NUMERICAL SIMULATIONS OF PLASMAS IN GALAXY CLUSTERS

By

Forrest W. Glines

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Astrophysics and Astronomy – Doctor of Philosophy

2022

ABSTRACT

NUMERICAL SIMULATIONS OF PLASMAS IN GALAXY CLUSTERS

By

Forrest W. Glines

As the largest gravitationally bound objects in the universe, galaxy clusters are a unique probe of large scale cosmological structure. Determining the distribution of galaxy clusters and their virial masses may be key to constraining properties of dark energy and dark matter. Since 90% of a typical galaxy cluster's mass is comprised of non-radiating dark matter, however, determining the virial mass of galaxy clusters depends on inference from the radiating baryonic matter. 90% of this baryonic matter is contained in the intracluster medium (ICM) – a hot, diffuse, magnetized plasma permeating the galaxy cluster. While the baryonic matter is the only emitter of observable electromagnetic emissions from galaxy clusters, the complex behavior of the ICM as a turbulent magnetized plasma makes constraining the virial mass of the cluster with observable signatures difficult. Numerical simulations are essential tools for advancing understanding of the ICM and for tying galaxy cluster observables to virial masses. *The goal of this dissertation is to explore and enable simulations of galaxy clusters and magnetized plasmas via a number of different avenues.*

I first explore self-regulation of feedback from active galactic nuclei (AGN) preventing over-cooling in cool-core (CC) clusters – galaxy clusters with anomalously high central thermal emission which should cool on shorter timescales than they persist. In the idealized galaxy cluster simulations with a thermal abstraction of AGN feedback, we find that the thermal-only heating kernels we test are unable to offset cooling while maintaining a realistic structure, suggesting exploration of more complex AGN feedback mechanisms such as those including magnetic fields and turbulence.

We then explore how kinetic and magnetic energy thermalizes in the ICM by studying decaying magnetized turbulence with simulations of the magnetized compressible Taylor-Green vortex. Using a shell-to-shell energy transfer analysis, we find that the magnetic fields facilitate a significant amount of the energy flux that is not seen in hydrodynamic turbulence. Although the full cascade will not be directly captured in ICM simulations for the foreseeable future, higher resolution simulations enabled by larger computational resources can diminish such effects.

Different novel many-core architectures have emerged in recent years on the way toward larger supercomputers in the exascale era. Performance portability is required to prevent repeated non-trivial refactoring of a code for different architectures. To address the need for a performance portable magnetohydrodynamics (MHD) code, we combined ATHENA++, an existing MHD CPU code, with KOKKOS, a performance portable framework, into K-ATHENA to allow efficient simulations on multiple architectures using a single codebase. K-ATHENA has also inspired the PARTHENON performance portable adaptive mesh refinement (AMR) framework. Using this framework, we developed the performance portable AMR MHD code ATHENAPK.

Galaxy clusters contain significant magnetic fields, although their origin and role is still under investigation. Numerical modeling is essential for the inference of their properties. One aspect is whether magnetic AGN feedback models can self-regulate. I present work-in-progress simulations with ATHENAPK of magnetized galaxy clusters slated for exascale supercomputers later this year.

With the higher resolutions enabled by exascale systems, galaxy cluster simulations with relativistic jet velocities will be possible. Robust methods for relativistic plasmas will be needed. With this goal, I present a discontinuous-Galerkin (DG) method for relativistic hydrodynamics. We include an exploration of different methods to recover the primitive variables from conserved variables, a new operator for enforcing a physically permissible conserved state, and numerous tests of the method. This method has been used at Sandia National Laboratories to study terrestrial plasmas and will inform relativistic MHD methods for ATHENAPK.

Finally, I cover the future directions of the work in this dissertation, including the many codes enabled by PARTHENON, additions to the magnetized galaxy cluster simulations with ATHENAPK, and the large body of projects at Los Alamos National Laboratory to explore binary black hole mergers embedded within AGN accretion disks as a possible formation channel of the massive black holes observed by LIGO. The work in this dissertation to develop performance portable plasma simulations will enable ground-breaking simulations for years to come.

ACKNOWLEDGEMENTS

This work was completed at Michigan State University from 2016-2022 and at Sandia National Laboratories from 2019-2022. This research has been made possible by the Michigan State University Distinguished Fellowship, by NASA through Astrophysics Theory Program grant No. NNX15AP39G and Hubble Theory grant HST-AR-13261.01-A, by the NSF through grant AST-1514700, by the NCSA through the 2019 Blue Waters Graduate Fellowship, and by the Michigan Institute of Plasma Science and Engineering through the MIPSE Graduate Fellowship.

This dissertation has made extensive use of computing resources at the Michigan State University High Performance Computing Center (operated by the Institute for Cyber-Enabled Research), the NASA Pleiades supercomputer through allocation SMD-16-7720, the Blue Waters Supercomputer at the National Center for Supercomputing Applications (NCSA), the Texas Advanced Computing Center (TACC) at The University of Texas at Austin, at OLCF on Titan through allocation AST133, on Summit through allocation AST146, on ALCF Theta through allocation athena_performance, and on XSEDE Comet through allocation TG-AST090040.

The work in this dissertation has benefited extensively from the work on AGN feedback by Greg Meece (Meece et al., 2017), the Athena++ code (Stone et al., 2020a), the Kokkos library (Carter Edwards et al., 2014; Trott et al., 2022), and the Parthenon library (Grete et al., 2022).

I give thanks to the the MSU Department of Physics and Astronomy and the Department of Computational Mathematics, Science and Engineering for fostering a supportive environment. Thank you to the many graduate students for welcoming me to Lansing and making me feel comfortable at MSU, including the previous cohort of students who welcomed me to MSU including Austin Edmister, Rachel Frisbie, Dana Koeppe, and Jennifer Ranta; my fellow cohort whom I came to love MSU including Jessica Maldonado and Carl Fields; the cohorts following me who made MSU home and whom I wish the best including Justin Grace, Adam Kawash, Claire Koppenhaufer, Michael Pajkos, Brandon Barker, Eric Britt, CJ Llorente, Teresa Panurach, and Joshua Shields.

Thanks to Kim Crosslan for keeping the Physics and Astronomy department running smoothly

and helping me through the administrative hurdles over graduate school.

Thanks to the postdoctoral researchers Elias Aydi, Brian Clark, Chelsea Harris, Sumit Sarbadhicary, and and Abbie Stevens and professors Wolfgang Kerzendorf and Daniel Hayden to whom I've gone to for academic and professional guidance moving onto the the postdoctoral stage and beyond.

Thank you to the members of my committee, Sean Couch, Tyce DeYoung, Megan Donahue, and Mark Voit for providing guidance and direction for my dissertation research.

Many thanks to the incredible computational structure research group, especially Deovrat Prasad, whose discussions and insights into contemporary galaxy cluster research have been essential for my understanding of the field, and Philipp Grete, whose contributions to computing have been instrumental to our successes with K-ATHENA, PARTHENON, ATHENAPK, and my future work with Los Alamos National Laboratory, and whose fervent pursuit of computational plasma research I aspire to match.

Special thanks you to Kristian Beckwith who helped me have a fulfilling internship, stuck with my research projects, and brought them towards publication despite all obstacles.

Special thanks to my new found friends in and around Michigan who have helped me discover and embrace my identity.

Special thanks to my family and parents, who have encouraged me throughout my graduate career when I have been supported me especially when I felt inadequate.

Special thanks above all to my advisor, Brian O'Shea, whose endless patience with me to pick a topic of study to match all my interests in astrophysics, plasmas, and computing. Without his continual support in academic pursuits, professional aspirations, and personal trowth this dissertation would not be possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Galaxy Clusters

Galaxy clusters are the largest gravitationally bound objects in the universe, beyond which the expansion of space due to dark energy exceeds gravity (Longair, 2008; Mo et al., 2010). With virial masses on the order of $10^{14} - 10^{15}$ $\mathrm{M}_{\odot}$ and radii ~ 1 Mpc, by mass they are primarily composed of dark matter – typically 90% of a galaxy cluster's mass is contained in a dark matter halo. The remaining 10% is baryonic matter, 90% of which is contained in the intracluster medium (ICM), a hot diffuse X-ray emitting plasma permeating the cluster with temperatures on the order of $1 - 10$ keV ($10^7 - 10^8$ K) and particle densities on the order of $10^{-4} - 10^{-2}$ cm^{-3}. The remaining 10% of the baryonic matter, constituting 1% of the total galaxy cluster mass, is contained within 10-100 galaxies (Longair, 2008; Mo et al., 2010).

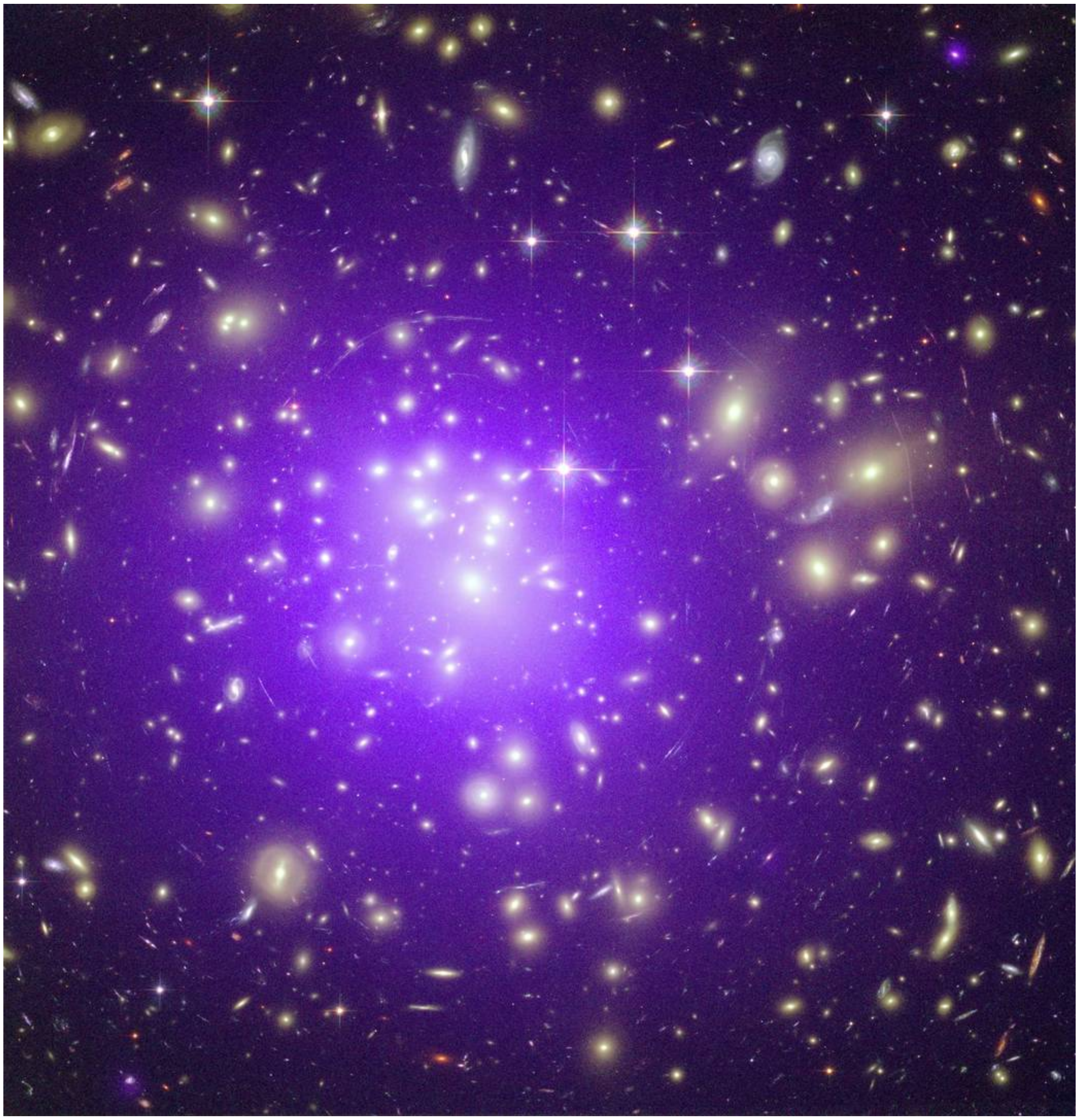
In their unique role as the largest gravitationally bound objects in the universe, galaxy clusters serve as key probes of cosmological properties of the universe (Lima et al., 2003; Wang et al., 2004; Basilakos et al., 2010; Pratt et al., 2019; Allen et al., 2011). Specifically, they trace the structure of the largest overdensities of dark matter, revealing the power spectrum of mass distribution through the universe on the largest scales. Determining this structure is essential for characterizing an equation of state for dark energy (Lima et al., 2003), the observed but as yet relatively uncharacterized force that drives the accelerating expansion of the universe. More precisely, we need the number density of galaxy clusters as a function of their virial mass and redshift (see Voit, 2005; Allen et al., 2011, for a thorough review).

However, virial masses are not directly measured, but instead must be inferred by observable properties such as gravitational lensing and the electromagnetic radiation emitted by the baryonic matter.

The most straightforward method to determine a galaxy cluster's mass is via *strong gravitational* *lensing*, where the mass of the galaxy clusters (primarily its dark matter halo) bends the trajectory of light from a background source behind the galaxy cluster following General Relativity (Kochanek, 2006; Hoekstra et al., 2013; Bartelmann, 2010), creating multiple images of the background source around the galaxy cluster. Although strong gravitational lensing can be used to determine galaxy cluster masses with minimal assumptions, it requires a background source directly behind the cluster and must be near enough to observe the multiple images, thus limiting its application to a small number of systems. Strong lensing is also only useful for estimating the mass near the cluster core, so the virial mass of the galaxy cluster still needs to be inferred (Hoekstra et al., 2013). In contrast, *weak gravitational lensing* from the deflection of light in the entire sky by multiple sources gives a statistical measure of the distribution of mass in the universe (Bartelmann & Schneider, 2001).

The virial masses and number densities of galaxy clusters can also be determined from multi-wavelength observations – the electromagnetic radiation emitted by the baryonic matter in galaxy clusters and observed in multiple wavelengths. Of particular interest to galaxy clusters and the ICM, the X-ray emission from the hot diffuse ICM measures the gas density and temperature (see Figure 1.1), which can be used to estimate the galaxy cluster mass assuming the gas is in hydrostatic equilibrium (HSE; Sarazin, 1988; Allen et al., 2011). However, the ICM is disrupted from HSE by AGN feedback, magnetic fields, turbulence, cosmic ray pressure, and any other non-thermal support (Fabian et al., 2003; Carilli & Taylor, 2002; Dennis & Chandran, 2005; Loewenstein et al., 1991). Likewise, the optical emissions from galaxies within the galaxy cluster can be used to estimate the cluster mass by assuming dynamical equilibrium (Binney & Tremaine, 1987; Carlberg et al., 1997). Galaxies can similarly be disrupted from dynamical equilibrium by large scale structure interactions in the universe (White et al., 2010). The Sunyaev–Zeldovich (SZ) effect – the upscattering of the cosmic microwave background (CMB) to higher energies via inverse Compton scattering with high-energy electrons (Sunyaev & Zel'dovich, 1980) – can also be used to estimate gas density and temperature of the electron population of galaxy clusters.

Although assumptions of HSE and dynamical equilibrium can be used to coarsely estimate



Abell 1689 in X-ray: NASA/CXC/SAO/E.Bulbul, et al. and Optical: NASA/STScI

Figure 1.1: Galaxy Cluster Abell 1689 in X-ray (purple) as captured by Chandra with optical from Hubble underneath. The galaxy cluster has sufficient mass to bend light from background galaxies around the galaxy cluster core, smearing background sources into duplicated arcs around the galaxy cluster core. This strong gravitational lensing permits estimates of the galaxy cluster's mass (Kochanek, 2006; Hoekstra et al., 2013; Bartelmann, 2010). The Intracluster Medium – the hot, diffuse plasma comprising most of the baryonic mass but a relatively smaller portion of the total mass – is responsible for the majority of the X-ray emissions.

galaxy cluster masses, more precise mass proxies relying on electromagnetic observables depend on more precise understanding of the dark matter and ICM out of HSE and dynamical equilibrium. Numerical simulations of both components of galaxy clusters individually and simultaneously have been essential for recent improvements of galaxy cluster mass proxies (Pratt et al., 2019). N-body simulations of the dark matter halos of galaxy clusters can inform more realistic halo mass profiles (Navarro et al., 2004; Gao et al., 2012). Galaxy cluster simulations including the complex plasma physics of the ICM, however, are rapidly evolving (Walker et al., 2019), with the magnetized plasma nature of the ICM and AGN feedback under particular recent scrutiny (Donnert et al., 2018; Morganti, 2017).

Understanding the ICM is key to developing accurate mass proxies to discern the the virial mass of galaxy clusters in large X-ray surveys that can characterize the structure of dark matter in the universe and the equation of state of dark energy. At present, the forefront of our understanding of galaxy clusters is limited by our understanding of the ICM as a complex plasma.

1.2 Plasmas

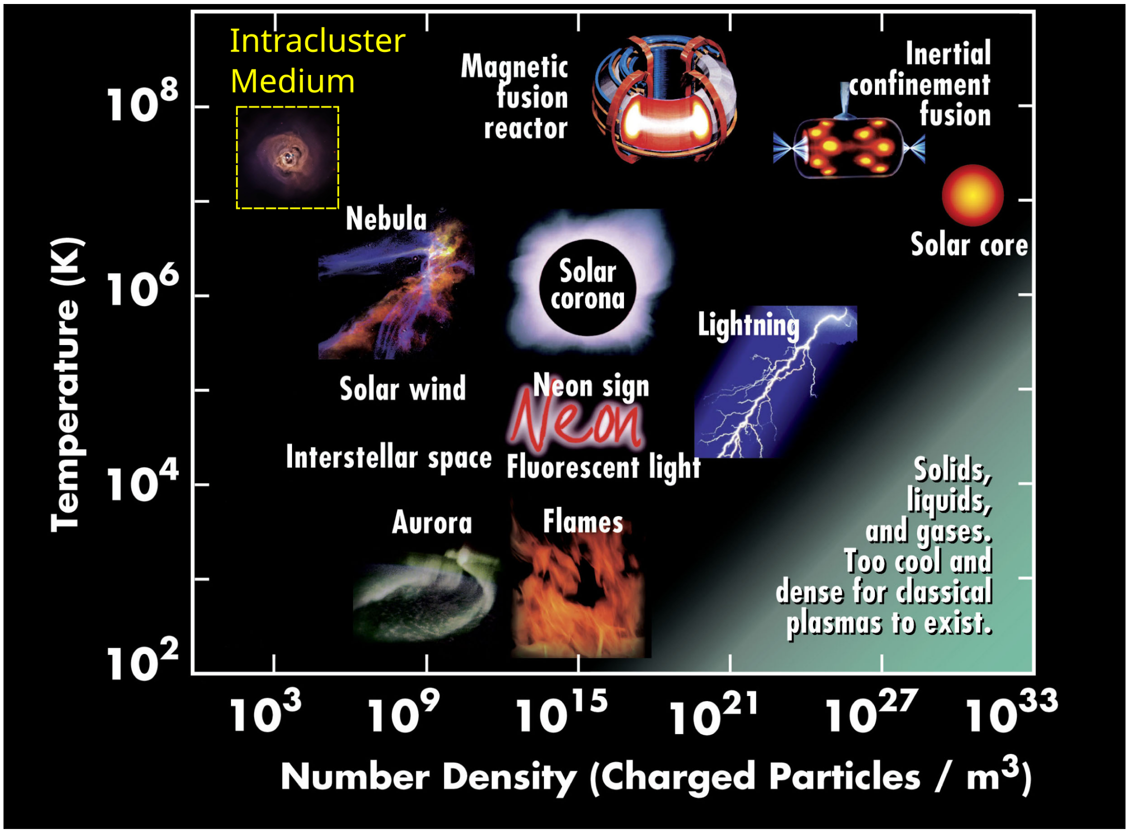
“Plasma” is a state of matter where a portion or all of the electrons are decoupled from the ions, creating a sea of charged particles (Chen & Chen, 1984; Bittencourt, 2004; Bellan, 2008). These charged particles facilitate currents and thus magnetic fields within the matter. The bulk kinetic motion of the charged particles exerts forces on these currents and magnetic fields and vice versa, leading plasmas to exhibit behaviors unseen in other states of matter. With these properties, plasmas can behave quite differently from unionized baryonic matter.

Although rare on Earth, plasmas are ubiquitous in the universe, comprising the vast majority of baryonic matter. We can divide most plasmas into two broad categories: terrestrial plasmas, which occur or are created on Earth, and astrophysical plasmas, which occur beyond the Earth’s atmosphere.[1]

[1] Plasmas in the upper Earth’s atmosphere and magnetosphere are known as space plasmas and have characteristics of both terrestrial plasmas and astrophysical plasmas, being more diffuse and longer lived than terrestrial plasmas but not as hot as astrophysical plasmas (Baumjohann & Treumann, 2012; Treumann & Baumjohann, 1997).

Most terrestrial plasmas, except for naturally occurring lighting, are either created for industrial applications or for a wide variety of scientific experiments (Chen & Chen, 1984). Chief among these experiments are prototype fusion devices, including magnetic confinement fusion (MCF; Ongena et al., 2016) devices such as the International Thermonuclear Experimental Reactor (ITER; Aymar et al., 2002) where the plasma is confined by self-sustaining magnetic fields, and inertial confinement fusion (ICF; Craxton et al., 2015) devices such as the National Ignition Facility (NIF; Miller et al., 2004; Zylstra et al., 2022) and the Z-Machine (Sinars et al., 2020), where the fusion target is unconfined but a burst of energy from lasers or large currents heats the plasma quickly enough to allow inertia to confine the plasma for long enough to attain pressures and temperatures sufficient to undergo fusion. MCF plasmas are typically maintained for seconds to tens of seconds (Ongena et al., 2016), with the expectations for minutes-long lived plasmas being created for fusion devices in the near future, while ICF plasmas persist for picoseconds to microseconds (Zylstra et al., 2022).

Astrophysical plasmas, in comparison, are typically hotter, often more diffuse, and much longer lived (Chiuderi & Velli, 2015, see Figure 1.2 for examples of astrophysical and terrestrial plasmas). Stars, the interstellar medium (ISM) between them, and the ICM are all plasmas. Except for the dense plasmas found within compact objects such as stars, the majority of matter in the universe is within diffuse plasmas such as the ISM and ICM. These plasmas are also typically much longer lived than terrestrial plasmas, where present day temperatures of the ISM and ICM lead to partially or fully ionized plasmas. However, the dynamics of the fluid and the coupling with magnetic fields in astrophysical plasmas are governed by the same physical laws as terrestrial plasmas. Although not as physically accessible as a plasma created in a laboratory, the ubiquity and longevity of astrophysical plasmas allows convenient study of plasma physics via our observations of astrophysical plasmas. Thus, knowledge in plasma physics gained from studying astrophysical plasmas can improve understanding of terrestrial plasmas and vice versa.



Typical plasmas ©2022 Contemporary Physics Education Project
Perseus Cluster in X-ray: NASA/CXC/SAO/E.Bulbul, et al.

Figure 1.2: Charged particle number densities on the x-axis and temperatures on the y-axis for different astrophysical and terrestrial plasmas. The comparatively hot and diffuse plasma of the ICM is marked in yellow, with the Perseus cluster as seen in X-ray by Chandra. Diagram by the https://www.cpepphysics.org.

1.2.1 Plasma Regimes

The behavior of plasmas and the theories and equations that best describe them depend on many of the properties of the plasma system in question (see Figure 1.3; Kramer et al., 2020). These properties include the particle composition, the degree of ionization, the thermodynamics, the kinematics, the electrodynamics, the scale of the system of interest relative to other scales in the system, and countless other properties. Fortunately, to simplify categorization different plasma models can be broadly divided based on a few quantifiable properties.

Different models of plasmas can be broadly divided into kinetic and fluid methods based on the Knudsen number Kn of the plasma system

$$\mathrm{Kn} = \frac{\lambda}{L}, \tag{1.1}$$

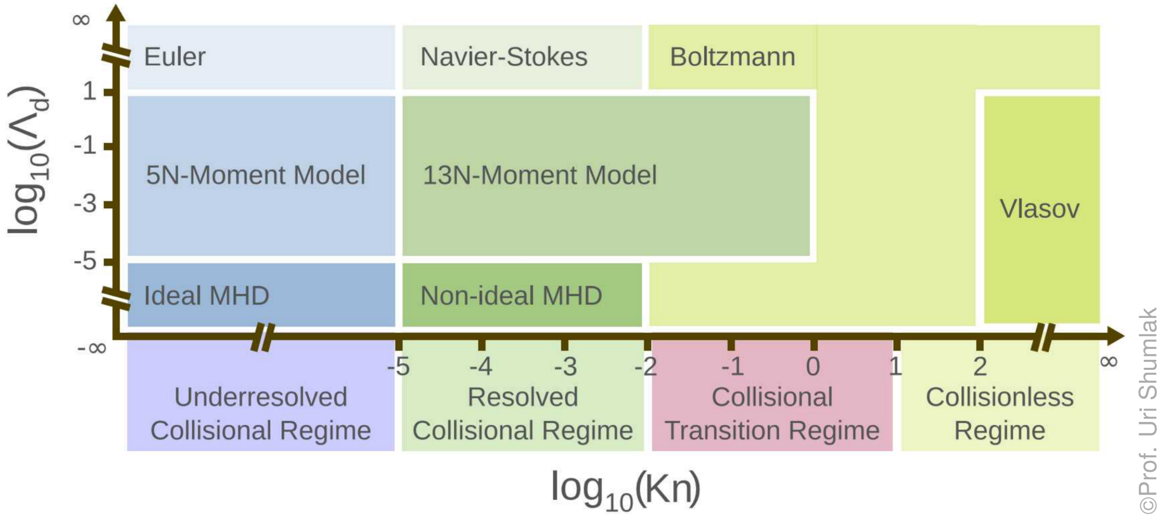


Figure 1.3: Spectrum of appropriate plasma models for different regimes, as determined by the Knudsen number Kn and the charge separation distance Λ_d. Fluid models appear to the left and kinetic models appear to the right while models where electromagnetics are important appear towards the bottom and models where electromagnetics unimportant appear towards the top. Systems and simulations explored within astrophysics typically use models from the 4 extremes: Euler, Boltzmann, ideal MHD, and Vlasov models. The plasma model best describing the ICM would be a non-ideal MHD model on the galaxy cluster scale and a Vlasov model on the plasma instability, particle acceleration scale. Created by Uri Shumlak for a presentation at Sandia National Laboratories (Shumlak, 2015) and appearing in Kramer et al. (2020).

which is the ratio of of the mean free path of particles λ to the length scale of interest L. The Knudsen number depends on the size of the system examined – i.e. Mpcs for the plasma comprising galaxy clusters. Smaller size systems exist, however, within the larger system. For example, in the ICM plasma instabilities and particle acceleration across shocks and turbulence happen at a much smaller scale (on the order of km to pc), whereas the mean free path of particles is the same, resulting in a larger Knudsen number a system studying plasma instabilities within the ICM versus studying the ICM of an entire galaxy cluster (Marcowith et al., 2020)[2]. Although the particle acceleration physics with high Knudsen numbers still occur within the larger physics of the ICM

[2] The effective Knudsen number of the ICM is complex since the mean free path of particles via Thompson scattering in the ICM (10 – 1000 kpc) is significant to the size of the galaxy cluster, but the length scale on which plasma instabilities can introduce dissipation (~ km) is quite small. Thus, the applicability of fluid models appropriate for small Knudsen numbers to the ICM is under debate. These issues may be addressed using non-ideal MHD (Kunz et al., 2011; Schekochihin et al., 2009).

of a galaxy cluster, their effects are typically secondary to the larger scale dynamics.

Even though the mean free paths and length scales of astrophysical and terrestrial plasmas can differ greatly, their ratio and thus Knudsen numbers can be quite similar, allowing them to be studied with shared models. For example, while the km-scale plasma instabilities in the ICM happen on much larger scales than plasma instabilities in magnetically insulated transmission lines (MITL) used to deliver power to accelerators (Ottinger & Schumer, 2006; Kramer et al., 2020; Luo et al., 2019), both systems have similar Knudsen number and thus can be studied with similar plasma models.

High Knudsen number plasmas are best described with kinetic theory, where the plasma is described by a statistical distribution of particles in phase space. Each particle species of the plasma is described by a density function in phase space that evolves over time. Following Kramer et al. (2020), let $N_s(\mathbf{r}, \mathbf{v}, t)$ be the phase density function containing every particle in the plasma of species s, where $\mathbf{r}$ is a position, $\mathbf{v}$ is a velocity, and t is a time. The microscopic evolution of this density functions is exactly described the Klimontovich equation (Klimontovich, 1994)

$$\frac{\partial N_s}{\partial t} + \mathbf{v} \cdot \frac{\partial N_s}{\partial \mathbf{r}} + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial N_s}{\mathbf{v}} = 0 \tag{1.2}$$

where q_s and m_s are the particle species charge and mass and where $\mathbf{E}$ and $\mathbf{B}$ are the local microscopic electric and magnetic fields which are governed by Maxwell's equations. The density function N_s encompasses every individual particle of the plasma, however, which is rarely useful or tractable for modeling whether by theory or numerical simulation.

If we instead consider a probability distribution function (PDF) $f_s(\mathbf{r}, \mathbf{v}, t)$ of each particle species and consider an averaged macroscopic electric and magnetic field, we obtain the Boltzmann equation (Chen & Chen, 1984; Bittencourt, 2004; Bellan, 2008)

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial N_s}{\partial \mathbf{r}} + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial N_s}{\mathbf{v}} = \left. \frac{f_s}{\partial t} \right|_{\text{Coulomb}}, \tag{1.3}$$

where the rightmost term is a source and sink term for Coulomb collisions. Specific operators for the Coulomb collision terms give the Fokker-Planck equation and the Vlasov equation. The particles are coupled to the electromagnetic fields via Maxwell's equations, which can be written in Lorentz-Heaviside units as

$$\frac{1}{c}\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{B} - \frac{4\pi}{c}\mathbf{J} \tag{1.4}$$

$$\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{1.5}$$

$$\nabla \cdot \mathbf{E} = q \tag{1.6}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{1.7}$$

where the current and charge densities are defined as

$$\mathbf{J} \equiv \sum_s q_s n_s \mathbf{v} \tag{1.8}$$

$$q \equiv \sum_s q_s n_s \tag{1.9}$$

and n_s is the zeroth moment of the distribution function,

$$n_s = \int f_s d\mathbf{v}. \tag{1.10}$$

Examples of high Knudsen number systems in astrophysics include the microphysics of particle acceleration via shocks and magnetized turbulence to create cosmic rays and the magnetosphere surrounding many planets.

Since the equations in kinetic theories have high dimension – 6D PDFs are needed for each species even with the statistical simplifications used for the Boltzmann and Vlasov equations – numerical approaches are often limited. Monte Carlo (MC) methods (Metropolis et al., 1953), which rely on random sampling to approximate solutions, are generally more useful for highly dimensional systems compared to other methods such as finite volume or finite element (see Section 1.2.4 and Humpherys et al., 2017). The most widely used method for kinetic theories is the Particle-in-Cell method (PIC), where the distributions are species are randomly sampled by super-particles representing populations of particles that are then used to approximate electromagnetic fields across a mesh of cells (Harlow et al., 1955; Dawson, 1983; Tskhakaya et al., 2007). The electromagnetic fields are then used to update the positions and velocities of the super-particles, evolving the fields and then particles with a leapfrog integration method. As an MC method, PIC converges slowly with increased super-particle count, improving accuracy at a $n^{1/2}$ rate where n is the number of super-particles (Myers et al., 2016). For large systems or long evolution times this slow convergence makes PIC a resource-intensive, cumbersome, and sometimes infeasible computational method, depending on the system of interest (Harlow, 1962; Liu et al., 2019). Fortunately, larger systems necessarily mean smaller Knudsen numbers, for which more computationally amendable approaches exist.

Low Knudsen number plasmas are best described with fluid theories, assuming continuum particle distributions in thermodynamic equilibrium (or close to thermodynamic equilibrium, with corrections). Although the kinetic theories and associated equations are still valid for low Knudsen number plasmas, their high dimensionality leads us to use approximations of these equations that are appropriate for a continuum particle distribution. We assume a thermodynamic distribution of the fluid, such as the Maxwell-Boltzmann distribution, which implies thermodynamic equilibrium, so that PDFs are not directly evolved. Taking the first three moments from the Boltzmann equation – multiplying Equation 1.3 by m_s, $m_s\mathbf{v}$, and $m_s v^2/2$ respectively and integrating over all velocity space – yields equations for conservation of mass, momentum, and energy (Kramer et al., 2020; Bittencourt, 2004). If we apply these for a single species non-relativistic fluid, ignoring other iteractions such as viscosity and electromagnetic fields, we obtain the Euler equations (Toro, 2009; Chen & Chen, 1984; Bittencourt, 2004; Bellan, 2008):

$$\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} = 0 \tag{1.11}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p = 0 \tag{1.12}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \mathbf{v} \left(\varepsilon + p \right) = 0 \tag{1.13}$$

where ρ is the density, $\mathbf{v}$ is the flow velocity, p is the pressure, ε is the energy density including kinetic and thermal contribution, and I is the identity matrix. A viscosity stress tensor can be added to the momentum equation to give the Navier-Stokes equations.

Electromagnetic field can be coupled to the Euler equations via Maxwell's equations to give models that better suit plasmas, where electromagnetic fields can influence the medium. In the *ideal plasma* limit, where currents are instantaneous and resistance is zero (leading to zero electric fields), we get the ideal magnetohydrodynamics (MHD) equations (Toro, 2009; Bittencourt, 2004; Bellan, 2008):

$$\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{v} = 0 \tag{1.14}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} - \mathbf{B} \otimes \mathbf{B}) + \nabla \left(p + B^2/2\right) = 0 \tag{1.15}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left[\mathbf{v}\left(\varepsilon + p + B^2/2\right) - \mathbf{B}\left(\mathbf{B} \cdot \mathbf{v}\right)\right] = 0 \tag{1.16}$$

with only two components remaining from Maxwell's equations due to vanishing electric fields

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) \,. \tag{1.17}$$

Although the ideal MHD equations provide a good model for many plasmas, they can be extended to include a variety of second order plasma effects.

Including other electromagnetic effects leads to non-ideal MHD equation sets. Resistivity can be included in this model via Ohm's Law to arrive at the resistive MHD equations, which support magnetic reconnection in the modeled plasma, while including anisotropic diffusion and thermal conduction along magnetic field lines gives Braginksii MHD (Braginskii, 1965).

The appropriate kinetic or fluid approximation depends on both on the Knudsen number and on the charge separation distance

$$\Lambda_d \equiv \frac{k_D}{L} \tag{1.18}$$

which is the degree to which electric fields are relevant to the plasma, and where k_D is the Debye length

$$k_D^2 \equiv \frac{4\pi n q^2}{k_B T}, \tag{1.19}$$

which is the how far the charged particles' comprising the plasma net electrostatic effect persists, where n is the number density of the particles, q is their elementary charge, k_B is the Boltzmann constant, and T is their temperature. If the Debye length is small compared to the system size then the system is *electrically well-screened*, so that the electric fields from discrete charges are unimportant compared to macroscale electromagnetic fields (Bittencourt, 2004; Bellan, 2008). In the extreme high Λ_d and low Knudsen number limit the fluid is neutral, and standard fluid dynamics governed via Euler's equations are relevant (Kramer et al., 2020). As the Knudsen number is increased and the dissipation scale becomes closer to the system scale, the Navier-Stokes equations become more appropriate. In the low Λ_d and low Knudsen number limit the plasma is an ideal plasma where the ideal MHD equation are best applicable. As the system size is shrunk dissipation scales and small scale plasma instabilities become more relevant, leading to non-ideal MHD approximations such as resistive MHD (Bonafede et al., 2011) and Braginskii MHD (St-Onge et al., 2020) becoming more applicable.

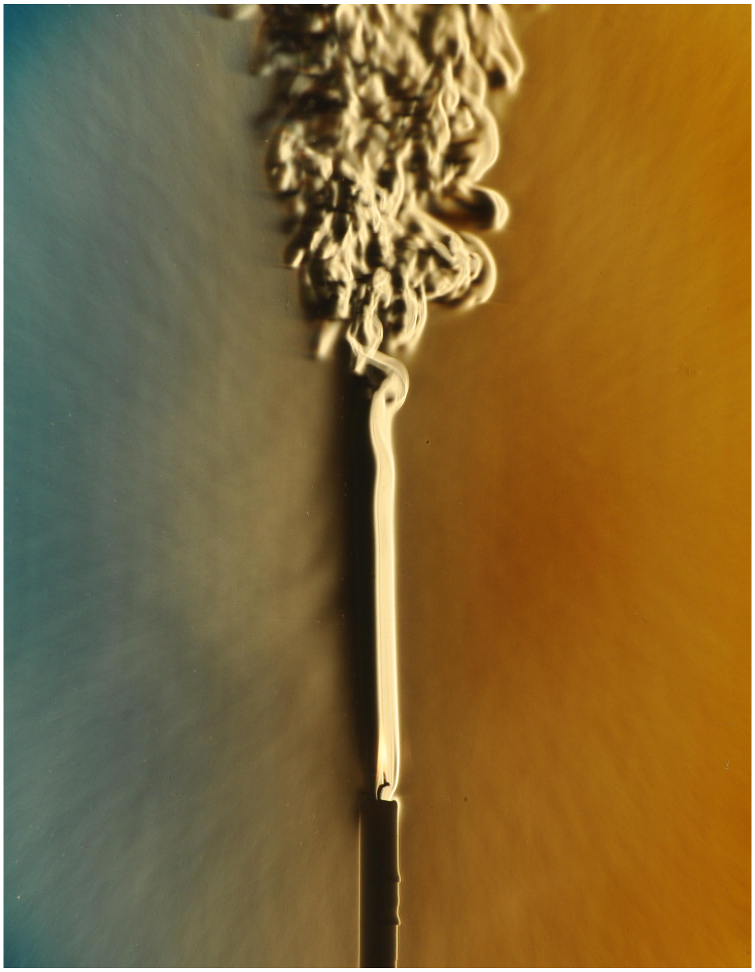
1.2.2 Turbulence in Plasmas

Turbulence is the chaotic flow, density, and pressure structures that form in all fluids when the kinetic or magnetic energy in the fluid exceeds dampening due to viscosity, which is the internal friction or resistance to flow within the fluid (see Figure 1.4; McComb, 1990). Being formally chaotic, the evolution of turbulent flows cannot be predicted exactly, but are better understood statistically on a macroscopic and microscopic level.

The onset of turbulence in a fluid can be predicted by the dimensionless *Reynolds number* (Stokes, 1851; Sommerfeld, 1909; Reynolds, 1883; Rott, 1990), which is defined as

$$\mathrm{Re} \equiv \frac{vL}{\nu} \tag{1.20}$$

where v is the fluid velocity, L is the characteristic length scale that depends on the size of the system examined, and ν is the kinematic viscosity. Although the transition point is fuzzy and depends on the fluid, flow structure, and system in question, fluids with a Reynolds number above $10^3 - 10^4$ exhibit instabilities in smooth (*laminar*) flows that disrupt them into turbulent flows (see Figure 1.5, Incropera & DeWitt, 1981). In terms of fluids encountered in everyday life, air has a low viscosity and thus higher Reynolds numbers for similar velocities and scales compared to water and honey, which have comparatively higher viscosities and thus lower Reynolds number and are less prone to turbulent flows. Viscosity in both liquids and gases arise from molecular interactions but the origin of these forces can be quite different (Bird et al., 2006). As relevant to this dissertation, viscosity within gases arises primarily from molecular diffusion, where the relevant scales are on



©Dr. Gary Settles

Figure 1.4: Schlieren photograph showing the thermal plume of a lit candle, showing the smooth rising flow starting from the base of the flame that transitions into turbulence at the top of the flame. As a gas, the viscosity in smoke and air is low; thus, the velocity of the uplifted heated gas is sufficient to create a high Reynolds number flow, with Re $\gtrsim 10^3$, which is prone to fluid instabilities. The laminar flow originating from the flame decays into turbulence as these instabilities grow further down the flow.

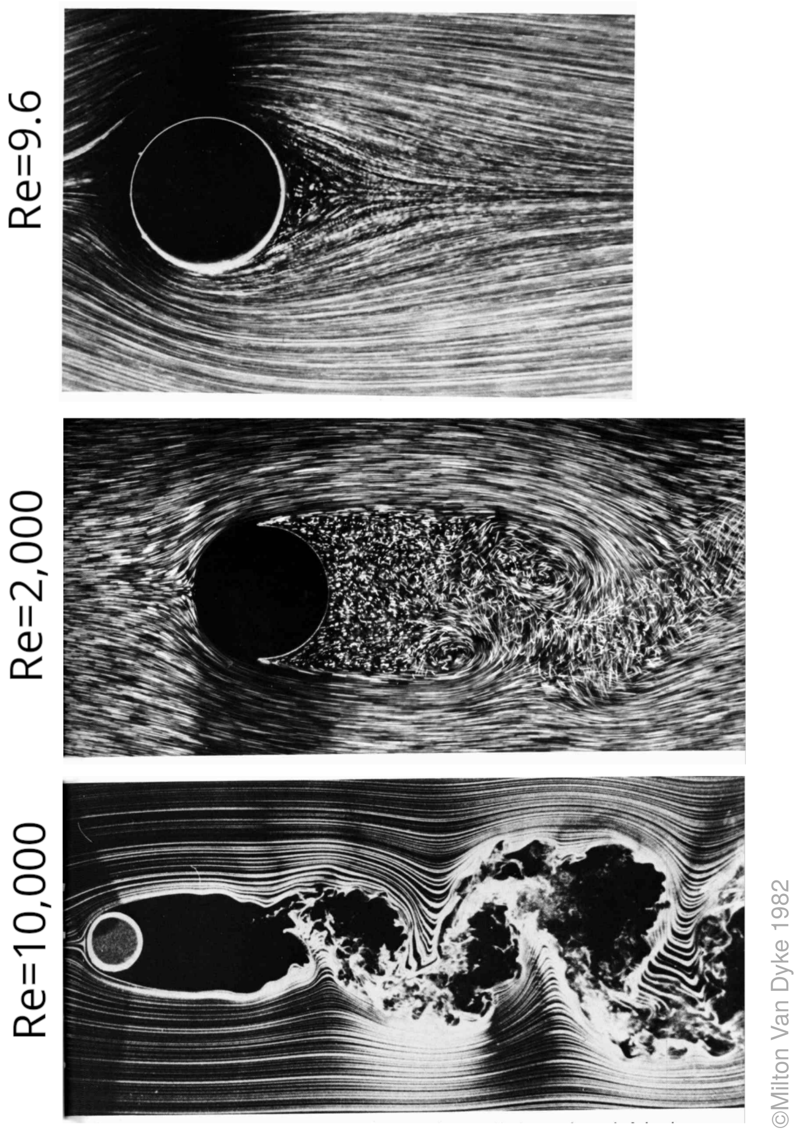


Figure 1.5: Photographs of a cylinder moving through a tank of water containing aluminum powder (van Dyke, 1982). The higher the velocity of the water flow relative to the cylinder the higher the Reynolds number, showing flows from top to bottom with Re = 9.6, Re = 2,000, and Re = 10,000. As the Reynolds number is increased beyond $\sim 10^3$, the flow becomes prone to fluid instabilities which grow non-linearly as the flow moves past the cylinder. These instabilities develop into the turbulent flow beyond the cylinder, as best seen on the right hand side with the Re = 10,000 flow.

the order of the mean free path of particles. This length scale at which dissipation becomes relevant is known as the *dissipation scale* or the *Kolmogorov length scale*. Since the system scales of gases studied are often much larger than the dissipation scale, the Reynolds number is often quite high for gas systems and thus they are usually turbulent. Since viscosity serves as a dampening force against kinetic flow, it converts macroscopic kinetic energy in the fluid into thermal energy at the dissipation scale.

