamma+1}{4}$, $C_1^+ = C_3^- = D$ (viscosity when $\nabla \cdot \boldsymbol{u} > 0$, and heat flux when $\nabla \cdot \boldsymbol{u} < 0$)



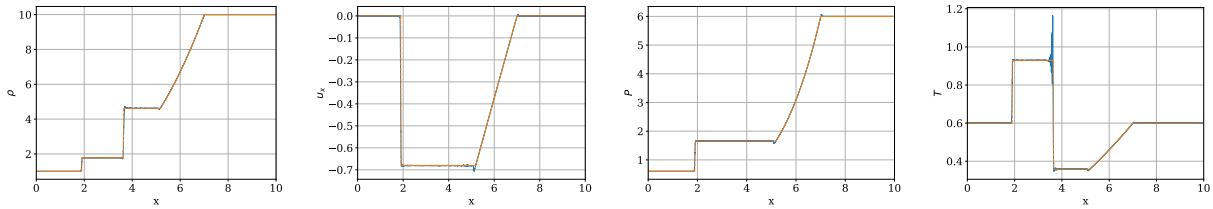
13. $C_0^- = C_2 = \frac{\gamma+1}{4}$, $C_1^- = C_3 = D$ (viscosity when $\nabla \cdot \boldsymbol{u} < 0$, and heat flux in the entire domain)



14. $C_0^- = C_2^+ = \frac{\gamma+1}{4}$, $C_1^- = C_3^+ = D$ (viscosity when $\nabla\cdot\boldsymbol{u} < 0$, and heat flux when $\nabla\cdot\boldsymbol{u} > 0$)



15. $C_0^- = C_2^- = \frac{\gamma+1}{4}$, $C_1^- = C_3^- = D$ (viscosity when $\nabla\cdot\boldsymbol{u} < 0$, and heat flux when $\nabla\cdot\boldsymbol{u} < 0$)



C.2 Effects of using density and internal energy floors

Consider the test parameters

$$\frac{\rho_R}{\rho_L} = 17 \qquad P = \frac{\rho}{\gamma} \qquad \gamma = \frac{5}{3} \qquad A = 1 \qquad \rho_L = 1$$

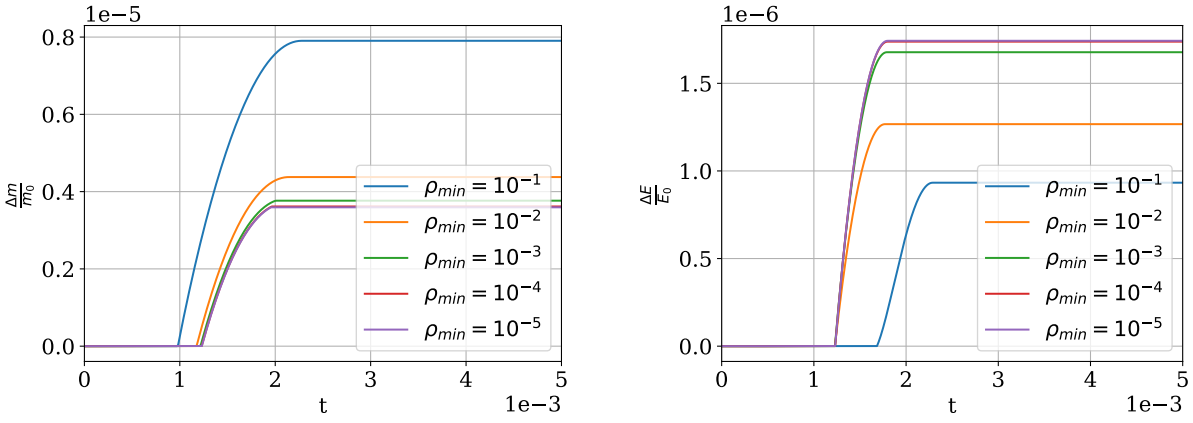
The domain is discretized with 400 equally sized elements, using a degree 3 polynomial basis. This is approximately the maximum shock jump which was observed to be possible using the artificial dissipation given by Eqs. (5.7)-(5.10) without using any artificial floors.