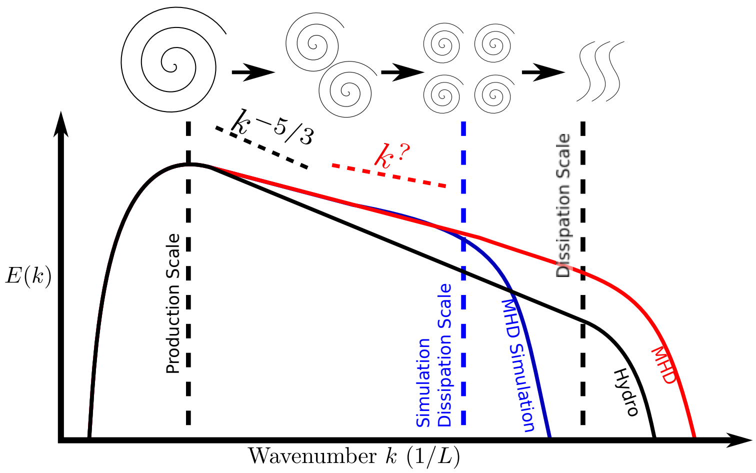


Figure 1.6: Diagram of the energy spectra of a turbulent plasma denoting the hydrodynamic turbulent cascade and the effects of magnetic fields and limited simulations resolution on the energy spectra. Wavenumber increases along the $x-$axis, with larger length scales to the left and smaller length scales to the right. Energy contained in the plasma at a certain wavenumber is plotted along the $y-$axis. The black solid line shows the kinetic energy spectrum of a plasma with no magnetic fields, where kinetic energy is introduced into the plasma at the production scale (marked by the leftmost vertical dashed black line) and dissipates into thermal heating at the dissipation scale (marked by the rightmost vertical dashed black line). Between these scales, turbulent plasmas follow a $k^{-5/3}$ power law in the kinetic energy spectrum. With the addition of magnetic fields, in the resulting kinetic energy spectrum (shown in red) the power law is flattened or broken, with more energy at smaller scales. In simulations without an explicit viscosity, the smallest cell size introduces a dissipation length scale (the vertical dashed blue line) potentially larger than the physical length scale, which truncates the energy spectrum (in solid blue). Increased resolution decreases the dissipation imposed by numerics.

Energy distribution in a turbulent plasma can be further understood via its energy power spectrum (Taylor, 1938) as shown in Figure 1.6. With this approach, we examine the energy (which may be kinetic, magnetic, thermal, or a total energy) contained in the plasma at every length scale. This can be computed from a 3D plasma via a Fourier transform into spectral space or also from structure functions, which are the real-space equivalent of the power spectrum (Arenas & Chorin, 2006), as used by Kolmogorov (1941). These methods will give the energy power spectrum as a function of wavenumber k or wavelength $\lambda = 2\pi/k$. Smaller k pertains to larger length scales while higher k pertain to smaller length scales.

The energy spectrum of a turbulent fluid reveals how turbulence transfers energy from larger scales to smaller scales via the *Kolmogorov* cascade model of turbulence (Richardson, 1922; Beresnyak, 2019; Kolmogorov, 1941). When energy is introduced to the plasma by external forces at a certain length scale, it produces eddies and flows at these injection scales. In the ICM, large scale production includes contraction from the initial conditions, galaxy cluster mergers, and at a smaller scale ($\sim 10 - 100$ kpc) AGN feedback. These large scale energy injections lead to large eddies that break up into smaller eddies (higher wavenumber). As eddies break up, less kinetic energy is transferred from the large eddies to smaller eddies. Eventually, the eddies become small enough that viscous effects disallow smaller eddies and instead the kinetic energy dissipates into thermal heating, i.e., turbulent dissipation or turbulent heating. In the ICM this dissipation occurs due to plasma instabilities with length scales on the order of the cyclotron radius, which is typically on the order of 1 km. If the small-scale turbulent motions are statistically isotropic, then the energy spectrum between the injection scale and dissipation scale follows a power law $E(k) \propto k^{-\gamma}$ with spectral index $\gamma = 5/3$, as predicted by Kolmogorov (1941) for incompressible hydrodynamic turbulence.

The addition of magnetic fields to a turbulent plasma greatly complicates models of turbulence and has been under intense research and debate over the last decade (Beresnyak, 2019; Schekochihin, 2020). Not only do magnetic fields introduce an additional energy reservoir with its own energy spectrum apart from the kinetic energy spectrum, but magnetic fields also confine and collimate kinetic flows while the kinetic motions twist and wind magnetic fields, exchanging energy between these reservoirs (Grete et al., 2017, 2018, 2021b; Glines et al., 2021) and generally disrupting the assumptions of Kolmogorov turbulence.

Non-ideal MHD effects due to particle interactions near the particle scale lead to additional dissipation in plasmas, leading to the *magnetic Reynolds number* (Beresnyak, 2019)

$$\mathrm{Re}_m \equiv \frac{vL}{\eta} \tag{1.21}$$

where v and L are again the velocity and length scale of the scale of interest, and η is the magnetic diffusivity

$$\eta = \frac{c^2}{4\pi\sigma} \tag{1.22}$$

where c is the speed of light and σ is the conductivity. Magnetic fields likewise dissipate on small scales due to these same particle interactions, giving the *Lundquist number* (Beresnyak, 2019)

$$S \equiv \frac{v_A L}{\eta} \tag{1.23}$$

where v_A is the Alfvén speed

$$v_A \equiv \frac{B}{\sqrt{4\pi\rho}} \tag{1.24}$$

where B is the magnetic field strength. Similar to how high Reynolds number lead to fluids more prone to turbulence, high magnetic Reynolds numbers and Lundquist numbers (such as in the ICM) lead to plasmas that are more prone to magnetized turbulence (Beresnyak, 2019).

In the presence of a strong mean-field magnetic field, meaning there is a significant large scale magnetic field with associated Alfvèn speed much greater than velocity perturbations, perturbations with wavevectors perpendicular to the magnetic fields are well favored over parallel wavevectors, producing anisotropic turbulent motions in conflict with the assumptions of Kolmogorov turbulence (Montgomery & Turner, 1981; Shebalin et al., 1983).

Turbulence may also play a significant role in the amplification of magnetic fields in the ICM via the *small-scale turbulent dynamo* (Roh et al., 2019; Tobias, 2021). In this dynamo, the twisting and folding of magnetic fields by the turbulent motions in small eddies leads to an increase in the magnetic fields on small scales (Schekochihin et al., 2004; Steinwandel et al., 2021). Magnetic tension in the plasma in some cases can also accelerate or hinder the growth of turbulence in the magnetic and kinetic spectra at different rates (Glines et al., 2021; Bambic et al., 2018). It is currently unknown what the true spectral index of a magnetized plasma is, or if the energy spectrum is a power law between the production and dissipation scales (Grete et al., 2017, 2018; Glines et al., 2021; Grete et al., 2021b).

Magnetized turbulence in the ICM (and the applicability of MHD to the ICM in general) is complicated due to the ICM being weakly collisional: the mean-free path in the ICM, on the order of $1 - 10^5$ pc[3], is not much smaller than the system scales of the ICM, which is a requirement of a collisional plasma and an assumption in most theories of turbulence. Small scale plasma instabilities may instead make up for the lost dissipation from collisions, although this is an area of open research (Lyutikov, 2007; Rosin et al., 2011; Berlok & Pessah, 2015). The pressure anisotropy of weakly collisional plasmas should be accounted for in models of the ICM and may have an effect on turbulence dissipation in the ICM (Kunz et al., 2011).

Previous theoretical studies have estimated the turbulent dissipation in galaxy clusters, showing that an RMS turbulence velocity within 100 to 300 km s^{-1} can produce sufficient turbulence to match cooling within clusters (Dennis & Chandran, 2005). Observational studies have estimated the turbulent heating by inferring a power spectrum of density fluctuations in cool core galaxy clusters imprinted on high-resolution Chandra images (Zhuravleva et al., 2014, 2019; Li et al., 2020; Vidal-García et al., 2021). Although these studies have shown that turbulent heating may be sufficient to counteract overcooling, they have approximated the turbulence within the ICM as non-magnetized. It is also unclear whether processes in the ICM such as AGN feedback are sufficient to drive this turbulence, or whether multiple cycles of jet feedback are required (Heinrich et al., 2021). Generally, better understanding of magnetized turbulent dissipation within diffuse astrophysical plasmas such as the ICM is needed, and understanding of this phenomenon can be expanded via numerical simulations.

1.2.3 The Simulation of Plasmas as a Research Tool

Although plasmas are ubiquitous throughout the universe and are often created in laboratories, recreating exact astrophysical plasma conditions (or their scaled-down equivalent) and observing

[3] Mean free path of Coulomb collisions in the ICM (Spitzer, 1956, 1978)

them in a laboratory can be challenging and prohibitively expensive. Astrophysical plasmas span huge distances, both high and low densities, and extreme energies that are nearly impossible to recreate in a lab. Observing certain characteristics of astrophysical plasmas such as the magnetic fields and small scale turbulence can also be difficult due to the lack of direct electromagnetic emissions and limited resolution of telescopes. The complex and often non-linear nature of the equations governing these plasmas also makes pen and paper theoretical work limited. In both terrestrial and astrophysical plasmas, numerical simulations bridge the gaps between theory, observations, and experimental design. Numerical simulations of plasmas serves as a simplified, affordable, and accessible experimental stand-in for real plasmas, giving insight to both observations and experiments.

Simulating *turbulent* plasmas comes with its own complexities. Numerical methods implicitly but unavoidably add a numerical viscosity, which introduces a dissipation scale on the order of the resolution of the simulation. If a system can be fully resolved with elements smaller than the physical dissipation scale, then the entire turbulent cascade can be directly captured with an explicitly included realistic viscosity. Since turbulence in the ICM is driven on scales of kpc but dissipates on the scale of km, spanning several orders of magnitude, fully resolving the turbulent cascade of the magnetized plasma is infeasible for the foreseeable future due to the enormous volume of data that would be required to simulate a galaxy cluster down to km scales. As a result, the dissipation scale is artificially large and the turbulent dissipation is stronger in simulated clusters. This over-powered turbulent dissipation can be diminished by increasing the spatial resolution of simulations, although numerical dissipation will exceed the true dissipation using supercomputing resources available in the near to intermediate future. This translates to a difference in Reynolds number between the simulated plasma and the target system. Simulations on current supercomputers can achieve Reynolds numbers up to $\mathrm{Re} \sim 10^3 - 10^4$ (Ritos et al., 2018) whereas Reynolds numbers in the ICM could be as high as $\mathrm{Re} > 10^{12}$ (Miniati, 2014, 2015; Egan et al., 2016). Although larger supercomputers will enable higher resolution and lower dissipation, the achieved Reynolds number of simulations is unlikely to reach the true Reynolds number of the

ICM for the foreseeable future.

1.2.4 Numerical Methods for Plasmas in the Fluid Approximation

At its core, simulating plasmas in the fluid approximation amounts to evolving approximate solutions to the partial differential equations describing the plasma. Plasmas in the fluid regime have been simulated via many classes of methods developed for computational fluid dynamics (CFD) but extended to include magnetic fields for MHD or non-ideal MHD (Trac & Pen, 2003; Lind et al., 2020; Ledvina et al., 2008). Although not an exhaustive list, these methods include:

- Finite difference (FD) methods, where the partial differential equations are approximated via finite differences on a mesh of cells (Trac & Pen, 2003; Brandenburg & Dobler, 2010)
- Finite volume (FV) methods, where the fluid equations are converted to surface integrals constituting fluxes between cells (Toro, 2009; Stone & Norman, 1992; Stone et al., 2008a; Bryan et al., 2014; White et al., 2016a)
- Finite element (FE) methods, which comprise a variety of other methods (including discontinuous-Galerkin methods, DG) where the plasma is also discretized into a mesh of cells (Meier, 1999)
- Smoothed particle hydrodynamics(SPH), where a mesh is forgone and the fluid is represented by particles with overlapping spatially smoothed density functions (Katz et al., 1996; Springel et al., 2001; Wadsley et al., 2004; Springel, 2005, 2010)
- Pseudo-spectral methods, where the equations are solved in a spectral basis (such as with Fourier transforms) and with an additional basis to quickly convert to a spatial grid (Simon, 1992; Burns et al., 2020)

These fluid methods can be broadly divided by their specification of the fluid flow into Eulerian and Lagrangian specifications. Lagrangian specifications follow *along* with a parcel of the fluid, whether that be a mass or volumetric discretization (See Hopkins, 2014, for a Lagrangian code that implements both mass and volumetric discretizations), whereas Euler specifications follow fluid motion as it moves *through* a discretization of space. In a simple analogy of the flow of a river, a Lagrangian specification would follow the water as a boat moving with the river while an Eulerian specification would follow the water from a bridge stationary to the river.

Codes using Lagrangian specifications typically discretize using particles representing discrete masses or volumes within the domain. SPH is historically the most used Lagrangian method within astrophysics (Katz et al., 1996; Springel et al., 2001; Springel, 2005), although recent methods have innovated beyond SPH by including corrections to better capture shocks like an Eulerian specification (Hopkins, 2014) or to use a moving mesh where a Godunov-like scheme (explained below) can be applied to a Lagrangian code (Weinberger et al., 2020).

Codes using an Eulerian specification typically discretize the fluid domain into a mesh of cells within which properties of the fluid are tracked. In the case of FV (Toro, 2009; Stone & Norman, 1992; Stone et al., 2008a; Bryan et al., 2014; White et al., 2016a) and FD (Trac & Pen, 2003; Brandenburg & Dobler, 2010) methods, the cell averages of variables such as density, momentum, pressure, and energy are tracked. For other Eulerian methods such as DG methods, a linear combination of polynomials of these same variables are tracked, in addition to the cell averages evolving quadratic, cubic, and higher order spatial terms.

The theoretical basis for FV plasma methods begins the *strong form* of the fluid equations, where the conservation laws for the conserved quantities such as density, momentum, energy, etc. are expressed in terms of divergence of fluxes and source terms, i.e.

$$\frac{\partial}{\partial t}\mathcal{U} + \nabla \cdot \mathcal{F}\left(\mathcal{U}\right) = \mathcal{S}, \tag{1.25}$$

where $\mathcal{U}$ are the conserved variables, $\mathcal{F}$ are flux terms, and $\mathcal{S}$ are source terms. This strong form of the equations holds absolutely for the plasma. This strong form of the equations is converted to the *weak form* of the equation set using the divergence theorem, leading to a set of surface integrals to be satisfied (LeVeque, 2002), i.e.

$$\int_\Omega \frac{\partial}{\partial t}\mathcal{U} d\Omega + \int_\Omega \nabla \cdot \mathcal{F}\left(\mathcal{U}\right) d\Omega = \int_\Omega \mathcal{S} d\Omega \tag{1.26}$$

where Ω is the domain of a single cell from the discretized mesh. Assuming $\mathcal{U}$ and $\mathcal{F}$ are sufficiently smooth over Ω allows us to apply the divergence theorem to obtain the weak formulation

$$\frac{\partial}{\partial t}\int_{\Omega}\mathcal{U}d\Omega+\int_{\partial\Omega}\mathcal{F}\left(\mathcal{U}\right)\cdot\mathbf{n}dA=\int_{\Omega}\mathcal{S}d\Omega. \tag{1.27}$$

The advantage of the weak formulation is that it permits discontinuous solutions *between cells or different* Ω *volumes* where the divergence is not defined; i.e., the fluid can be approximated with a mesh of cells between which the fluid description is discontinuous. In a FV method, the cell averages of fluid quantities are tracked in each cell while these surface integrals become fluid fluxes between neighboring cells. Most FV methods for CFD are Godunov-like schemes (Godunov, 1959; Toro, 2009), where the fluxes are determined by solving or approximating a solution to a local Riemann problem at each cell interface. In a typical Godunov-like scheme, the fluid state at both sides of each cell interface is first *reconstructed* using an interpolation from the cell averages in surrounding cells. At each cell interface, the two fluid states from each side creates a Riemann problem that can be solved to determine the fluid flux into each cell. This computed flux is then used in the numerical integration to advance the state of the fluid in time.

In a DG method, solutions to the weak form of the equation set take the form of linear combinations of polynomials (such as the Legendre polynomials), which allow higher order representations of the fluid compared to FV methods (Reed & Hill, 1973; Cockburn et al., 2005; Chen & Liu, 2013). A 0th order DG method, which carries a constant contribution across is each cell, is equivalent to a FV method carrying cell averages. The method order for DG can be increased arbitrarily, however, just by carrying more polynomial terms. Reconstruction of fluid states at cell interfaces is computed using the polynomials internal to each cell while the Riemann problems solved in DG are equivalent to those solved in FV methods. Exact integration of surface integrals is facilitated by Gaussian quadrature. DG methods are also potentially better suited for upcoming hardware by being more arithmetically intensive, i.e., by executing more floating point operations per byte of data loaded or written from memory, which pairs well with hardware advances improving computational throughput faster than memory bandwidth (Klöckner et al., 2009, ; see discussion of changing supercomputer architectures in Section 1.4).

1.3 The Intracluster Medium – Plasma Physics Applied to Galaxy Clusters

The ICM, a hot diffuse plasma, comprises the majority of baryonic matter in galaxy clusters and is the primary emitter of cluster X-rays. Thus, the ICM has a profound effect on both how clusters evolve and how we observe them. Modeling and understanding the plasma physics governing the ICM allows better characterizations of galaxy clusters as a whole, one ultimate goal being to refine the luminosity-mass relation for galaxy clusters. This would enable surveys of galaxy cluster number densities that would reveal properties of dark matter and dark energy and the large scale structure of the universe.

Additionally, the ICM provides a unique plasma laboratory that can inform terrestrial plasmas. The high temperatures and low densities of the ICM are impractical to achieve on Earth, restricting their study to astrophysical observations, theory, and simulation. However, the ICM is likely very turbulent (Brüggen & Vazza, 2015; Zhuravleva et al., 2014; Simionescu et al., 2019), allowing study of magnetized turbulence that directly affects applications of plasmas on Earth. Turbulence triggered by the onset of plasma instabilities is a fundamental obstacle for achieving net power-generating fusion in both ICF (Casner, 2021) and MCF (Boozer, 2005; Sanchez & Newman, 2015), as it disrupts plasmas from being long-lived enough to achieve fusion. By studying the long-lived turbulent plasmas in astrophysical contexts via observations, we can better understand magnetized turbulence in laboratory plasmas and potentially develop more effective plasma devices (Ryutov & Remington, 2002; Chatterjee et al., 2017).

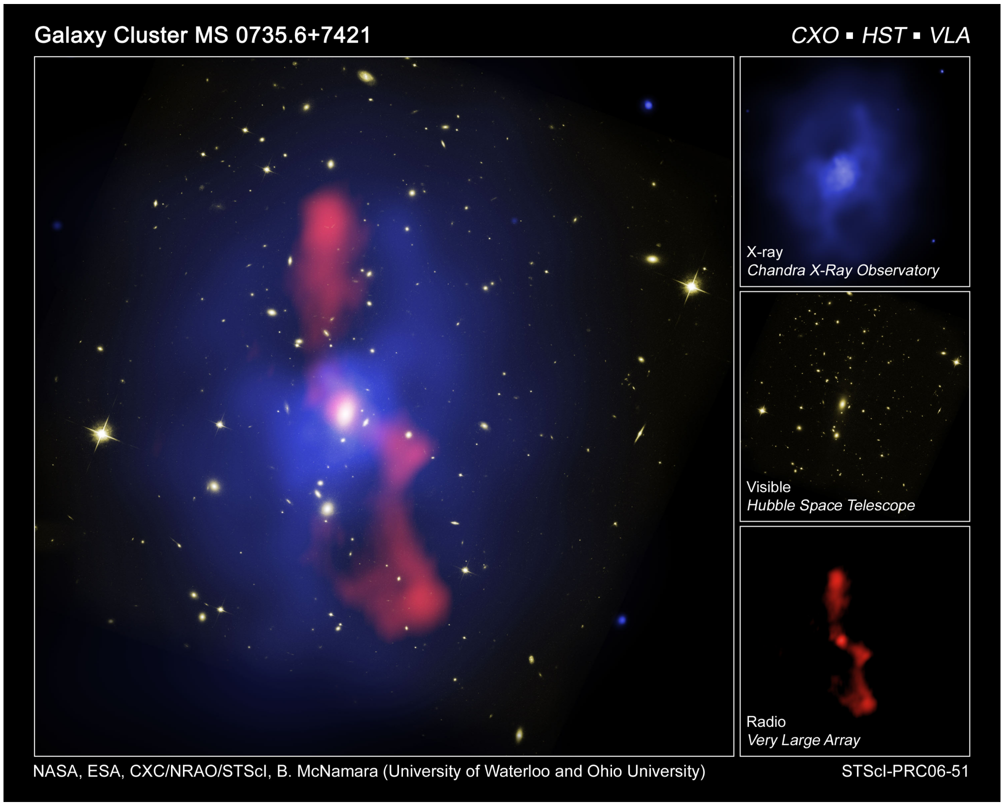
Conversely, since laboratory plasmas can be examined in closer detail and their experimental parameters changed, they can be used to study astrophysical plasmas as long as results are scaled appropriately (Ryutov & Remington, 2002). The magnetized supersonic flows, shocks, jets, and the development of plasma instabilities in these systems can be studied in laboratory high energy density plasmas (HEDP; Giuliani et al., 2012), which can inform understanding of these phenomena in the ICM (Beg, 2019). From a numerical perspective, methods, algorithms, and codes used for modeling laboratory plasmas can be repurposed for astrophysical plasmas (Howes et al., 2008) and vice versa (Beresnyak et al., 2018).

1.3.1 The cool core cluster problem

Approximately half of the galaxy clusters in the universe have high central X-ray surface brightnesses that would indicate significant radiative thermal loses within the inner several kpc (Fabian, 1994; Cavagnolo et al., 2009). Galaxy clusters with this property are known as cool-core (CC) clusters. Consequently, the centers of the galaxy clusters should quickly cool and collapse due to these energy losses within a few hundred million years in an event known as a "cooling catastrophe," which would be accompanied by massive rates of star formation. Historically, from a theoretical perspective these CC cluster centers would be replenished by massive inflows of gas known as cooling flows (Fabian, 1994). These cooling flows were never observed, however, nor were the elevated rates of star formation that would accompany the collapsing of the cold gas. Although X-rays are being emitted and energy is being radiated away, CC clusters are not cooling down - although not in HSE, they are apparently quasi-stable. Thus, some mechanism must offset or disrupt this cooling. Many potential mechanisms for doing so have been proposed.

Galaxy cluster mergers could disrupt this cooling since a large scale interaction such as a merger can inject sufficient energy into a CC cluster to offset central heating. However, galaxy cluster mergers are too infrequent to account for the abundance of quasi-stable CC clusters, occurring on the scale of 1 Gyr rather than 10 – 100 Myr cooling times observed. Thermal conduction, where thermal energy from the cluster outreaches is conducted along magnetic field lines to the cluster center, can offset some cooling but the effect is insufficient to offset all central cooling (Voigt et al., 2002; Ruszkowski & Begelman, 2002; Voigt & Fabian, 2004; Parrish et al., 2009). Stars collapsing into supernovae within the cluster can also inject heating but are likewise insufficient in power and frequency to offset cooling and also introduce metals, which promote cooling (Bregman & David, 1989; Domainko et al., 2004).

AGN feedback via jets excited by gas infalling onto the accretion disk of the AGN's central SMBH, however, is widely agreed to be sufficient to offset cooling (Fabian et al., 2000; McNamara


Galaxy Cluster MS 0735.6+7421
CXO ▪ HST ▪ VLA
X-ray
Chandra X-Ray Observatory
Visible
Hubble Space Telescope
Radio
Very Large Array
NASA, ESA, CXC/NRAO/STScI, B. McNamara (University of Waterloo and Ohio University)
STScI-PRC06-51

Composite image created by NASA
Hubble and Chandra Image: NASA, ESA, CXC, STScI, and B. McNamara (University of Waterloo)
Very Large Array Telescope Image: NRAO, and L. Birzan and team (Ohio University)

Figure 1.7: Bubbles inflated in by AGN jets in galaxy cluster MS0735.6+7421, as evidenced by X-ray cavities in the ICM and radio synchrotron emission from cosmic rays accelerated at the shock fronts around the bubble. Image made by NASA

et al., 2000; Gitti et al., 2012; Fabian, 2012). The capability of AGN jets to inject sufficient energy into the ICM to offset cooling was realized by bubbles inflated by AGN feedback, which appear as X-ray cavities indicating evacuated gas and radio lobes where cosmic rays are accelerated across shocks and emit radio synchrotron radiation at the bubble shock-front (Fabian et al., 2000; McNamara et al., 2000). Figure 1.7 shows said bubbles inflated by AGN jets in galaxy cluster MS0735.6+7421 as observed in X-ray and radio wavelengths. The energy injected by the AGN into the cluster can be estimated by the work done on the gas to inflate these bubbles; $W \sim PdV$ where W is the work done by the AGN, P is the pressure of the bubble, and dV is the size of the bubble (McNamara et al., 2000; Churazov et al., 2002; Blanton et al., 2010). The work done by AGN feedback is sufficient to offset the central cooling in CC clusters. In our current understanding of CC clusters, AGN feedback is widely believed to be the dominant mechanism preventing cooling flows and cooling catastrophes.

Many aspects of AGN feedback are still poorly understood (Morganti, 2017), including how AGN feedback is triggered, how AGN feedback deposits energy into the ICM, and how these two factors of AGN feedback combine to apparently maintain CC clusters in a thermodynamically unstable multiphase state (Gaspari et al., 2012b; Tümer et al., 2019). The AGN feedback is sufficient to offset cooling, prevent cooling flows, and quench star formation, but it is not so powerful as to evacuate gas from CC clusters. Instead, the cluster centers are maintained in a thermodynamically unstable multiphase state, with blobs of cold condensed gas amongst hot, rapidly cooling X-ray bright gas. Thus, AGN feedback is believed to be *self-regulating* – i.e. increased AGN feedback diminishes AGN triggering, thereby tempering further feedback. The multiphase nature of the AGN environment may be key to the self-regulation of AGN feedback in CC cores, which is explored in the precipitation model of self-regulating AGN feedback (Voit et al., 2015, 2017).

1.3.2 Self-Regulating AGN Feedback via Precipitation

Given the thermodynamically unstable nature of the multiphase medium of the AGN environment that is maintained in CC clusters, it may play a significant role in the AGN triggering mechanism. In the precipitation model of self-regulating feedback shown in Figure 1.8 this multiphase medium leads to cold gas condensing out of the ICM and falling inwards due to loss of buoyancy onto the AGN accretion disk. Since accretion of mass onto the SMBH is inefficient, much of the gravitational potential energy of this infalling mass is diverted into the jet driven by the accretion disk, feeding energy into the ICM. This feedback drives outflows that uplift condensed blobs of cold gas which would otherwise feed onto the AGN jet, regulating the feedback. Additionally, the energy deposited by the AGN into the outskirts of the cluster creates an entropy gradient sloping down towards the multiphase region of the cluster. As gas cools in this pow-law zone of the entropy curve of the cluster it loses buoyancy and falls into the isentropic zone, replenishing the gas (Voit et al., 2015, 2017).

From observations, the boundary between the isentropic zone and the power-law zone of the entropy profile is where the ratio of the cooling time t_{cool}, the time the plasma would take to cool

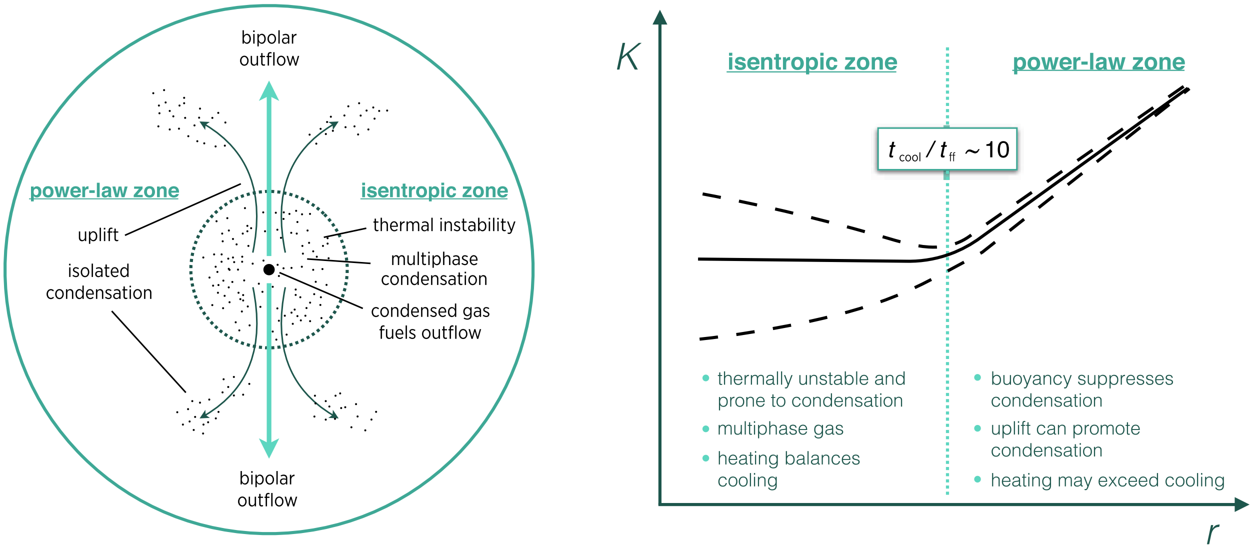


Diagram made by Voit et al. (2017)

Figure 1.8: Diagram of the self-regulating AGN feedback precipitation model from Voit et al. (2017), where the left panel shows a diagram of AGN feedback in a galaxy cluster and the right panel shows the entropy $K \equiv k_B T n_e^{-2/3}$ where n_e is the electron number density. In this model, cold gas condenses in the isentropic central region of the galaxy cluster and accretes onto the central SMBH, triggering feedback in the form of bipolar outflows that uplift condensed gas into the power-law zone of the entropy profile in the cluster outreaches, tempering the overcooling and condensation of gas. In this power-law zone, buoyancy suppresses condensation while uplift promotes condensation. Observationally, the transition between the isentropic and power-law zones of the entropy profile occurs where the ratio of cooling time to free fall time is $t_{\rm cool}/t_{\rm ff} \sim 10$, where the cooling time $t_{\rm cool}$ of a parcel of gas is the time it would take for it to radiative away all its energy at its current rate of radiative cooling and the free fall time $t_{\rm ff}$ of a parcel of gas is the time it would take to infall from rest to the cluster center due to gravity.

to $0K$ at its current rate of emission, and the freefall time $t_{\rm ff}$, the time the gas would take to fall to the cluster center from rest at its current radius, is approximately $t_{\rm cool}/t_{\rm ff} \sim 10$ (Cavagnolo et al., 2008; Rafferty et al., 2008; McCourt et al., 2012; Meece et al., 2015).

In this model, the AGN feedback and triggering mechanisms are intrinsically tied to the multiphase nature of the AGN environment. However, the model is not specific on the details of how AGN feedback couples to the ICM – how the AGN jet thermalizes energy into the ICM (Ho, 2004; Kunz et al., 2011; Morganti, 2017).

1.3.3 The nature of AGN Feedback

As gas accretes onto the AGN accretion disk around the central SMBH, the charged particles comprising the plasma of the accretion disk winds up magnetic fields that collimate into jets that emanate from both poles of the SMBH. Although these jets are likely the primary mechanism by which the AGN deposits energy into the ICM, it is still under debate how the magnetized, relativistic, tightly collimated jet thermalizes into heating and large scale outflows that can quench cooling in such a way to self-regulate AGN feedback and maintain a multiphase AGN environment (Young, 2010; Morganti, 2017).

One possible mechanism is turbulent dissipation incited by the AGN jet. Observational studies have estimated the turbulent heating by inferring a power spectrum of density fluctuations in cool core galaxy clusters imprinted on high-resolution Chandra images (Zhuravleva et al., 2014, 2019; Li et al., 2020; Vidal-García et al., 2021). By approximating the turbulence as purely hydrodynamic, velocity spectra can be inferred from these density perturbations and a $k^{-5/3}$ energy spectra turbulent cascade can be fit to the velocity spectra. This gives an observational estimate of the turbulent heating in the cluster that is sufficient to offset cooling. This estimate, however, does not account for the magnetic fields within the ICM which change the behavior of the turbulence.

This aspect of the ICM as a magnetized, potentially non-ideal MHD plasma may play a significant role in the thermalization of AGN feedback. The AGN accretion disk winds up strong magnetic fields that lead to the tight collimation of the AGN jet. These same fields may deposit significant energy into the ICM (Li et al., 2006). The AGN jet may also play a role in the amplification of existing magnetic fields within the galaxy cluster (Dubois et al., 2009) via a turbulent dynamo (Federrath, 2016). Anisotropic pressure in the ICM as a high-β plasma may trigger microscale instabilities in the plasma faster than if it were an ideal plasma, leading to higher turbulent dissipation that can more closely match radiative cooling (Kunz et al., 2011).

Numerical simulations are one cornerstone of our advancement in understanding AGN jets and how they interact with the ICM (Martí, 2019; Komissarov & Porth, 2021). Simulating the nature of AGN feedback is one of the ultimate goals of the methods presented in this dissertation. The current and future state of this work is explored in Chapter 6.

1.3.4 Simulation of Galaxy Clusters

The large dynamical range of the ICM requires vast computational resources to simulate accurately. The dynamical range of the ICM extends from the cluster scales on the order of 10 Mpc, down to the 1 pc scale of molecular clouds and star forming regions, and further down to the 1 km scale of plasma instabilities that drive dissipation in the diffuse plasma, spanning more than 20 orders of magnitude. Current world-class cosmological simulations can reach resolutions on the order of 100 pc (Pillepich et al., 2019), more than 15 orders of magnitude larger than the 1 km scale of plasma instabilities. In order to resolve said plasma instabilities directly in simulation we would need on the order of $\left(10^{15}\right)^3 = 10^{45}$ as many elements as used presently, and thus a supercomputer at least 10^{45} times larger than current supercomputers. Following the Courant–Friedrichs–Lewy (CFL) condition, the duration of timesteps Δt for this hypothetical simulation would need to satisfy

$$\frac{v\Delta t}{\Delta x} \leq C_{\rm CFL} \tag{1.28}$$

where v is the velocity (unchanged), Δx is the cell size (now 10^{15} times smaller than current simulations), and $C_{\rm CFL}$ is a constant to maintain stability that depends on the method (unchanged). Thus Δt would need to be 10^{15} times smaller than currently used timesteps and said simulation would require 10^{15} as many timesteps to complete. Since individual CPU core speeds have stagnated and are unlikely to increase in the near future (Leiserson et al., 2020), said supercomputer would need to be 10^{15} times larger again to complete the simulation in the same human time, on the order of months. In totality, we would need a supercomputer 10^{60} larger than present supercomputers (20 orders magnitude short of a "gogolFLOP" supercomputer). Assuming a variant of Moore's Law holds true for the indefinite future – that supercomputers will double in computational throughput every 2 years – this computer will come online in ~ 400 years.[4]

[4] If energy consumption per operation is the same for this hypothetical computer as current hardware, this supercomputer would need 10^{61} MW = 10^{70} erg s^{-1} of power. Over one day it would consume ~ 10^{79} erg ~ 10^{24} M$_\odot c^2$ in energy.

Since supercomputers in the foreseeable future are not capable of resolving the ICM down to plasma instability scales, all simulations of the ICM are necessarily an approximation. Unresolved key features of galaxy clusters such as the star forming regions and AGN must be approximated with subgrid model prescriptions that mimic the unresolved physics using a combination of observations and smaller scale simulations of plasmas and galaxy clusters. Within computational modeling, such simulations are referred to as multiphysics simulations as they incorporate many physical descriptions and scales into a single simulation. At its most basic, simulations of galaxy clusters are comprised of a model for gravity and dark matter, a model of the plasma, and any number of additional physics, feedback mechanisms, and subgrid models.

As the most massive component of the galaxy cluster, a treatment of the gravitational interactions of dark matter is essential for galaxy cluster simulations. For computational efficiency for idealized isolated galaxy clusters, this dark matter profile can be a fixed gravitational potential such as a Navarro–Frenk–White profile (NFW; Navarro et al., 1996). The gold-standard for dynamically evolving dark matter distributions, however, is to use an N-body method where the dark matter population is discretized into super-particles that can be evolved following gravity including the expansion of the universe (Aarseth et al., 1979). N-body simulations of dark matter have a long history that pre-dates computers (Holmberg, 1941) and continues to be researched today (Rogers & Peiris, 2021; Ebisu et al., 2022). To make robust predictions of the electromagnetic observations, however, requires coupling a treatment of the dark matter, whether that be a fixed gravitational potential or N-body simulation, to the baryonic matter.

This baryonic matter – the ICM – is a plasma that is reasonably approximated as a fluid[5]. This plasma can be modeled using methods from CFD that may include magnetic fields for higher

[5] The ICM is weakly collisional, in that the the mean free path of particle-particle interactions (via Coulomb collisions in the ICM) is long ($1-10^5$ pc) while the Debye length – $\lambda_D^2 = k_B T / 4\pi n q^2$, which is a measure of the scale on which the electric fields from individual charged particles in the plasma is relevant (Bellan, 2008) – is short. The ICM is thus electrically well screened, in that macroscale electric fields dominate over the fields from individual particles, but particle-particle collisions are infrequent. Non-ideal MHD models including pressure anisotropy and thermal conduction are more appropriate for weakly collisional plasmas such as the ICM (Braginskii, 1965; Berlok & Pessah, 2015).

fidelity. As discussed in Section 1.2.4 there are a wide variety of methods, but also a range of additional plasma physics that can be included. The ICM is potentially a non-ideal MHD plasma, so including non-ideal MHD effects such as resistivity (Bonafede et al., 2011), anisotropic diffusion (Berlok & Pessah, 2015), and thermal conduction (Narayan & Medvedev, 2001; Jubelgas et al., 2004; Wagh et al., 2014) along magnetic field lines can provide a more realistic simulation of the ICM.

The ICM also loses significant energy over time via free-free emission and line emission. Free-free emission, or Bremsstrahlung emission, is caused by the deceleration of charged particles, namely the electrons of the plasma, by the electric field of larger charged particles, specifically the ions of the plasma. This radiative cooling rate depends on the temperature and ion density. In a H/He plasma with hydrogen number density n_{H} and hydrogen mass fraction $X \approx 0.76$ such that $n_{\mathrm{H}} = X\rho/m_p$ where m_p is the proton mass, then the volumeteric free-free cooling rate is (Katz et al., 1996)

$$\Lambda_{\text{free-free}} \approx 2.5 \times 10^{-23} n_{\mathrm{H}}^2 \left(\frac{T}{10^8\ \mathrm{K}} \right)^{1/2} \mathrm{erg}\ . \tag{1.29}$$

Free-free emission only dominates cooling when the plasma is fully ionized, with ICM temperatures above $\sim 10^7$ K. At lower temperatures other processes become more important. These processes are collisional ionization, where atoms are ionized by collisions with electrons; recombination, where electrons combine with an ion, emitting a photon; and collisional excitation, where atoms are excited by collisions with electrons and then decay to a lower state (Mo et al., 2010). These processes depend on both the temperature and ion species within the plasma, where larger nuclei, or metals, lead to more cooling due to higher availability of electron orbitals. For numerical simulations these processes can be pre-computed for the ICM with a fixed metallicity (Schure et al., 2009) or with an evolving metallicity (Smith et al., 2017) combined with cooling tables to compute a radiative cooling rate (Ferland et al., 2013). These effects persist for temperatures down to 10^4 K, below which radiative losses are negligible for the dynamics of the ICM (Mo et al., 2010).

Beyond the basics of a gravitational or dark matter model and an ICM plasma model with radiative cooling, many important systems contributing to the dynamics of the ICM such as the

AGN, supernovae, and star forming molecular clouds, remain unresolved or underresolved due to limited computational resources. These phenomena can be included via *subgrid* models, which are prescriptions for the triggering and feedback of these systems on the ICM. For example, gas accretion onto the AGN accretion disk (which is approximately 10^{-2} pc Hawkins, 2007) occurs well below the 1 pc resolution of the current highest resolution isolated galaxy cluster simulations. AGN triggering can instead be included with a subgrid model following a Bondi-Hoyle accretion model (Bondi, 1952; Edgar, 2004), a boosted Bondi-Hoyle mode (Booth & Schaye, 2009), or a cold-gas mass triggered model informed by the precipitation theory (Meece Jr, 2016). The accretion disk physics that generate the AGN jet are likewise underresolved but various subgrid models of the AGN jet can be used to incorporate this feedback (Li et al., 2006; Meece Jr, 2016; Glines et al., 2020). Subgrid models for star formation, supernovae, turbulence (Schmidt & Federrath, 2011; Vlaykov et al., 2016; Grete et al., 2016), and cosmic rays can similarly improve the simulation of the galaxy cluster at the cost of complexity.

Despite these approximations, more resolution enabled by larger computational resources is always preferred for achieving higher fidelity simulations of the ICM as it reduces the dependency on artificial models and their free parameters. More complex multiphysics – including magnetic fields, self-gravity, cosmic rays, plasma microphysics, cooling, and more complex subgrid models for turbulence and AGN feedback – all impose additional computational expense, resolution constraints, and time step constraints to galaxy cluster simulations. Astrophysics simulations and especially simulations of the ICM are always wanting for more computational resources. In order to gain access to such resources, astrophysical simulation codes must evolve with the changing landscape of supercomputing hardware.

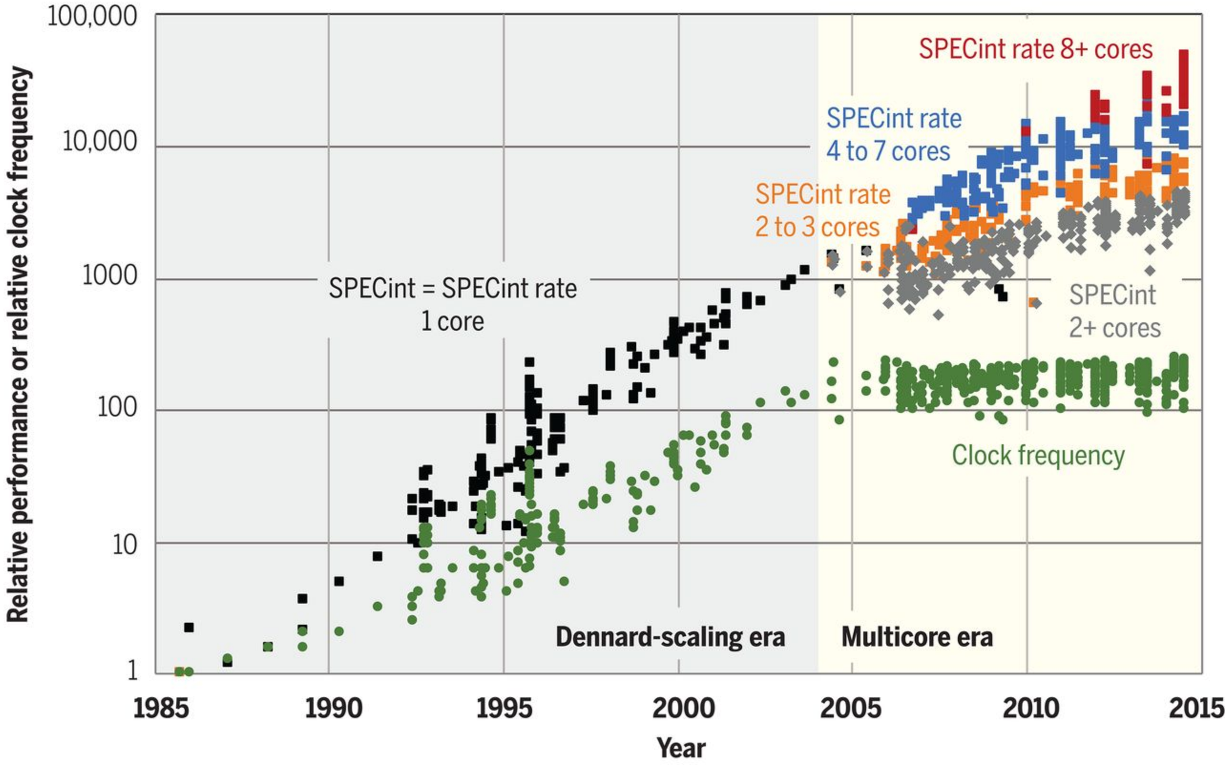
1.4 The Changing Supercomputer Architecture Landscape

Limitations to semiconductor manufacture have to led the predicted end of Moore's law – the trend in computer chip manufacturing observed over the last 50 years that transistor density has doubled every two years – which has previously driven the growth of supercomputing resources. Transistor dimensions are reaching the physical limitations of semiconductor manufacturing, with microchip features reaching 3 nm in the coming years, where the atomic radius of silicon is 0.1 nm, meaning transistors in microchips now span 10s of atoms[6]. Smaller microchips, which allow high clock speeds and thus faster computation, have become increasingly more difficult to develop over the last two decades (Iwai, 1999; Theis & Wong, 2017). The power consumed by these higher density microchips is likewise becoming more of an issue, since this power needs to be transported away from the chip to prevent heat damage (Landauer, 1988). More recent designs often trade computing speed for power efficiency, further limiting increases in computing resources (Leiserson et al., 2020). Alternative materials to silicon and other technologies such as optical transistors (Nolte & Nolte, 2001) may extend Moore's law for a few years but eventually microchip manufacture will reach hard physical limits of atomic radii. Although useful in some contexts, it is unclear whether quantum computers will impact astrophysical simulations since they have limited applications to CFD in general (Sammak et al., 2015; Steijl & Barakos, 2018).

Instead of relying on increasing clock speeds and processing cores to grow computing resources, supercomputer hardware has increase the size of processing chips by adding more cores or more parallelization to computer chips (See Figure 1.9; Leiserson et al., 2020). Whereas the Pentium Pro CPUs in ASCI Red (Top500, 2000), the fastest supercomputer in June 2000, had 1 core per CPU, the Xeon X5670 CPUS in Tianhe-1A (Top500, 2010), the fastest supercomputer in June 2010, had 12 cores per CPU, and the A64FX CPUs in Fugaku (Top500, 2020), the fastest supercomputer in June 2020 and at present, have 48 cores per CPU. Although individual core speeds have not improved since roughly 2005 (Leiserson et al., 2020), the increased core count permits higher computational throughput that is especially useful for CFD.

This trend in higher core counts on individual chips is taken to the extreme in hardware accelerators – computer chips designed for higher core counts and parallelization compared to traditional CPUs. Whereas a state-of-the-art Intel Xeon Platinum 8280 CPU used in Frontera, the current leading supercomputer where a majority of throughput is via traditional CPUs (Top500, 2021), has 28 cores per CPU with 2 threads per core (Intel, 2021) and provides over 2×10^{12} floating

[6] This limitation in the size of microchip features has long been predicted, including by Feynman in lectures on computation given during the 1980s (Feynman et al., 1998)



©Science (Leiserson et al., 2020)

Figure 1.9: Relative clock speeds of single core (black) and multicore (gray, orange, blue, and red, in order of increasing core counts) processors relative to the Intel 80386 CPU using the SPECint benchmark. The green round dots show processor clock frequencies, the frequency at which a single core can execute a clock cycle to execute one or several operations, relative to the Intel 80386. Although clock frequencies have stagnated since the mid 2000s, processors have increased performance by adding more cores. Future performance gains are increasingly dependent on higher core counts. Figure from (Leiserson et al., 2020).

point (64 bit) operations per second, or 2 TFLOPS; the state-of-the-art NVIDIA A100 graphics processing unit (GPU, Choquette et al., 2021) has 108 streaming microprocessors (SMs) with a total of 6912 cores, providing 9.7 TFLOPS of computational throughput for comparable price and energy consumption. Among the different accelerators, GPUs originally made to accelerate graphics rendering have been especially well-suited for high performance scientific computing (Du et al., 2011; Afzal et al., 2017; HajiRassouliha et al., 2018). GPU cores are designed for performing the same computational tasks simultaneously on large blocks of data as opposed to near complete independence between cores on CPUs. Although GPU cores are simpler than CPU cores, providing less features and less independence in execution, they are physically smaller in size and thus more GPU cores than CPU cores can be fit onto the same silicon die and for a similar cost. Thus, computational throughput can be expanded without depending on transistor manufacturing improvements, extending the growth of HPC past the end of Moore's Law (Leiserson et al., 2020).

GPUs' high core counts make them remarkably well suited for highly parallelizable tasks such as the methods used for CFD and plasma simulations (Griebel & Zaspel, 2010; Xu et al., 2015). All of the largest upcoming supercomputers being built in the US will use GPUs for the vast majority of their computational throughput. The US Department of Energy (DOE) is investing in new supercomputers to break the exascale barrier, executing 10^{18} floating point operations per second (FLOPS), an exaFLOP, on a single supercomputer. The goal is encapsulated in the Exascale Computing Project (ECP), which funds both the software and hardware for an exascale supercomputer (Messina, 2017). All US exascale supercomputers planned for the near future – Frontier, Aurora, and El Capitan – will use GPUs to achieve an exaFLOPS.

These hardware accelerators can be difficult to program for compared to traditional CPUs, however. This is not only because of their extreme vectorization and streamlined architecture that maximizes computational throughput, but also since they require different application programming interfaces (APIs). Traditional CPUs can be programmed using standard programming languages such as C, C++, and FORTRAN. GPUs, on the other hand, use APIs specific to each manufacturer (Patterson, 2010). NVidia GPUs, the historical leader in scientific computing with GPUs, uses the CUDA API, AMD uses ROCm and also provides the CUDA-like HIP interface, while Intel uses SYCL with its implementation named Data Parallel C++ (DPC++). Figure 1.10 shows a comparison between these different APIs. This state of APIs for GPUs is detrimental for scientific computing, as it requires rewriting code for each new API to use new computing resources. Said rewrites may introduce new bugs in different versions of the software, while making algorithmic improvements and additions to the the code requires updating the code for each API. New hardware architectures, such as Field Programmable Gate Arrays (FPGAs), would require additional versions and more development effort. Additionally, different architectures use different parallelization and memory layouts which might lead a code design to perform optimally on one machine but underperform on

```
#pragma omp simd
for( int i=0; i<n; i++){
    z[i] = a*x[i] + y[i];
}
```

(a) C/C++ example, where OpenMP is used for vectorization.

```
__global__
void vec_add(int n,float a,
  float *x,float *y,float *z){
  int i = threadIdx.x
    + blockIdx.x*blockDim.x;
  if (i < n)
    z[i] = a*x[i] + y[i];
}
...
vec_add<<<(n+255)/256, 256>>>
  (n, a, d_x, d_y, d_z);
```

(b) CUDA Example, where the arrays d_x, d_y, and d_z are allocated as CUDA arrays within GPU memory.

```
__global__ void
vec_add(int n, float a,
  const float* __restrict__ x,
  const float* __restrict__ y,
  float* __restrict__ z) {
    int i = hipThreadIdx_x
      + hipBlockDim_x*
         hipBlockIdx_x;
    if (i < n)
      z[i] = a*x[i] + y[i];
}
...
hipLaunchKernelGGL(vec_add,
  dim3((n+255)/256), dim3(256),
  0, 0,
  n, a, d_a, d_x, d_y, d_z);
```

(c) HIP Example, where the arrays d_x, d_y, and d_z are allocated as HIP arrays within GPU memory.

```
...
parallel_for(n,
  kernel_functor(
      [ = ](id<> item) {
    int i =
      item.get_global(0);
    d_z[i] = a*d_x[i] + d_y[i];
}));
```

(d) SYCL Example, where the arrays d_x, d_y, and d_z are allocated within GPU memory.

Figure 1.10: Example code to execute `z[i]=a*x[i]+y[i]` with different programming APIs. Even with this simple code example, there are significant differences in the implementation with different APIs. Each API also requires different code outside of this snippet to manage memory and execution on the GPU, along with a myriad of performance concerns.

others, wasting computing resources. The duplicated code for different hardware leads to higher development costs in terms of scientific researchers' time, which could otherwise be used to pursue science goals. As algorithmic and method changes are made and as bugs are found in the code, the different versions of the code written for the different architectures becomes out of sync, multiplying the development cost for each new architecture. Generally, needing to rewrite code with different APIs for each new hardware architecture limits scientific computing on these upcoming exascale supercomputers.

1.4.1 Performance Portability

Performance portability APIs have been developed to address the issue of different APIs for each hardware architecture (Reguly & Mudalige, 2020). Performance portability APIs provide portability – code written with the framework can be run on multiple hardware architectures without modification – and portable performance – the code executes with high performance, efficiently using hardware resources and features, on multiple architectures with differing memory and parallelization layouts. Within a performance portability framework, algorithms are written with more abstraction from parallelization and memory management details. This approach allows the API and the compiler to assemble a program for multiple hardware architectures from a single version of the code, vastly cutting down code duplication and software development for the scientist. The API can also vary the memory layout and parallelization strategy between different architectures, optimizing for each with minimal effort on the part of the scientist. Recent performance portability solutions include the libraries OCCA (Medina et al., 2014), Kokkos (Carter Edwards et al., 2014; Trott et al., 2022), and RAJA (Beckingsale et al., 2019), the OpenMP API with the "target offloading" capabilities beginning with OpenMP 4.5, and specifically for AMR applications, the AMReX library (Zhang et al., 2019). With a single code version using these APIs, the API backend can handle the execution of code and management of memory on both CPUs and GPUs from the different manufacturers currently producing the world's largest supercomputers.

The implementation of performance portability is an emerging field in scientific computing

(Deakin et al., 2019). The construction of exascale supercomputers with each of the different GPU manufacturers necessitates developing new performance portable astrophysics codes that can adapt to these upcoming architectures as well as to future computers. Research into performance portability strategies as well as quantifying performance portability across different hardware architectures (Pennycook et al., 2016) is needed to better facilitate adoption of performance portability APIs in scientific computing.

1.5 Outline of Dissertation

The remaining chapters of this dissertation are composed of first a series of four peer-reviewed papers where I am either the first author or an equal co-first author, one chapter consisting of current projects, and a final chapter for future directions of my work.

In Chapter 2 I explore the energy deposition requirements for self-regulating AGN feedback triggered by cold gas accretion using thermal only abstractions of AGN feedback. This chapter originally appeared as the published paper Glines et al. (2020).

In Chapter 3 I explore magnetized turbulence from decaying large scale flows, as might be created by large scale infrequent events in the ICM such as AGN outbursts and galaxy cluster mergers, using simulations of the magnetized Taylor-Green vortex. This chapter originally appeared as the published paper Glines et al. (2021).

In Chapter 4 I present the implementation and profiling of the performance portable magnetohydrodynamics code K-Athena, which was used for the simulations in chapter 3. This chapter originally appeared as the published paper Grete et al. (2021a), on which I am equal co-first author.

In Chapter 5 I present a new DG method for relativistic hydrodynamics. This chapter originally appeared as Glines et al. (2022), which has been submitted to the Astrophysical Journal Supplements.

In Chapter 6 I present in-progress simulations of magnetized AGN feedback in galaxy clusters, coming full circle to the nature of self-regulating AGN feedback.

Finally, in Chapter 7 I summarize the dissertation and discuss future directions of the methods, codes, and scientific results presented in this dissertation.

CHAPTER 2

TESTS OF AGN FEEDBACK KERNELS IN SIMULATED GALAXY CLUSTERS

This chapter first appeared as the published paper Glines et al. (2020). I include the original abstract as the introduction to this chapter.

Chapter Abstract

In cool-core galaxy clusters with central cooling times much shorter than a Hubble time, condensation of the ambient central gas is regulated by a heating mechanism, probably an active galactic nucleus (AGN). Previous analytical work has suggested that certain radial distributions of heat input may result in convergence to a quasi-steady global state that does not substantively change on the timescale for radiative cooling, even if the heating and cooling are not locally in balance. To test this hypothesis, we simulate idealized galaxy cluster halos using the Enzo code with an idealized, spherically symmetric heat-input kernel intended to emulate. Thermal energy is distributed with radius according to a range of kernels, in which total heating is updated to match total cooling every 10 Myr. Some heating kernels can maintain quasi-steady global configurations, but no kernel we tested produces a quasi-steady state with central entropy as low as those observed in cool-core clusters. The general behavior of the simulations depends on the proportion of heating in the inner 10 kpc, with low central heating leading to central cooling catastrophes, high central heating creating a central convective zone with an inverted entropy gradient, and intermediate central heating resulting in a flat central entropy profile that exceeds observations. The timescale on which our simulated halos fall into an unsteady multiphase state is proportional to the square of the cooling time of the lowest entropy gas, allowing more centrally concentrated heating to maintain a longer lasting steady state.

2.1 Introduction

Cool-core (CC) clusters have X-ray surface brightness profiles with sharp central peaks produced by substantial radiative losses of thermal energy from gas within the central few tens of kpc (Fabian, 1994). Given the observed rates of energy loss, CC clusters should be capable of radiating away their central thermal energy in less than 1 Gyr. If uncompensated, such a rapid cooling rate would lead to a cooling catastrophe in which multiphase condensation of ambient gas into cold clouds fuels star formation rates much greater than those observed. However, CC clusters are generally not observed to experience such dramatic cooling catastrophes (McDonald et al., 2019). They apparently remain close to thermal balance for billions of years and are common, representing about half of all galaxy clusters at the present time. Consequently, some mechanism must be counteracting central radiative cooling, and active galactic nuclei (AGN) are currently believed to be the responsible energy sources (Fabian et al., 2000; McNamara et al., 2000; Fabian et al., 2006; McNamara & Nulsen, 2007; Panagoulia et al., 2014; Gaspari, 2015).

Many other heat sources have been explored, including galaxy cluster mergers (Roettiger et al., 1997; Gómez et al., 2002; ZuHone et al., 2010), supernovae (Ciotti & Ostriker, 1997; Wu et al., 1998; Voit & Bryan, 2001; Domainko et al., 2004; Short et al., 2013), thermal conduction (Chandran & Cowley, 1998; Narayan & Medvedev, 2001; Malyshkin & Kulsrud, 2001; Voigt et al., 2002; Jubelgas et al., 2004; Brüggen, 2003a; Smith et al., 2013), gravitational heating (Khosroshahi et al., 2004; Dekel & Birnboim, 2007), and gas sloshing (Ritchie & Thomas, 2002; Markevitch et al., 2001; ZuHone et al., 2010). Most either do not provide enough heat to offset the observed cooling or do not adjust to the radiative cooling rate on a short enough time scale. Core cooling times in many CC clusters are $<$ 1 Gyr (Cavagnolo et al., 2009; Pratt et al., 2009), much less than the lifetimes of these clusters, suggesting that any heating mechanism coupled to cooling must react on shorter timescales. The gas accretion rate onto the central supermassive black hole (SMBH) would therefore need to couple to the radiative cooling rate with a lag time no greater than several hundred Myr.

Feedback from the central galaxy and AGN was explored numerically as early as Tabor & Binney (1993), Metzler & Evrard (1994), and Binney & Tabor (1995). More recently, Sijacki et al. (2007), Gaspari et al. (2011), Li et al. (2015), Meece et al. (2017), Prasad et al. (2015, 2017, 2018), and many others (Fabjan et al., 2010; Dubois et al., 2010; Short et al., 2013; Yang & Reynolds, 2016a) have demonstrated in hydrodynamic simulations of idealized galaxy clusters that AGN can plausibly regulate the high cooling rate in CC clusters. Simulated AGN self-regulate by coupling feedback energy output to the ambient gas density or cold-gas accretion rate around the AGN and inject that energy through either thermal deposition around the AGN or bipolar outflows from the AGN or a combination of the two. In addition to regulating the cooling rate and the condensation of cold gas clouds within the cluster, some of these AGN simulations produce temperature, density, and entropy profiles that resemble observations, including the multiphase cores observed in the central 100 kpc of galaxy clusters (Gaspari et al., 2012b; Meece et al., 2017; Prasad et al., 2018).

The simulations that most successfully resemble observations rely on cold-gas accretion to fuel the AGN and bipolar outflows to distribute the feedback energy (Gaspari et al., 2017; Gaspari & Sądowski, 2017; Voit et al., 2017; Meece et al., 2017). Ambient gas at the center of the system is nearly isentropic and therefore convectively unstable, resulting in the formation of a complex multiphase medium in which cold clumps of gas condense out of the ambient gas and precipitate onto the black hole. As the precipitation increases, so does the output of feedback energy, which raises the central cooling time and ultimately reduces the rate of precipitation. The resulting coupling suspends the ambient medium in a transitional state on the verge of a cooling catastrophe. Condensation outside of the isentropic center is marginally suppressed by buoyancy, and gas lifted out of the center by bipolar jets and buoyant bubbles forms multiphase filaments (Revaz et al., 2008; Li & Bryan, 2014a,b), in general agreement with observations (McDonald et al., 2010; Russell et al., 2016, 2017). However, even these idealized simulations do not track all of the physical processes that might be transporting and thermalizing AGN feedback energy, which range from turbulent heat diffusion (Ruszkowski et al., 2011; Zhuravleva et al., 2014), viscous dissipation of waves generated by the AGN (Ruszkowski et al., 2004), and cosmic rays created by the AGN heating the plasma via small scale fluid instabilities (Böehringer & Morfill, 1988; Loewenstein et al., 1991; Rephaeli & Silk, 1995; Colafrancesco et al., 2004; Pfrommer et al., 2007; Jubelgas et al., 2008).

Incorporating all of these mechanisms and processes into a cosmological simulation of galaxy cluster formation is currently prohibitively complex. Typically, the minimum spatial resolution in simulations modeling hot jets that interact with the intracluster medium is 200 pc. The finer resolution of the gas along which the jet deposits energy leads the jet to drill a hole through the ICM, allowing energy from the AGN to be deposited at further radii (Meece et al., 2017; Li et al., 2015). These resolution constraints are not always feasible for large cosmological simulations, because the computational effort needed to model these AGN jets exerts unacceptable drag on the evolution of the entire system. Therefore, simplified subgrid models are still needed to represent AGN feedback in cosmological simulations.

The results we present here emerged from an effort to develop a simple heat-input kernel to serve as an acceptable proxy for the much more complex process of AGN feedback. We sought a kernel that would satisfy three criteria:

1. The simulated hot-gas atmospheres of clusters balanced by AGN feedback should remain nearly thermally steady, meaning that they should not dramatically change because of cooling and feedback for periods of several billion years.

2. The central entropy of the hot gas in such a quasi-steady cluster halo should not exceed the values observed in CC clusters.

3. The feedback process should be computationally efficient, requiring neither very high resolution nor extremely small time steps that would make implementation in a current cosmological simulation prohibitively costly.

The first criterion requires the heating kernel to prevent a cooling catastrophe, which we define for the purposes of this paper to be a factor of 10 increase in radiative cooling within 10 Myr, accompanied by a rapid increase in the amount of cold (10^4 K) gas. As the central cooling time becomes short, compensating thermal feedback is needed to prevent runaway overcooling.

The second criterion requires that the kernel not overheat the central region, which would elevate or invert the central entropy profile. Such centrally concentrated AGN feedback can produce both non-cool core (NCC) clusters or observationally unreasonable galaxy clusters with large central entropy peaks. Furthermore, buoyancy is unable to suppress runaway thermal instabilities in systems with centrally flattened entropy profiles, making them prone to multiphase condensation (e.g., Voit et al., 2017) Simultaneously satisfying both this criterion and the first one proved to be difficult, even though observations show that CC clusters can remain remarkably close to a cooling catastrophe without producing an overabundance of cold gas and young stars.

Finding a way to satisfy the third criterion along with with the other two was the main motivator for this paper. Tracking the rapid formation of a complex multiphase medium approaching a cooling catastrophe requires high resolution and small time steps. Furthermore, if feedback energy output is directly linked to condensation of cold clouds, the approach of a cooling catastrophe leads directly to rapid central heating, computational requirements. We therefore sought a simple method that would avert a cooling catastrophe while still allowing the ambient central gas to remain in a low-entropy state.

In our search for a numerically simple heating kernel that would satisfy these three criteria, we investigated kernels with a power-law radial distribution of thermal feedback, normalized so that feedback heating globally equals radiative cooling within the galaxy-cluster halo. Use of such a heating kernel implicitly assumes that the most consequential feature of more complex AGN feedback mechanisms is the radial distribution of heat input. Depositing heat into the gas according to a kernel that depends only on radius is numerically simple and efficient to incorporate into cosmological simulations, and it does not require high spatial resolution as long as the feedback method can maintain the hot halo gas in a thermally steady state without overcooling. In order to create a tunable model, we also modified the radial power law with an inner truncation radius to limit central feedback and an outer exponential cutoff radius to constrain the bulk of the AGN heating to gas with shorter and more relevant cooling times. These additional parameters gave us a numerically simple but tunable model to search for an adequate AGN feedback kernel. We heuristically explored different values of the inner truncation radius that avoided central entropy peaks and different values of the outer cutoff radius that kept the majority of the feedback inside the region of the halo where gas cools within a hubble time. We discuss the model in more detail later in the paper.

Section 2.2 discuses the simulation setup and AGN feedback prescription and heating kernel in detail. Section 2.3 shows simulation results, describing in detail the results of three heating kernels that broadly represent the whole set of simulations, and examining the impact of different heating kernel parameters. Section 2.4 discusses the adequacy of the heating kernels tested, the robustness of the resulting feedback model, and the possible implications of these simulations for our understanding of AGN feedback in general. Lastly, Section 2.5 summarizes the results and conclusions of this work.

2.2 Methodology

This work builds upon simulations by Meece et al. (2017), using the same initial conditions from that work, described in §2.2.1, but using an AGN feedback kernel that is adapted to deposit energy at large radii as described in §2.2.2.

2.2.1 Simulation Setup

We ran several simulations of idealized galaxy cluster halos with a simplified AGN heating model using the hydrodynamics code Enzo (Bryan et al., 2014).

We used initial conditions approximating the Perseus Cluster, following the approach from Li & Bryan (2012) and Meece et al. (2017). The ICM begins as a hydrostatic sphere of gas in a fixed gravitational potential.

The gravitational potential has two components: a dark matter halo profile and a BCG with a mass profile with parameters chosen to match the Perseus cluster. The dark matter follows the NFW profile (Navarro et al., 1997), using $M_{200c} = 8.5 \times 10^{14} \mathrm{M}_{\odot}$ for the mass within the virial radius and a concentration parameter $c = 6.81$. The dark matter density from the NFW profile takes the form

$$\rho^{\mathrm{NFW}}(r) = \frac{\rho_0^{\mathrm{NFW}}}{(r/R_s)\left(1 + \frac{r}{R_s}\right)^2} \tag{2.1}$$

where the scale density ρ_0^{NFW} is defined by

$$\rho_0^{\mathrm{NFW}} = \frac{200}{3}\rho_c \frac{c^3}{\ln(1+c) - c/(1+c)}, \tag{2.2}$$

where $\rho_c = 3H^2/(8\pi G)$ is the critical density and the scale radius R_s can be found from

$$M_{200c} = 4\pi\rho_0^{\mathrm{NFW}} R_s^3 \left[\ln(1+c) - c/(1+c)\right]. \tag{2.3}$$

The BCG mass profile, following Meece et al. (2017), has the form

$$M_*(r) = M_4 \left[\frac{2^{-\beta_*}}{(r/4\ \mathrm{kpc})^{-\alpha_*}\,(1 + r/4\ \mathrm{kpc})^{\alpha_* - \beta_*}}\right], \tag{2.4}$$

where $M_4 = 7.5 \times 10^{10} M_\odot$ is the stellar mass within 4kpc and $\alpha_* = 0.1$ and $\beta_* = 1.43$ are constraints.[1] does not substantially affect our results.

The initial pressure was computed from the temperature and density assuming an ideal gas with $\gamma = 5/3$ in hydrostatic equilibrium with the gravitational potential. Cosmological expansion is neglected in these simulations. We used a vanilla ΛCDM model to get the virial mass of the NFW halo and to set its gas temperature. We set redshift $z = 0$ at initialization with $\Omega_M = 0.3$, $\Omega_\Lambda = 0.7$, and $H_0 = 70$ km s^{-1}. We note that the precise details of the cosmological model do not impact the results presented in later sections of this paper, which pertain to baryonic physics in the halo core.

The entropy profile of the gas, using the form

$$K \equiv \frac{k_b T}{n_e^{2/3}} \tag{2.5}$$

[1] Due to a programming error, the simulations use an incorrect initial mass profile for the BCG, which leads to the central 1 kpc being initialized out of hydrostatic equilibrium, with an absence of baryonic mass by less than a factor of two. However, the central halo gas either relaxes to hydrostatic equilibrium within 50 Myr or AGN feedback quickly drives it further from equilibrium, depending on the heating kernel parameters. Consequently, this error in the initial conditions

for the specific entropy, where k_b is Boltzmann's constant, T is the temperature, and n_e is the electron density, was initialized to a power law

$$K(r) = K_0 + K_{100}\,(r/100\text{ kpc})^{\alpha_K}\,, \tag{2.6}$$

following the power law fits used in the ACCEPT database (Cavagnolo et al., 2009). Here, r is the radius from the halo center and $K_0 = 19.38$ keV cm^2, $K_{100} = 119.87$ keV cm^2, and $\alpha_K = 1.74$ are fitting parameters corresponding to the core entropy, entropy slope and exponential increase, chosen to approximate the Perseus Cluster.

The simulations were run on a cartesian grid in a cubic volume with side length of 3.2 Mpc, with 64^3 cells in the base grid of the AMR hierarchy and a maximum of 8 levels of refinement, making the resolution of the finest cells approximately 195 pc. The mesh was refined based on the magnitude of gradients in fluid quantities and high baryon density. Additionally, a cubic grid with side length 4 kpc around the simulation center and was fixed at the maximum level of refinement with 195 pc resolution.

Each simulation was allowed to run for 16 Gyr or until excessive AGN feedback during a cooling catastrophe either created unphysical cell values or led to intractably small timesteps (see Section 2.4.2). To give context to the simulation duration, consider that the sound speed of gas with a temperature of $T = 2 \times 10^7$ K is $c_s = \sqrt{\gamma k_B T/\mu m_{\rm H}} \approx 0.70$ Mpc Gyr^{-1}, where $m_{\rm H}$ is the mass of hydrogen and $\mu = 0.6$ is the mean mass per particle in units of $m_{\rm H}$, meaning that the approximate sound crossing time across the inner $R = 0.5$ Mpc, where the majority of the dynamics of the galaxy cluster halo evolves, is approximately 1.4 Gyr.

We used the ZEUS solver for hydrodynamics (Stone & Norman, 1992) due to its robustness to evolve through discontinuities in the fluid around the AGN due to sharply peaked thermal injection. ZEUS is a relatively diffusive solver and requires an artificial viscosity, which may affect the accuracy of the hydrodynamics simulation (Stone & Norman, 1992; Meece Jr, 2016). Tabulated cooling was used to model radiative cooling following Schure et al. (2009), assuming a metallicity of 0.5 $Z_\odot$. The cooling table has a temperature floor of 10^4 K; any processes below this temperature will take place on a smaller scale than can be accurately explored with our spatial resolution.

Simulation results were analyzed using `yt` (Turk et al., 2011).

2.2.2 AGN Feedback Kernels

In our simplified AGN feedback model, thermal energy is deposited in a spherically symmetric distribution around the halo center by an assumed AGN, with the total amount of heating set equal to the total cooling in the halo every 10 Myr. Heating per unit volume $\dot{e}(r)$ is distributed following a power law in radius so that $\dot{e}(r) \propto r^{-\alpha}$. This basic power-law functional form has several numerical and practical issues. Most critically, these issues are a volumetric heating rate that diverges to infinity at the halo center, a "long tail" of heating at the halo outskirts where cooling is too slow to be relevant, and an unrealistic hard cutoff at the simulation boundaries. These latter two issues are compounded by observations that suggest AGN feedback is generally constrained to be within a few hundred kpc of the halo center. To address these issues and to create a more tunable and effective heating kernel, we added two parameters: a minimum truncation radius r_s (effectively a smoothing length) and an exponential decay cutoff radius r_c. To avoid having the feedback stop at a simulation boundary at $x, y, z = \pm 1.6$ Mpc, the AGN feedback is contained within a radius of $R = 1.5$ Mpc and set to zero outside this radius. Since the heating leading up to R is negligible compared to the cooling at far radii and the cooling time of the gas is much longer than the simulation time at that radius, we do not expect the value of R to have an impact on the outcome of the simulation. The full form of the feedback kernel defining the heating rate per unit volume $\dot{e}(t)\,[\mathrm{erg\ s^{-1}\ cm^{-3}}]$ is

$$\dot{e}(r,t) = \frac{\dot{E}(t)}{A} \begin{cases} \left(\frac{r_s}{r_c}\right)^{-\alpha} \exp\left(-\frac{r_s}{r_c}\right), & r \leq r_s \\ \left(\frac{r}{r_c}\right)^{-\alpha} \exp\left(-\frac{r}{r_c}\right), & r_s < r \leq R \\ 0, & R < r \end{cases} . \tag{2.7}$$

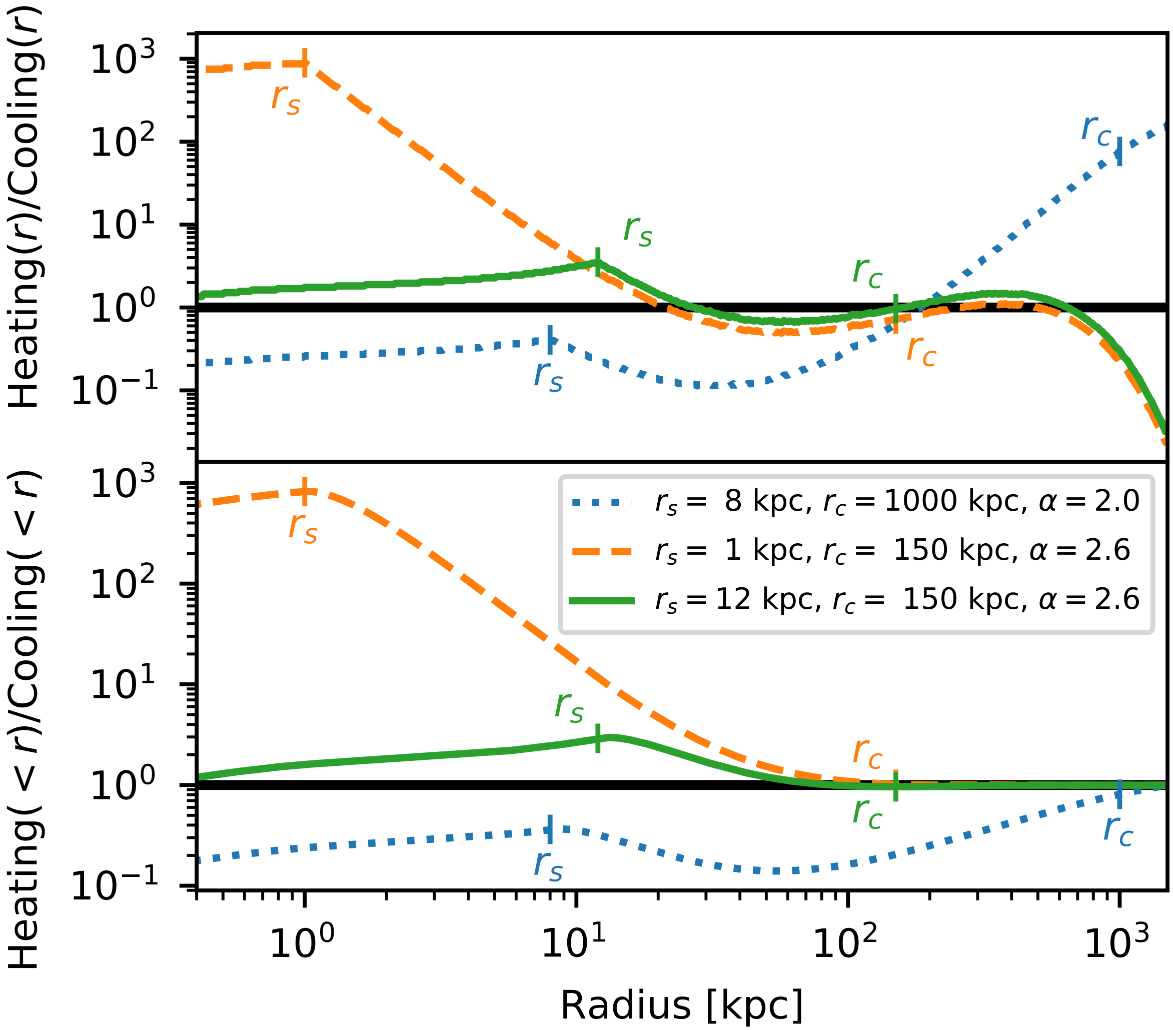


Figure 2.1: **Top:** Local ratio of heating to cooling as a function of radius (r) at the beginning of several representative simulations. The dotted blue line shows a simulation with low central heating and heating kernel parameters $\alpha = 2.0$, $r_s = 8$ kpc, and $r_c = 1000$ kpc. The dashed orange line shows a simulation with high central heating and heating kernel parameters $\alpha = 2.6$, $r_s = 1$ kpc, and $r_c = 150$ kpc. The solid green line shows a simulation with intermediate central heating and heating kernel parameters $\alpha = 2.6$, $r_s = 12$ kpc, and $r_c = 150$ kpc. **Bottom:** Cumulative ratio of heating to cooling within r for the same simulations. At large radii, all of the cumulative heating curves converge to the cumulative cooling rate because total heating is normalized to equal to total cooling rate at $R = 1.5$ Mpc.

The scalar A $[\mathrm{cm}^3]$ is defined by

$$\begin{aligned} A &= \int_0^{r_s} 4\pi r^2 dr \left(\frac{r_s}{r_c}\right)^{-\alpha} \exp\left(-\frac{r_s}{r_c}\right) \\ &\quad + \int_{r_s}^{R} 4\pi r^2 dr \left(\frac{r}{r_c}\right)^{-\alpha} \exp\left(-\frac{r}{r_c}\right) \qquad (2.8) \\ &= \frac{4\pi}{3} \exp\left(-\frac{r_s}{r_c}\right) r_s^3 \left(\frac{r_s}{r_c}\right)^{-\alpha} \\ &\quad + 4\pi r_c^3 \left[-\Gamma\left(3-\alpha, \frac{R}{r_c}\right) - \Gamma\left(3-\alpha, \frac{r_s}{r_c}\right)\right], \qquad (2.9) \end{aligned}$$

where $\Gamma(s,x) = \int_x^\infty t^{s-1} e^{-t} dt$ is the upper incomplete gamma function, normalizes $\dot{e}(t)$ so that the integral of $\dot{e}(t)$ over the volume of the simulation matches $\dot{E}(t)$. Higher values of α correspond to more centralized feedback around the AGN. Without the inner smoothing length, a heating kernel with $\alpha \geq 3$ is not normalizable, because integration over a volume containing the origin diverges.

The total heating rate $\dot{E}(t)$ is set to the total cooling rate within the cluster halo. Since the total cooling rate can be difficult to compute on-the-fly due to the nature of the AMR hierarchy's timestep update, it is recomputed only every 10 Myr. Although the cooling rate increases exponentially leading up to a cooling catastrophe, the increase is slow enough that the heating rate does not fall behind the true cooling rate by more than a few percent except immediately within a Myr before the catastrophe, at which point the simulation has already demonstrated that the particular heating kernel being tested is inadequate.

Note that the short time scale over which heating reacts to cooling in our model is not physical. Heat deposition resulting from AGN feedback does not instantaneously happen far from the AGN. We therefore probed heating kernels with a 50 Myr lag time between heating and cooling as well as averaging cooling over the same time period to smooth out jumps in heating. However, adding lag time led to more cold gas forming due to the lack of immediate feedback to counter condensation and more explosive feedback overall.

This study tested 91 different heating kernels with a range of parameters: different radial exponents $\alpha \in [2.0, 3.2]$, smoothing lengths $r_s \in 1, 4, 8, 10, 12, 16, 20, 40$ kpc, and exponential cutoff radii $r_c \in 100, 150, 200$ kpc. We began our exploration of the parameter space by setting

Table 2.1: List of combinations of inner smoothing radius r_s [kpc], outer cutoff radius r_c [kpc], and exponent α used. The rightmost column lists all values of α explored for the given combination of r_s and r_c in the leftmost and middle column.

r_s [kpc]	r_c [kpc]	α
1	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
1	1000	2.0, 2.1, 2.2, 2.3, 2.35, 2.375, 2.4, 2.425, 2.45, 2.5, 2.525, 2.55, 2.575, 2.6, 2.65, 2.7, 2.8, 2.9, 3.0
4	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
8	150	2.0, 2.2, 2.4, 2.6, 2.8, 2.9, 2.95, 3.0, 3.2
8	1000	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
16	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
10	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
10	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
12	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
16	100	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
16	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
20	100	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2
40	150	2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2

$r_s = 1$ kpc and $r_c = 1500$ kpc and sampled the range of α before trying different values of r_s and r_c with a smaller number of α values, seeking parameter combinations that seemed closest to an optimal kernel. Figure 2.1 presents a representative sampling of heating kernels showing the initial ratio of heating to cooling as a function of radius, including both the local ratio at each radius and the cumulative ratio within each radius. Table 2.1 lists all combinations of parameters explored.

2.3 Results

All the heating kernels we explored resulted either in cooling catastrophes within a few Gyr, central entropy levels greater than observations, or both. Simulations that eventually formed cold, condensed gas all went through cooling catastrophes. In those simulations, the minimum entropy drops over time, eventually leading to multiphase condensation. As cold clumps of gas form and runaway cooling begins, the requirement for total heating to match total cooling causes the heating

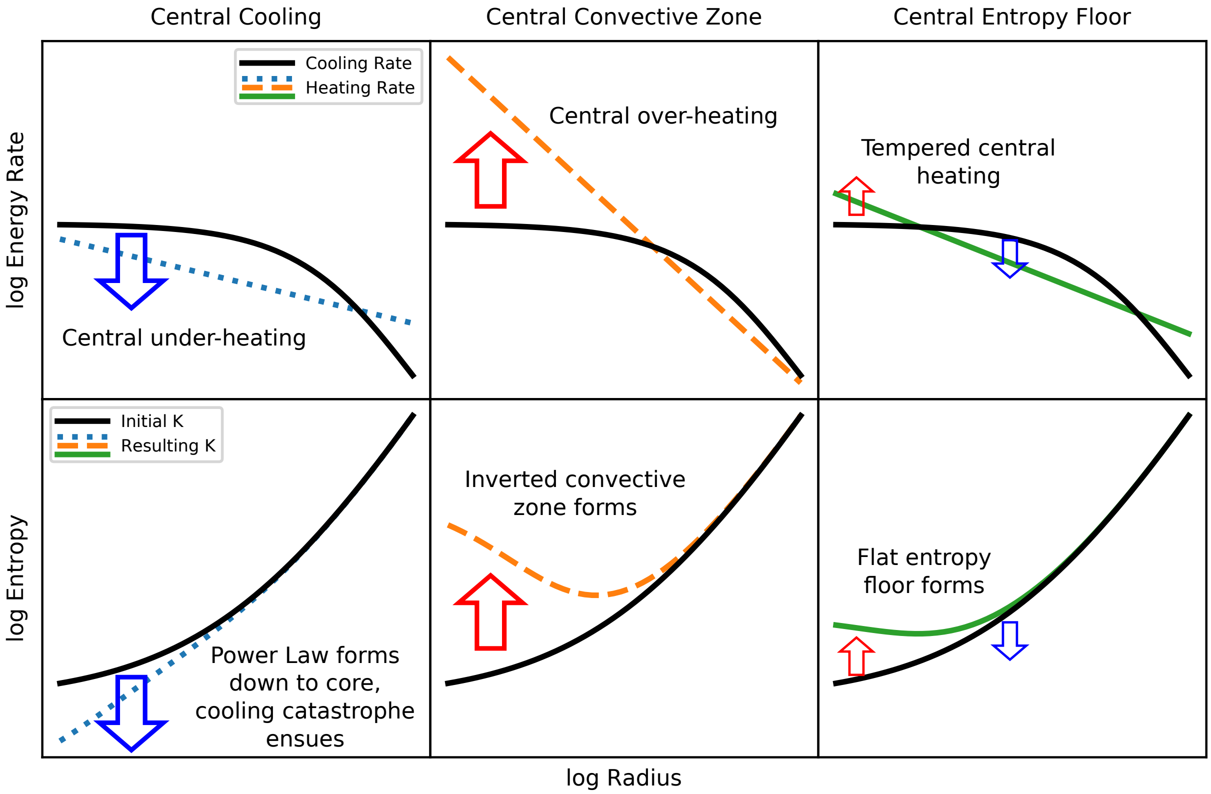


Figure 2.2: Schematic illustrations of how different AGN heating kernels affect the entropy profile of a simulated galaxy cluster. In each case, the total heating rate is set equal to the total cooling rate. **Top:** Radial profiles of radiative cooling and AGN heating per unit volume, with the initial median cooling rate in black and the AGN heating kernel in color. **Bottom:** Response of the median entropy profile to heat input. The initial median profile in black and the response is in color. The left column shows a heating kernel with central heating that falls below central cooling. The entropy profile in this case tends to follow a power law down to the origin and eventually leads to a central cooling catastrophe. The center column shows a heating kernel with excessive central heating, which elevates central entropy, inverts the entropy profile, and produces a central convective zone. The right column shows a heating kernel with intermediate central heating, which slightly raises the central entropy and produces a flat core. Due to the high initial entropy and long cooling time at outer radii, the power-law at the outer radii changes very slowly with under- and over-heating.

rate to spike. The time required for cold gas to form is roughly correlated with the smallest radius at which cooling exceeds heating. If central cooling exceeds central heating, the halo quickly forms cold gas and experiences a cooling catastrophe. Simulations with higher central heating tend to have high central entropy, similar to observations of NCC clusters. If the heating exceeds cooling out to radii of several tens of kpc, then the simulations persist for many Gyr without forming cold gas. Under- and over-heating at outer radii beyond 100 kpc is inconsequential since the time scale of heating is much longer than the dynamical time scale of the system due to the large specific energy and entropy at initialization.