Varying levels of density and internal energy floors are applied to evaluate how much total mass and energy is introduced into the system. Two specific strategies for reducing the floors are tested:

1. Reducing the density floor, but keeping the internal energy floor at $U_{min} = 10^{-15}$

2. Reducing the density floor, and choosing the internal energy floor as $U_{min} = \frac{\rho_{min}}{10\gamma(\gamma-1)}$

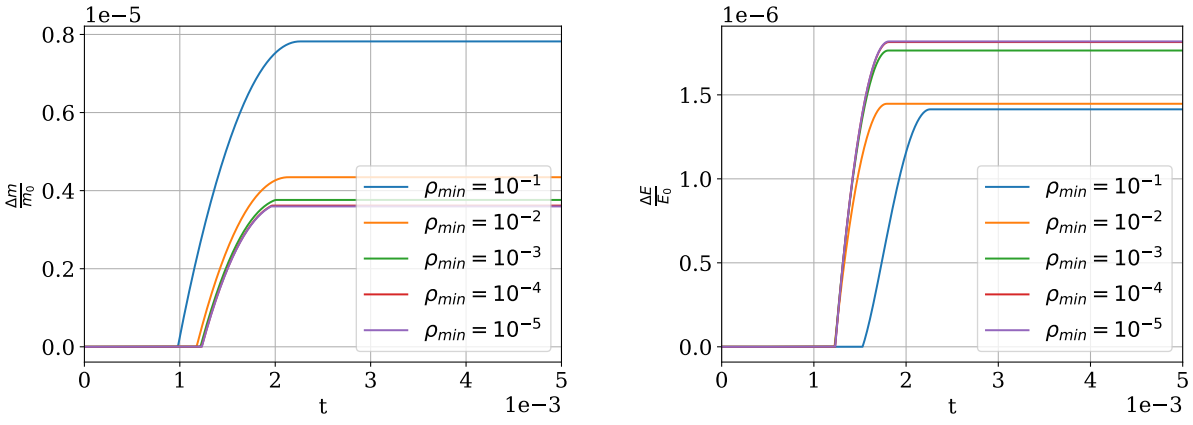
The density floor is chosen from the series $\rho_{min} \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The baseline uses no artificial floors what-so-ever.



(a) Relative change in mass

(b) Relative change in energy

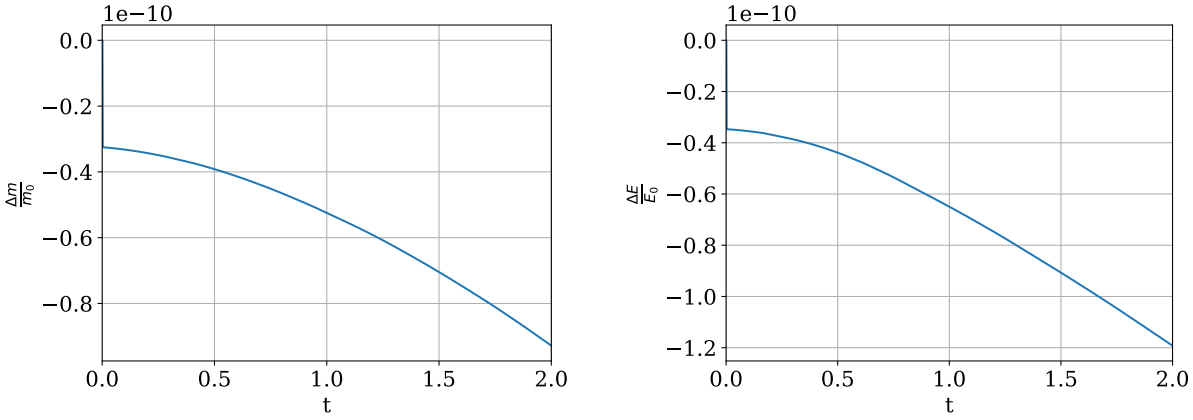
Figure C.13: $U_{min} = 10^{-15}$. Plots are zoomed in on the temporal scale to the only period when floors are actually being applied.



(a) Relative change in mass

(b) Relative change in energy

Figure C.14: $U_{min} = \frac{\rho_{min}}{10\gamma(\gamma-1)}$. Plots are zoomed in on the temporal scale to the only period when floors are actually being applied.