Figure 2.2 schematically shows the general behavior of the different heating kernels. The three heating kernel examples in Figure 2.1 have colors that match the corresponding schematics in Figure 2.2. Figure 2.3 shows mass density profiles of cooling rate, heating rate, and entropy at later moments in simulations employing the same three heating kernels as in Figure 2.1.

2.3.1 Categorization of Simulations

The results of our simulations can be grouped according to the morphology of the entropy profiles that develop within the central 100 kpc:

1. **Central Cooling.** The entropy profiles of simulated cluster halos with heating that is insufficient to balance radiative cooling at small radii develop central cooling flows with a positive entropy gradient at all radii. They undergo a central cooling catastrophe relatively quickly, in which runaway multiphase condensation at small radii brings the simulation to a halt.

2. **Central Convective Zone.** The entropy profiles of simulations with high central heating form an inner convective zone with high central entropy and a negative central entropy gradient. Those simulations persist the longest before undergoing cooling catastrophes.

3. **Central Entropy Floor.** Simulations with intermediate central heating can maintain a nearly flat entropy gradient within the central $\sim$ 10 to 20 kpc.

For the purposes of our analysis, we define these categories based on the entropy within the inner 25 kpc. We categorize as Central Cooling those simulations whose average minimum entropy remains below 12 keV cm^2 (2/3 of the the initial minimum central entropy of 18 keV cm^2) . The Central Convective Zone simulations are defined to have maximum central entropy above 50 keV cm^2 (equal to the initial mean entropy of the inner 100 kpc). No simulation meets both of these criteria, so there is no overlap of these first two groups. The remaining simulations, which have minimum central entropies above 12 keV cm^2 and maximum central entropies below 50 keV cm^2, are categorized as Central Entropy Floor simulations.

The schematic diagrams in Figure 2.2 illustrate the general behavior of the different categories. Figure 2.3 shows representative snapshots of both cooling rate and entropy versus radius. Some of our simulations exhibit behavior from multiple categories at different times in their evolution. The following subsections describe each category in more detail.

2.3.1.1 Central Cooling

Simulations with low α, large r_c, or large r_s tend to have central cooling exceeding central heating, which quickly leads to a cooling catastrophe. The left column in Fig. 2.3 shows an example of such a simulation. Within the inner 10 kpc, the heating rate ranges from half the cooling rate to more than an order of magnitude less than the cooling rate. Because the central heating is insufficient to counteract a growing mass of strongly cooling gas at the halo center, the simulation produces a cooling catastrophe within 2 Gyr. However, up to the moment at which a substantial quantity of cold gas forms, the entropy profile remains close to the initial state and similar to the cool-core clusters in the ACCEPT data set.

2.3.1.2 Central Convective Zone

Heating rates within the central ~ 10 kpc of simulations with high α, small r_c, or small r_s tend to greatly exceed radiative cooling. The middle column in Fig. 2.3 shows an example. Excess central heating leads to a central entropy peak and an inverted entropy profile that drives convection.

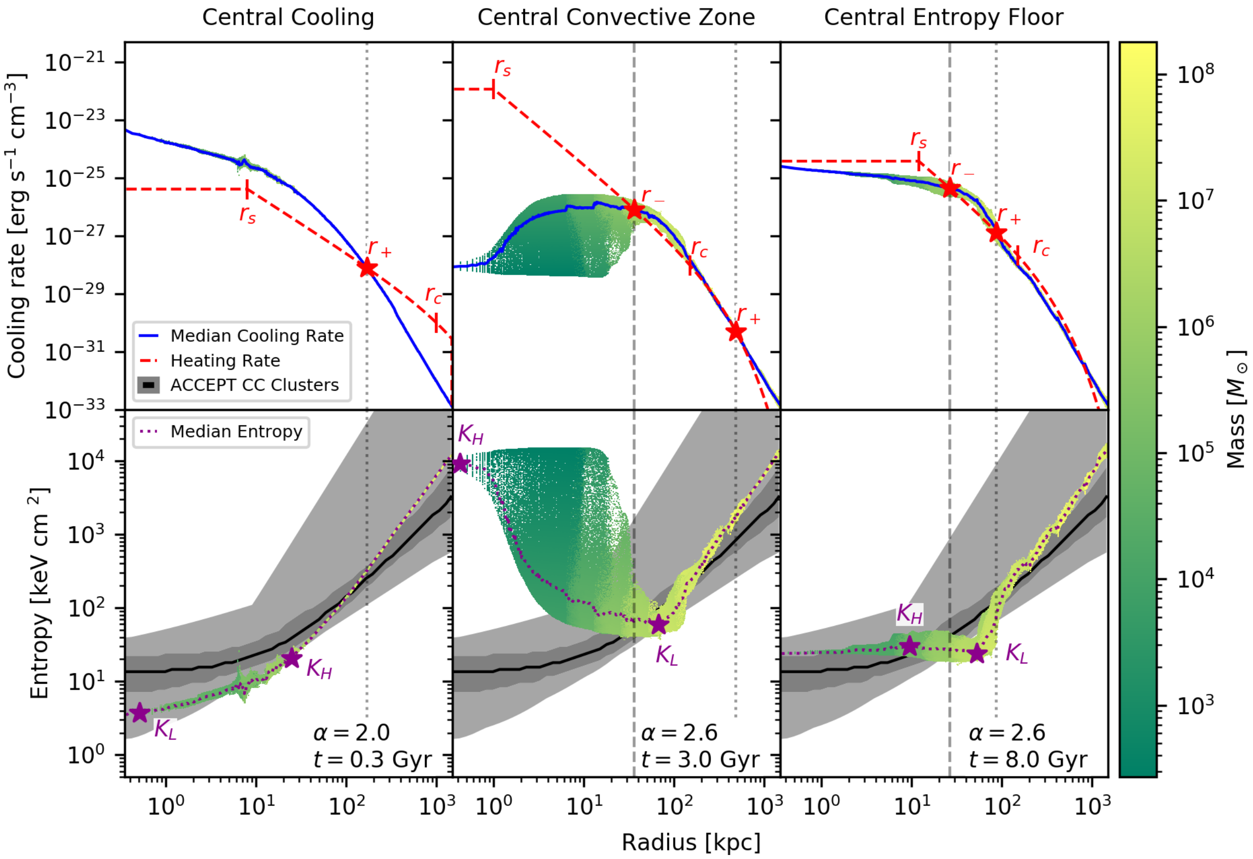


Figure 2.3: Mass density plots of cooling and heating rate (**top**) and entropy (**bottom**) versus radius, with color representing the total mass of all simulation cells from a 2D histogram of cooling rate and entropy versus radius. Across the three columns we show three simulations at different times that broadly represent the whole set of simulations, as differentiated by the behavior of the inner tens of kpc. The left column shows a simulation (with $\alpha = 2.0$, $r_s = 8$ kpc, and $r_c = 1000$ kpc at $t = 0.3$ Gyr) with low central heating which allows excess central cooling that quickly undergoes a cooling catastrophe. The middle column shows a simulation (with $\alpha = 2.6$, $r_s = 1$ kpc, and $r_c = 150$ kpc at $t = 3.0$ Gyr) with high central heating that maintains a convective zone in the inner 100 kpc with a high central entropy peak. The right column shows a simulation (with $\alpha = 2.6$, $r_s = 12$ kpc, and $r_c = 150$ kpc at $t = 8.0$ Gyr) with an intermediate amount of central heating and that holds a flat entropy floor slightly elevated from the initial conditions and observational data on the entropy of the inner tens of kpc. On the entropy plots, observational entropy data of clusters from the ACCEPT data set are displayed in grayscale showing the range (light grey), 68% confidence interval (dark grey), and median (black line) of the dataset. The median entropy is also marked by a magenta line, and the minimum (K_L) and maximum (K_H) values of the entropy median within the inner 25 kpc are marked by stars. On the cooling rate plots, the heating rate is marked by a red line and the median cooling rate is marked by a blue line. The crossover radii r_- and r_+ as defined in the text are marked by stars in the simulations where they can be defined.The heating curve parameters r_s and r_c are also annotated with finely dashed and dashed gray lines.

Low-entropy gas at the minimum entropy point sinks toward the center, but is reheated there and eventually rises to larger radii. Such a convective configuration can persist for many Gyr without producing multiphase condensation, because the minimum entropy and minimum cooling time are both large.

A few of the simulations in this category do form multiphase gas. When that happens, condensation first appears at the minimum of the entropy profile and rapidly leads to a cooling catastrophe. Although these simulations have large central heating rates, the heating rate still falls below cooling at intermediate radii (near the entropy minimum), allowing large clumps of cold gas to form there. In all cases in which a convective central zone forms, the central entropy is excessive compared with observed CC clusters, in some cases being more typical for a NCC.

2.3.1.3 Central Entropy Floor

Simulations with intermediate central heating, corresponding to a narrow range of combinations of α, r_s, and r_c, are able to maintain quasi-stable flat entropy profiles out to radii exceeding 10 kpc. The right column in Fig. 2.3 shows an example. Central heating within the inner 10 kpc of these simulations is typically several times the central cooling rate, sufficient to offset runaway cooling but not great enough to produce a large entropy inversion. Only some of these simulations form cold gas, and typically do so at larger radii and later times than in the Central Cooling simulations. However, the central heating in these simulations is still great enough to elevate the central entropy above the values observed in CC clusters.

2.3.2 Important radii: r_L, r_H, r_-, r_+, and $r_{\mathbf{multi}}$

To help with the analysis of the simulations, we identify several quantities that proved to be useful for interpreting their behavior. Those quantities are labeled in Figure 2.3.

The maximum and minimum entropy levels in the central regions turn out to be closely related to the time it takes for a cooling catastrophe to manifest. To quantify those extremes we first determine the median entropy at each radius, illustrated by the purple dotted lines in Figure 2.3.

We then define K_L to be the minimum of the median entropy profile and r_L to be the radius at that point. Outside of r_L the median entropy profile is stable to convection, but inside of r_L it is convectively unstable. In simulations with low central heating, r_L is close to the center. We define K_H to be the maximum of the median entropy profile within 25 kpc of the simulation center and r_H to be the radius at that point. We use the 25 kpc cutoff to exclude cosmologically heated gas at large radii from the analysis in order to focus on the effects of feedback heating. The initial entropy at 25 kpc is just below 30 keV cm^2, so a persistent K_H above 30 keV cm^2 indicates that heating has elevated the central entropy, making it too great for a CC cluster and possibly producing a central convective zone.

The entropy extrema K_L and K_H and the corresponding radii r_L and r_H evolve over time as feedback alters the median entropy profile. We denote the cooling times at those radii by $t_c\ (r_L)$ and $t_c\ (r_H)$. The value of $t_c(r_L)$ is closely linked to the time required for condensation to begin. The relationship between how the heating kernel parameters affect K_H and K_L along with the associated radii and cooling times is explored in sections 2.3.3, 2.3.4, and 2.4.1.

The radii at which heating equals cooling are special and come in two types. For one type, the net heating rate goes from positive to negative as r increases. We define r_- to be the smallest such radius. Excess heating within that radius tends to raise the median entropy while excess cooling at large radii causes the median entropy to decline. The result is flattening and sometimes inversion of the median entropy profile, which drives convection and ultimately makes the system prone to condensation near r_-. However, if cooling dominates heating in the central regions, then r_- is undefined. Some relationships between r_- and the simulation outcomes are explored in Section 2.3.3.

At the other type of heating-cooling equality radius, the net heating rate goes from negative to positive as r increases. We define r_+ to be the largest such radius. Outside of r_+, net heating raises the median entropy and suppresses condensation. Within r_+, net cooling lowers the median entropy. Together, these effects produce a positive entropy gradient in the vicinity of r_+.

While the median cooling rate may exceed the heating rate at very large radii (on the order

Table 2.2: Brief definition of variables described in full in text and used in later figures. "Median" here refers to the median of the distribution of a variable (e.g. entropy, cooling rate, etc.) at given radius.

K_L	Lowest median entropy
K_H	Highest median entropy within 25 kpc of the simulation center
r_L	Radius of lowest median entropy
r_H	Radius of highest median entropy within 25 kpc of the simulation center
r_-	Inner radius within which median heating exceeds median cooling
r_+	Outer radius outside of which median heating exceeds median cooling
$t_c(r_x)$	Median cooling rate at radius r_x
t_{multi}	Simulation time at which multiphase gas first forms
r_{multi}	Radius at which multiphase gas first forms

of hundreds of kpc), cooling times at those radii are so long that cold gas does not form on an astrophysically significant time scale. During a given simulation, the radii r_- and r_+ do not stay fixed, but rather shift as heating and cooling change the median cooling rate. We denote the cooling time at those radii as $t_c(r_-)$ and $t_c(r_+)$.

The heating kernel parameters also affect when cold gas forms in the simulations and at what radius the cold gas first appears. We define t_{multi} to be the time from the beginning of the simulation to the moment when multiphase condensation produces cold gas. In our analysis, we use 10^5 K as the temperature cutoff for cold, although gas around these temperatures will rapidly cool to colder temperatures. Our temporal resolution of t_{multi} is limited by the frequency of output to disk, which is every 10 Myr. We define r_{multi} to be the radius at which cold gas first appears, using the innermost radius if cold gas appears simultaneously at multiple radii. The relationship between r_{multi}, r_H, t_{multi}, and $t_c(r_-)$ is explored in Section 2.3.3.

Table 2.2 summarizes the variables defined in this section. These variables are used in figures and analysis in later sections.

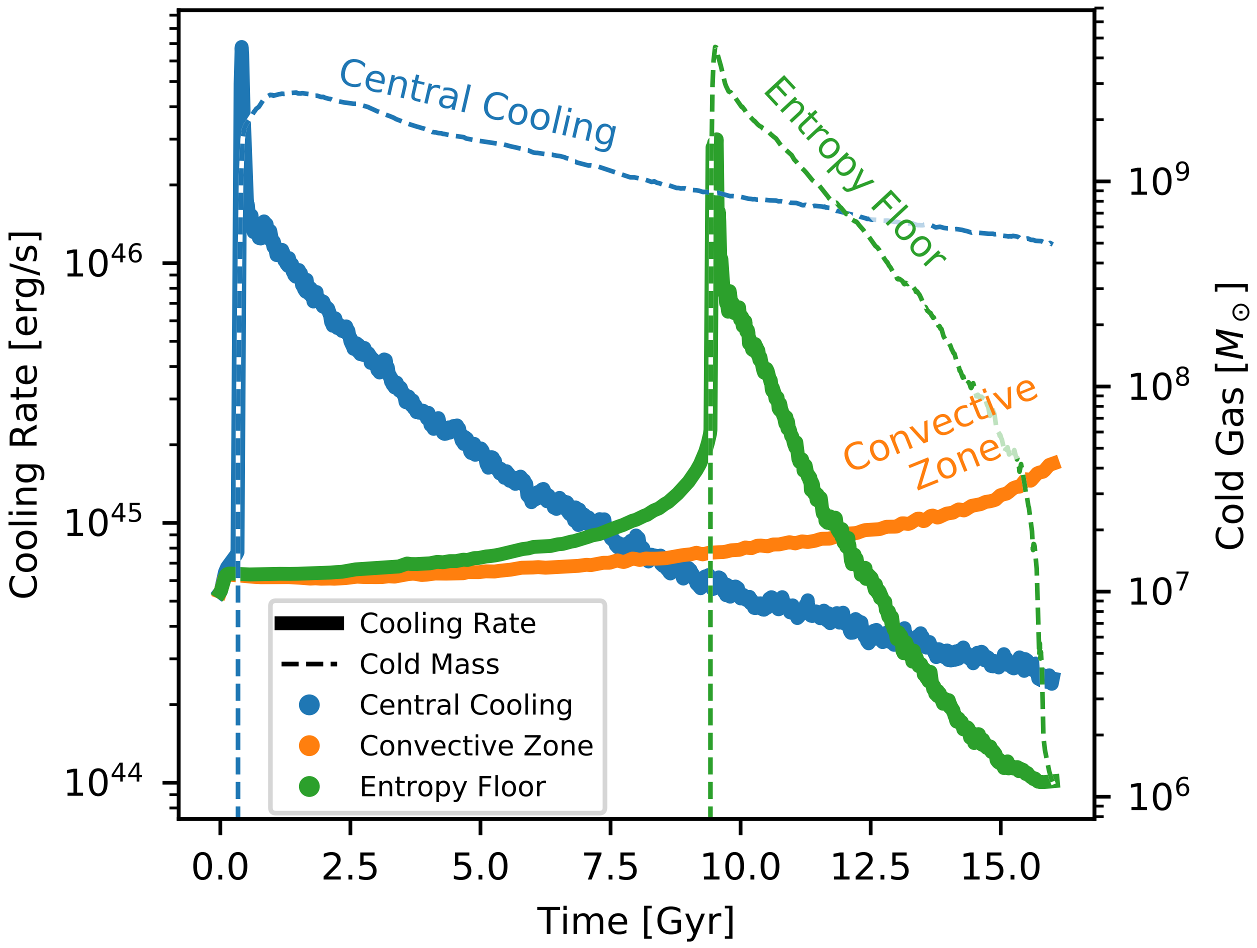


Figure 2.4: Time dependence of total cooling rate (solid lines) and total mass of condensed gas under 3×10^4 K (dashed lines) for the three simulations shown in Figure 2.3. The blue points show a simulation with low central heating and excess central cooling (α = 2.0, r_s = 8 kpc, r_c = 1000 kpc) that experiences an early cooling catastrophe. Orange points show a simulation with high central heating (α = 2.6, r_s = 1 kpc, r_c = 150 kpc) that forms a quasi-stable central convective zone. Green points show a simulation with intermediate central heating (α = 2.6, r_s = 12 kpc, r_c = 150 kpc) that maintains a flat entropy core for almost 10 Gyr before undergoing a late cooling catastrophe. In simulations that form a multiphase gas through a cooling catastrophe, the formation of cold gas is preceded by a rise and then a sharp peak in the total cooling rate.

2.3.3 Condensation of Cold Gas

Multiphase condensation forms cold gas in many of the simulations, in each case leading to a cooling catastrophe. Cold gas starts forming near r_L, then falls toward the center, displacing buoyantly rising warmer gas. The location of r_L depends on the heating kernel parameters and is related to r_-.

However, when gas at r_L cools enough to transition into the cold phase, it sharply raises the total cooling rate of the halo. That event immediately boosts the heating rate by the same factor, because our AGN feedback prescription forces the total heating rate to equal the total cooling rate. This heat is distributed across the halo and is not concentrated on the cooling gas, and thus the AGN feedback does not halt the cooling catastrophe.

In many cases, rapid heating of lower-density gas during the cooling catastrophe produces such great sound speeds and creates such large discontinuities in the fluid that the simulation becomes infeasible to continue due to the Courant condition. At that point the heating input greatly exceeds the AGN activity observed in real CC clusters, meaning that the chosen heating kernel has become physically unrealistic. In simulations that managed to evolve through this catastrophic event, the heat input leads to drastically elevated entropy in the ambient gas, which slowly reheats the embedded cold gas and prevents more cold gas from forming. After the cooling catastrophe, the core entropy is left much higher than before the catastrophe. Figure 2.4 illustrates the timeline of a catastrophe resulting from an increasing cooling rate that leads the formation of cold gas.

Our simulation set generally demonstrates that the radii r_{multi} and r_L are both related to r_-. Figure 2.5 shows the relationships among the values of those three radii. We average these quantites over time from the simulation outputs, which have 10 Myr frequency, in order to produce one data point per heating kernel. Larger $\langle r_- \rangle$ corresponded to a larger $\langle r_L \rangle$, as shown in top right panel, meaning that the radius of lowest entropy corresponds to the inner radius inside of which heating exceeds cooling. The top right panel shows that larger $\langle r_- \rangle$ corresponds to larger r_{multi}, meaning that the radius of lowest entropy corresponds to the inner radius inside of which heating exceeds cooling roughly determines where cold gas first forms. In the bottom left panel, $\langle r_L \rangle$ also corresponds to larger r_{multi}, showing that multiphase gas typically first forms around the entropy minimum. The relationship between r_- and the formation of cold gas is most apparent in the plot of $t_c\,(r_-)$ versus t_{multi} in the bottom right panel. When r_- is larger, so that cooling first exceeds heating at a larger radius, the cooling time at r_- is longer, which leads to cold gas forming later in the simulation. The timescale on which cold gas forms is closely tied to the cooling time of this

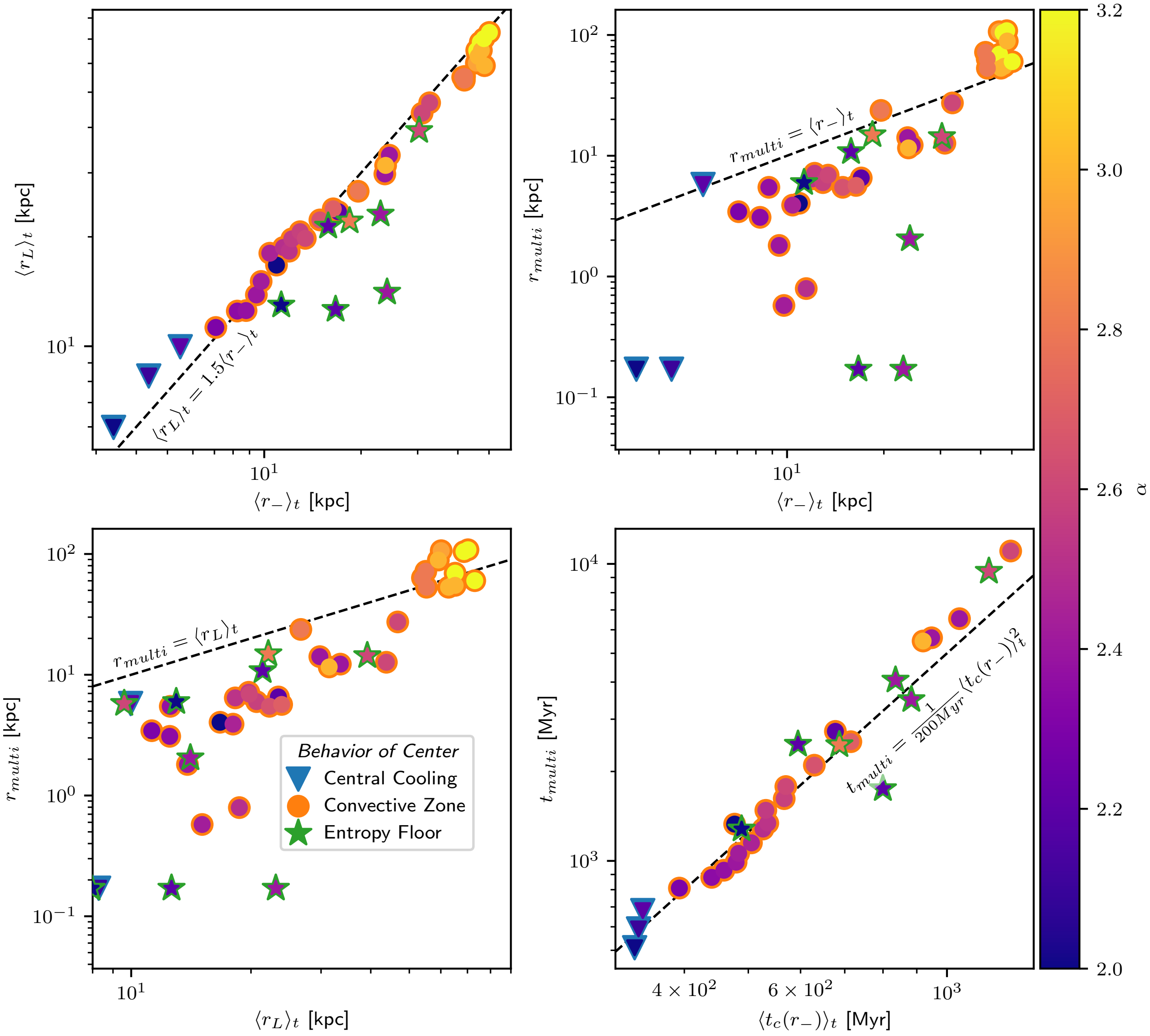


Figure 2.5: Plots of relationships between r_-, the radius at which the gas switches from net heating to net cooling, and other features of the simulations. **Top left:** Time averaged radius of the minimum of the median entropy profile (r_L) versus the time average of r_- up to the formation of a multiphase gas. (Includes only simulations in which r_- can be defined for at least 50 Myr.) **Top right:** Radius at which multiphase gas first forms versus the time averaged r_-. (Includes only simulations in which r_- can be defined for more than one time step.) **Bottom left:** Radius at which multiphase gas first forms versus the time averaged value of r_L for all simulations. **Bottom right:** The time required for a simulation to form multiphase gas versus the time averaged value of the cooling time at r_-. (Includes only simulations that form multiphase gas and in which r_- can be defined for at least 50 Myr.) Shapes in each panel denote the general behavior of the central region of the simulation. Blue highlighted triangles denote Central Cooling simulations, orange highlighted circles denote Central Convective Zone simulations. Green highlighted stars denote Entropy Floor simulations. Colors show the heating kernel parameter α, with greater α generally corresponding to heating that is more centrally concentrated.

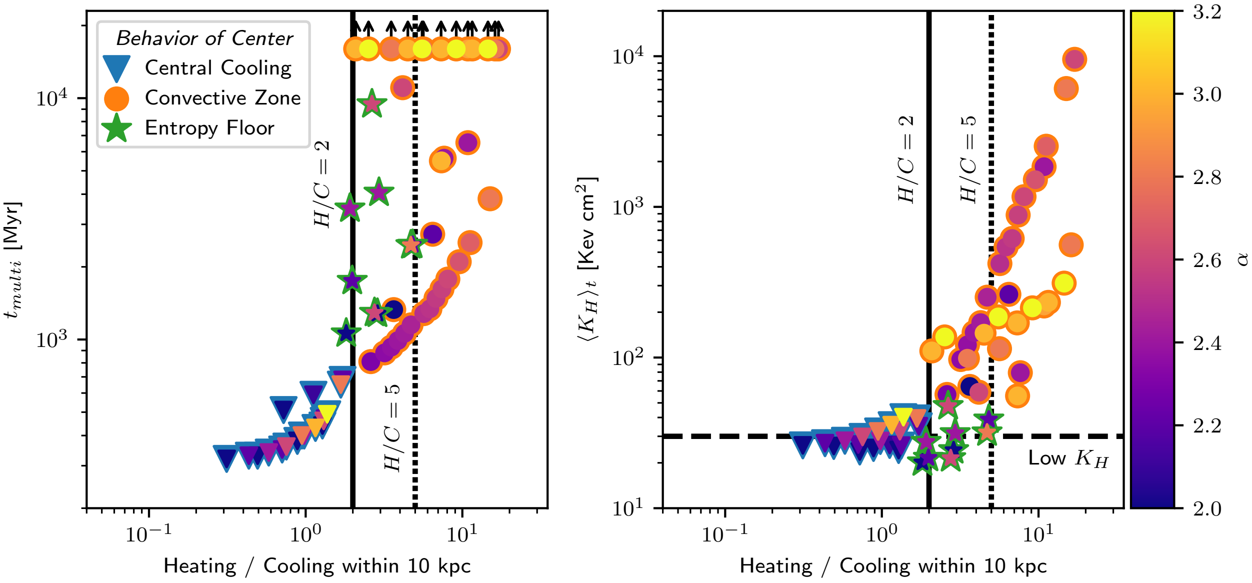


Figure 2.6: **Left:** Time required to form multiphase gas in a simulation versus the ratio of heating to cooling within the inner 10 kpc at the first time step. **Right:** Maximum of the median entropy within the inner 25 kpc, versus the ratio of heating to cooling within the inner 10 kpc at the first time step. In both panels, a solid line marks a heating to cooling ratio of 2, and a dashed line marks a heating to cooling ratio of 5. A ratio of at least 2 is required to avoid multiphase condensation within 1 Gyr. In the right panel, a dashed line marks the maximum central entropy that is observationally expected for a CC cluster.

gas. Interestingly, the relationship is non-linear, following

$$t_{\text{multi}} = \frac{\langle t_c\left(r_{\text{multi}}\right)\rangle^2}{200\text{ Myr}}. \tag{2.10}$$

This result is consistent with previous work by Meece et al. (2015) exploring the condensation of gas in the central ICM of galaxy clusters. Meece et al. (2015) found in thermally balanced ICM simulations with varying initial ratios of cooling time to freefall time that gas with a greater initial ratio remains nearly homogeneous for a larger number of cooling times before condensing into a multiphase gas, suggesting a non-linear relationship between cooling time and the formation of a multiphase medium.

2.3.4 Central Heating

The heating kernel parameters also affect the central entropy of the cluster halo, in some cases resulting in unreasonably high levels for a CC cluster and in other cases allowing cold gas to quickly condense and collect in the halo center. The central entropy and general behavior of the core is directly related to the amount of heating compared to cooling in the halo center. A certain amount of heating in the center is necessary to offset the central cooling but an excess of heating in the halo center causes central entropies higher than observed in CC clusters.

To explore this behavior, we track the ratio of the total heating within the inner 10 kpc of the halo to the total cooling within the same volume.[2] Figure 2.6 shows $t_{\rm multi}$ and the time average of K_H versus the initial central heating to cooling ratio. A ratio of heating-to-cooling of approximately two is needed to maintain quasi-stability for any significant amount of time, while a ratio greater than five always leads to high central entropies. Inside this range of ratios of heating to cooling, different heating kernels produce all three categories of central entropy behaviors.

When the integrated heating in the inner region is less than twice the cooling in the same region, a cooling catastrophe happens within 1 Gyr. For simulations with less heating than cooling in the central region, cooling quickly causes the central entropy profile to approximate a power law down to the halo center. Cooling gas then flows down the entropy gradient, collecting in the center, and forming multiphase gas. In simulations with average heating one to two times the average cooling rate in the center, density inhomogeneities in the gas allow cooling to exceed heating in some locations. As the cooling of that gas increases, the total heating rate rises but is insufficient to counter the localized increase in cooling, thus leading a runaway cooling catastrophe. Additionally, as central entropy falls and density increases in the lead up to the catastrophe, central pressure increases and compresses clumps of cooling gas. This further accelerates their cooling during the runaway catastrophe. With simulations having heating-to-cooling ratios above two in the center region, the central cooling is more successfully countered so that the formation of multiphase gas happens on a longer timescale connected to $t_c(r_L)$ and $t_c(r_-)$, as discussed in Section 2.3.3. The left plot in Figure 2.6 also shows this distinction in behavior.

When central heating rates are more than two times greater than the cooling rate, excess heating

[2] The inner 10 kpc volume was chosen to coincide with the region within which the initial entropy profile is nearly flat. We also tested this analysis using the inner 20 kpc volume and found similar results.

leads to central entropies that are higher than what is observed for CC clusters. The right plot in Figure 2.6 shows the relationship between the ratio of central heating to cooling and the maximum entropy in the central region averaged over time. Some simulations with two to five times heating to cooling in the center stay under the typical 30 keV cm^2 specific entropy for CC clusters, but all of the simulations with heating-to-cooling ratios of greater than five produce unrealistically high entropies. With values of K_H above the 30 keV cm^2 specific entropy where the isentropic entropy profile changes into power law, these simulations form an inverse convective zone where hot gas collects in the halo center and cold gas collects at r_L at intermediate radii.

2.4 Discussion

2.4.1 No Adequate Heating Kernel

None of the 91 heating kernels we simulated meet all three of the adequacy criteria specified in Section 2.1. The failure modes we observe in the simulations can be discussed in terms of the same behavioral categories listed in Section 2.3.1 for the central entropy profile:

1. **Central Cooling.** Heating kernels with low central heating fail to meet our first criterion by producing a cooling catastrophe within ~ 1 Gyr that radically changed the structure of the ambient medium.

2. **Central Convective Zone.** Heating kernels with high central heating produces central convective zones that fail to meet our second criterion by producing central entropy levels greatly exceeding those observed among typical CC clusters. Some of the simulations in this group also fail our longevity criterion because the heating kernel is unable to prevent an early cooling catastrophe due to insufficient heating at intermediate radii.

3. **Central Entropy Floor.** The heating kernels closest to being adequate, according to our criteria, were those with intermediate central heating that exceeds central cooling, but not by a large factor. Those simulations maintain a quasi-stable entropy floor and prevents cooling catastrophe for billions of years. However, the central entropy profiles of those simulations,

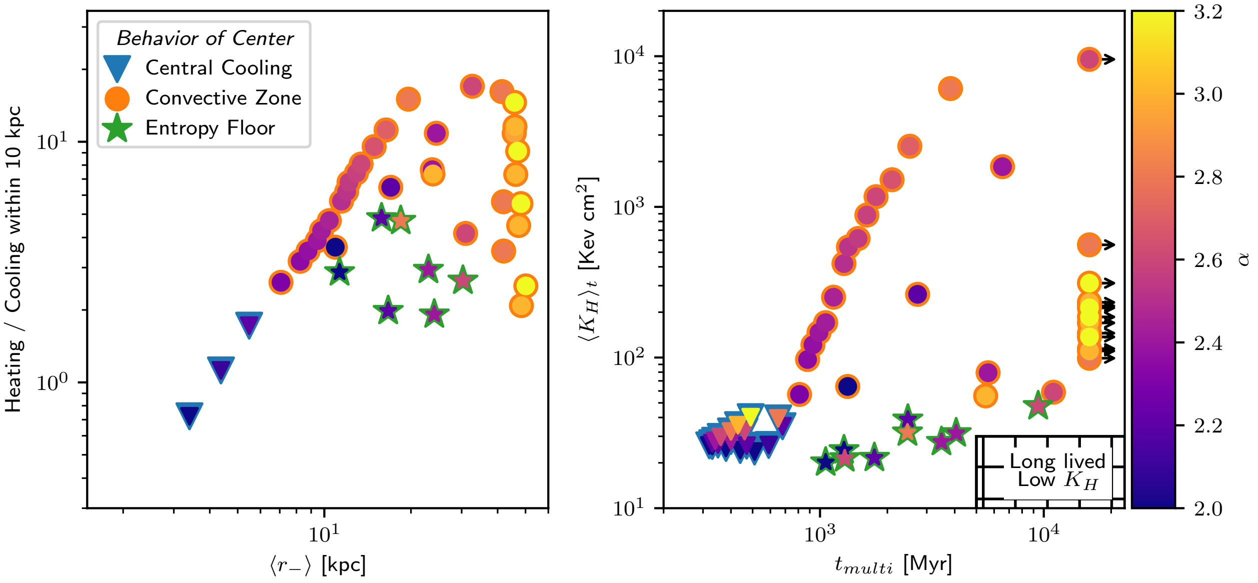


Figure 2.7: **Left:** Relationships between the initial ratio of heating to cooling averaged over the inner 10 kpc and the time-averaged radius $\langle r_- \rangle$ beyond which cooling begins to dominate over heating. Only those simulations in which r_- can be defined for at least 50 Myr are included. The box in the lower right shows hypothetical simulations with an average r_- over 30 kpc and an inner heating to cooling ratio under five. **Right:** Relationships between the time average of K_H (the maximum level of the median entropy profile within the inner 25 kpc) and the time t_{multi} until multiphase gas forms in the simulation. The plot includes all simulations, assigning $t_{\mathrm{multi}} = 16$ Gyr to simulations that do not form cold gas by that time. An empty box in the lower right corner indicates where points representing heating kernels satisfying adequacy criteria would fall, by persisting for more than 5 Gyr before forming multiphase gas while maintaining a maximum entropy level < 30 keV cm^{-2} within 25 kpc. However, no heating kernel we tested satisfies those those criteria.

while lower than those in the previous category, were still elevated compared to observed CC clusters and thus do not meet our second criterion. Lowering the central heating rates in an attempt to bring their entropy profiles more in line with observation also causes cold gas to form much more quickly. The simulation that provides results closest to a realistic cluster (with kernel parameters $r_s = 12$ kpc, $r_l =$ kpc, and $\alpha = 2.4$) maintains a flat entropy core of 30 keV cm^2 and lasts for just under 4 Gyr, which may be sufficiently long to maintain a CC cluster between external heating events.

No heating kernel we tested is able to maintain a low entropy floor close to observations of CC clusters for longer than 4 Gyr. Figure 2.7 summarizes the failure modes of the heating kernels probed in this study. The right panel shows K_H versus t_{multi}, a measure of the longevity of the simulation before a cooling catastrophe strongly altered it. Some simulations prevent a multiphase cooling catastrophe for many Gyr while others maintain low central entropy, but no heating kernel accomplished both aims. The left panel shows the ratio of central heating to cooling versus r_-, the two parameters that most strongly influenced the central entropy and longevity, respectively.

2.4.2 Robustness of Feedback Algorithm

The ultimate obstacle to finding an adequate thermal heating kernel is the difficulty of preventing gas in the halo center from overcooling while still maintaining a reasonably low entropy profile. In order to prevent a cooling catastrophe, central heating must be sufficient to raise the median entropy profile enough to keep the lowest-entropy gas from undergoing runaway cooling. Our simulations show that an integrated central heating rate within the inner 10 kpc that is approximately two times the cooling rate in that same region is necessary. Otherwise, too large a proportion of the gas within the central region ends up with cooling exceeding heating, causing a rapid increase in the total radiative cooling rate.

The consequences of that rapid rise in cooling are dramatic, because the total heating rate is set equal to the radiative cooling rate and rises just as rapidly. However, that heat input is distributed more evenly across a large volume and cannot counteract radiative cooling of localized dense gas clumps. As a result, the ambient pressure sharply rises, compressing the dense clumps of low-entropy gas, causing both radiative cooling and the matching heating rate to increase. That coupling therefore causes the cooling/heating rate to spike to unphysically high levels during a cooling catastrophe (see Figure 2.4). Central internal energies and velocities then rapidly rise and create discontinuities in the fluid. Due to the Courant condition, the time steps sometimes became too small to continue evolving the simulations. In other cases, those discontinuities lead to negative densities and/orz internal energies in the hydro solver, ultimately ending the simulation.

In reality, CC clusters can form cold gas (as is evident from observed star formation rates ranging from 1 to 100 $\text{M}_\odot$ per year), and so a physically accurate model should accommodate the formation of moderate amounts of cold gas. However, a heating kernel that immediately responds by injecting compensating thermal energy with a fixed spatial distribution appears unable to accommodate multiphase condensation without causing excessive heating.

2.4.3 Comparison to Observations

Figure 2.8 shows the time-averaged median entropy profile and projected X-ray surface brightness profile, along with the 1σ dispersion in the median profiles. It also shows the median entropy profile of observed CC clusters in the ACCEPT dataset (Cavagnolo et al., 2009), along with the 1σ dispersion and the full range. The dispersion in the simulated profiles is computed in radial bins over the lifetime of each simulation up until the formation of cold gas or the end of the simulation. The dispersion in the ACCEPT data is generated from a table of power-law fits to the entropy profiles. Only CC clusters from ACCEPT with $K_0 < 30$ keV cm^2 are used.

No quasi-stable simulation maintains a central entropy close to the majority of the CC clusters in the ACCEPT dataset. Heating kernels that keep low entropies within the range of the ACCEPT CC clusters are not steady for more than 1 Gyr, and all experience central cooling catastrophes. Heating kernels that form central convective regions have higher central entropies than the ACCEPT CC clusters. Simulations that form a central entropy floor have lower entropies than the central convective zone simulations and are steady for longer periods than the low central heating kernels, but still have higher central entropies than the majority of observed CC clusters in the ACCEPT dataset.

The differences among the X-ray surface brightness profiles are more subdued, with more centralized feedback corresponding to a lower central surface brightness. The median central surface brightness of the simulation shown here with a central catastrophe is within an order of magnitude of the simulations that form a convective zone. Additionally, the surface brightness profiles from the simulations fall inside the 1σ interval of the CC clusters from ACCEPT.

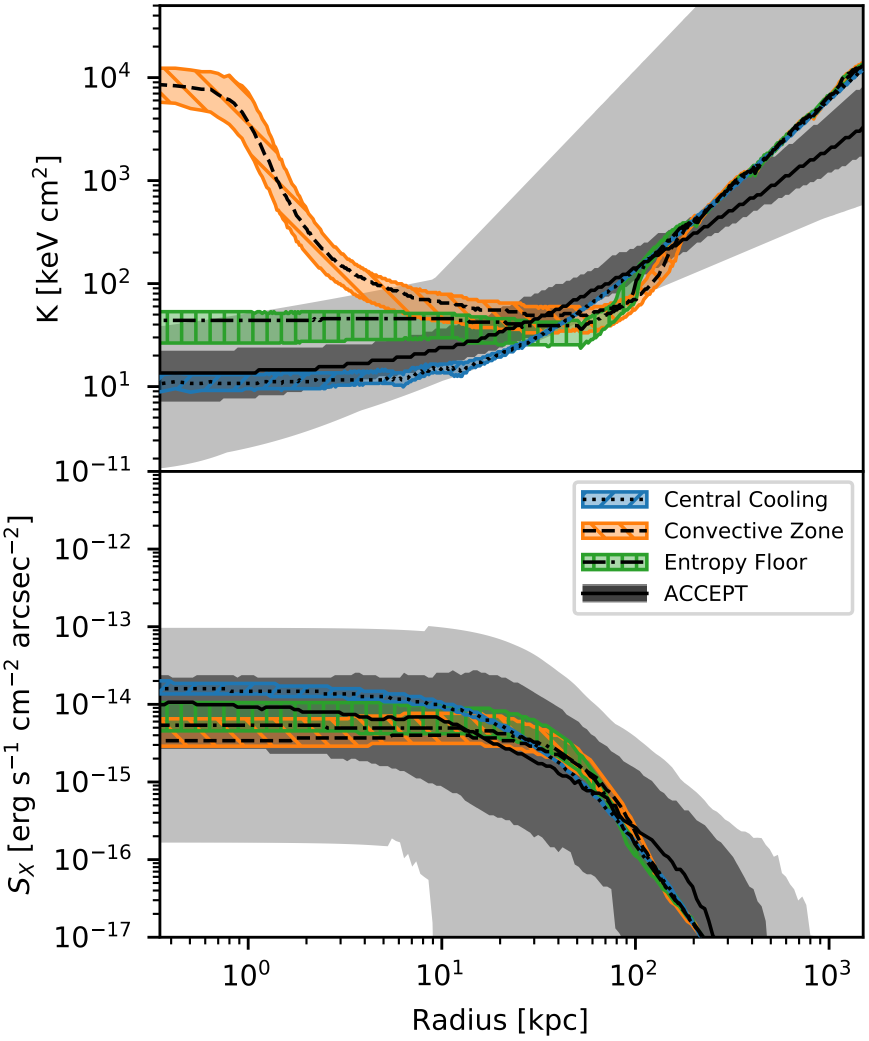


Figure 2.8: **Top:** Time-averaged median entropy profiles of the simulated cluster halos in Figure 2.3. The dotted line shows the simulation with low central heating (α = 2.0, r_s = 8 kpc, r_c = 1000 kpc), and the blue shaded region around it shows the 1σ dispersion of its median profile over time. The dashed line shows the simulation with high central heating (α = 2.6, r_s = 1 kpc, r_c = 150 kpc), and the orange shaded region around it shows its 1σ dispersion. The dot-dashed line shows the simulation with intermediate central heating (α = 2.6, r_s = 12 kpc, r_c = 150 kpc), and the green shaded region around it shows its 1σ dispersion. In each case, entropy is weighted by the x-ray luminosity in the 0.5–2.0 keV band, to mimic data obtainable with *Chandra*. The median, 1σ interval, and full extent of the entropy profiles of clusters with less than 30 keV cm^2 from ACCEPT are shown in grayscale, using the broken power law fits from Cavagnolo et al. (2009) for the entropy profiles. **Bottom:** X-ray surface brightness in the 0.5–2.0 keV band for the same simulated halos, with shaded regions showing the 1σ dispersion and black lines showing the median. The median, 1σ interval, and full extent of the entropy profiles of CC clusters from ACCEPT are shown in grayscale, using surface brightness profiles derived from electron density and temperature profiles.