(a) Relative change in mass

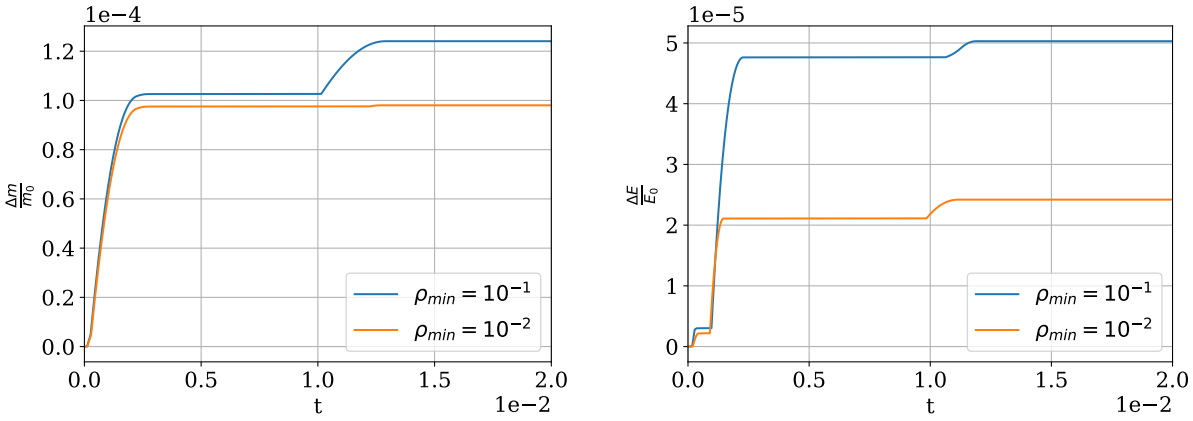
(b) Relative change in energy

Figure C.15: baseline

Despite orders of magnitude reductions in ρ_{min} and U_{min}, the amount of total mass or energy change appears to only be reduced by a relatively small amount, roughly 5 orders of magnitude below the initial total system mass total system energy. These levels are separated from the baseline of not using any artificial floors by 5 orders of magnitude respectively.

Avoiding the use of artificial floors is certainly desirable, however there may be cases where this isn't possible. This is tested by increasing the shock jump to test the limits where the combination of using artificial dissipation with artificial density/energy floors can be used.

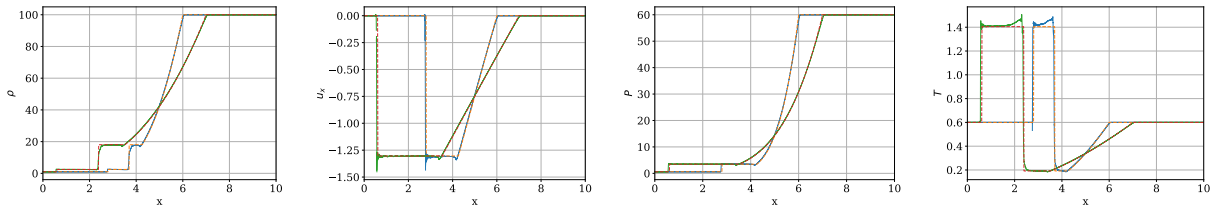
For a 100:1 jump, there is approximately a 0.01% increase in mass and 0.002% increase in energy when $\rho_{min} = 10^{-2}$ and $U_{min} = 10^{-15}$. The increase in mass is slightly higher when ρ_{min} is increased to 10^{-1} to 0.012%, however the total system energy increases by 0.005%. Comparing the simulation vs. the analytical solution demonstrates reasonable matching for the density and pressure profiles, with the exception of some slight rounding at the contact discontinuity for the density. However, there's a distinct anomalous oscillation in u_x at the shock front, as well as some strange increases in the temperature profile between the shock front and contact discontinuity.



(a) Relative change in mass

(b) Relative change in energy

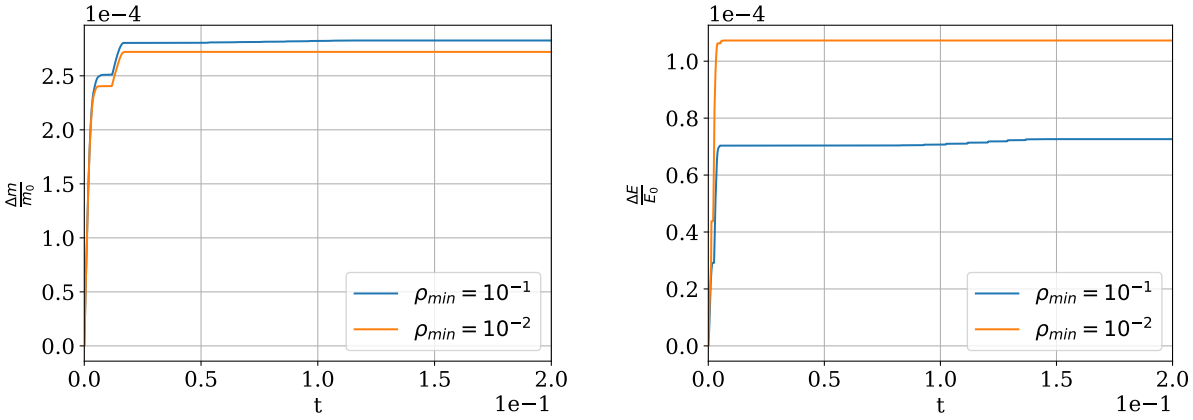
Figure C.16: Effects of artificial floors on conservation properties for a 100:1 Jump. $U_{min} = 10^{-15}$



(a) Density (b) u_x (c) Pressure (d) Temperature

Figure C.17: 100:1 Jump simulation comparison vs. the analytical solution, plotted at $t = [1, 2]$. Solid lines are simulation results, dashed lines are the analytical solution. $\rho_{min} = 10^{-2}$, $U_{min} = 10^{-15}$

For a 1000:1 jump, using the Q-H artificial dissipation model demonstrates significantly reduced performance. Attempts at using several different artificial floor levels including $\rho_{min} = 10^{-1}$ and $U_{min} = 10^{-15}$ were insufficient at producing a viable simulation with a degree 3 basis. Reducing the basis degree to 2 allows the simulation to run to completion, with approximately 0.01% increases in mass and energy.



(a) Relative change in mass (b) Relative change in energy

Figure C.18: Effects of artificial floors on conservation properties for a 1000:1 Jump with a degree 2 basis and Q-H artificial dissipation. $U_{min} = 10^{-15}$

Comparing the Q-H artificial dissipation simulation vs. analytical solution profile demonstrates significant rounding near the contact discontinuity, as well a lower than expected density/higher than expected temperature between the shock front and contact discontinuity. This higher than expected temperature results in a faster shock speed.

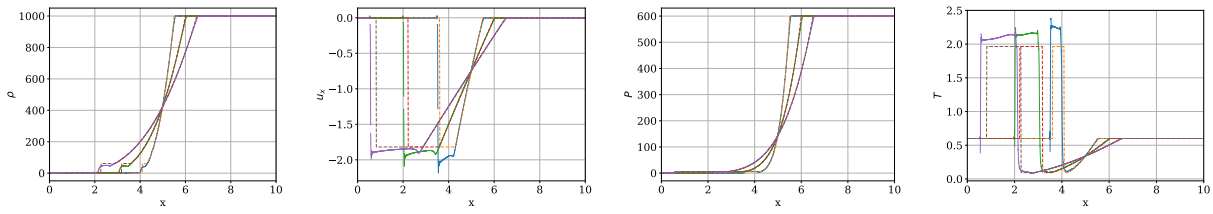
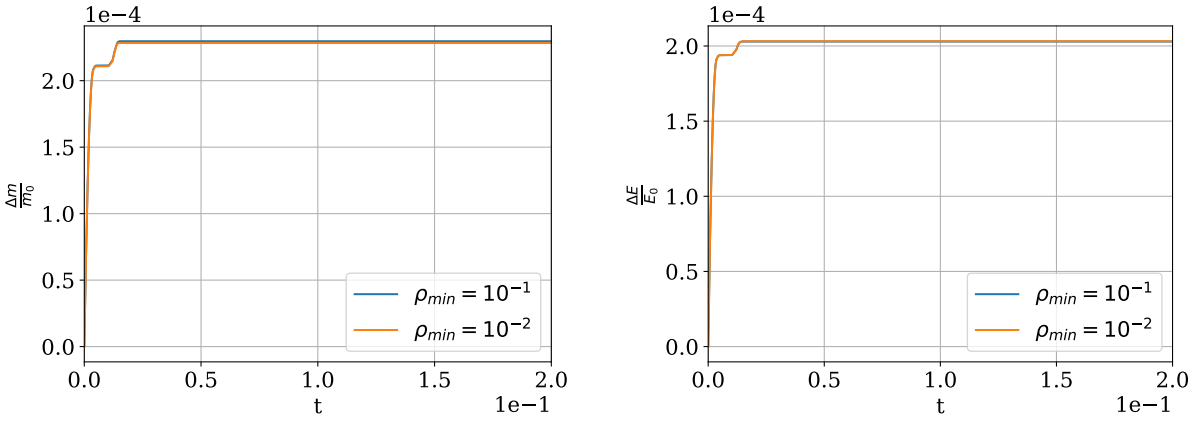