2.4.4 Comparison to Other Simulations

Thermal regulation of galaxy clusters by AGN jets has been studied previously through numerical simulation using many different models of AGN feedback. These approaches include injection of buoyant bubbles (Brüggen, 2003b; Hillel & Soker, 2016), magnetic fields (Li et al., 2006; Nakamura et al., 2006, 2007; Huarte-Espinosa et al., 2012), kinetic jets (Wu et al., 2015; Martizzi et al., 2016; Hahn et al., 2017; Meece et al., 2017), stochastic momentum feedback (Weinberger et al., 2017; Nelson et al., 2019), cosmic rays (Jubelgas et al., 2008; Butsky & Quinn, 2018), and turbulent heating (Gaspari et al., 2012a; Zhuravleva et al., 2014; Banerjee & Sharma, 2014), either explicitly or implicitly driven by the central SMBH. Some simulations have also used purely thermal feedback models like the model used in this work, to which we can compare.

Meece et al. (2017), the predecessor to this work, tested a AGN feedback model consisting of a precessing bipolar jet that injected kinetic and thermal energy. They tested different fractions of AGN feedback going into thermal heating versus the kinetic jet. For triggering the feedback they tested three different models: a cold gas triggering model from Li & Bryan (2014a), a boosted Bondi-like triggering, and a Booth and Schaye accretion model (Booth & Schaye, 2009). Like this work, Meece et al. (2017) found that AGN models with purely thermal feedback led to an overabundance of cold gas in the simulation core. However, their thermal feedback was limited to a small region around the AGN, less than 1 kpc in diameter. In their simulations, hot bubbles inflated via AGN heating at the cluster center buoyantly rose a short distance out of the center to 10 – 30 kpc and created a flatter entropy profile that was unstable to multiphase condensation and therefore failed to suppress large accumulations of multiphase gas. Many of the heating kernels tested in this paper rectify the problem of overly centralized heating but result in elevated core entropy beyond what is reasonable for a CC cluster. globally our heating prescription is no longer robust to the formation of cold gas.

The Rhapsody-G simulations of galaxy clusters explored cosmological zoom-in simulations with star formation and feedback (SFF) and supermassive black hole (SMBH) formation and feedback, using the Ramses Eulerian AMR code (Wu et al., 2015; Teyssier, 2002). In their AGN feedback prescription, mass accreted onto the SMBH following a density-boosted Bondi-Hoyle accretion rate (Booth & Schaye, 2009). Thermal energy was deposited into a small radius around the SMBH (Martizzi et al., 2016). Compared to CC cluster entropy profiles from the ACCEPT catalogue, CC clusters in the Rhapsody-G had lower central entropies, showing overcooling in the inner tens of kpc (Hahn et al., 2017).

Tremmel et al. (2017) presented the Romulus galaxy simulations using the ChaNGa smoothed particle hydrodynamics code and includes SMBH feedback and SFF models tuned to observations. Their SMBH feedback model had two free parameters: (1) the efficiency of the accretion rate onto the SMBH and (2) the gas coupling efficiency ϵ_c. These parameters were calibrated to produce galaxies with observed values of the stellar-mass to halo ratio, HI gas fraction as a function of stellar mass, galaxy specific angular momentum versus stellar mass, and the SMBH to stellar mass relation. Their simulations used a thermal-only feedback model that deposited feedback energy into the 32 gas particles nearest to the SMBH. Mass accretion was governed by a modified Bondi accretion rate. Gas cooling was suppressed when heated by the SMBH for a time step equal to the time step of the SMBH. This allowed energy to escape away from the SMBH, although it may not be physically realistic. This feedback model produced galaxies with regulated SFF compared to observation.

In the follow-up paper Tremmel et al. (2019) on the cosmological RomulusC simulations, the same SFF and SMBH feedback models were used in a zoom-in simulation of a single halo. In an isolated halo, purely thermal feedback from the SMBH led to a conic structure with a highly collimated jet-like outflow. The outflows evolved over time, changing in shape and direction with the angular momentum of the gas near the SMBH. Energy was carried out to large radii through the outflows, which suppressed cooling at large radii. Star formation rates were regulated and matched observed rates in clusters. Additionally, the entropy profile of the clusters was within the range of observed profiles in CC clusters. Although the outflows were not explicitly introduced by their feedback prescription, their ability to transport AGN feedback energy tens of kiloparsecs from the center without inverting the large-scale entropy profile and overstimulating thermal instability is the key to proper thermal regulation of their simulated CC cluster.

2.4.5 Implications

Since the heating kernels explored here failed to produce quasi-stable CC clusters with realistic entropy profiles, extrapolations to real CC clusters may not be accurate. However, a few lessons can be drawn from these simulations:

- In the context of purely thermal AGN feedback, feedback that is highly centrally concentrated and tied directly to the global radiative cooling rate produces cores with entropy levels that greatly exceed those of observed CC clusters and in some cases are physically unreasonable.
- When the total heating rate is directly tied to the total cooling rate in the halo, rapid cooling of gas into cold clumps causes the heating rate to reach unphysically high levels. In comparison, in simulations using Bondi accretion or cold gas accretion such as in Meece et al. (2017) AGN feedback increases more gradually with the formation of cold gas, allowing feedback energy output to tune itself to physically reasonable values.
- The heating kernels considered here, in which heating per unit volume had a fixed radial distribution, were unable to maintain thermal stability of the cluster halo. In cases where a cold clump of gas formed, the purely thermal AGN feedback was insufficient to disrupt the clump without injecting unphysically high amounts of energy. The thermal heating in these simulations was unable to reproduce the effects caused by kinetic outflows from AGN jets such as in Meece et al. (2017).

A spherically symmetric heating kernel for purely thermal feedback that satisfies all of our criteria may exist but would need to have different parameters than are explored here. Such an idealized heating kernel would be useful to efficiently include AGN feedback in cosmological simulations.

2.4.6 Other Models Investigated

In search of a satisfactory heating kernel, we investigated several extensions to the spherically symmetric ones described in Section 2.2. First, we applied a polar angle dependence of $\cos^2\theta$ to mimic the conical distribution of heat from a kinetic jet. Total heating remained linked to total cooling. However, decreased heating near the equatorial plane leads to cold gas forming several tens of Myr sooner than for the corresponding spherical kernel and did not change the general behavior of the cooling catastrophe. Next, we tried a model in which cold gas was removed from the center of the simulation as it formed, to decrease the central density, potentially avoid fluid discontinuities in the fluid solver, and allow robust simulations with the formation of cold gas. However, explosive heat input triggered by the formation of cold gas still causes the hydrodynamics solver to fail. We also tested equating total heating to total cooling of only the warm gas, testing separately temperature thresholds of $10^{6.5}$ K and 10^7 K, to exclude the rapid cooling of cold gas and avoid explosive AGN feedback. However, this filtering of cold gas in the calculation of the heating rate leads to more cold gas forming and the leftover warm gas having an elevated central entropy. In some cases the heat input is still great enough to halt the simulation because of the Courant condition. Lastly, we tried smoothing out the rise in AGN heating by setting the total feedback to the average of the cooling rate over the last 50 Myr, in essence implementing a temporal kernel as well as a spatial kernel. However, this approach also leads to high rates of formation of cold gas due to the delayed heating response, as well as an eventual spike in AGN heating since the cooling catastrophe ultimately is not counteracted.

2.4.7 Future Models

There remain conceivable modifications to this heating kernel approach that we did not investigate, but which could produce more physically realistic CC clusters. For example, total heating could be capped at a physically reasonable value to avoid the overheating that coincides with the formation of cold gas. Additionally, we could investigate a radially piecewise conic feedback kernel in which AGN heating is spherically symmetric at small radii and conical at large radii. Another alternative would be a kernel with a spatial distribution that depends on the total heat input, adjusting to spikes in heating/cooling by distributing increased heating over a larger volume, as would happen with an increase in total jet power.

2.5 Summary

We have presented simulation results for simplified models of AGN feedback using heating kernels for purely thermal feedback. In those kernels, heat input has a spatial dependence following a radial power law $\dot{e} \propto r^{-\alpha}$ having a smoothing length r_s at small radii, an exponential cutoff radius r_c at large radii, and a total heating rate set equal to the total cooling rate measured within the cluster halo. This approach differs from previous simulations approximating feedback rates using Bondi and cold gas accretion models, which can temper the feedback response but are computationally more expensive. Our intention was to identify a heating kernel that would be both computationally inexpensive and able to maintain the hot atmosphere of a galaxy cluster in realistic quasi-steady state.

All of the heating kernels we tested failed to maintain a quasi-steady state with an entropy profile consistent with those observed among cool-core clusters (see Figures 2.3 and 2.7). We compared entropy profiles from our simulations to observational data from the ACCEPT dataset. Some simulations exhibit small to large central peaks in entropy that differ significantly from the entropy profiles seen in the ACCEPT sample. The central entropy peaks are most pronounced in simulations with highly centralized feedback. Simplified AGN models with overly centralized thermal heating therefore do not produce realistic entropy profiles.

A few lessons can be drawn from this work. Thermalization of AGN feedback energy must occur over a large region in order for the entropy profiles of simulated clusters to agree with those of observed cool-core clusters. However, it is difficult to distribute thermal feedback over a large region while also preventing a cooling catastrophe. Also, requiring total heating to equal total cooling becomes particularly problematic near the onset of a cooling catastrophe, because the increased cooling rate during the formation of large clumps of cold gas raises the heating rate to very high levels.

No configuration of purely thermal feedback explored here achieved thermal stability nor prevented a run away collapse into a cold clump, in contrast to simulations that introduce feedback energy in the form of kinetic jets. A heating kernel for purely thermal AGN feedback that produces realistic CC clusters may still exist but would need to significantly differ from the kernels we tested. Such a heating kernel that functions as an accurate and efficient proxy for more complex AGN feedback physics would allow larger cosmological simulations without increasing resolution.

CHAPTER 3

MAGNETIZED DECAYING TURBULENCE IN THE WEAKLY COMPRESSIBLE TAYLOR-GREEN VORTEX

This chapter first appeared as the published paper Glines et al. (2021). I include the original abstract as the introduction to this chapter.

Chapter Abstract

Magnetohydrodynamic turbulence affects both terrestrial and astrophysical plasmas. The properties of magnetized turbulence must be better understood to more accurately characterize these systems. This work presents ideal MHD simulations of the compressible Taylor-Green vortex under a range of initial sub-sonic Mach numbers and magnetic field strengths. We find that regardless of the initial field strength, the magnetic energy becomes dominant over the kinetic energy on all scales after at most several dynamical times. The spectral indices of the kinetic and magnetic energy spectra become shallower than $k^{-5/3}$ over time and generally fluctuate. Using a shell-to-shell energy transfer analysis framework, we find that the magnetic fields facilitate a significant amount of the energy flux and that the kinetic energy cascade is suppressed. Moreover, we observe nonlocal energy transfer from the large scale kinetic energy to intermediate and small scale magnetic energy via magnetic tension. We conclude that even in intermittently or singularly driven weakly magnetized systems, the dynamical effects of magnetic fields cannot be neglected.

3.1 Introduction

Magnetized turbulence is present in many terrestrial and astrophysical plasmas. Turbulence in magnetohydrodynamics (MHD) has been studied extensively over recent decades, from experimental, theoretical, and numerical perspectives, as the field continues to work towards a full understanding of magnetized turbulent plasmas. However, much of the theoretical and numerical work focuses on continuously driven plasmas, where a continuous (although potentially stochastic) force adds energy to the plasma, resulting in stationary turbulence. In many natural systems, the turbulence can be intermittently driven by infrequently occurring events or initialized from the initial conditions. For example, in the circumgalactic medium (CGM), the hot diffuse gas surrounding galaxies, or in the intracluster medium (ICM), the plasma in galaxy cluster that accounts for the majority of baryonic mass, turbulence can be introduced by various mechanisms. These include mergers with other galaxies, brief increases in the birth rate of stars, temporary outflows from jets driven by gas accreting onto supermassive black holes, supernovae, and many more transient events (Norman & Bryan, 1999; Larson, 1981; Britzen et al., 2017; Korpi et al., 1999). In pulsed power plasmas such as in a z-pinch, the plasma is driven by a single initial event and then allowed to decay into turbulence as kinetic and magnetic energy in the plasma dissipate into heat (Rudakov & Sudan, 1997; Kroupp et al., 2018). Therefore, to bridge the gap between observed, intermittently driven turbulent systems and theories of stationary MHD turbulence, we can study the behavior of decaying magnetized turbulence in an idealized environment.

In decaying turbulence, the turbulent flow arises purely from the initial conditions in the absence of a continuous driving force that injects energy. Essentially, the driving force is a delta function forcing at the initialization of the flow. The absence of external forces can avoid some of the shortfalls of driven turbulence simulations. As an example of these shortfalls, previous studies have shown that seemingly unimportant driving parameters such as the autocorrelation time and normalization of the driving field can bias plasma properties in turbulence simulations, in some cases affecting the scaling of the energy spectra (Grete et al., 2018). In addition, the driving forces contaminate the driven scales, making studies of turbulent plasma properties on those scale difficult to interpret. Simulations of decaying turbulence with fixed initial conditions avoid these issues since there are no driving forces.

The Taylor-Green (TG) vortex provides a useful set of smooth initial conditions that devolve into a turbulent flow. It was first proposed by Taylor & Green (1937) as an early mathematical exploration of the development of the turbulent cascade in a three dimensional hydrodynamic fluid. In the modern era, it is a canonical transition-to-turbulence problem also used for validation and verification of numerical schemes (Wang et al., 2013). From a physics point of view, the TG vortex has been explored from numerous angles, including numerical simulations of inviscid and viscous incompressible hydrodynamics with an emphasis on the development of small scale structures through vortex stretching (Brachet et al., 1983). Multiple configurations for TG vortices with magnetic fields were proposed in Lee et al. (2008) in order to study decaying turbulence in incompressible MHD. The new magnetic field configurations maintain all of the symmetries of the original hydrodynamic flow (Lee et al., 2008), and later works (Lee et al., 2010; Pouquet et al., 2010; Brachet et al., 2013) used these symmetries to save computational resources and allow more highly resolved simulations of the vortex. These simulations produced differing k^{-2}, $k^{-5/3}$, and $k^{-3/2}$ spectra depending on the initial magnetic field, where the k^{-2} spectra was speculated to be due to weak turbulence. Later work by Dallas & Alexakis (2013a,b) investigated the mechanism behind the different spectra. They concluded that the k^{-2} spectra produced by one configuration of the magnetic field was due to magnetic discontinuities in the plasma and not weak turbulence as previously thought. In Dallas & Alexakis (2013c), perturbations added to the initial conditions lead the symmetries of the TG vortex to break and the k^{-2} spectra to dissipate to shallower $k^{-5/3}$ spectra. A similar problem using the hydrodynamic initial configuration of the TG vortex but with an Orszag-Tang magnetic field was studied in imcompressible resistive MHD by Vahala et al. (2008), where a $k^{-5/3}$ energy spectra was found in their simulations.

All of these studies are concerned with incompressible turbulence, whereas many astrophysical systems (such as the interstellar, circumgalactic, intracluster, and intergalactic media) are comprised of compressible magnetized plasmas. To our knowledge, the formulation of the TG vortex from Lee et al. (2008) remains unexplored in the compressible MHD regime. Moreover, there have been recent advances in analytical tools to study the transfer of energy between reservoirs in compressible MHD (Yang et al., 2016; Grete et al., 2017). Energy transfer analysis enables measurement of the flux of energies between length scales within and between the kinetic, magnetic, and thermal energies of the plasma. In a compressible ideal MHD plasma, energy can be redistributed within the kinetic and within the magnetic energy budget via advection and compression. Moreover, magnetic tension can facilitate energy transfer between kinetic and magnetic energies as vortical motion in the turbulent plasma contributes to magnetic fields and magnetic fields constrain the motion of the plasma. In turbulent flow, intra-budget energy transfers via advection and compression typically manifest from a larger scale to a smaller but similar scale (i.e., “down scale-local"), defining the turbulent cascade. Inter-budget energy transfer via, e.g., magnetic tension, complicates the picture of a turbulent cascade as it moves energy between reservoirs and potentially allows for nonlocal transfer of energy from large scales directly to much smaller scales. Given the transient nature of the TG vortex, we expect the energy transfers to change over time as, e.g., the ratio of kinetic to magnetic energy evolves over time or due to the development of increasingly small-scale structure. This is in contrast to stationary turbulence where the dynamics remain constant over time in a statistical sense.

For these reasons, we focus on a detailed study of the dynamics in the magnetized, weakly compressible Taylor-Green vortex. Moreover, to explore magnetized decaying turbulence in different regimes we present nine simulations of the TG vortex probing all combinations of three different initial ratios of kinetic to magnetic energy (1, 10, and 100, corresponding to initial Alfvénic Mach numbers of $\mathcal{M}_A = \{1, 3.2, 10\}$) and three different initial fluid velocities (initial root mean squared, or RMS, sonic Mach numbers of $\mathcal{M}_{s,0} = \{0.1, 0.2, 0.4\}$). Thus, we explore strongly and weakly magnetized, subsonic plasmas in which density perturbations are present but limited.

To summarize our results, we find that magnetic fields significantly influence the decaying turbulence in the plasma regardless of the initial field strength. In all cases, we find that at late times the magnetic dynamics dominate kinetic dynamics even if the initial magnetic energy is 100 times smaller than the kinetic energy. Moreover, the spectral indices of the kinetic and magnetic energies are not fixed in time but evolve from steep $\simeq k^{-2}$ spectra at earlier times to shallower $\simeq k^{-4/3}$ spectra at later times. Using the energy transfer analysis, we see that most energy transfer is dominated by magnetic field dynamics. This includes both energy flux from kinetic to magnetic energy via magnetic tension and the flux of energy within the magnetic energy budget via compression and advection. Overall, the kinetic energy cascade is effectively absent and the initial sonic Mach number ($\mathcal{M}_{s,0}$) only weakly affects the observed dynamics. We also see several transient phenomena during the transition to turbulence, including temporary inverse turbulent cascades in both the magnetic and kinetic energies and large nonlocal energy transfers between scales separated by up to two orders of magnitude from the kinetic to the magnetic energy.

We organize the paper as follows. In Section 3.2, we describe the simulation and analysis setup including numerical methods, detailed Taylor-Green vortex initial conditions, and the energy transfer analysis. In Section 3.3, we present results of the simulations (focusing on $\mathcal{M}_{s,0} = 0.2$) such as the bulk properties of the plasma, the evolution of the energy spectra, and the transient behaviors seen through the energy transfer analysis:Section 3.4, we discuss our findings in the broader context of magnetized turbulence and astrophysical plasmas and conclude in Section 3.5 with a summary of our key findings. The online supplementary materials for this paper contain detailed plots of the results of all initial $\mathcal{M}_{s,0}$.

3.2 Method

3.2.1 MHD Equations and Numerical Method

The equations for compressible ideal MHD plasma can be written as a hyperbolic system of conservation laws. In differential form the ideal MHD equations are

$$\begin{aligned}
\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0 \\
\partial_t \rho \mathbf{u} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \mathbf{B} \otimes \mathbf{B}) + \nabla \left(p + \mathbf{B}^2/2\right) &= 0 \\
\partial_t \mathbf{B} - \nabla \times (\mathbf{u} \times \mathbf{B}) &= 0 \\
\partial_t E + \nabla \cdot \left[\left(E + p + \mathbf{B}^2/2\right) \mathbf{u} - (\mathbf{B} \cdot \mathbf{v})\, \mathbf{B}\right] &= 0
\end{aligned}$$

where ρ is the density, $\mathbf{u}$ is the flow velocity, $\mathbf{B}$ is the magnetic field (that includes a factor of $1/\sqrt{4\pi}$), p is the thermal pressure, and E is the total energy density. We close the system of equations with the equation of state for an adiabatic ideal gas with

$$p = \rho\,(\gamma - 1)\,e$$

where γ is the ratio of specific heats and e is the internal energy found from

$$E = \rho\left(\frac{1}{2}\mathbf{u}\cdot\mathbf{u} + \frac{1}{2}\mathbf{B}\cdot\mathbf{B} + e\right).$$

We use the open source K-Athena Grete et al. (2021a) astrophysical MHD code, which is a performance portable version of Athena++ Stone et al. (2020a) using the Kokkos performance portability library Carter Edwards et al. (2014). K-Athena uses an unsplit finite volume Godunov scheme to evolve the ideal MHD equations originally presented and implemented in Athena Stone & Gardiner (2009). The method consists of a second-order Van Leer predictor-corrector integrator with piecewise linear reconstruction (PLM) and HLLD Riemann solver, and constrained transport to preserve a divergence-free magnetic field.

3.2.2 Magnetized TG Vortex

The TG vortex was first proposed by Taylor & Green (1937) as a mathematical exploration of the development of hydrodynamic turbulence in 3D. The initial flow was made to be periodic and symmetrical in order to accommodate simple approximations to a solution. There exist a number of different formulations. We follow the setup described in Wang et al. (2013) for the hydro variables and Lee et al. (2008) for the initial magnetic field configuration.

The simplest hydrodynamic setup of a TG vortex begins with a periodic field of fluid velocity in the xy-plane and periodic pressure and density field with constant sound speed throughout the domain. Using a cubic periodic domain with side length $2\pi L$, the initial fluid velocity is set to

$$\begin{aligned}
u_x &= u_0 \sin\frac{x}{L}\cos\frac{y}{L}\cos\frac{z}{L}\\
u_y &= -u_0 \cos\frac{x}{L}\sin\frac{y}{L}\cos\frac{z}{L}\\
u_z &= 0
\end{aligned}$$

where u_0 is the maximum initial velocity. Note that in this formulation the initial flow velocity is confined to the xy-plane. The initial pressure and density are set to

$$\begin{aligned} P &= P_0 + \frac{\rho_0 u_0^2}{16}\left(\cos\frac{2x}{L} + \cos\frac{2y}{L}\right)\left(\cos\frac{2z}{L} + 2\right) \\ \rho &= P\rho_0/P_0 \end{aligned}$$

so that P and ρ are proportional to each other. This means that the sound speed

$$c_s = \sqrt{\gamma P/\rho} = \sqrt{\gamma P_0/\rho_0}$$

is initially constant throughout the domain.

The root mean square (RMS) of the initial Mach number is related to u_0 by

$$\mathcal{M}_{s,0} = \frac{u_0}{2c_s}.$$

For simplicity, we set $P_0 = 1$ and $\rho_0 = 1$. We assume the fluid is a monatomic ideal gas with an adiabatic index $\gamma = 5/3$. The resulting total initial kinetic energy is

$$E_{U,0} = \rho_0 u_0^2 (\pi L)^3 \quad . \tag{3.1}$$

Magnetic fields were first added to the TG vortex in Lee et al. (2008) with the express constraint of preserving the same symmetries of the hydrodynamic flow. Here, we follow the proposed insulating configuration so that currents are confined to πL boxes, e.g., the cube $[0, \pi L]^3$ forms an insulating box. The corresponding initial magnetic fields are given by

$$\begin{aligned} B_x &= B_0 \cos\frac{x}{L}\sin\frac{y}{L}\sin\frac{z}{L} \\ B_y &= B_0 \sin\frac{x}{L}\cos\frac{y}{L}\sin\frac{z}{L} \\ B_z &= -2B_0 \sin\frac{x}{L}\sin\frac{y}{L}\cos\frac{z}{L} \end{aligned}$$

where B_0 is the initial magnetic field strength. In practice, we initialize the magnetic field from the magnetic vector potential $\mathbf{A}$

$$\begin{aligned} A_x &= -B_0 \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right) \\ A_y &= B_0 \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right) \\ A_z &= 0 \end{aligned}$$

using $\mathbf{B} = \nabla \times \mathbf{A}$. This guarantees $\nabla \cdot \mathbf{B} = 0$ to machine precision in the initial conditions, which is then preserved by the constrained transport algorithm throughout the simulation. The total initial magnetic energy is

$$E_{B,0} = 3B_0^2 (\pi L)^3 \tag{3.2}$$

so that the initial ratio of kinetic to magnetic energy is

$$\frac{E_{U,0}}{E_{B,0}} = \frac{\rho_0 u_0^2}{3B_0^2}. \tag{3.3}$$

Since the magnetic field is zero is some regions of the domain, the Alfvénic Mach number $\mathcal{M}_A = u\sqrt{\rho}/B$ is also undefined in some regions. For this reason, we use a proxy based on the mean energies for the Alfvénic Mach number

$$\mathcal{M}_A := \sqrt{\langle E_U \rangle / \langle E_B \rangle} \tag{3.4}$$

throughout the rest of the paper. We also adopt a similar proxy for the plasma β (ratio of thermal to magnetic pressure)

$$\beta := \frac{2}{\gamma}\frac{\mathcal{M}_A^2}{\mathcal{M}_s^2} \tag{3.5}$$

where $\mathcal{M}_s$ is the RMS of the sonic Mach number.

The hydrodynamic and magnetic initial conditions exhibit a number of symmetries that are maintained throughout the simulation. In each of the three dimensions there are two planes across which the fluid is antisymmetric. For our setup, these are planes through $x = 0$ and $x = \pi L$; planes through $y = 0$ and $y = \pi L$; and planes through $z = 0$ and $z = \pi L$. Additionally, the flow is rotationally symmetric through a rotation of π around the two axes $x = z = \pi L/2$ and $x = z = \pi L/2$ and rotationally symmetric through a rotation of $\pi/2$ around the axis $x = y = \pi L/2$. These symmetries are more thoroughly explored in Lee et al. (2008).

We explore the transition to magnetized turbulence and the following decay in different regimes with our simulation suite of TG vortices and focus on two parameters: the initial RMS Mach number using $\mathcal{M}_{s,0} = \{0.1, 0.2, 0.4\}$ and the initial ratio of kinetic to magnetic energy using $E_{U,0}/E_{B,0} = \{1, 10, 100\}$, or alternatively, the initial RMS Alfvénic Mach number $\mathcal{M}_{A,0} = \{1, 3.2, 10\}$. We simulate all nine combinations of the three values of these two parameters. Throughout the rest of the text, we use `MsX` to refer to simulations with $\mathcal{M}_{s,0} = X$ and `MaY` to refer to simulations with $\mathcal{M}_{A,0} = Y$.

The initial magnetic field amplitude B_0 is obtained from Equation 3.3 using given a specific value of $\mathcal{M}_{s,0}$ and $\mathcal{M}_{A,0}$. All simulations employ a cubic $[-0.5, 0.5]^3$ domain with periodic boundaries, with $L = \frac{1}{2\pi}$ to be consistent with the definition of the initial condition that is presented above. We use a uniform Cartesian grid with $1{,}024^3$ cells. The characteristic length scale of the initial vortices is πL, so that we define

$$T = \frac{\pi L}{u_0}$$

as the dynamical time [1] In order to evolve the simulations for sufficient time to allow a turbulent flow to form and evolve, we run each simulation for ≈ 6 dynamical times.

In our results, we present all measurements of time in terms of the dynamical time T and all measurements of wavenumber in terms of $1/L$. Unless otherwise noted, all other results are in terms of simulation units.

3.2.3 Energy Transfer Analysis

In order to probe the movement of energy between different energy reservoirs, we use the shell-to-shell energy transfer analysis from Grete et al. (2017), which extends the framework presented

[1] Note that other works such as Wang et al. (2013); Pouquet et al. (2010) use a nondimensional time, $t^* = L/u_0$, in contrast to the dynamical time used here.

in Alexakis et al. (2005) to the compressible regime.

The total transfer of energy from some shell Q in energy reservoir X to some shell K in reservoir Y is denoted by

$$\mathcal{T}_{XY}(Q,K) \quad X,Y \in [U,B] \tag{3.6}$$

where we use U and B to denote the kinetic and magnetic energy reservoirs, respectively.

In this work we focus on the energy transfer within the kinetic and magnetic energy reservoirs via advection and compression which are respectively

$$\begin{aligned}
\mathcal{T}_{UU}(Q,K) = & -\int \mathbf{w}^K \cdot (\mathbf{u}\cdot\nabla)\, \mathbf{w}^Q d\mathbf{x} \\
& -\tfrac{1}{2}\int \mathbf{w}^K \cdot \mathbf{w}^Q \nabla \cdot \mathbf{u} d\mathbf{x} \\
\mathcal{T}_{BB}(Q,K) = & -\int \mathbf{B}^K \cdot (\mathbf{u}\cdot\nabla)\, \mathbf{B}^Q d\mathbf{x} \\
& -\tfrac{1}{2}\int \mathbf{B}^K \cdot \mathbf{B}^Q \nabla \cdot \mathbf{u} d\mathbf{x}
\end{aligned}$$

and the energy transferred from kinetic energy to magnetic energy via magnetic tension (and vice versa) given by

$$\mathcal{T}_{UBT}(Q,K) \;=\; \int \mathbf{B}^K \cdot \nabla \left(\mathbf{v}_A \otimes \mathbf{w}^Q\right) d\mathbf{x} \tag{3.7}$$

$$\mathcal{T}_{BUT}(Q,K) \;=\; \int \mathbf{w}^K \cdot (\mathbf{v}_A \cdot \nabla)\, \mathbf{B}^Q d\mathbf{x}\,. \tag{3.8}$$

Here we use the mass weighted velocity $\mathbf{w} = \sqrt{\rho}\mathbf{u}$ so that the spectral energy density is positive definite Kida & Orszag (1990), and $\mathbf{v}_A$ is the Alfvénic wave speed.

The velocity $\mathbf{w}^K$ and magnetic field $\mathbf{B}^K$ in a shell K (or Q) are obtained using a sharp spectral filter in Fourier space. The shell bounds are logarithmically spaced and given by 1 and $2^{n/4+2}$ for $n \in \{-1, 0, 1, \ldots, 32\}$. Shells (uppercase, e.g., K) and wavenumbers (lowercase, e.g., k) obey a direct mapping, i.e., $K = 24$ corresponds to the logarithmic shell that contains $k = 24$, i.e., $k \in (22.6, 26.9]$.

3.3 Results

In this section we present results of the Taylor-Green vortices we simulated, showing bulk properties of the fluid (Section 3.3.1), including the evolution of the different energy spectra. These results demonstrate that the kinetic, magnetic, and thermal energy reservoirs interact with each other in a manner that depends significantly on the initial strength of the magnetic field. The energy spectra evolves to a turbulent cascade over 1-2 dynamical times and then stays there for the remainder of the simulation. In Section 3.3.2, we examine in detail the transfer of energy between different energy reservoirs, including the transient behaviors we observed in the simulations. We see robust transfer of energy at all scales within the kinetic and magnetic energy reservoirs when examined separately, as well as complex and time-varying nonlocal transfer of energy between the kinetic and magnetic energy reservoirs, including evidence for an intermittent inverse turbulent cascade. Since the initial Mach number had much less of an effect on the results compared to the initial ratio of kinetic to magnetic energy, we focus on results using only the three `Ms0.2` simulations as reference. We provide more complete plots of all nine simulations spanning all Mach numbers in the online supplements.

Starting with a visual demonstration of the TG vortex, Figure 3.1 shows the sonic Mach number and magnetic pressure from the `Ms0.2_Ma10` simulation after 0.77 dynamical times and after 5.16 dynamical times in a slice in the $xy-$plane through the origin. Only one quadrant of the xy-place is shown, as it exhibits symmetry across 4 quadrants in the xy-plane. From the slice plot, we can see that the TG vortex begins as a smooth vortical flow and magnetic field. After several dynamical times, the smooth flow devolves into a chaotic magnetized turbulent flow. Kinetic and magnetic structures at all scales persist throughout the simulation, as will be shown in energy spectra later in this work.

3.3.1 Bulk Properties

3.3.1.1 Evolution of energy reservoirs

Figure 3.2 shows the total kinetic, magnetic, and thermal energies and the dimensionless RMS sonic Mach number $\mathcal{M}_s$, Alvénic Mach number $\mathcal{M}_A$, and plasma beta β of the `Ms0.2` simulations as a function of time. In this figure, we can see that in all simulations kinetic and magnetic energy convert into thermal energy over time. This decay into thermal energy is not immediate; rather, it requires at least one dynamical time to begin (i.e., it is observed to occur at a minimum of $t = 1T$ in all simulations). In the `Ma1` simulations, due to the initial conditions there is even a small transient transfer of thermal energy into kinetic and magnetic energies. After $t = 2T$, all simulations dissipate kinetic and magnetic energy into thermal energy. The sonic Mach number generally decreases by less than a factor of 4 over time from its initial 0.2 value, and β remains high (from $\gtrsim 20$ for `Ms0.2_Ma1` to $\gtrsim 100$ for `Ms0.2_Ma10`) throughout the simulations.

In all cases, the flow becomes dominated by magnetic energy (i.e., become sub-Alfvénic with $\mathcal{M}_A < 1$) at different dynamical times depending on the initial ratio of kinetic to magnetic energy and mostly independent of the initial Mach number. In other words, even for the simulations with initially 100 times more kinetic than magnetic energy (`Ma10`), in the final state the magnetic energy dominates over the kinetic energy. This already highlights the importance of kinetic to magnetic energy transfer. The initial growth of magnetic energy is characteristic of the insulating magnetic field configuration and is seen in other works on the TG vortex Lee et al. (2010). This behavior of the magnetic field is likely due to the magnetic fields and vorticity beginning parallel to each other everywhere. All simulations experience a peak in the magnetic energy evolution before $t = 3T$ depending on the initial magnetic energy. At $t = 6T$, all simulations are still losing total kinetic and magnetic energy to thermal energy, although the rate of energy dissipation is slowing by the simulation end. The magnetic and kinetic energies also become similar in magnitude, cf., $\mathcal{M}_A \simeq 1$.

The `Ms0.2_Ma1` simulation displays notably different behavior than those where the kinetic energy initially dominates. In particular, we observe periodic exchanges of energy between these two reservoirs before the bulk of the energy is converted into heat, rather than a smooth transfer of energy from the kinetic to magnetic reservoir, followed by a decline of both as the flow thermalizes. At approximately $t = 1T$, more than five times as much energy is stored in the magnetic reservoir as compared to the kinetic reservoir, which is in stark contrast with other calculations. These results suggest that the large initial magnetic field facilitates a more rapid transfer of kinetic energy, which will be examined in more detail later in this paper. For reference, we also plot the temporal evolution of the energies in the incompressible, magnetized Taylor-Green vortex with `Ma=1` presented in

Pouquet et al. (2010) in the top left panel of Fig. 3.2 next to our `Ms0.2_Ma1` results. The evolution in (Pouquet et al., 2010) covers the first oscillation and is in good agreement with our simulation. Finally, the oscillations observed in the energy reservoirs for the `Ma1` simulations in general have a period that depends on the initial Mach number, which can be seen in the figures that we leave for the online supplements.

3.3.1.2 Energy Spectra

Figure 3.3 shows the temporal evolution of the kinetic and magnetic energy spectra of the three `Ms0.2` simulations, compensated by $k^{4/3}$, which demonstrates how both the kinetic and magnetic energy spectra change from the smooth initial large scale flow to fully developed turbulence. The top row shows the three simulations earlier in the evolution ($t = 0.77T$), when the spectra are still steep with large scale structure from the initial conditions. In the case of the strongest initial magnetization (`Ma1`), the magnetic energy is larger than the kinetic energy on all scales and their spectral scaling is comparable. For `Ma3.2` and `Ma10` the kinetic energy spectrum is steeper than the magnetic one. The spectra cross at $k \simeq 7$ and $k \simeq 20$, respectively, so that the kinetic energy is still dominant on large scales. The middle row in Figure 3.3 shows intermediate times with `Ms0.2_Ma1` at $t = 1.29T$, which is the time that is discussed in Section 3.3.2.2 and `Ms0.2_Ma3.2` and `Ms0.2_Ma10` simulations at $t = 1.81T$, which is the time is discussed in Section 3.3.2.1. Note that the spectra are still evolving at this intermediate stage. In the `Ms0.2_Ma10` simulation at $t = 1.81T$, the magnetic spectra has reached a $k^{-4/3}$ spectrum while the kinetic spectra shows a broken power law with excess energy at larger length scales. In both `Ma1` and `Ma3.2` the magnetic energy is now dominant on effectively all scales (with the exception of the noisy part of the spectrum at the largest scales, $k \lesssim 4$). The bottom row shows all three `Ms0.2` simulations at $t = 5.16T$. Here, the magnetic energy is effectively dominant on all scales in all simulations and the kinetic and magnetic spectra exhibit a scaling close to $k^{-4/3}$. The spectral indices still fluctuate, which we explore in Section 3.3.1.3.

In Figure 3.4 we show the kinetic and magnetic energy at specific wavenumbers and compensated by $k^{4/3}$ plotted over time. At early times (before $t = 2T$) the large scale ($k = 8$) kinetic energy shows the fastest growth rate compared to smaller scales as expected from an initial entirely large scale configuration. The kinetic energy at $k = 8$ peaks between $t = 1T$ and $t = 2T$ with larger initial magnetic field leading to an earlier peak. The magnetic energy at $k = 8$ in the `Ms0.2_Ma1` simulation oscillates throughout the duration of the simulation, with the kinetic energy oscillating once. No oscillatory behavior is observed in `Ms0.2_Ma3.2` and `Ms0.2_Ma10` for these quantities. From this plot we can also see that the small scale ($k = 128$) energies saturate at $t \simeq 1T$, $t \simeq 1.5T$, and $t \simeq 2.5T$, respectively.

3.3.1.3 Spectral Index

We measured the spectral indices of the kinetic and magnetic energy spectra α by fitting a power-law $E \propto k^{\alpha}$ to the energy spectra of each reservoir at each time step. For the inertial range of wavenumbers across which we fit the power-law to the spectra, we used wavenumbers $k = 10$ to $k = 32$. We chose this inertial range because very little large scale structure persists below $k = 10$ and wavenumbers above $k = 32$ are not entirely free of numerical dissipation any more. The kinetic and magnetic spectral indices measured across the inertial range are not fixed in time across the different simulations, with the most variation being due to initial magnetic energy. Figure 3.5 shows the spectral indices of the kinetic, magnetic, and sum of kinetic and magnetic energy spectra over time for the `Ms0.2` simulations. In all simulations, the spectral index evolves over time, decaying from the initial steep spectral index ($\alpha \lesssim -2$) as energy is transferred to small scales. The kinetic and magnetic spectral indices evolves separately in the calculations until the magnetic energy exceeds the kinetic energy, after which the spectral indices of the separate and combined reservoirs fluctuate within $\Delta\alpha \simeq 0.2$. The crossover of kinetic and magnetic energies happens immediately in the `Ms0.2_Ma1` simulation, early in the `Ms0.2_Ma3.2` simulation before $t = 2T$, and later in the `Ms0.2_Ma10` simulation at $t \simeq 4T$. After the kinetic and magnetic spectral indices reach rough parity and the magnetic field becomes dominant, both spectral indices reach comparable values and reach a rough constant $1 - 2$ dynamical times later, although they continue to vary over time. Since the magnetic fields in the Ma1 simulations immediately become dominant, the spectral indices reach a rough constant at $t \simeq 2T$, while in the Ma3.2 simulations they reach a rough constant at $t \simeq 4T$ and in the Ma10 simulations this happens at $t \simeq 5T$. The Ms0.2_Ma3.2 simulation experiences a brief peak in the spectral index around $t \simeq 1.5T$ while the flow is still in transition. This is also reflected in the large uncertainty of the spectral index during that time, e.g., the index of the kinetic energy spectrum varies between -1 and -2.25 by choosing slightly different fitting ranges (as indicated by the shaded blue bands in Fig. 3.5). Note that in the Ma10 case, the magnetic spectrum flattens and the spectral index reaches a roughly constant value much sooner than in the other two cases, at $t \simeq 2T$ when the kinetic energy still dominates. Later on in the Ma10 simulations, the kinetic spectral index becomes comparable to the magnetic spectral index. For the high initial magnetic field simulations, the spectral index levels out at about $\alpha \simeq -5/3$ while the initially kinetically dominated simulations level out at $\alpha \simeq -4/3$.

The final spectral indices depend on the initial ratio of kinetic to magnetic energy, with more magnetic energy leading to shallower magnetic spectra. The Ma1 simulations end with $\alpha \simeq -1.7$ (close to $-5/3$), Ma3.2 ends with $\alpha \simeq -1.3$ (close to $-4/3$), and Ma10 ends with slightly lower values of $\alpha \simeq -1.2$. In the presence of the stronger magnetic fields in the Ma1 simulations, the flattening of the spectra seems to be suppressed. Before the kinetic and magnetic spectral indices become comparable in each simulation, there is also greater variance in the spectral slope when measured using different inertial ranges. This indicates that a power-law might be a poor fit for the spectra at those early times, showing that the spectra is not fully developed until the magnetic energy is dominant. For example, as seen in Figure 3.3, the kinetic energy spectra appears as a broken power law at intermediate times, which is especially evident in the Ms0.2_Ma10 simulation at $t = 1.81T$ to a lesser extent the Ms0.2_Ma1 simulation at $t = 1.29T$ and the Ms0.2_Ma3.2 simulation at $t = 1.81T$. Oscillations in the spectral index of the Ma1 simulations also appear, whose period seems to be linked to the initial Mach number, with larger Mach numbers leading to a smaller period of oscillation.

We note that between the three values of $\mathcal{M}_A$, the simulations shown here exhibit a wide variety of behaviors, highlighted by the spectral indices in Fig. 3.5. More simulations with intermediate values of $\mathcal{M}_A$ would be required to determine if the transition between these behaviors is smooth or abrupt.

3.3.2 Energy Transfer

While the total energy and spectra of the kinetic and magnetic reservoirs can broadly describe the isolated behavior of the different energy reservoirs, examining the energy transfer within and between reservoirs using the analysis described in Section 3.2.3 can provide deeper insights into the physical phenomena, including demonstrating the mechanisms that are responsible for the transfer of energy. The shell-to-shell energy transfer fluxes examined in this section demonstrate the flux from wavenumber Q to wavenumber K within and between energy reservoirs via different pathways.

Figure 3.6 shows the energy transfer *within* the kinetic (left) and magnetic (right) energy reservoirs via advection and compression in the `Ms0.2_Ma1` simulation at $t = 0.77T$ (top) and at $t = 5.16T$ (bottom). This plot encapsulates the energy transfer of a turbulent cascade. Near the beginning of the simulation in the top panels, most of the energy is in large scale modes, with energy from larger Q wavenumbers moving to smaller K wavenumbers. Note that the energy transfer is constrained to the diagonal because the bulk of the energy transfer is local, occurring between comparable scales of Q to K. White space fills the off-diagonals because very little nonlocal energy transfer occurs internally within reservoirs. The energy transfer shown in this figure is solely within the kinetic and magnetic reservoirs – there is no energy transfer shown between these reservoirs (although it is occurring, as will be discussed in the next paragraph). In the simulation shown here, the magnetic energy transfer is larger in magnitude than the kinetic energy transfer. In all simulations, the magnetic energy transfer extends to higher wavenumbers more rapidly than the kinetic energy. After the flow has decayed into turbulence (as shown in the bottom panels), energy transfer to smaller local scales happens across the resolved modes down to numerical dissipation scales. At large wavenumbers ($Q > 16$), the energy transfers are scale-local and of comparable magnitude. This phenomenon continues to at least $Q \simeq 200$ in both the kinetic and magnetic energy transfer – i.e., to much larger wavenumbers than an inertial range is observed (see, e.g., Figure 3.3). Thus, the effective (numerical) viscosity and resistivity are not affecting the turbulent cascade encoded by these transfers to a significant degree.