Figure C.19: 1000:1 Jump simulation with $Q-H$ artificial dissipation comparison vs. the analytical solution, plotted at $t = [0.5, 1, 1.5]$. Solid lines are simulation results, dashed lines are the analytical solution. $\rho_{min} = 10^{-2}$, $U_{min} = 10^{-15}$, degree 2 basis.

Using only artificial viscosity, there is a similar level of mass and energy increase compared to using $Q - H$ artificial dissipation with a degree 2 basis.



(a) Relative change in mass

(b) Relative change in energy

Figure C.20: Effects of artificial floors on conservation properties for a 1000:1 Jump with a degree 2 basis and artificial viscosity. $U_{min} = 10^{-15}$

The artificial viscosity simulation has similar deviations from the analytical solution as the Q-H artificial dissipation simulation, with slightly larger temperature oscillations at the contact discontinuity.

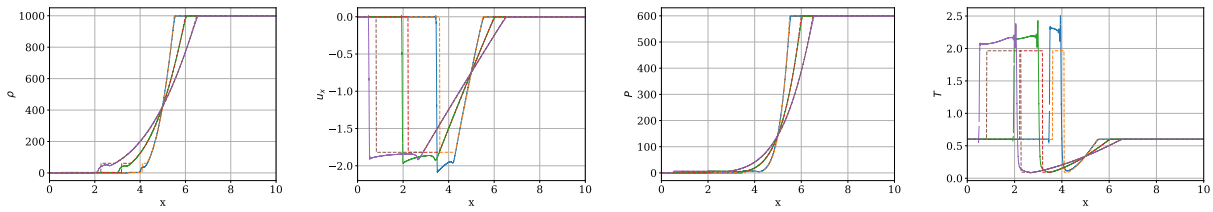


Figure C.21: 1000:1 Jump simulation with artificial viscosity comparison vs. the analytical solution, plotted at $t = [0.5, 1, 1.5]$. Solid lines are simulation results, dashed lines are the analytical solution. $\rho_{min} = 10^{-2}$, $U_{min} = 10^{-15}$, degree 2 basis.

Despite very similar results, applying artificial heat flux and artificial viscosity resulted in roughly 10x as many timesteps.

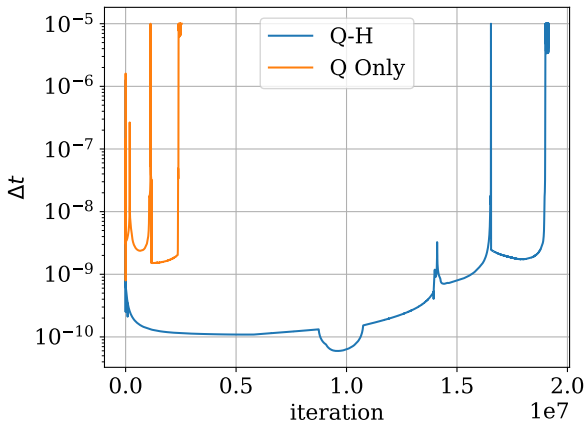


Figure C.22: Comparing allowed timestep for the Q-H artificial dissipation model vs. artificial viscosity. $\rho_{min} = 10^{-2}$, $U_{min} = 10^{-15}$, degree 2 basis.