Figure 3.7 shows the energy transfer within the kinetic (top) and magnetic (bottom) energy reservoirs in the `Ms0.2_Ma1` simulation at $t = 1.29T$ (just before the magnetic energy peaks). Energy transfer within the kinetic and magnetic reservoirs briefly reverses directions and moves energy from smaller local scales to larger local scales (note the purple color indicating energy loss above the diagonal and orange color below the diagonal, which is in contrast to Fig. 3.6). This constitutes a transient inverse cascade. Additionally, the inverse cascade is present throughout most scales of the magnetic energy ($K, Q \lesssim 100$) but only apparent at large scales in the kinetic energy ($K, Q \lesssim 16$). As seen in Figure 3.4, at this early time the turbulent flow is just beginning to saturate the smallest scales while the large scale energy oscillates, so the energy transfer inversion lasts less than a dynamical time (see Section 3.3.2.2 for further exploration of the duration).

Figure 3.8 shows the energy transfer *between* the kinetic to magnetic energy reservoirs due to magnetic tension at $t = 1.81T$ in the `Ms0.2_Ma10` simulation. This Figure displays nonlocal transfer from kinetic to magnetic energy. Unlike the advection- and compression-driven modes within the magnetic and kinetic energy reservoirs, energy transfers from kinetic to magnetic reservoirs via tension can support nonlocal energy transfers. The nonlocal transfer happens from large kinetic scales to much smaller magnetic scales, spanning more than an order of magnitude downward in spatial scale from the largest kinetic modes. The nonlocal energy transfer between kinetic and magnetic energy was significant in simulations with lower initial magnetic energy, and especially in the `Ma10` simulations where the magnetic field is dynamically unimportant at early times. Kinetic energy moves significant energy to all magnetic scales from early times at $t \simeq 1.5T$ to intermediate times at $t \simeq 4T$ in these simulations, although some energy continues to flow via this mechanism at later times. Additionally, since the transfer of energy via tension is between two different reservoirs, the energy transfer can transfer at equivalent scales from one reservoir to the other. This is shown as non-zero transfer along the diagonal of the plot.

3.3.2.1 Nonlocal Energy Transfer

Like in some driven turbulence simulations Alexakis et al. (2005); Grete et al. (2017), these decaying turbulence simulations also demonstrate significant nonlocal energy transfer between kinetic and magnetic energy reservoirs. Unlike in driven simulations, the energy transfers in this work are solely due to the fluid flow and not due to externally-applied driving forces. Figure 3.9 shows the total local, nonlocal, and equivalent-scale energy transfers via magnetic tension in the `Ms0.2` simulations over time. We obtain these quantities by integrating the transfer functions over different sets of scales:

$$\begin{array}{cl}
\text{Nonlocal lower} & \displaystyle\sum_{Q} \sum_{K\in[1,2^{-\ell}Q)} \mathcal{T}_{XY}\left(Q,K\right) \\
\text{Local-Lower} & \displaystyle\sum_{Q} \sum_{K\in[2^{-\ell}Q,Q)} \mathcal{T}_{XY}\left(Q,K\right) \\
\text{Equivalent} & \displaystyle\sum_{Q} \sum_{K=Q} \mathcal{T}_{XY}\left(Q,K\right) \\
\text{Local-Higher} & \displaystyle\sum_{Q} \sum_{K\in(2^{\ell}Q,Q]} \mathcal{T}_{XY}\left(Q,K\right) \\
\text{Nonlocal Higher} & \displaystyle\sum_{Q} \sum_{K\in(2^{\ell}Q,\infty]} \mathcal{T}_{XY}\left(Q,K\right)
\end{array}$$

where ℓ is a parameter for differentiating local versus nonlocal separation of wavenumbers in log space. In Figure 3.9, we show the analysis using $\ell = 5/4$ with a solid line, which corresponds to 5 logarithmic bins above or below Q (see 3.2.3 for the description of the binning), and show the extent of the fluxes if $\ell = 5/4 \pm 1/4$ is used in shaded regions. As seen in this figure from the red line, the nonlocal energy transfer from large scale kinetic modes to small scale magnetic modes ("downscale" transfer) is present in all simulations but is only dominant when the initial kinetic energy exceeds the initial magnetic energy – this nonlocal energy transfer is more significant in the `Ma3.2` and `Ma10` simulations. Nonlocal energy transfer downscale (red line) peaks depending on the initial magnetic field and in all cases before the total magnetic energy peaks. The nonlocal transfer helps fill out the magnetic energy spectrum faster than the kinetic energy spectrum, especially in the `Ma10` simulations, which is consistent with the spectral index shown in Figure 3.3 and the turbulent cascades shown in the shell-to-shell energy transfer in Figure 3.6. By the time the magnetic energy has exceeded the kinetic energy in the Ma3.2 and Ma10 simulations, nonlocal energy transfer is largely diminished due to the lack of kinetic energy to feed the transfer.

Local energy transfer downscale (orange line) depends more strongly on the initial magnetic field, with local transfer to smaller scales reaching double the nonlocal transfer in the Ma1 simulation and being less than half in other cases. Local energy transfer upscale (blue line) is positive for some early times in the Ma1 and Ma3.2 simulations.

The Ma1 simulations also display two different oscillatory behaviors, with a low frequency oscillation in the local energy transfer and a high frequency oscillation clearly visible in the equivalent energy transfer but also present in local and nonlocal down scale transfer.

3.3.2.2 Inverted Turbulent Cascades

At early times during the evolution of the Ma1 simulations, a temporary inverse cascade forms within the kinetic and magnetic energy reservoirs where small scale energy transfers to larger spatial scales. Figure 3.10 shows the local and nonlocal energy transfers within the kinetic and magnetic energies to both smaller and larger length scales. In the Ma1 simulations, the local energy transfer from larger to smaller length scales temporarily reverses into an inverse cascade in both the kinetic and magnetic energy reservoirs shortly after peak magnetic energy is reached. The inversion appears with all three sonic Mach numbers simulated, with the longest inversion appearing in the Ms0.1_Ma1 simulation for $\simeq 1T$ and shortest in the high Ms0.4_Ma1 simulation for $\simeq 0.5T$. For the Ms0.1_Ma1 simulation, the kinetic energy reservoir briefly reverses to the normal configuration, moving energy from large scales to scales while the magnetic energy is in an inverted cascade, before returning to the inverted cascade, lingering longer than the magnetic field in the inverted state and finally transitioning into a turbulent cascade for the rest of the simulation. As seen in Figure 3.7, the movement of energy to larger scales is not limited to any region of the spectra – it is present at all length scales. The Ma1 simulations, which are the only simulations to exhibit an inverse cascade, are also the only ones in which the total kinetic energy increases during any period. After peak magnetic energy in the `Ma1`, the magnetic energy increases while the kinetic energy increases for $\simeq 1T$; the inverse cascade appears during this same period.

3.3.2.3 Cross-Scale Flux

With additional analysis of the shell-to-shell transfer, we can extract more insight into the movement of energy. We can measure the cross-scale flux of energy from scales below a wavenumber k to scales above a wave number k by integrating the transfer function

$$\Pi^{X<}_{Y>}(k) = \sum_{Q \leq k} \sum_{K \geq k} \mathcal{T}_{XY}(Q, K) \tag{3.9}$$

Figure 3.11 shows the cross-scale fluxes via different transfer mechanisms for the simulations with `Ms0.2`. The top row shows cross-scale fluxes early in the simulation at $t = 0.77T$, when the large scale flow is still decaying into smaller scales. The magnetic cross-scale flux at low wavenumbers predictably depends on the initial magnetic energy, while the kinetic energy cross-scale flux is largely the same between simulations at a given sonic Mach number. For example, for `Ma10` the cross-scale flux is strongly dominated by $\Pi^{U<}_{U>}$, whereas for `Ma3.2` it is still the most significant contribution to the cross-scale flux, but substantial contributions are also seen from $\Pi^{U<}_{B>}$ ($\simeq 60\%$ of $\Pi^{U<}_{U>}(4)$), $\Pi^{B<}_{B>}$ ($\simeq 30\%$), and $\Pi^{B<}_{U>}$ ($\simeq 20\%$). For the strongest initial magnetization (`Ma1`) the early cross-scale flux is dominated by magnetic tension-mediated transfers from the kinetic-to-magnetic budget ($\Pi^{U<}_{B>}$) on all scales having a non-zero cross-scale flux ($k \lesssim 64$), with a similar contribution by the magnetic cascade on intermediate scales ($9 \lesssim k \lesssim 64$). The kinetic cascade is suppressed on all scales, generally contributing less than 10% to the total cross-scale flux.

At later times ($t = 5.16T$, bottom row of Fig. 3.11), magnetic energy dominates both the energy budget and cross-scale energy flux. Cross-scale energy flux via kinetic interactions is near zero across the inertial range of the spectrum, and thus does not significantly contribute to the total cross-scale energy flux. Only the magnetic fields facilitate down scale cross-scale flux at intermediate scales, both within the magnetic energy and from kinetic to magnetic energy. Moreover, the relative contributions of the individual transfer $\Pi^{U<}_{B>}$, $\Pi^{B<}_{B>}$, $\Pi^{B<}_{U>}$, and $\Pi^{U<}_{U>}$ (in order of decreasing contribution) on intermediate scales ($16 \lesssim k \lesssim 64$) is the same independent of initial magnetization. This continuous cross-scale flux is consistent with the evolving spectral index discussed in Section 3.3.1.3. Cross-scale flux through large physical scales is irregular, variable, and sometimes negative due to the lack of structure and driving forces at large scales.

3.4 Discussion

3.4.1 Comparison to driven turbulence simulations

The Taylor-Green vortex provides an interesting study of a freely evolving transition to decaying turbulence. In other words, no external force is applied to the simulation as is the case in driven turbulence simulations. This external force may introduce unintended dynamics to the flow (Grete et al., 2018). For example, in a simulation that is mechanically driven at large scales, energy may still be injected on intermediate scales both in the incompressible regime (Domaradzki et al., 2010) as well as in the compressible regime due to density coupling (Grete et al., 2017). Moreover, mechanical driving generally results in an excess of energy on the excited, kinetic scales that presents a barrier for magnetic field amplification on those scales in cases without a dynamically relevant mean magnetic field. This barrier is often expressed in the lack of a clear power law regime in the magnetic spectrum and resembles an inverse parabolic shape. At the same time, the magnetic energy spectrum drops below the kinetic one on the driving scales (see, e.g., Figure 1 in (Grete et al., 2021c) and references therein). In the simulations presented here no such barrier is observed. Both kinetic and magnetic energy spectra exhibit a (limited) regime where power law scaling is observed once a state of developed turbulence is reached.

Another important question raised from driven turbulence simulations pertains the locality of energy transfers. While there is agreement that $\mathcal{T}_{UU}$ and $\mathcal{T}_{BB}$ mediated transfers, i.e., within a budget, are highly local, the energy transfers between budgets (here, $\mathcal{T}_{UBT}$) have been observed to be weakly local and/or contain a nonlocal component from the driven scales (Alexakis et al., 2005; Yang et al., 2016; Grete et al., 2017). Here, we show that in the absence of the driving force the energy transfer mediated by magnetic tension contains both a local component as well as nonlocal component. The latter directly transfers large-scale kinetic energy to large and intermediate scales in the magnetic energy budget. Thus, the nonlocal component is not an artifact of an external driving force.

Finally, we recently showed that the kinetic energy spectra in driven turbulence simulations follow a scaling close to $k^{-4/3}$, i.e., shallower than Kolmogorov scaling, and explained this by the suppression of the kinetic energy cascade due to magnetic tension (Grete et al., 2021c). This is in agreement with our findings in the work presented here, where the same dynamics are observed at late times when turbulence is fully developed.

Naturally, this does not demonstrate that the same physical mechanisms are causing the similar slopes. Nevertheless, the late time evolution of the simulations presented here is still comparable to a limited degree to driven simulation of stationary turbulence. For example, even at late times (see, e.g., $t = 5.16T$ in Fig. 3.6), energy is still cascading down from the largest scales ($k \lesssim 8$) but the cascade is weaker than its initial magnitude. The reduction in strength of the cascade on large scale is directly linked to the decay of the large initial vortices. Nevertheless, even at late times the overall energy balance is still dominated by the largest scales, cf., the spectra shown in Fig. 3.3 when taking into account the $k^{4/3}$ compensation used in the plot. Overall, while here the inertial range shrinks and becomes weaker (to a limited degree) over time as the large scale modes lose energy, the dynamics within the inertial range is similar to driven turbulence simulations.

3.4.2 Comparison to previous results

In general, our results in the weakly compressible MHD regime are in agreement with the $\alpha \simeq -2$ spectrum reported by previous works on the TG vortex in Pouquet et al. (2010); Lee et al. (2010); Dallas & Alexakis (2013a,b) in the imcompressible MHD regime using the insulating magnetic field configuration. We see the same $\alpha \simeq -2$ spectrum early in the evolution before $t = 2T$, which corresponds to the time period near maximum energy dissipation that these other studies focused on. In all cases that we simulated the spectra became shallower at later times, independent of the initial magnetization (whereas these other works focused on $E_U/E_B = 1$, i.e., $\mathcal{M}_{A,0} = 1$, configurations, which are in good agreement with the `Ms0.2_Ma1.0` simulation presented here, see top left panel of Fig. 3.2). As noted by Dallas & Alexakis (2013a), the $\alpha \simeq -2$ spectrum is likely due to discontinuities in a small volume of the flow that can be disrupted by symmetry breaking at either large or small scales Dallas & Alexakis (2013c). According to Dallas & Alexakis (2013c), a simulated Taylor-Green vortex with sufficiently high Reynolds number should show symmetry breaking at the small scales at late times in the evolution, causing a break from the -2 power law at large wavenumbers. Since our simulations do not impose symmetries on the flow, this is a possible explanation for the observed behavior. However, we see an $\alpha \simeq -4/3$ inertial range scaling at late times, instead of the $\alpha \simeq -2$ and $\alpha \simeq -5/3$ broken power law theorized by Dallas & Alexakis (2013c).

Finally, work done in Lee et al. (2010); Brachet et al. (2013); Dallas & Alexakis (2013b) shows that the behavior of the magnetic field and spectra changes with the initial magnetic field configurations. With the insulating initial magnetic fields that we use, the vorticity begins parallel to the magnetic field. This facilitates the early energy flux from kinetic to magnetic energy. The insulating case tends towards stronger large magnetic fields compared to the other magnetic field configurations. Both of the other initial magnetic fields result in different energy spectra, with the conducting magnetic field setup leading to a $k^{-3/2}$ spectra and the alternative insulating field setup leading to spectra interpreted as either a $k^{-5/3}$ or k^{-2} spectra as argued by Lee et al. (2010) and Dallas & Alexakis (2013b) respectively.

3.4.3 Implication of results

In all of our simulations, we see magnetic fields and effects facilitated by the magnetic fields dominating the evolution of the decaying turbulence, even when the initial kinetic energy exceeds the magnetic energy by a factor of 100 in the `Ma10` simulations. Energy transfer from kinetic to magnetic energy via tension and energy transfer within the magnetic energy far exceed energy flux via the kinetic turbulent cascade at later times. Energy transfer from kinetic to magnetic energy at earlier times leads to the magnetic energy dominating over kinetic energy in all cases in both total magnitude as well as in terms of the scale-wise budget, cf., magnetic versus kinetic energy spectra. This is similar to what has been found in incompressible (Alexakis et al., 2005) and compressible simulations (Grete et al., 2017, 2021c) of driven turbulence. Thus, even in intermittently-driven systems one can expect the magnetic field to significantly influence the dynamics after a few dynamical times.

Our simulations exhibit a magnetic energy spectra with a measurable power law after the turbulent flow is realized. The inertial range is short, from approximately $k = 10$ to $k = 32$, due to the resolution of these simulations. Nevertheless, within this region we can reasonably fit a power law to both the kinetic and magnetic spectra, which is often not possible in driven turbulence simulation without a dynamically relevant mean magnetic field, cf., Sec. 3.4.1. Thus, freely evolving and driven turbulence simulations complement each other and both are required to disentangle environmental from intrinsic effects.

From an observational point of view, we demonstrated that the spectral indices evolve over time and fluctuate even for similar parameters. Therefore, the derived spectral indices from observation (e.g., velocity maps in astrophysics), which represent individual snapshots in time, need to be interpreted with care when trying to infer the "nature" of turbulence (e.g., Kolmogorov or Burgers) in the object of interest.

Finally, the observed nonlocal energy transfer has implications on the dynamical development of small scale structures from intermittent or singular energy injection events. Within the context of natural astrophysical and terrestrial plasmas, the nonlocal energy transfer from kinetic to magnetic energies suggests that small magnetic field structures develop before small scale kinetic structures.

3.4.4 Limitations

While our analysis showed that the results are generally robust (e.g., with respect to varying the fitting range in the spectral indices or varying range in the definition of scale-local in the energy transfers), higher resolution simulations are desirable. With higher resolution in an implicit large eddy simulation (ILES) the dynamic range is increased and, thus, the effective Reynolds numbers of the simulated plasma are raised.

Similarly, due to the nature of ILES the effective magnetic Prandtl number in all simulations is Pm $\simeq 1$. However, in natural systems (both astrophysical and terrestrial/experimental) Pm is either $\gg 1$ or $\ll 1$, motivating the exploration of these regimes in the future as well.

All of our simulations started with subsonic initial conditions, leaving the supersonic regime unexplored. The additional shocks, discontinuities, and strong density variations that may arise in a supersonic flow could alter the energy transfer as the flow transitions into turbulence. In the simulations we present here, the Mach number generally did not significantly affect the growth and behavior of the turbulence. In a supersonic flow, however, the transitory effects such as the nonlocal energy transfer and inverse cascade may be altered or suppressed in addition to generally richer dynamics related to compressive effects and effective space-filling of turbulent structures (Federrath, 2013).

Figure 5 indicates that the spectral index of both the kinetic and magnetic energy cascades evolves as a function of magnetic field strength (i.e., initial $\mathcal{M}_A$.) It is unclear whether there is a threshold of $\mathcal{M}_A$ above which the spectra become shallower, or whether there is a continuum of behavior as the initial $\mathcal{M}_A$ is increased. While we would like to engage in a more thorough exploration of the dependence of these behaviors on $\mathcal{M}_A$, the simulations in question are computationally expensive and it is infeasible to do so at present. Exploration of this transition is a promising venue for future work. Finally, the shell decomposition used here to study energy transfer has been shown to violate the inviscid criterion for decomposing scales in the compressible regime (Zhao & Aluie, 2018). However, this only pertains to flows with significant density variations and, thus, is effectively irrelevant for the subsonic simulations presented here.

3.5 Conclusions

We have presented in this work nine simulations of the Taylor-Green vortex using the insulating magnetic field setup from Lee et al. (2008) to study magnetized decaying turbulence in the compressible ideal MHD regime using the finite volume code K-Athena. As a first for the Taylor-Green vortex, we have also presented an energy transfer analysis to show the movement of energy between scales and energy reservoirs as facilitated via different mechanisms. Our key results are as follows:

- Magnetic fields significantly affect the evolution of the decaying turbulence, regardless of initial field strength. Energy flux from kinetic energy to magnetic energy leads to the magnetic energy dominating the energy budget, even in simulations where the magnetic energy is initially very small.
- The Taylor-Green vortex simulations explored here display a power law in both the kinetic and magnetic energy spectra with a measurable spectral index, which is in contrast with the lack of a power law in the magnetic energy spectrum seen in driven turbulence calculations without a significant mean field.
- Decaying turbulent flows do not exhibit a spectral index that is constant in time in either the kinetic nor magnetic energy reservoirs – these spectra continually evolve over time. The spectral indices of the kinetic and magnetic energies become comparable and roughly constant around 1 – 2 dynamical times after the magnetic energy has become dominant. This can happen as early as $t = 2T$ when the initial magnetic energy equals initial the kinetic energy, and as late as $t = 5T$ when initial kinetic energy exceeds the magnetic by a factor of 100. For simulations with more initial kinetic energy than magnetic energy, the spectral indices reach a rough constant slightly steeper than $\alpha \simeq -4/3$.
- Before the turbulent flow fully develops, an inverse cascade within the kinetic and magnetic energy reservoirs is intermittently observed. This intermittent behavior moves energy from smaller scales to larger scales, and is possible when the magnetic energy is comparable to the kinetic energy.
- Analysis of energy transfer within and between reservoirs indicates that within fully-developed turbulence, the cross-scale flux of energy in both the kinetic and magnetic cascades are dominated by energy transfer mediated by the magnetic field.

- Magnetic tension facilitates nonlocal transfer from larger scales in the kinetic energy to smaller scales in the magnetic energy, and is particularly prominent in simulations where the magnetic field is initially weak.

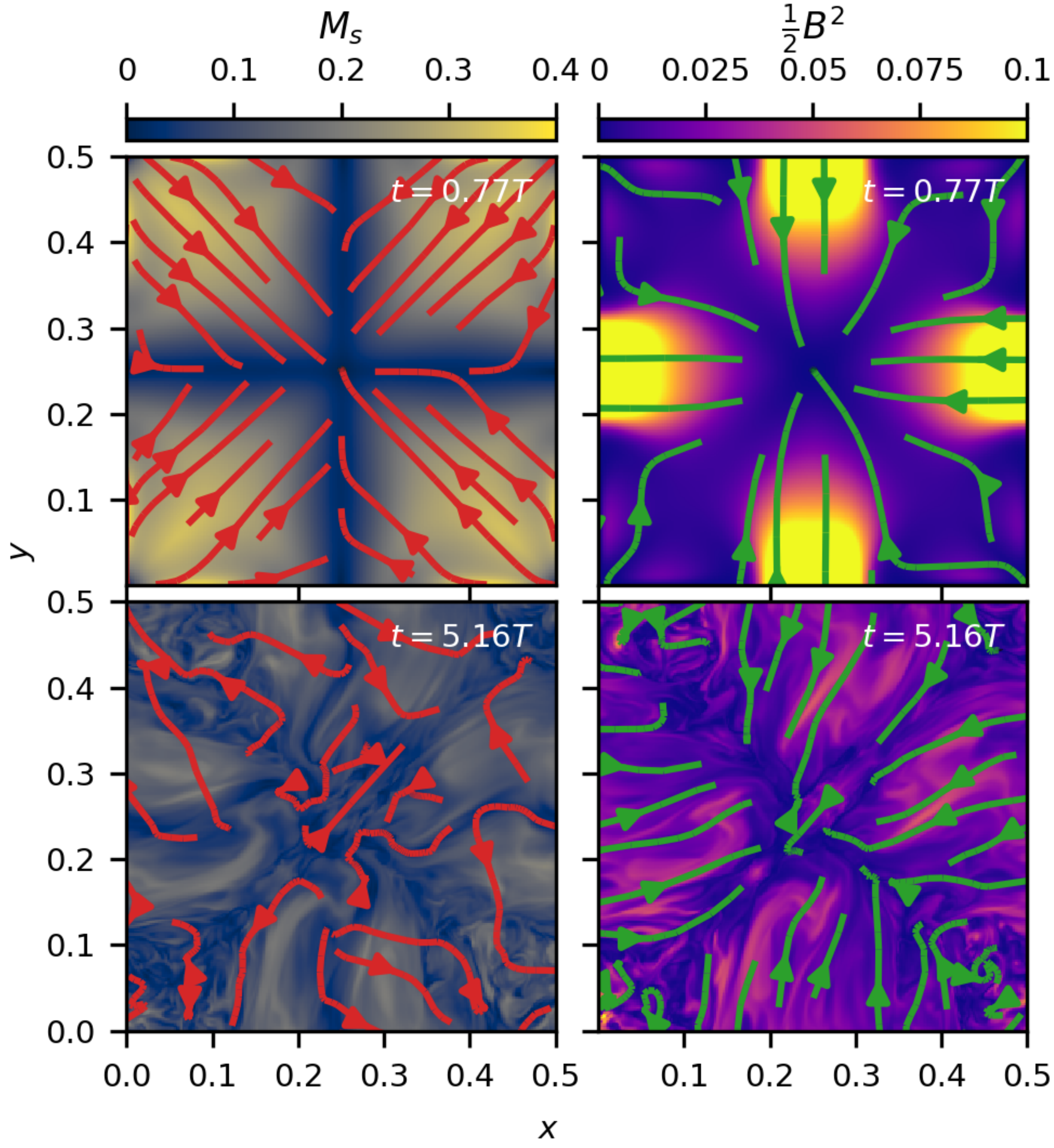


Figure 3.1: Slices of sonic Mach number (left) and magnetic pressure (right) at $t = 0.77T$ and $t = 5.16T$ in the $xy-$plane through $z = \frac{\pi}{2}L$, with streamlines on the left showing the direction of flow and streamlines on the right showing the direction of the magnetic fields, plotting only the 1st quadrant from the `Ms0.2_Ma10` simulation, demonstrating the transition of the flow into turbulence.

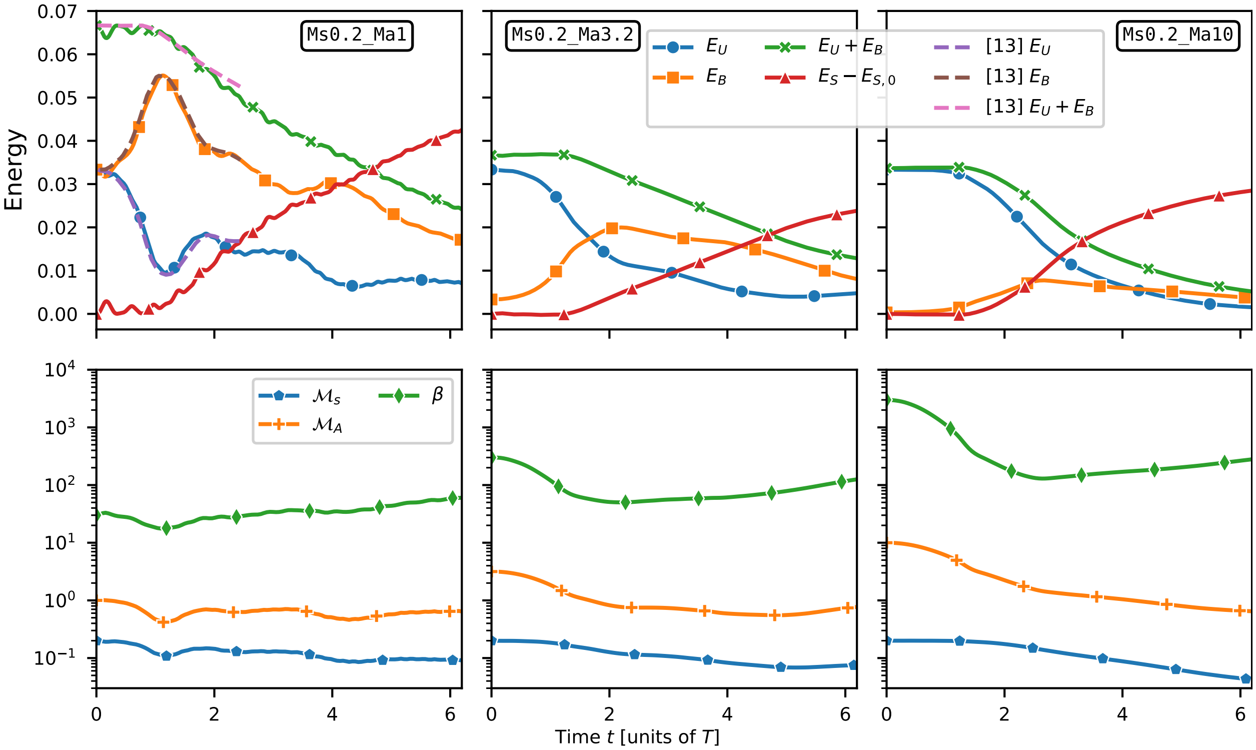


Figure 3.2: Mean energies over over time in the top row with kinetic energy (solid blue), magnetic energy (solid orange), the sum of kinetic and magnetic energies (solid green), and the change in thermal energy since the simulation start (solid red), and dimensionless numbers over time in the bottom row with RMS sonic Mach number $\mathcal{M}_s$ (blue), Alvénic Mach number $\mathcal{M}_A$ (orange), and plasma beta β (green) for the `Ms0.2` simulations. Energy over time from the simulation from Fig. 3a in Pouquet et al. (2010) (adjusted to the normalization used here), which matches the setup of the `Ms0.2_Ma1` simulation, is shown with dashed lines in the upper left panel for reference. Energies and mach numbers for all nine simulations are shown in the online supplements.

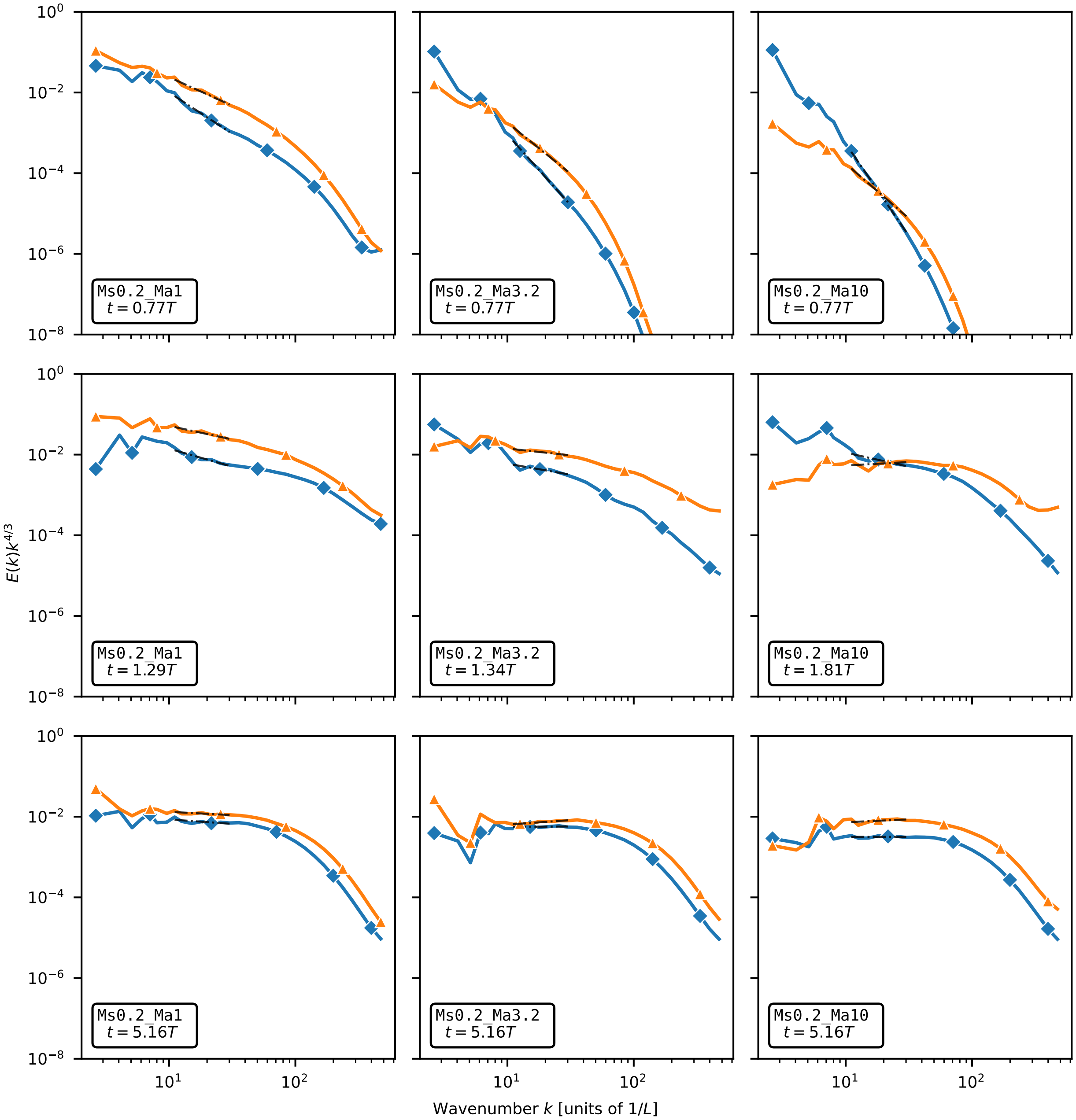


Figure 3.3: Kinetic energy spectra (in solid blue) and magnetic energy spectra (in solid orange) compensated by $k^{4/3}$, with black dashed lines showing the power law fit to the spectral to obtain a spectral index. In the left column we show the `Ms0.2_Ma1` simulation, in the middle column we show the `Ms0.2_Ma3.2` simulation, and in the right column we show the `Ms0.2_Ma10` simulation. In the top row we show all simulations at $t = 0.77T$, in the middle row we show the three simulations at different times ($t = 1.29$, $t = 1.81T$, $t = 1.81T$) when the simulations are displaying interesting behavior discussed in sections 3.3.2.2 and 3.3.2.1, and in the bottom row we show all simulations at $t = 5.16T$ when the initial flow has completely decayed into turbulence and both energy spectra fluctuate around a $k^{-4/3}$ spectrum.

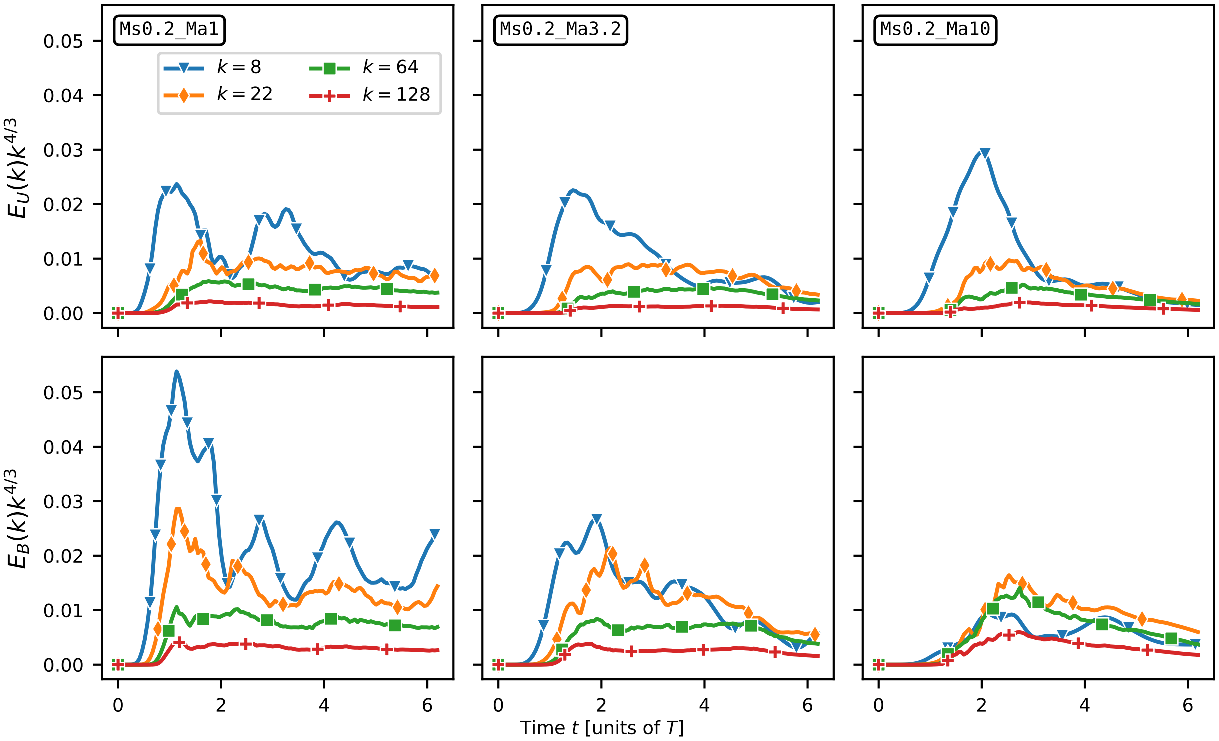


Figure 3.4: The kinetic energy (top) and magnetic energy (bottom) at wavenumbers $k = 8, 22, 64, 128$ plotted separately in different colors versus time, where the energy at each wavenumber has been compensated by $k^{4/3}$ to make them comparable. In the left column we show the `Ms0.2_Ma1` simulation, in the middle column we show the `Ms0.2_Ma3.2` simulation, and in the right column we show the `Ms0.2_Ma10` simulation. Energy at the smallest length scales in both reservoirs saturates at $t \simeq 1T$, $t \simeq 1.5T$, and $t \simeq 2.5$ in the `Ms0.2_Ma1`, `Ms0.2_Ma3.2`, and `Ms0.2_Ma10` simulations respectively, showing approximately when the turbulence has developed at all scales.

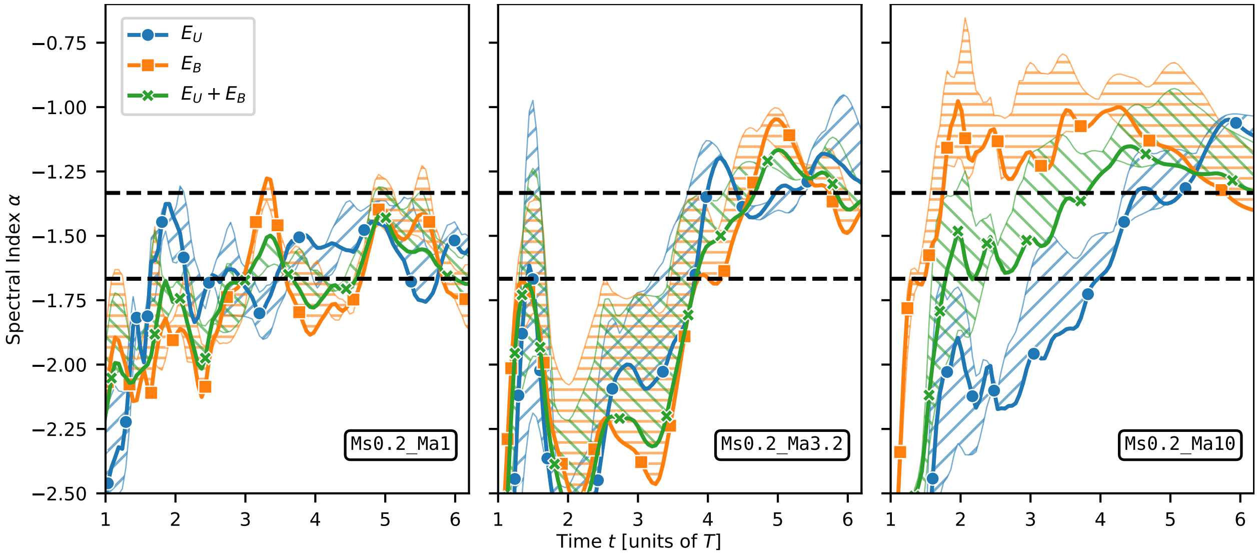


Figure 3.5: Evolution of the spectral indices of the kinetic (blue), magnetic (orange), and sum of kinetic and magnetic energy (green) spectra over time for the `Ms0.2` simulations. The slope is computed from a least squares fit of the energy spectra limited to wavenumbers $k \in [10, 32]$ which is approximately the inertial range. Shaded bands show how the fitted slope differs if a range $k \in [8, 34]$, $k \in [10, 32]$, or $k \in [12, 30]$ is used. Note that the spectral index using the range $k \in [10, 32]$ is not guaranteed to be bounded by the spectral indices obtained using $k \in [8, 34]$, $k \in [10, 32]$ and $k \in [12, 30]$, which is especially evident in the `Ms0.2_Ma3.2` and `Ms0.2_Ma10` simulations from $t \simeq 2T$ to $t \simeq 4T$. Horizontal dashed lines show $-4/3$ and $-5/3$ spectral indices. The slope is only shown after $t = 1T$ as the initial flow conditions dominate the spectra at early times, leading to steep spectra. We include the spectral indices versus time for all nine simulations in the online supplements.

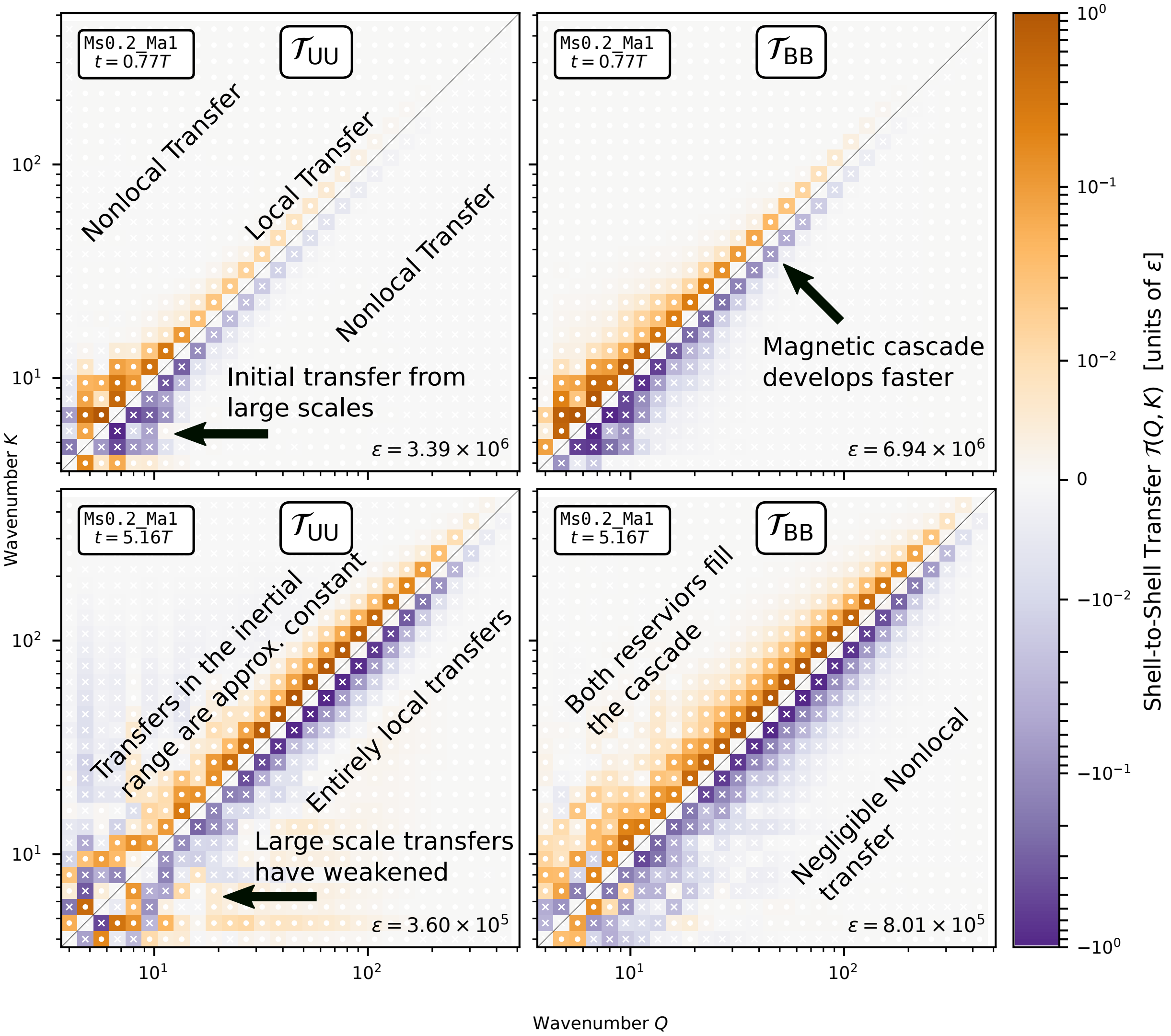


Figure 3.6: Shell-to-shell energy transfer plots for the energy transfer within the kinetic (left) and magnetic (right) energy reservoirs via advection and compression at $t = 0.77T$ (top) and $t = 5.16T$ (bottom) from the simulations with Ms0.2_Ma1, showing the development of the kinetic and magnetic turbulent cascades. Annotations on the figure highlight key features of the energy transfer that are characteristic of a developing turbulence cascade. Each bin shows the flux of energy from shell Q to shell K, where orange with white circles showing a positive flux of energy, so that K is gaining energy, and purple with white x's showing a negative flux, so that K is losing energy. The energy flux in each bin is normalized by $\varepsilon = \max_{Q,K} |\mathcal{T}_{XY}(Q, K)|$ so that a higher ε means a higher energy flux. The solid black line shows equivalent scale transfers. As the turbulent cascade develops in the magnetic and kinetic energy reservoirs, more energy transfers along the diagonal fill out the energy spectrum down to numerical dissipation scales.

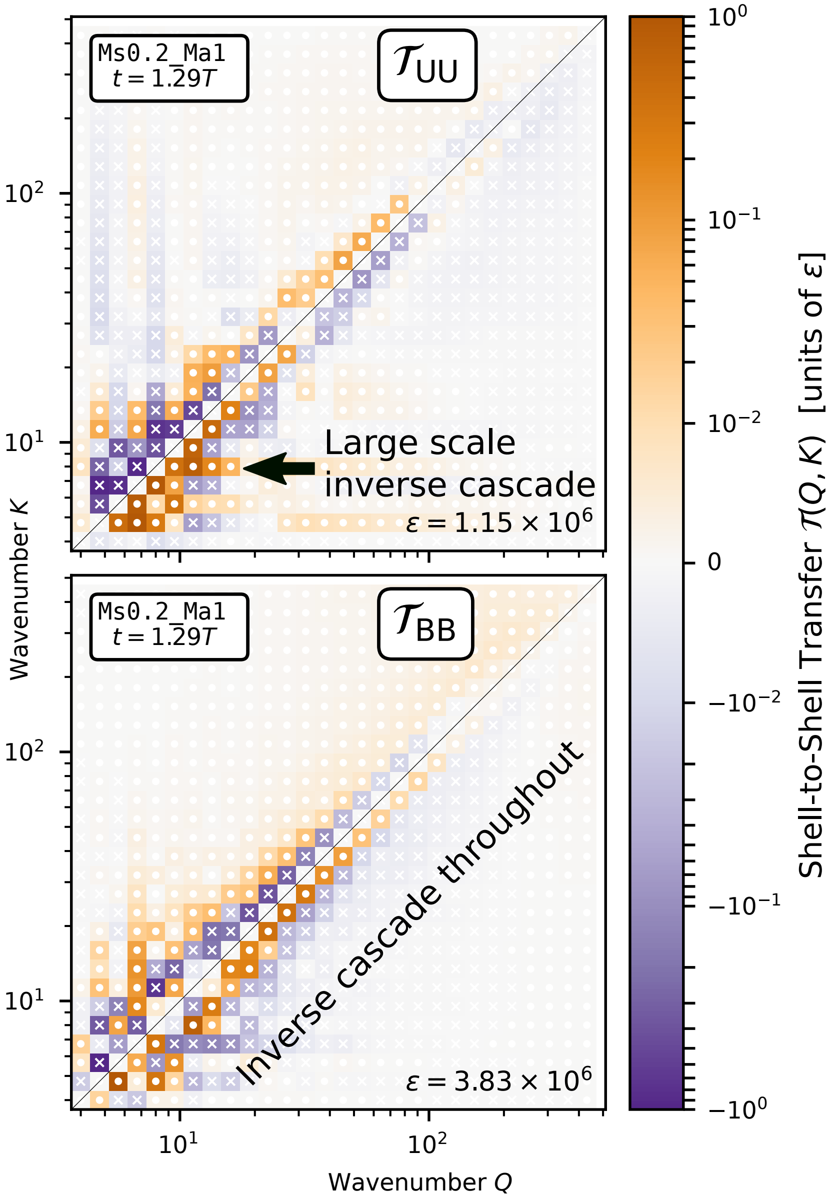


Figure 3.7: Shell-to-shell energy transfer plots for the energy transfer within the kinetic (top) and magnetic (bottom) energy reservoirs via advection and compression at $t = 1.29T$ from the Ms0.2_Ma1 simulation, showing a transient inverse cascade within the magnetic energy reservoir (on all scales $K, Q \lesssim 100$) and kinetic energy reservoir (on large scales $K, Q \lesssim 16$). Annotations show where along the diagonal the inverse cascade is present.

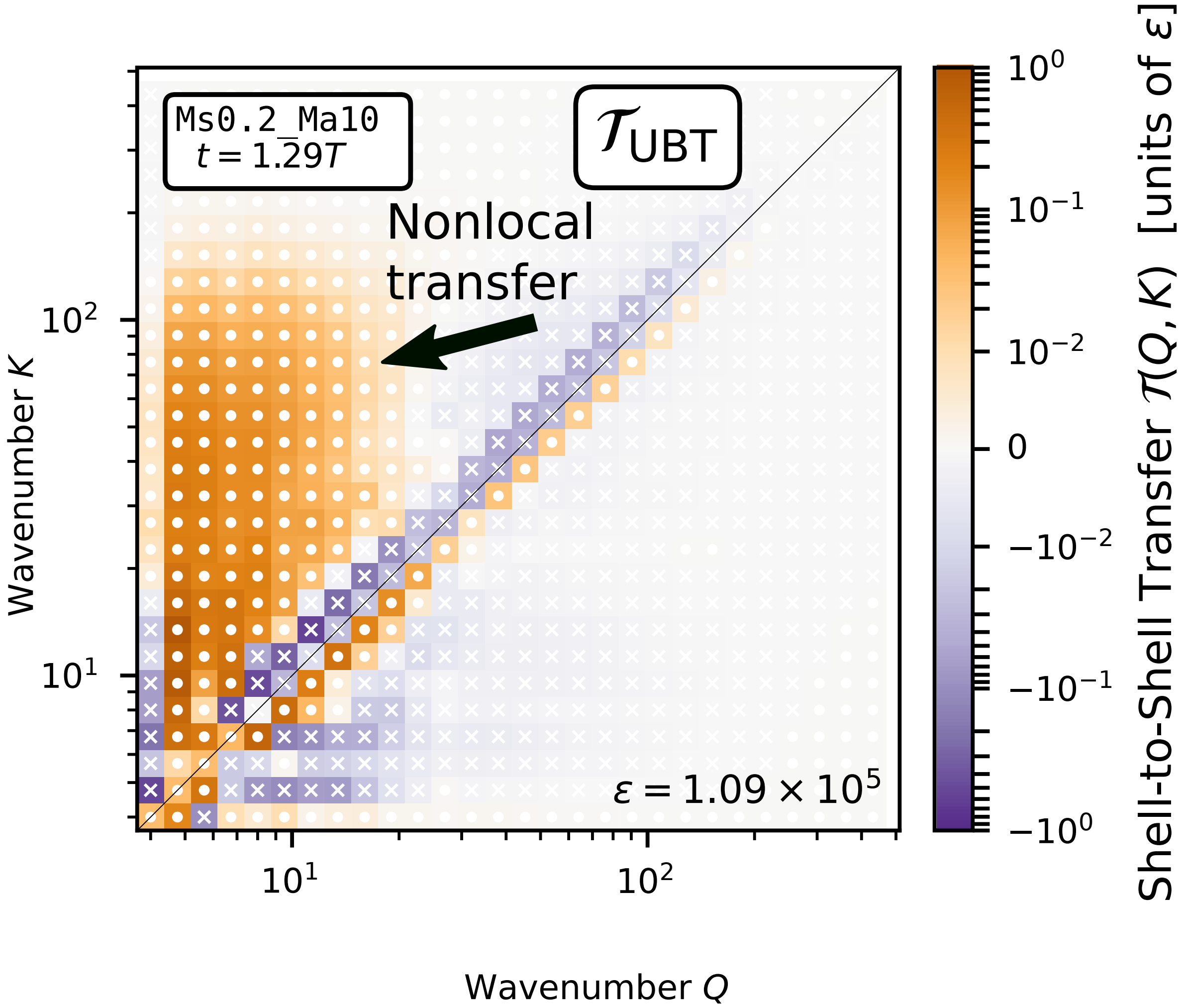


Figure 3.8: Shell-to-shell energy transfer plots for the energy transfer from kinetic to magnetic energy via magnetic tension at $t = 1.81T$ from the `Ms0.2_Ma10` simulation, showing the nonlocal energy transfer from large kinetic scales to many smaller magnetic scales. Annotations show where the nonlocal transfer is present.

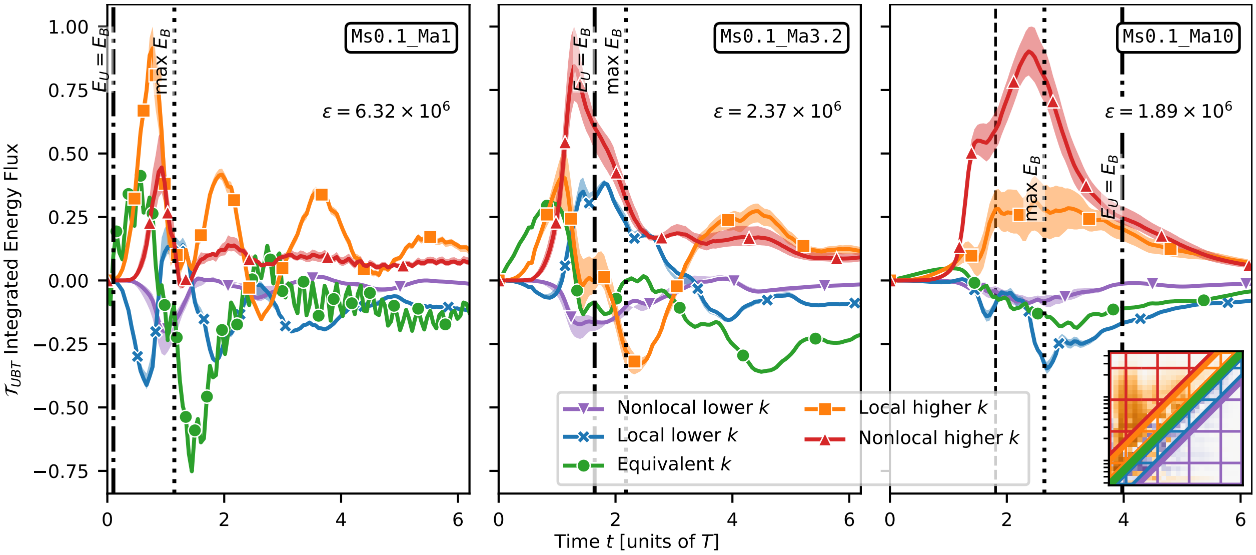


Figure 3.9: Integrated energy flux over time from kinetic to magnetic energy via tension from larger wavenumbers to smaller nonlocal wavenumbers (purple), from larger wavenumbers to smaller local wavenumbers (blue), between equivalent wavenumbers (green), from smaller wavenumbers to larger local wavenumbers (orange), and from smaller wavenumbers to larger nonlocal wavenumbers (red) in the `Ms0.2` simulations. We normalize the energy flux in each panel so that the absolute maximum of all of the flux bins is 1.0, where ε is the normalization factor use in each panel. Comparisons of the relative strength of energy fluxes in different simulations must consider ε. The inset plot in the lower right panel shows the color coded regions that are integrated to calculate each line at a single time for the same shell-to-shell transfer from Figure 3.8. Solid lines show the integrated flux if “local" wavenumbers as defined as 5 logarithmic bins away from the equivalent wavenumber. The shaded regions show the integrated flux if 4 or 6 bins are used, showing that the behavior is robust if the range “local" wavenumbers is defined closer or further away from transfer between equivalent scales. We include the integrated flux from kinetic to magnetic energy via tension for all nine simulations in the online supplements

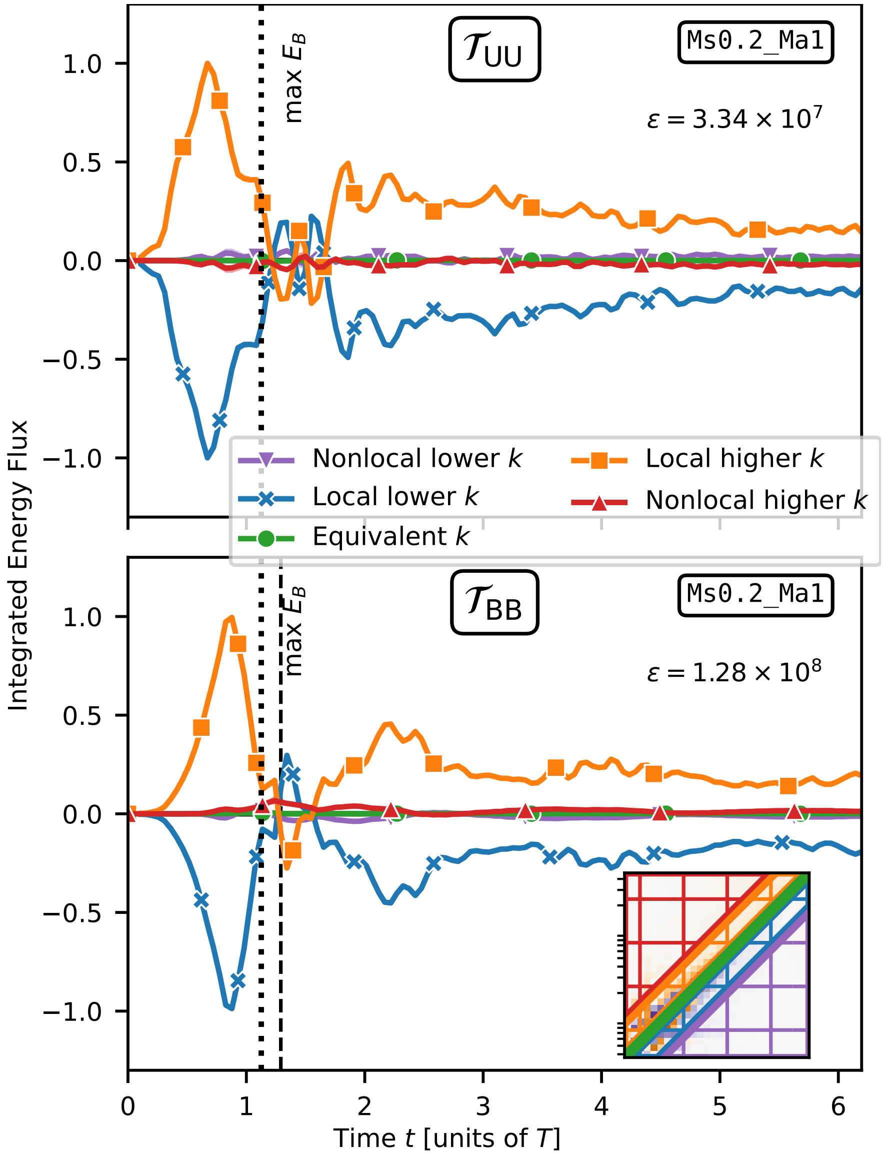


Figure 3.10: Integrated energy flux over time within the kinetic energy (top) and within the magnetic energy (bottom) from larger wavenumbers to smaller nonlocal wavenumbers (purple), from larger wavenumbers to smaller local wavenumbers (blue), between equivalent wavenumbers (green), from smaller wavenumbers to larger local wavenumbers (orange), and from smaller wavenumbers to larger nonlocal wavenumbers (red) in the Ms0.2_Ma1 simulation. The inset plot in the lower middle panel demonstrates the color coded regions that are integrated to calculate each line at $t = 1.29T$ from the shell-to-shell transfer from Figure 3.7. Solid lines show the integrated flux if "local" wavenumbers as defined as 5 logarithmic bins away from the equivalent wavenumber. The results change very little if 4 or 6 bins are used. We include the integrated flux within the kinetic energy and magnetic energy for all nine simulations in the online supplements.

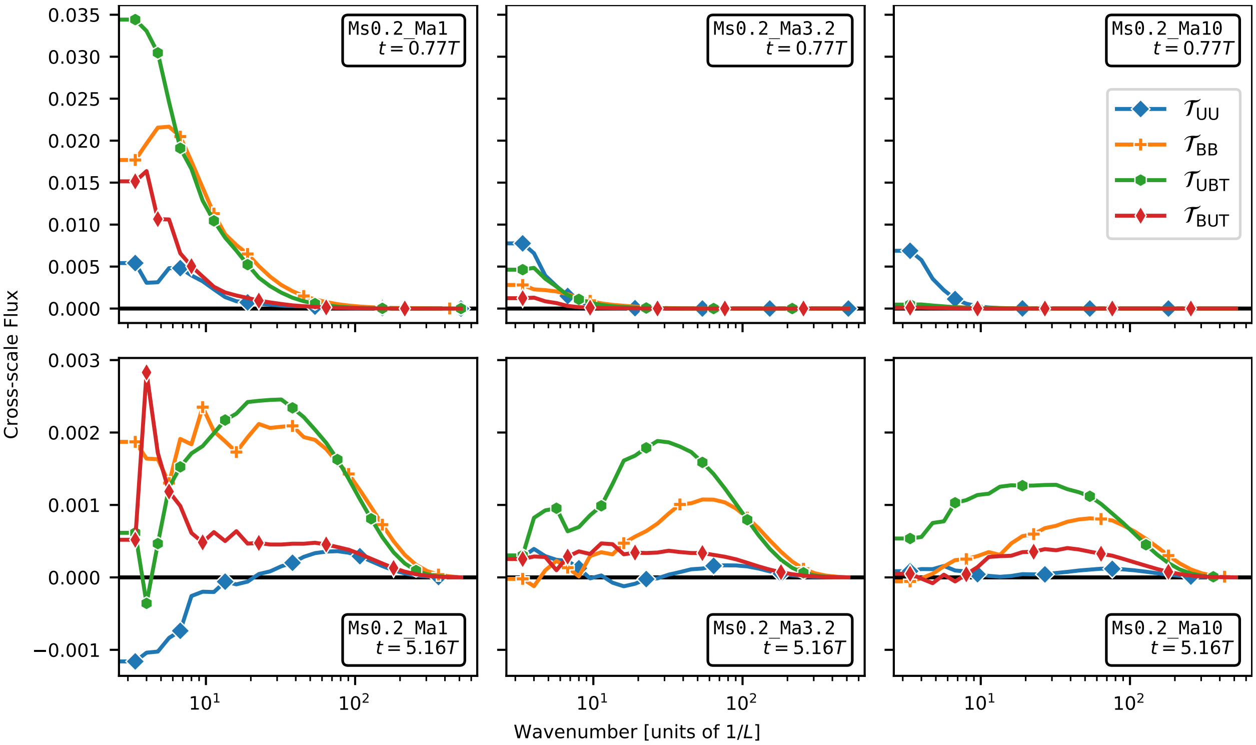


Figure 3.11: Cross-scale flux within the kinetic energy (blue line), within the magnetic energy (orange line), and from kinetic to magnetic energy via tension (green line) in the three `Ms0.2` simulations across columns and at dynamical time $t = 0.77T$ (top) and later at dynamical time $t = 5.16T$. Note that the cross-scale fluxes at later times are an order of magnitude less than early cross-scale fluxes. Positive values of this quantity denote energy transfer from larger to smaller scales.

CHAPTER 4

K-ATHENA: A PERFORMANCE PORTABLE STRUCTURED GRID FINITE VOLUME MAGNETOHYDRODYNAMICS CODE

This chapter first appeared as the published paper Grete et al. (2021a), on which I am equal co-first author. I include the original abstract as the introduction to this chapter.

Chapter Abstract

Large scale simulations are a key pillar of modern research and require ever-increasing computational resources. Different novel manycore architectures have emerged in recent years on the way towards the exascale era. Performance portability is required to prevent repeated non-trivial refactoring of a code for different architectures. We combine Athena, an existing magnetohydrodynamics (MHD) CPU code, with Kokkos, a performance portable on-node parallel programming paradigm, into K-Athena to allow efficient simulations on multiple architectures using a single codebase. We present profiling and scaling results for different platforms including Intel Skylake CPUs, Intel Xeon Phis, and NVIDIA GPUs. K-Athena achieves $> 10^8$ cell-updates/s on a single V100 GPU for second-order double precision MHD calculations, and a speedup of 30 on up to 24,576 GPUs on Summit (compared to 172,032 CPU cores), reaching 1.94×10^{12} total cell-updates/s at 76% parallel efficiency. Using a roofline analysis we demonstrate that the overall performance is currently limited by DRAM bandwidth and calculate a performance portability metric of 62.8%. Finally, we present the implementation strategies used and the challenges encountered in maximizing performance. This will provide other research groups with a straightforward approach to prepare their own codes for the exascale era. K-Athena is available at `https://gitlab.com/pgrete/kathena`.

4.1 Introduction

The era of exascale computing is approaching. Different projects around the globe are working on the first exascale supercomputers, i.e., supercomputers capable of conducting 10^{18} floating point operations per second. This includes, for example, the Exascale Computing Initiative working with Intel and Cray on Aurora as the first exascale computer in the US in 2021, the EuroHPC collaboration working on building two exascale systems in Europe by 2022/2023, Fujitsu and RIKEN in Japan working on the Post-K machine to launch in 2021/2022, and China who target 2020 for their first exascale machine. While the exact architectural details of these machines are not announced yet and/or are still under active development, the overall trend in recent years has been manycore architectures. Here, manycore refers to an increasing number of (potentially simpler) cores on a single compute node and includes CPUs (e.g., Intel's Xeon Scalable Processor family or AMD's Epyc family), accelerators (e.g., the now discontinued Intel Xeon Phi line), and GPUs for general purpose computing. MPI+OpenMP has been the prevailing parallel programming paradigm in many areas of high performance computing for roughly two decades. It is questionable, however, whether this generic approach will be capable of making efficient use of available hardware features such as parallel threads and vectorization across different manycore architectures and between nodes.

In addition to extensions of the MPI standard such as shared-memory parallelism, several approaches in addition to MPI+OpenMP exist and are being actively developed to address either on-node, inter-node, or both types of parallelism. These include, for example, partitioned global address space (PGAS) programming models such as UPC++ Zheng et al. (2014), or parallel programming frameworks such as Charm++ or Legion, which are based on message-driven migratable objects Kale & Krishnan (1993); Bauer et al. (2012).

Our main goal is a performance portable version of the existing MPI+OpenMP finite volume (general relativity) magnetohydrodynamics (MHD) code Athena++ White et al. (2016b); Stone et al. (2020b). This goal includes enabling GPU-accelerated simulations while maintaining CPU performance using a single code base. More generally, performance portability refers to achieving consistent levels of performance across heterogeneous platforms using as little architecture-dependent code as possible. Given the uncertainties in future architecures (and the broad availability of different architecture already today) performance portability is an active field of research in many areas Straatsma et al. (2017); Bennett et al. (2015). This includes (but is not limited to) idealized benchmarks and miniapps Heroux et al. (2009); Martineau et al. (2017); Deakin et al. (2018); Hammond & Mattson (2019), algorithm libraries Heroux & Willenbring (2012), structured mesh codes Holmen et al. (2019), or particle in cell codes Artigues et al. (2019).

In order to keep the code changes minimal, and given the MPI+OpenMP basis of Athena++, we decided to keep MPI for inter-node parallelism and focus on on-node performance portability. For on-node performance portability several libraries and programming language extensions exist. With version 4.5 OpenMP Dagum & Menon (1998) has been extended to support offloading to devices such as GPUs, but support and maturity is still highly compiler and architecture dependent. This similarly applies to OpenACC, which has been designed from the beginning to target heterogeneous platforms. While these two directives-based programming models are generally less intrusive with respect to the code base, they only expose a limited fraction of various platform-specific features. OpenCL Stone et al. (2010) is much more flexible and allows fine grained control over hardware features (e.g., threads), but this, on the other hand, adds substantial complexity to the code. Kokkos Edwards et al. (2014) and RAJA Hornung et al. (2015) try to combine the strength of flexibility with ease of use by providing abstractions in the form of C++ templates. Both Kokkos and RAJA focus on abstractions of parallel regions in the code, and Kokkos additionally provides abstractions of the memory hierarchy. At compile time the templates are translated to different (native) backends, e.g., OpenMP on CPUs or CUDA on NVIDIA GPUs. A more detailed description of these different approaches including benchmarking in more idealized setups can be found in, e.g., Martineau et al. (2017); Deakin et al. (2018).

We chose Kokkos for the refactoring of Athena++ for several reasons. Kokkos offers the highest level of abstraction without forcing the developer to use it by setting reasonable implicit platform defaults. Moreover, the Kokkos core developer team actively works on integrating the programming model into the C++ standard. New, upcoming features, e.g., in OpenMP, will replace manual implementations in the Kokkos OpenMP backend over time. Kokkos is already used in several large projects to achieve performance portability, e.g., the scientific software building block collection Trilinos Heroux et al. (2005) or the computational framework for simulating chemical and physical reactions Uintah Holmen et al. (2017). In addition, Kokkos is part of the DOE's Exascale Computing Project and we thus expect a backend for Aurora's new Intel Xe architecture when the system launches. Finally, the Kokkos community, including core developers and users, is very active and supportive with respect to handling issues, questions and offering workshops.

The resulting K-Athena code successfully achieves performance portability across CPUs (Intel, AMD, and IBM), Intel Xeon Phis, and NVIDIA GPUs. We demonstrate weak scaling at 76% parallel efficiency on 24,576 GPUs on OLCF's Summit, reaching 1.94×10^{12} total cell-updates/s for a double precision MHD calculation. Moreover, we calculate a performance portability metric of 62.8% across Xeon Phis, 6 CPU generations, and 3 GPU generations. We make the code available as an open source project[1].

The paper is organized as follows. In Section 4.2 we introduce Kokkos, Athena++, and the changes made and approach chosen in creating K-Athena. In Section 4.3 we present profiling, scaling and roofline analysis results. Finally, we discuss current limitations and future enhancements in Sec. 4.4 and make concluding remarks in Sec. 4.5.

4.2 Method

4.2.1 Kokkos

Kokkos is an open source[2] C++ performance portability programming model Edwards et al. (2014). It is implemented as a template library and offers abstractions for parallel execution of code and data management. The core of the programming model consists of six abstractions.

[1] K-Athena's project repository is located at https://gitlab.com/pgrete/kathena.

[2] See https://github.com/kokkos for the library itself, associated tools, tutorial and a wiki.

First, *execution spaces* define where code is executed. This includes, for example, OpenMP on CPUs or Intel Xeon Phis, CUDA on NVIDIA GPUs, or `ROCm` on AMD GPUs (which is currently experimental). Second, *execution patterns* are parallel patterns, e.g. `parallel_for` or `parallel_reduce`, are the building blocks of any application that uses Kokkos. These parallel regions are often also referred to as kernels as they can be dispatched for execution on execution spaces (such as GPUs). Third, *execution policies* determine how an execution pattern is executed. There exist simple range policies that only specify the indices of the parallel pattern and the order of iteration (i.e., the fastest changing index for multidimensional arrays). More complicated policies, such as team policies, can be used for more fine-grained control over individual threads and nested parallelism. Fourth, *memory spaces* specify where data is located, e.g., in host/system memory or in device space such as GPU memory. Fifth, the *memory layout* determines the logical mapping of multidimensional indices to actual memory location, cf., `C` family row-major order versus `Fortran` column-major order. Sixth, *memory traits* can be assigned to data and specify how data is accessed, e.g., atomic access, random access, or streaming access.

These six abstractions offer substantial flexibility in fine-tuning application, but the application developer is not always required to specify all details. In general, architecture-dependent defaults are set at compile time based on the information on devices and architecture provided. For example, if CUDA is defined as the default execution space at compile time, all `Kokkos::Views`, which are the fundamental multidimensional array structure, will be allocated in GPU memory. Moreover, the memory layout is set to column-major so that consecutive threads in the same warp access consecutive entries in memory.

4.2.2 Athena++

Athena++ is a radiation general relativistic magnetohydrodynamics (GRMHD) code focusing on astrophysical applications White et al. (2016b); Stone et al. (2020b). It is a rewrite in modern C++ of the widely used Athena `C` version Stone et al. (2008b). Athena++ offers a wide variety of compressible hydro- and magnetohydrodynamics solvers including support for special and relativistic (M)HD, flexible geometries (Cartesian, cylindrical, or spherical), and mixed parallelization with OpenMP and MPI. Apart from the overall feature set, the main reasons we chose Athena++ are a) its excellent performance on CPUs and KNLs due to a focus on vectorization in the code design, b) a generally well written and documented code base in modern C++, c) point releases are publicly available that contain many (but not all) features[3], and d) a flexible task-based execution model that allows for a high degree of modularity.

Athena++'s parallelization strategy evolves around so-called `meshblocks`. The entire simulation grid is divided into smaller `meshblocks` that are distributed among MPI processes and/or OpenMP threads. Each MPI processes (or OpenMP thread) owns one or more `meshblocks` that can be updated independently after boundary information have been communicated. If hybrid parallelization is used, each MPI process runs one or more OpenMP threads that each are assigned one or more `meshblock`. This design choice is often referred to as coarse-grained parallelization as threads are used at a block (here `meshblock`) level and not over loop indices. In general, Athena++ uses persistent MPI communication handles in combination with one-sided MPI calls to realize asynchronous communication. Moreover, each thread makes its own MPI calls to exchange boundary information. As a result, using more than one thread per MPI process may increase overall on-node performance due to hyperthreading but also increases both the number of MPI messages sent and the total amount of data sent. The latter may result in overall worse parallel performance and efficiency, as demonstrated in Sec 4.3.3.2.

Given the coarse-grained OpenMP approach over `meshblocks` the prevalent structures in the code base are triple (or quadruple) nested `for` loops that iterate over the content of each `meshblock` (and variables in the quadruple case). A prototypical nested loop is illustrated in Listing 4.1. Generally, all loops (or kernels) in Athena++ have been written so that OpenMP `simd pragmas` are used for the innermost loop. This helps the compiler in trying to automatically vectorize the loops resulting in a more performant application.

[3] Our code changes are based on the public version, Athena++ 1.1.1, see `https://github.com/PrincetonUniversity/athena-public-version`

Listing 4.1: Example triple for loop for a typical operation in a finite volume method on a structured mesh such as in a code like Athena++, where ks, ke, js, je, is, and ie are loop bounds and u is an athena_array object of, for example, an MHD variable.

```
for( int k = ks; k < ke; k++){
  for( int j = js; j < je; j++){
    #pragma omp simd
    for( int i = is; i < ie; i++){
      /* Loop Body */
      u(k,j,i) = ...
}}}
```

4.2.3 K-Athena = Kokkos + Athena++

In order to combine Athena++ and Kokkos, four major changes in the code base were required: 1) making Kokkos::Views the fundamental data structure, 2) converting nested for loop structures to kernels, 3) converting “support” functions, such as the equation of state, to inline functions, and 4) converting communication buffer filling functions into kernels.

First, Views are the Kokkos’ abstraction of multidimensional arrays. Thus, the multidimensional arrays originally used in Athena++, e.g., the MHD variables for each meshblock, need to be converted to Views so that these arrays can transparently be allocated in arbitrary memory spaces such as device (e.g., GPU) memory or system memory. Athena++ already implemented an abstract athena_array class for all multidimensional arrays with an interface similar to the interface of a View. Therefore, we only had to add View objects as member variables and to modify the functions of athena_arrays to transparently use functions of those member Views. This included using View constructors to allocate memory, using Kokkos::deep_copy or Kokkos::subview for copy constructors and shallow slices, and creating public member functions to access the Views. The latter is required in order to properly access the data from within compute kernels.

Second, all nested for loop structures (see Listing 4.1 need to be converted to so-called kernels, i.e., parallel region that can be dispatched for execution by an execution space. As described in Sec. 4.2.1 multiple execution policies are possible, such as a multidimensional range policy (see

Listing 4.2: Example for loop using Kokkos. The loop body is reformulated into a lambda function and passed into Kokkos::parallel_for to execute on the target architecture. The class Kokkos::MDRangePolicy specifies the loop bounds. The array u is now a Kokkos::View, a Kokkos building block that allows transparent access to CPU and GPU memory. The loop body, i.e., the majority of the code, remains mostly unchanged.

```
parallel_for( MDRangePolicy<Rank<3>>
    ({ks,js,is},{ke,je,ie}),
  KOKKOS_LAMBDA(int k, int j, int i){
    /* Loop Body */
    u(k,j,i) = ...
});
```

Listing 4.3: Same as Listing 4.2 but using a one dimensional Kokkos::RangePolicy (implicit through default template parameter) with explicit index calculation.

```
int nk = ke-ks, nj = je-js, ni = ie-is;
parallel_for(nk*nj*ni,
  KOKKOS_LAMBDA(int idx){
    int k = idx / (nj*ni);
    int j = (idx - k*(nj*ni) / ni;
    int i = idx - k*(nj*ni) - j*ni;
    /* Loop Body */
    u(k,j,i) = ...
});
```

Listing 4.2), a one dimensional policy with manual index mapping (see Listing 4.3), or a team policy that allows for more fine-grained control and nested parallelism (see Listing 4.4).

Generally, the loop body remained mostly unchanged. Given that it is not a priori clear what kind of execution policy yields the best performance for a given implementation of an algorithm, we decided to implement a flexible loop macro[4]. That macro allows us to easily change the execution policy for performance tests – see profiling results in Sec. 4.3.3.1 and discussion in Sec. 4.4, and this intermediate abstraction is similar to the approach chosen in other projects Holmen et al. (2019).

Third, all functions that are called within a kernel need to be converted into inline functions (here, more specifically using the KOKKOS_INLINE_FUNCTION macro). This is required because if

[4] Note, that in newer versions of the code we replaced the macro with a template.

Listing 4.4: Another approach using Kokkos' nested team-based parallelism through the Kokkos::TeamThreadRange and Kokkos::ThreadVectorRange classes. This interface is closer to the underlying parallelism used by the backend such as CUDA blocks on GPUs and SIMD vectors on CPUs.

```
parallel_for(team_policy(nk, AUTO),
  KOKKOS_LAMBDA(member_type thread) {
    const int k = thread.league_rank() + ks;
    parallel_for(
      TeamThreadRange<>(thread,js,je,
        [&] (const int j) {
        parallel_for(
          ThreadVectorRange<>(thread,is,ie,
          [=] (const int i) {
            /* Loop Body */
            u(k,j,i) = ...
});});});
```

the kernels are executed on a device such as a GPU, the function need to be compiled for the device (e.g., with a __device__ attribute when compiling with CUDA). In Athena++, this primarily concerned functions such as the equation of state and coordinate system-related functions.

Fourth, Athena++ uses persistent communication buffers (and MPI handles) to exchange data between processes. Originally, these buffers resided in the system memory and were filled directly from arrays residing in the system memory. In the case where a device (such as a GPU) is used as the primary execution space and the arrays should remain on the device to reduce data transfers, the buffer filling functions need to be converted too. Thus, we changed all buffers to be Views and converted the buffer filling functions into kernels that can be executed on any execution space. In addition, this allows for CUDA-aware MPI– GPU buffers to be directly copied between the memories of GPUs (both on the same node and on different nodes) without an implicit or explicit copy of the data to system memory.

In general, the first three changes above are required in refactoring any legacy code to make use of Kokkos. We note that the original Athena++ design made it mostly straightforward to implement those changes, e.g., because of the existence of an abstract array class and the prevailing tightly nested loops already optimized for vectorized instructions. More broadly, we expect that structured grid fluid codes will require similar changes and that other algorithms and application may require more subtle refactoring in order to achieve good performance. The fourth change was required more specifically for Athena++ due to the existing MPI communication patterns.

Finally, for the purpose of the initial proof-of-concept, we only refactored the parts required for running hydrodynamic and magnetohydrodynamic simulations on static and adaptive Cartesian meshes. Running special and general relativistic simulations on spherical or cylindrical coordinates is currently not supported. However, the changes required to allow for these kind of simulations are straightforward and we encourage and support contributions to re-enable this functionality.

Throughout the development process, we continuously measured the code performance in detail using so-called Kokkos profiling regions as well as the automated profiling of all Kokkos kernels. Moreover, we employed automated regression testing using GitLab's continuous integration features and included specific tests to address changes related to Kokkos (such as running on different architectures and testing different loop patterns).

4.3 Results

If not noted otherwise, all results in this section have been obtained using a double precision, shock-capturing, unsplit, adiabatic MHD solver consisting of Van Leer integration, piecewise linear reconstruction, Roe Riemann solver, and constrained transport for the integration of the induction equation (see, e.g., Stone & Gardiner (2009) for more details). The test problem is a linear fast magnetosonic wave on a static, structured, three-dimensional grid. In GPU runs there is no explicit data transfer between system and GPU memory except during problem initialization, i.e., the exchange of ghost cells is handled either by direct copies between buffers in GPU memory on the same GPU or between buffers in GPU memory on different GPUs using CUDA-aware MPI. Similarly, there is also no implicit data transfer as unified memory was not used. Generally, we used the Intel compilers on Intel platforms, and `gcc` and `nvcc` on other platforms as we found that (recent) Intel compilers are more effective in automatic vectorization than (recent) `gcc` compilers. We used the identical software environment and compiler flags for both K-Athena and Athena++ where

Table 4.1: Software Environment and Compiler Flags Used In Scaling Tests.

Machine	Compiler	Compiler flags	MPI version
Summit GPU	GCC 6.4.0 & Cuda 9.2.148	`-O3 -std=c++11 -fopenmp -Xcudafe -diag_suppress=\ esa_on_defaulted_function_ignored -expt-extended-lambda -arch=sm_70 -Xcompiler`	Spectrum MPI 10.2.0.11
Summit CPU	GCC 8.1.1	`-O3 -std=c++11 -fopenmp-simd -fwhole-program -flto -ffast-math -fprefetch-loop-arrays -fopenmp -mcpu=power9 -mtune=power9`	Spectrum MPI 10.2.0.11
Titan GPU	GCC 6.3.0 & Cuda 9.1.85	`-O3 -std=c++11 -fopenmp -Xcudafe -diag_suppress=\ esa_on_defaulted_function_ignored -expt-extended-lambda -arch=sm_35 -Xcompiler`	Cray MPICH 7.6.3
Titan CPU	GCC 6.3.0	`-O3 -std=c++11 -fopenmp`	Cray MPICH 7.6.3
Theta	ICC 18.0.0	`-O3 -std=c++11 -ipo -xMIC-AVX512 -inline-forceinline -qopenmp-simd -qopenmp`	Cray MPICH 7.7.3
Electra	ICC 18.0.3	`-O3 -std=c++11 -ipo -inline-forceinline -qopenmp-simd -qopt-prefetch=4 -qopenmp -xCORE-AVX512`	HPE MPT 2.17

possible. Details are listed in Table 4.1. We used Athena++ version 1.1.1 (commit `4d0e425`) and K-Athena commit `73fec12d` for the scaling tests. Additional information on how to run K-Athena on different machines can be found in the code's documentation.

4.3.1 Profiling

In order to evaluate the effect on performance of the different loop structures presented in Sec. 4.2.3 we compare the timings of different regions within the main loop of the code. The results using both an NVIDIA V100 GPU and an Intel Skylake CPU for a selection of the computationally most expensive regions are shown in Fig. 4.1. The `1DRange` loop structure refers to a one dimensional range policy over a single index that is explicitly unpacked to the multidimensional indices in the code (cf. Listing 4.3). While this `1DRange` is the fastest loop structure for all regions on the GPU, it is the slowest for all regions on the CPU. According to the compiler report this particular one dimensional mapping prevents automated vectorization optimizations. All other loop structures tested, i.e., `simd-for` (cf. Listing 4.1), `MDRange` (cf. Listing 4.2), and `TeamPolicy` (cf. Listing 4.4) logically separate the nested loops and, thus, make it easier for the compiler to auto-

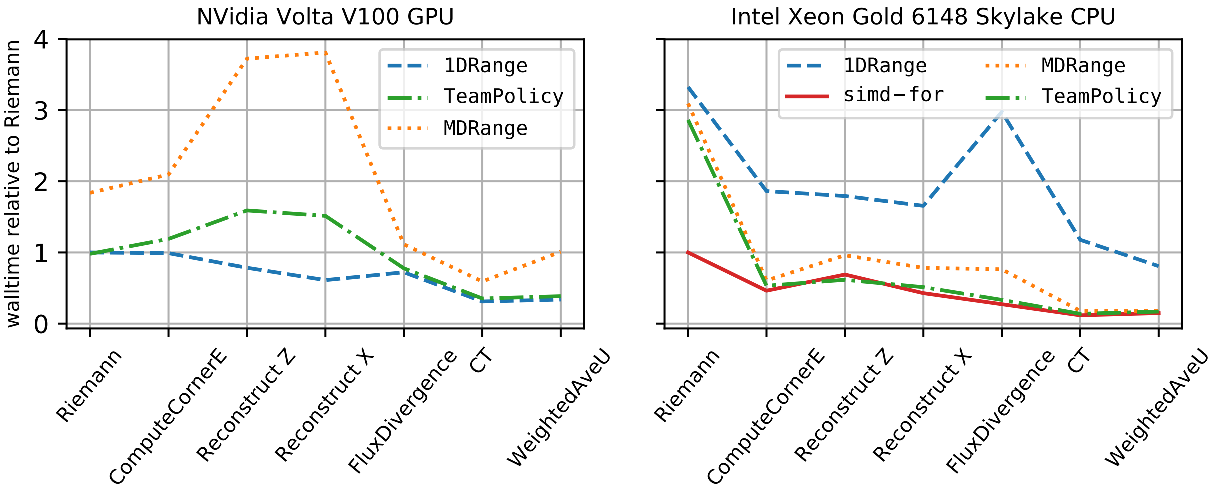


Figure 4.1: Profiling results on a GPU (left) and CPU (right) for selected regions (x-axis) within the main loop of an MHD timestep using the algorithm described in Sec. 4.3. The different lines correspond to different loop structures, see Sec. 4.2.3 and the timings are normalized to the fastest Riemann region in each panel.

matically vectorize the innermost loop. This also explains why the results for `simd-for`, `MDRange`, and `TeamPolicy` are very close to each other for all regions except the Riemann solver. The Riemann solver is the most complex kernel in the chosen setup so that the compiler is not automatically vectorizing this loop despite the `#pragma ivdep` in Kokkos' `MDRange` and `TeamPolicy`. Only the more aggressive explicit `#pragma omp simd` results in a vectorized loop. The aggregate performance differences (all kernels of a cycle combined) to the fastest `simd-for` pattern are 0.78 (`TeamPolicy`), 0.71 (`MDRange`), and 0.51 (`1DRange`).

On the GPU, `MDRange` is the slowest loop structure, being several times (2x-4x) slower than the `1DRange` across all regions. `TeamPolicy` is on par with `1DRange` for half of the regions shown. Here, the aggregate performance differences to the fastest `1DRange` pattern are 0.75 (`TeamPolicy`) and 0.078 (`MDRange`). As discussed in more detail in Sec. 4.4, we expected these non-optimized raw loop structures to not cause any major differences in performance.

The results shown here for V100 GPUs and Skylake CPUs equally apply to other GPU generations and other CPUs (and Xeon Phis), respectively. For all tests conducted in the following, we use the loop structure with the highest performance on each architecture, i.e., `1DRange` on GPUs and `simd-for` on CPUs and Xeon Phis.

4.3.2 Performance portability

Our main objective for writing K-Athena is an MHD code that runs efficiently on any current supercomputer and possibly any future machines. A code that runs efficiently on more architectures is said to be performance portable. Determining what is meant by "efficient code" can be vague, especially when comparing performance across different architectures. The memory space sizes, bandwidths, instruction sets, and arrangement of cores on different architectures can all affect how efficiently a code can utilize the hardware.

In order to make fair comparisons of K-Athena's performance across different machines (see Sec. 4.3.2.1), we used the roofline model Williams et al. (2009), described in Sec. 4.3.2.2, to compute on several architectures the architectural efficiency of K-Athena, or the fraction of the performance achieved compared to the theoretical performance as limited hardware. We then used the architectural efficiencies to compute the performance portability metric from Pennycook et al. (2019), described in Sec. 4.3.2.3, to quantify the performance portability of K-Athena.

4.3.2.1 Overview of architectures used

In total, we created roofline models for six Intel CPUs, Intel Xeon Phis, and three NVIDIA GPUs. The CPU models roughly follow Intel's tick-tock production model and, thus, span pairs of three different instructions sets (AVX, AVX2, and AVX512) with one CPU introducing a new instruction set and the other an increase in cores and/or clock rate with the same instruction set. The Intel Xeon Phi (Knights Landing) also supports AVX512 instructions and differs from the CPUs at the highest level by an increased core count, lower clock rate, and access to MCDRAM. The three different NVIDIA GPUs span three different microarchitectures (Kepler, Pascal, and Volta), which also translates to an increased core count in the GPUs used. L1 data caches are also implemented differently across the three microarchitectures. On Kelper and Volta GPUs, the L1 cache is physically in the same memory device as CUDA "shared" memory while on Pascal GPUs

Table 4.2: Technical specifications for devices used in the performance portability metric. Cache size and core counts for CPUs specify the aggregate sizes and counts for a two-socket node while numbers for GPUs show the aggregate for a single device. For the Tesla K80, the cache size and core count is for just one of the two GK210 chips in the GPU. For DRAM bandwidth (BW) we use the empirically measured bandwidth of the DRAM on CPUs and the global memory on GPUs. Data for Intel devices comes from Intel Corporation (2016) and data for NVIDIA devices comes from NVIDIA Corporation (2014, 2016, 2017); Jia et al. (2018).

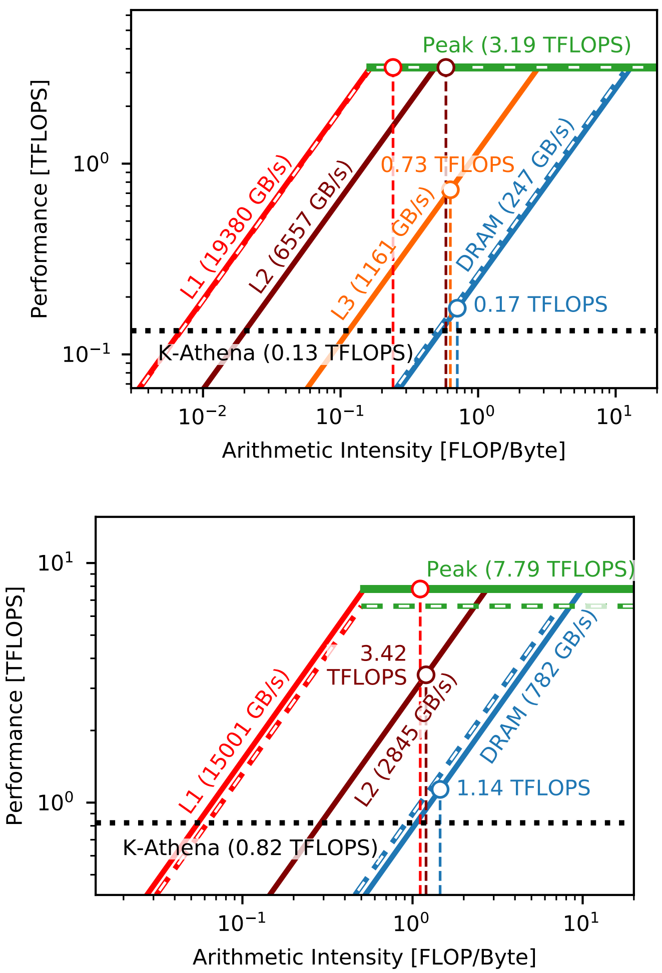
Manufacturer	Intel	Intel	Intel	Intel	Intel	Intel	Intel	NVIDIA	NVIDIA	NVIDIA
Family	Xeon E5	Xeon E5	Xeon E5	Xeon E5	Xeon Gold	Xeon Gold	Xeon Phi	Tesla	Tesla	Tesla
Microarchitecture	Sandy Bridge	Ivy Bridge	Haswell	Broadwell	Skylake	Cascade Lake	Knights Landing	Kepler	Pascal	Volta
Model	2670	2680v2	2680v3	2680v4	6148	6248	7250	K80	P100	V100
Instruction Set	AVX	AVX	AVX2	AVX2	AVX512	AVX512	AVX512			
CUDA Capability								3.7	6.0	7.0
Clock Rate (GHz)	2.6	2.8	2.5	2.4	2.4	2.5	1.4	0.562	1.328	1.29
Num. Cores	16	20	24	28	40	40	68	832	1792	2560
Max L1 Cache (KB)	512	640	768	896	1280	1280	2176	1456	1344	10240
Total L2 Cache (KB)	4096	2560	5120	7168	40000	40000	34000	1536	4096	6144
Total L3 Cache (MB)	40	50	60	70	55	55				
DRAM BW (GB/s)	97.9	121	139	147	246	247	494	195	521	782

the L1 cache is combined with texture memory NVIDIA Corporation (2014, 2016, 2017). Load throughput to L1 cache on Pascal GPUs achieves lower bytes/cycle compared to Kelper and Volta GPUs Jia et al. (2018), which led to K-Athena maintaining a higher fraction of peak L1 bandwidth. An comparative overview of the technical specifications for all architectures is given in Table 4.2.

4.3.2.2 Roofline model

The roofline model is a graphical tool to demonstrate the theoretical peak performance of an application on an architecture by condensing the performance limits imposed by the bandwidth of each memory space and peak throughput of the device into a single plot. In a roofline model plot, peak throughputs and bandwidths of the hardware are plotted on a log Performance [FLOPS] versus log arithmetic intensity [FLOP/B] axis so that throughputs are horizontal lines and bandwidths as $P \propto I$ lines (since bandwidth-limited $P = B \times I$), where P [FLOPS] is performance[5], I [FLOP/B]

[5] In this work we consider double precision throughput and count FMA instructions as two FLOP on architectures that support it.



[Cascade Lake CPU Roofline]

[Tesla V100 GPU Roofline]

Figure 4.2: Roofline models of a 2 socket Intel Xeon Gold 6248 "Cascade Lake" CPU node on NASA's Aitken (4.2a) and a single NVIDIA Tesla V100 "Volta" GPU on MSU HPCC (4.2b). Theoretical L1 and DRAM bandwidths and theoretical peak throughputs according to manufacturer specifications are shown in dashed line. for For both cases shown here and all other architectures we tested, DRAM bandwidth (or MCDRAM bandwidth for KNLs) is the limiting bandwidth for K-Athena's performance.

is arithmetic intensity (the operations executed per byte read and written), and B [B/s] is the bandwidth. The arithmetic intensities of each memory space for a specific application appear as vertical lines, extending up where the bandwidth of the memory space limits performance.

The maximum theoretical performance of an application is limited by the bandwidth and throughput ceilings displayed in the roofline model. For the given device and application, the maximum obtainable performance in FLOPS is limited by

$$P_{\max}(a,p,i) \leq \min_{m \in M} \{\min \left[\, T_{\text{Peak}}(i), \right. \\ \left. B(i,m) \times I(a,p,i,m) \right] \} \,, \tag{4.1}$$

where $P_{\max}(a,p,i)$[FLOPS] is the maximum possible FLOPS obtainable by application a solving problem p on architectural platform i, $T_{\text{Peak}}(i)$[FLOPS] is the peak throughput on the platform, M is all the memory spaces on the device (L1 cache, L2 cache, DRAM, etc.), and $I(a,p,i,m)$[FLOP/B] is the arithmetic intensity the application solving the problem on the memory space m, or the number of FLOP executed per number of bytes written and read to and from m. We can also mark the actual performance of application with a horizontal dashed line, indicating the actual average FLOPS achieved. Figures 4.2a and 4.2b show roofline models of K-Athena solving a 256^3 linear wave on an Intel Cascade Lake CPU node on NASA's Aitken and a single NVIDIA Volta V100 GPU on MSU's HPCC.

Using the roofline model, we can quantify the architectural efficiency of the K-Athena, or the fraction of performance achieved compared to the theoretical maximum performance of the algorithm as limited by bandwidth. In this work, we further distinguish multiple architectural efficiencies per platform as limited by the bandwidth of different memory spaces. The architectural efficiency $e(a,p,i,m)$ of the application a solving the problem p on platform i as limited by the bandwidth of the memory space m on platform i is

$$e(a,p,i) = \frac{\varepsilon(a,p,i)}{\min\left(T_{\text{Peak}}(i), B(i,m) \times I(a,p,i,m)\right)} \tag{4.2}$$

where $\varepsilon(a,p,i)$ is the achieved performance of the application a for solving the problem p on the platform i, $B(i,m)$ is the peak DRAM bandwidth on the platform, and $I(a,p,i,m)$ is the arithmetic intensity of the for solving the problem on that platform. For example, on Summit's Volta V100s, K-Athena achieves 0.82 TFLOPS while the DRAM bandwidth limits performance to 1.13 TFLOPS, giving to a 72.5% architectural performance as limited by DRAM bandwidth.

Although bandwidths and throughputs can be obtained from vendor specifications and arithmetic intensities can be computed by hand, empirical testing more accurately reflects the actual performance. Acquiring these metrics requires a variety of performance profiling tools on the different architectures and machines. For gathering the bandwidths and throughputs on GPUs, we used GPUMembench Konstantinidis & Cotronis (2017) for measuring the L1 bandwidth and the Empirical Roofline Tool (Version 1.1.0) Lo et al. (2015) for measuring all other bandwidths and the peak throughput. For computing arithmetic intensities on GPUs, we used NVIDIA's nvprof (CUDA Toolkit 9.2.88 on MSU HPCC, 9.2.148 on SDSC Comet) to measure memory usage to calculate arithmetic intensities and total FLOP count to estimate FLOP per finite volume cell update. To measure memory usage of the different caches, we specifically measured total memory transactions from global memory to the SMs (`gld_transactions` and `gst_transactions`, as a rough proxy for L1 usage), transactions to and from L2 cache (`l2_read_transactions` and `l2_write_transactions`), and transactions to and from DRAM/HBM (`dram_read_transactions` and `dram_write_transactions`). Since we do not use atomic memory operations, texture memory, or shared memory, we measured zero transactions from these memory spaces. For Intel CPUs and KNLs, we used Intel Advisor's (version 2019 update 5) built-in hierarchical roofline gathering tools to collect memory bandwidths, throughputs, and arithmetic intensities Marques et al. (2017) using the arithmetic intensity from the cache-aware roofline model for the roofline of the highest memory level. For both CPUs and GPUs, we use total memory transactions to cores and SMs as a surrogate for L1 cache usage due to limitations in the memory transaction metrics available. Although some of the memory transactions may not be through L1 cache, in a best case performance scenario the memory transactions to the registers are limited by the fastest cache bandwidth, which is the L1 cache bandwidth.

We used a 3D linear wave on a 256^3 cell grid for benchmarking K-Athena's performance and arithmetic intensities for the roofline model. Our metric for CPU machines are for two sockets on a node while the metric for KNLs and GPUs are for a single device, or a single GK210 chip for the Tesla K80. In all cases we found that K-Athena's performance is limited by the main memory space that accommodates the data for a single MPI task. For GPUs, this is on device DRAM/HBM, for CPUs this is the DDR3/DDR4 DRAM, and for KNLs this was the MCDRAM. This result is expected, since the finite volume MHD method in K-Athena is implemented as a series of simple triple or quadruple for-loop kernels that loop over the data in a task without explicitly caching data. Since the data can only fit in its entirety in DRAM, it must be loaded from and written to DRAM within each kernel. Future improvements can be made to K-Athena to explicitly cache data in smaller 1D arrays and kept in higher level caches. This would raise the DRAM arithmetic intensity and facilitate faster throughput Glines et al. (2015). Similar improvements have already been implemented upstream in Athena++. A more complete solution would involve fusing consecutive kernels into one kernel to reduce DRAM accesses. Given the virtually identical performance between Athena++ and K-Athena on CPUs (cf. 4.3.3.1) we expect the roofline model of Athena++ to be practically indistinguishable from K-Athena on non-GPU platforms.

4.3.2.3 Performance portability metric

Performance portability is at present nebulously defined. It is generally held that a performance portable application can execute wide variety of architectures and achieve acceptable performance, preferably maintaining a single code base for all architectures. In order to make valid comparisons between codes, an objective metric of performance portability is needed.

The metric proposed by Pennycook et al. (2019) quantifies performance portability by the harmonic sum of the performance achieved on each platform, so that

$$P(a,p,H) = \begin{cases} \dfrac{|H|}{\sum_{i\in H} \frac{1}{e(a,p,i)}} & \text{if } i \text{ is supported } \forall i \in H \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where H is the space of all relevant platforms and $e(a,p,i)$ is the performance efficiency of application a to solve the problem p on a platform i. If an application does not support a platform, then it is not performance portable across the platforms and is assigned a metric of 0. The performance efficiency can also be defined as either the application efficiency, the fraction of the performance of the fastest application that can solve the problem on the platform; or as the architectural efficiency, the achieved fraction of the theoretical peak performance limited by the hardware that we computed in Sec. 4.3.2.2. Since we did not have MHD codes implementing the same method as K-Athena on all architectures, we used the architectural efficiencies obtained from the roofline model to compute the performance portability metric. For completeness, we considered the architectural efficiencies as limited by the both the L1 cache and DRAM bandwidths to compute separate performance portability metric against both memory spaces.

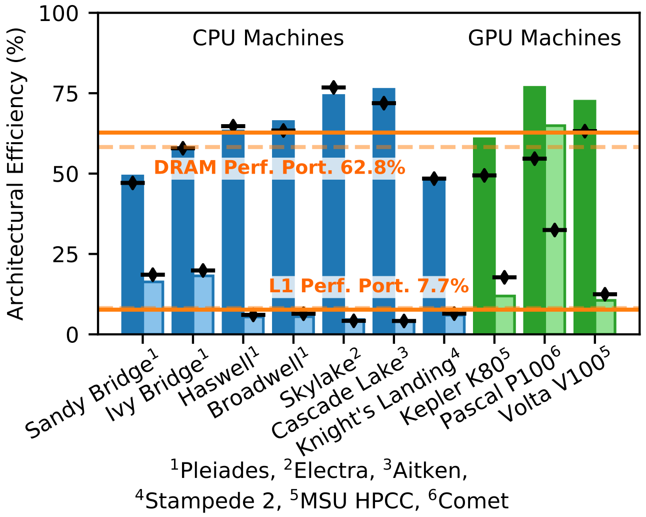


Figure 4.3: Performance Portability plot of several CPU and GPU machines with different architectures. Individual bars show the performance of K-Athena compared to the theoretical peak performance limited by the empirically measured DRAM and L1 bandwidths. Black bars with diamonds denote the theoretical performance limited by the manufacturer reported bandwidths. The performance portability metrics across all architectures for DRAM and L1 are shown with horizontal orange lines where solid orange used the empirically measured bandwidths and dashed orange uses manufacturer reported bandwidths.[6]

In Fig. 4.3, the architectural efficiencies as measured against the DRAM bandwidth and L1 cache bandwidth are shown with the computed performance portability metrics. K-Athena achieved 62.8% DRAM performance portability and 7.7% L1 cache performance portability, measured across a number of CPU and GPU architectures. In general, K-Athena achieved higher efficiencies on newer architectures.

[6] The high L1 efficiency on the NVIDIA Tesla Pascal P100 is due to a lower obtainable bytes loaded to L1 per cycle compared to the Kepler and Volta GPUs Jia et al. (2018, 2019). The lower L1 cache performance makes it easier to obtain a higher efficiency.

4.3.3 Scaling

4.3.3.1 Single CPU and GPU performance

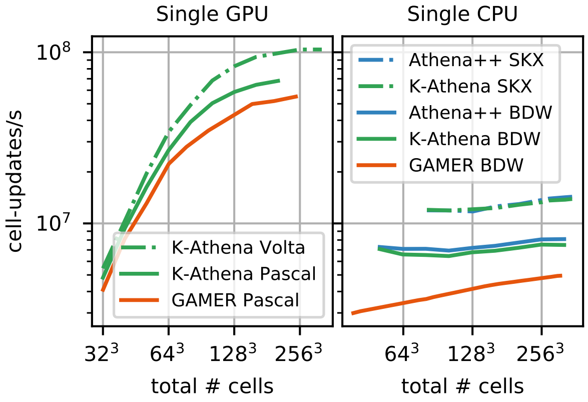


Figure 4.4: Raw performance for double precision MHD (algorithm described in Sec. 4.3) of K-Athena, Athena++, and GAMER on a single GPU (left) or CPU (right) for varying problem sizes. Volta refers to an NVIDIA V100 GPU, Pascal refers to an NVIDIA P100 GPU, BDW (Broadwell) refers to a 14-core Xeon E5-2680 CPU, and SKX (Skylake) refers to a 20-core Xeon Gold 6148 CPU. The GAMER numbers were reported in Zhang et al. (2018) for the same algorithm used here.

In order to compare the degree to which the refactoring of Athena++ affected performance we first compare Athena++ and K-Athena on a single CPU. The right panel of Fig. 4.4 shows the cell-updates/s achieved on an Intel Broadwell and an Intel Skylake CPU for both codes for varying problem size. Overall, the achieved cell-updates/s are practically independent of problem sizes reaching $\approx 8 \times 10^6$ on a single Broadwell CPU and $\approx 1.4 \times 10^7$ on a single Skylake CPU. Moreover, without any additional performance optimizations (see discussion in Sec. 4.4), K-Athena is virtually on par with Athena++, reaching 93% or more of the original performance. For comparison, we also show the results of GAMER Zhang et al. (2018). It is another recent (astrophysical) MHD code with support for CPU and (CUDA-based) GPU accelerated calculations and has directly been compared to Athena++ in Zhang et al. (2018). We also find that Athena++ (and thus K-Athena) is about 1.5 times faster than GAMER on the same CPU.

A slightly smaller difference (factor of ≈ 1.25) is observed when comparing results for GPU runs as shown in the left panel of Fig. 4.4. On a P100 Pascal GPU, K-Athena is about 1.3 times faster than GAMER, suggesting that the difference in performance is related to the fundamental code design and not related to the implementation of specific computing kernels. On a single V100 Volta GPU, K-Athena reaches a peak performance of greater than 10^8 cell-updates/s for large problem sizes. In general, the achieved performance in cell-updates/s is strongly dependent on the problem size. For small grids the performance is more than one order of magnitude lower than what is achieved for the largest permissible grid sizes that still fit into GPU memory. The plateau in performance on GPUs at larger grid sizes is due to DRAM bandwidth impeding K-Athena's performance, as discussed in Section 4.3.2.2.

4.3.3.2 Weak scaling

Weak scaling results (using the same test problem and algorithm as in Sec. 4.3.3.1) for K-Athena and the original Athena++ version on different systems and architectures are shown in Fig. 4.5. Note that the chosen problem setup (using a single meshblock per MPI process) is effectively not making use of of the asynchronous communication capabilities to allow for overlapping computation and communication.

Overall, the differences between K-Athena and Athena++ on CPUs and Xeon Phis are marginal. This is expected as K-Athena employed `simd-for` loops for all kernels that are similar to the ones already in Athena++. Therefore, the parallel efficiency is also almost identical between both codes, reaching ≈ 80% on NASA's Electra system with Skylake CPUs (first column in Fig. 4.5) and ≈ 70% on ALCF's Theta system with Knights Landing Xeon Phis (second column in Fig. 4.5) at 2,048 nodes each. Using multiple hyperthreads per core on Theta has no significant influence on the results given the intrinsic variations observed on that system[7].

The first major difference is observed on OLCF's Titan (third column in Fig. 4.5), where results for K-Athena on GPUs are included. While the parallel efficiency for both codes remains at 94% up to 8,192 nodes using only CPUs, it drops to 72% when using GPUs with K-Athena. However, the majority of loss in parallel efficiency already occurs going from 1 to 8 nodes using GPUs and

[7] According to the ALCF support staff, system variability contributes around 10% to the fluctuations in performance between identical runs.

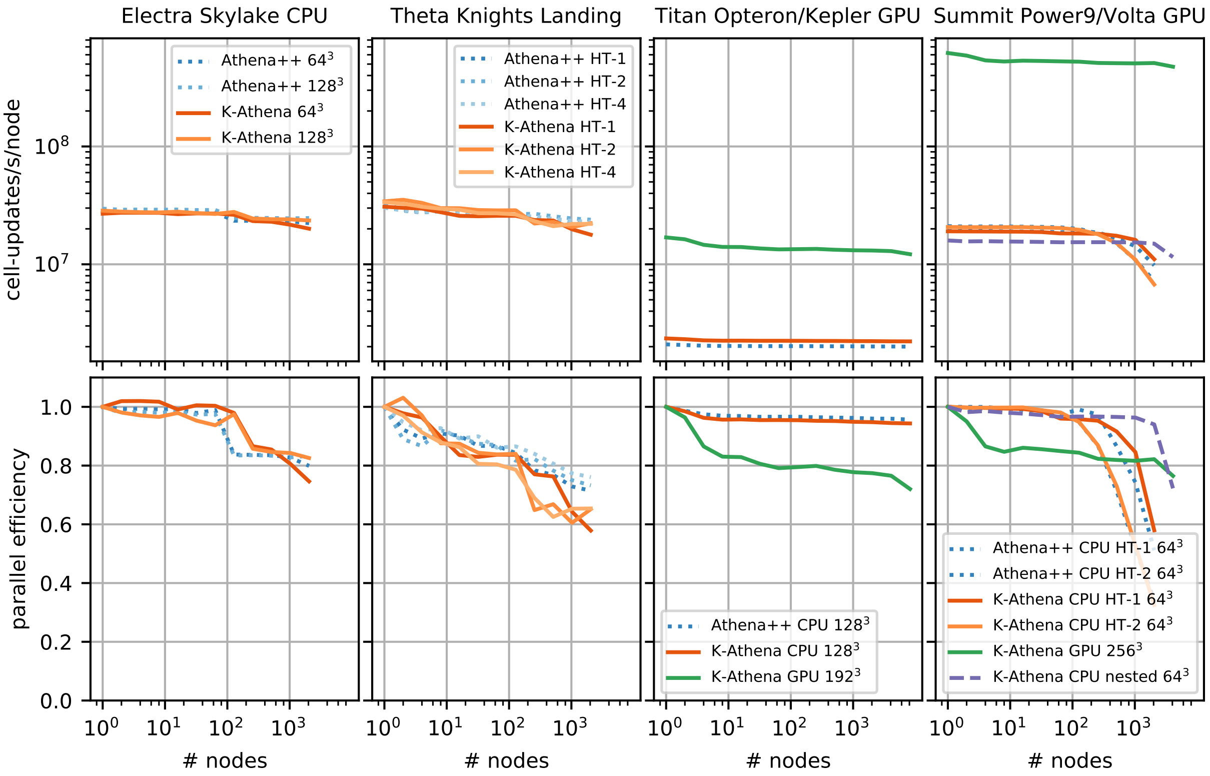


Figure 4.5: Weak scaling for double precision MHD (exact algorithm described in Sec. 4.3) on different supercomputers and architectures for K-Athena and the original Athena++ version. Numbers correspond to the 80th percentile of individual cycle performances of several runs in order to reduce effects of network variability. The top row shows the raw performance in number of cell-updates per second per node and can directly be compared between different system and architectures. The bottom row shows the parallel efficiency normalized to the individual single node performance. The first column contains results for a workload of 64^3 and 128^3 cells per core on NASA's Electra system using two 20-core Intel Xeon Gold 6148 processors per node. The second column shows results for a workload of 64^3 per core on ALCF's Theta system with one 64-core Intel Xeon Phi 7230 (Knights Landing) per node. HT-1, HT-2, and HT-4 refers to using 1, 2, and 4 hyperthreads per core, respectively. The third column shows results for a workload of 128^3 per CPU core and 192^3 per GPU on OLCF's Titan system with one AMD Opteron 6274 16-core CPU and one NVIDIA K20X (Kepler) GPU per node. The last column contains results for a workload of 64^3 per CPU core and 256^3 per GPU on OLCF's Summit system with two 21-core IBM POWER9 CPUs and six NVIDIA V100 (Volta) GPUs per node. On all systems the GPU runs used 1D loops and the CPU runs used `simd-for` loops with the the exception of the dashed purple line on Summit that used Kokkos nested parallelism, see Sec. 4.2.3 for more details.

afterwards remains almost flat. This behavior is equally present for CPU runs but less visible due to the higher parallel efficiency in general. The differences in parallel efficiency between CPU and GPU runs can be attributed to the vastly different raw performance of each architecture. On a single node the single Kepler K20X GPU is about 7 times faster than the 16-core AMD Opteron CPU. Given that the interconnect is identical for GPU and CPU communication, the effective ratio of computation to communication is worse for GPUs. Despite the worse parallel efficiency on GPUs the raw per-node performance using GPUs is still about 5.5 times faster than using CPUs at 8,192 nodes, which is overall comparable to the ratio of theoretical peak performances in both FLOPS and DRAM bandwidth.

K-Athena on OLCF's Summit system (last column in Fig. 4.5) with six Volta V100 GPUs and two 21-core POWER9 CPUs exhibits a GPU weak scaling behavior similar to the one observed on Titan. Going from 1 to 8 nodes results in a loss of 15% and afterwards the parallel efficiency remains almost flat to 76% on 4,096 nodes. The CPU weak scaling results for both codes using CPUs reveal properties of the interconnect. The weak scaling is almost perfect up to 256 nodes using 1 hyperthread per core and afterwards rapidly plummets. Using 2 hyperthreads per core (i.e., doubling the number of threads making MPI calls and doubling the number of MPI messages sent and received, as described in Sec. 4.2.2) the steep drop in parallel efficiency is already observed beyond 128 nodes. No such drop is observed using GPUs, which perform $42/6 = 7$ times fewer MPI calls (compared to using 1 hyperthread per core) with larger message sizes in general.

Naturally, this is tightly related to the existing communication pattern in Athena++, i.e., coarse grained threading over meshblocks with each thread performing one-sided MPI calls. Without making additional changes to the code base, we can evaluate the effect of reducing the number of MPI calls for a fixed problem size in a multithreaded CPU setup using Kokkos nested parallelism in K-Athena. More specifically, we use the triple nested construct illustrated in Listing 4.4 allowing multiple threads handling a single meshblock. As a proof of concept, the results for using using 1 MPI process per 2 cores each with one thread are shown in the purple dash line in the last column of Fig. 4.5. While the raw performance on a single node is slightly lower (about 16%), the improved communication pattern results in a higher overall performance for > 1,024 nodes. Similarly, the sharp drop in parallel efficiency has been shifted to first occur at 2,048 nodes.

At the single node level the six GPUs on Summit are tightly connected via NVLink. The weak scaling efficiency from one GPU to six GPUs on a single node is ≈ 99% (cf., > 6×10^8 cell-updates/s/node for a single node in the top right panel of Fig. 4.5). In addition, the host interconnect has a lower bandwidth and higher latency compared to NVLink. Thus, the intra-node parallel overhead is generally negligible in our analysis.

Finally, the raw per-node performance is overall comparable between Intel Skylake CPUs, Intel Knight Landing Xeon Phis, IBM POWER9 CPUs, and a single NVIDIA Kepler GPU, ranging between ≈ 1.5 – $3{\times}10^7$ cell-updates/s/node. The latest NVIDIA Volta GPU is a notable exception, reaching more than 10^8 cell-updates/s/GPU. This performance, in combination with six GPUs per node on Summit and a high parallel efficiency, results in a total performance of 1.94×10^{12} cell-updates/s on 4,096 nodes.

4.3.4 Strong scaling

Strong scaling results for K-Athena on Summit on both CPUs and GPUs are shown in Fig. 4.6 (same test problem and algorithm as in Sec. 4.3.3.1). Overall, strong scaling in terms of parallel efficiency is better on CPUs than on GPUs. For example, for a $1{,}408^3$ domain the parallel efficiency using CPUs remains > 83% going from 32 to 512 nodes whereas it drops to 45% for the similar GPU case ($1{,}536^3$ domain using 36 to 576 nodes). This is easily explained by comparing to the single CPU/GPU performance discussed in Sec. 4.3.3.1, which effectively corresponds to on-node strong scaling. The more pronounced decrease in parallel efficiency on the GPUs is a direct result of the decreased raw performance of GPUs with smaller problem sizes per GPU. The increased communication overhead of the strong scaling test plays only a secondary role. Therefore, the strong scaling efficiency of K-Athena in comparison to Athena++ is expected to be identical. Moreover, additional performance improvements, as discussed in the following Section, will greatly benefit the strong scaling behavior of GPUs in general. Nevertheless, the raw performance of the GPUs

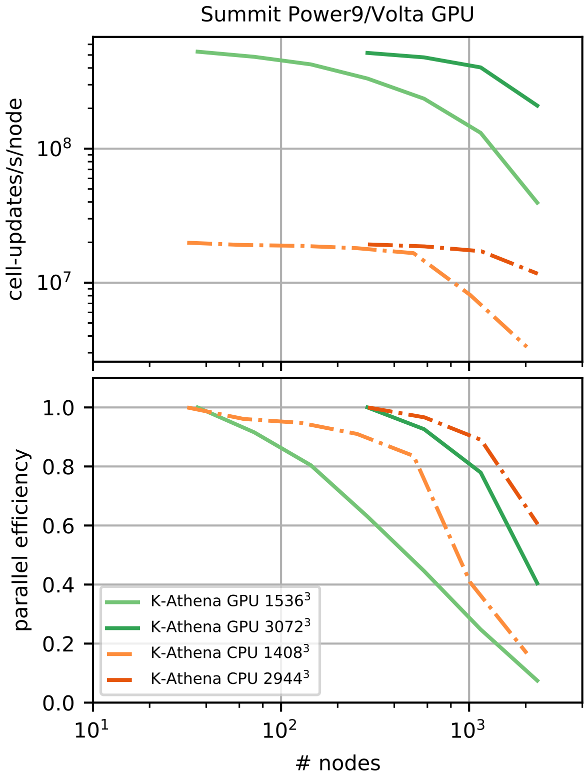


Figure 4.6: Strong parallel scaling for double precision MHD (algorithm described in Sec. 4.3) of K-Athena on NVIDIA V100 GPUs (6 GPUs per node; green solid lines) and IBM Power 9 CPUs (42 cores per node; orange/red dash dotted lines) on Summit. The top panel shows the raw performance in cell-updates per second per node and the bottom panel shows the parallel efficiency. The effective workload per GPU goes from 256^3 to 64^3 for the $1{,}536^3$ domain and from 256^3 to 128^3 for the 3072^3 domain. In the CPU case the effective workload per single Power9 CPU (21 cores) goes from 353^3 to 88^3 for the $1{,}408^3$ domain and from 353^3 to 177^3 for the $2{,}944^3$ domain. The resulting effective workloads per node are comparable (within few percent) between GPU and CPU runs.

still outperforms CPUs by a large multiple despite the worse strong scaling parallel efficiency. For example, in the case discussed above on Summit, the per-node performance of GPUs over CPUs is still about 14 times higher at > 512 nodes.

4.4 Current limitations and future enhancements

Our primary goal for the current version of K-Athena was to make GPU-accelerated simulations possible while maintaining CPU performance, and to do so with the smallest amount of code changes necessary. Naturally, this resulted in several trade-offs and leaves room for further (performance) improvements in the future.

For example, we are currently not making use of the memory hierarchy abstraction provided by Kokkos. This includes more advanced hardware features such as scratch spaces on GPUs. Scratch space can be shared among threads of a `TeamPolicy` and allows for efficient reuse of memory. We could use scratch space to reduce the number of reads from DRAM in stenciled kernels (like the fluid solver's reconstruction step). We could also fuse consecutive kernels to further reduce reads and writes to DRAM, although this would also increase register and possibly spill store usage. Moreover, complex kernels such as a Riemann solver could be broken down further by using `TeamThreadRange`s and `ThreadVectorRange`s structures that are closer to the structure of the algorithm. This is in contrast to our current approach where all kernels are treated equally, with the same execution policies independent of the individual algorithms within the kernels. The Riemann solver could also be split into separate kernels to reduce the number of registers needed, eliminate the use of spill stores on the GPU, and allow higher occupancy on the GPU.

Similarly, on CPUs and Xeon Phis we are currently not using a Kokkos parallel execution pattern. The macro we introduced to easily exchange parallel patterns replaces the parallel region on CPUs and Xeon Phis with a simple nested `for` loop including a `simd` pragma, as shown in Listing 4.1. This is required for maximum performance as the implicit `#pragma ivdep` hidden in the Kokkos templates is less aggressive than the explicit `#pragma omp simd` with respect to automated vectorization. We reported this issue and future Kokkos updates will address this by either providing an explicit tightly nested vectorized loop pattern and/or adding support for a `simd` property to the execution policy template.

Another possible future improvement is an increase in parallel efficiency by overlapping communication and computation. While Athena++ is already built for asynchronous communication through one-sided MPI calls and a task based execution model, more fine-grained optimizations are possible. For example, spatial dimensions in the variable reconstruction step that occurs after the exchange of boundary information could be split, so that the kernel in the first dimension could run while the boundary information of the second and third dimension are still being exchanged.

In addition, the next major Kokkos release will contain more support for architecture-dependent task based execution and, for example, will allow for the transparent use of CUDA streams.

CUDA streams may also help in addressing another current limitation of K-Athena on GPUs. Our minimal implementation approach currently limits all `meshblock`s to be allocated in a fixed memory space. This means that the total problem size that can currently be addressed with K-Athena is limited by the total amount of GPU memory available. An alternative approach is keeping the entire mesh in system memory, which is still several times larger than the GPU memory on most (if not all) current machines. For the execution of kernels individual `meshblock`s would be copied back and forth between system memory and GPU memory. Here, CUDA streams could be used to hide these expensive memory transfers as they would occur in the background while the GPU is executing different kernels. Theoretically, meshes larger than the GPU memory could already be used right now with the help of unified memory. However, given that the code is not optimized for efficient page migrations the resulting performance degradation is large (more than a factor of 10). Thus, using unified memory with meshes larger than the GPU memory is not recommended.

4.5 Conclusions

We presented K-Athena – a Kokkos-based performance portable version of the finite volume MHD code Athena++. Kokkos is a C++ template library that provides abstractions for on-node parallel regions and the memory hierarchy. Our main goal was to enable GPU-accelerated simulations while maintaining Athena++'s excellent CPU performance using a single code base and with minimal changes to the existing code.

Generally, four main changes were required in the refactoring process. We changed the underlying memory management in Athena++'s multi-dimensional array class to make transparently use of Kokkos's equivalent multi-dimensional arrays, i.e., `Kokkos::Views`. We exchanged all (tightly) nested `for` loops with the Kokkos equivalent parallel region, e.g., a `Kokkos::parallel_for`, which are now kernels that can be launched on any supported device. We inlined all support functions (e.g., the equation of state) that are called within kernels. We changed the communication buffers to be Views so that MPI calls between GPUs buffers are directly possible without going through system memory.

With all changes in place we performed both profiling and scaling studies across different platforms, including NASA's Electra system with Intel Skylake CPUs, ALCF's Theta system with Intel Xeon Phi Knights Landing, OLCF's Titan with AMD Opteron CPUs and NVIDIA Kepler GPUs, and OLCF's Summit machine with IBM Power9 CPUs and NVIDIA Volta GPUs. Using a roofline model analysis, we demonstrated that the current implementation of the MHD algorithms is memory bound by either the DRAM, HBM, or MCDRAM bandwidths on CPUs and GPUs. Moreover, we calculated a performance portability metric of 62.8% across Xeon Phis, and 6 CPU and 3 GPU generations.

Detailed Kokkos profiling revealed that there is currently no universal Kokkos execution policy (how a parallel region is executed) that achieves optimal performance across different architectures. For example, a one-dimensional loop with manual index matching from 1 to 3D/4D is fastest on GPUs (achieving $> 10^8$) double precision MHD cell-updates/s on a single NVIDIA V100 GPU) whereas tightly nested for loops with simd directives are fastest on CPUs. This is primarily a result of Kokkos's specific implementation details and expected to improve in future releases through more flexible execution policies.

Strong scaling on GPUs is currently predominately limited by individual GPU performance and not by communication. In other words, insufficient GPU utilization outweighs additional performance overhead with decreasing problem size per GPU.

Weak scaling is generally good, with parallel efficiencies of 80% and higher for more than 1,000 nodes across all machines tested. Notably, on Summit K-Athena achieves a total calculation speed of 1.94×10^{12} cell-updates/s on 24,567 V100 GPUs at a speedup of 30 compared to using the available 172,032 CPU cores.

Finally, there is still a great deal of untapped potential left, e.g., using more advanced hardware features such as fine-grained nested parallelism, scratch pad memory (i.e., fast memory that can be shared among threads), or CUDA streams. These are currently being addressed within the new Parthenon collaboration (https://github.com/lanl/parthenon), which is developing a performance portable adaptive mesh refinement framework based on the results presented here.

Nevertheless, we achieved our primary performance portability goal of enabling GPU-accelerated simulations while maintaining CPU performance using a single code base. Moreover, we consider the current results highly encouraging and will continue with further development on the project's GitLab repository at https://gitlab.com/pgrete/kathena. Contributions of any kind are welcome!

CHAPTER 5

RELATIVISTIC DISCONTINUOUS-GALERKIN HYDRODYNAMICS

This chapter has been submitted for publication as Glines et al. (2022). I include the original abstract as the introduction to this chapter.

Chapter Abstract

In this work, we present a discontinuous-Galerkin method for evolving relativistic hydrodynamics. We include an exploration of analytical and iterative methods to recover the primitive variables from the conserved variables for the ideal equation of state and the Taub-Matthews approximation to the Synge equation of state. We also present a new operator for enforcing a physically permissible conserved state at all basis points within an element while preserving the volume average of the conserved state. We implement this method using the Kokkos performance-portability library to enable running at performance on both CPUs and GPUs. We use this method to explore the relativistic Kelvin-Helmholtz instability compared to a finite volume method. Last, we explore the performance of our implementation on CPUs and GPUs.

5.1 Introduction

Many high energy astrophysical and terrestrial plasmas attain relativistic velocities and temperatures. Examples from astrophysics include jets from active galactic nuclei (Blandford et al., 2019), accretion flows onto black holes (Villiers et al., 2003), and gamma-ray bursts (Kumar & Zhang, 2015). In terrestrial systems, relativistic flows can also play a crucial role in a broad range of accelerator systems, including magnetically insulated transmission lines (MITLs) utilized in (for example) the Z machine at Sandia National Laboratories Sinars et al. (2020). In all of these plasmas, velocities close to the speed of light lead to an apparent increase of mass as measured by a stationary observer while relativistic particle velocities at high temperatures lead to a non-linear increase in pressure. Non-relativistic hydrodynamics are insufficient to model such flows – a relativistic treatment of the fluid is required. Numerical solutions for relativistic hydrodynamics were first pioneered in the 1960's and 1970's by May & White (1966) and Wilson (1972). High-resolution shock-capturing solutions followed suit, with an early review of those methods given by Martí & Müller (2003).

When modeling complex systems with small time step constraints, higher order methods are advantageous for efficiently achieving high accuracy. Discontinuous Galerkin methods have become standard in fluid dynamics for enabling high-order methods in complex geometries. High-order discontinuous-Galerkin methods afford enhanced data locality when compared with finite volume methods of similar order (Fuhry et al., 2014). Given the trend in compute performance outpacing memory performance in newer architectures such as graphics processing units (GPUs), the higher arithmetic intensity of discontinuous-Galerkin methods will permit higher computational efficiency due to higher arithmetic intensity algorithms using more of the growing computational throughput while using less of the stagnant memory bandwidth, enabling higher fidelity simulations compared to finite volume simulations for equivalent computational resources.

In this work, we present a robust, performance-portable discontinuous-Galerkin method for relativistic hydrodynamics. In §5.2.1 we present a formulation of the equations of relativistic hydrodynamics that allows for a range of equations of state; we present two such possibilities: (1) an ideal equation of state, which approximates a perfect gas but assumes a constant adiabatic index for a relativistic perfect gas, and (2) an approximation to the Synge gas from Mathews (1971), where the Synge equation of state models a relativistic perfect gas (Synge, 1957). We discuss the discretization of the system using a discontinuous-Galerkin technique and discuss strong-stability-preserving time discretization techniques. To enable robust higher order discretization, in §5.2.5 we present a new and novel physicality-enforcing operator for discontinuous-Galerkin methods for relativistic hydrodynamics. The method smooths conserved variables within individual cells to the cell volume averages until all basis points within the cell satisfy conditions for physicality (i.e. positive density and pressure and flow speed under the speed of light). We implement the method for relativistic hydrodynamics using the Kokkos performance portability library to enable running on both CPUs and GPU (Carter Edwards et al., 2014).

A key part of any algorithm for relativistic hydrodynamics is the method by which the non-linear relationship between primitive variables and the conserved state is solved. In §5.3, we compare analytical and iterative methods for recovering the primitive variables from the conserved variables for both equations of state, across a range of different hardware platforms and compilers as facilitated by Kokkos, finding that for the ideal gas our iterative method following Riccardi & Durante (2008) is faster, more robust, and more accurate than an analytical method, but the exact reverse is true for an approximation to the Synge gas.

We proceed to validate the method using several tests (discussed in detail in §5.4), exploring convergence of the method to analytical solutions of relativistic linear waves, convergence to high resolution reference solutions of a range of 1D shock tubes, evolution of 2D Riemann problems, and growth rates of the relativistic Kelvin-Helmholtz instability with two different initial perturbations. Using a 0th order basis, we find that the method performs comparably to 1st order finite volume methods, as expected. Using higher order bases we see the expected level of convergence for smooth flows. In fluid systems with shocks, the method requires the physicality-enforcing operator presented here and exhibits expected rates of convergence around shocks. Additionally, with the exploration of the growth rate of the Kelvin-Helmholtz problem, we show that using the more accurate HLLC Riemann solver (Mignone & Bodo, 2006) instead of the HLL solver (Schneider et al., 1993) has a greater impact on the growth rate than basis order or resolution. We further utilize this test problem to demonstrate a range of performance portability results in §5.4.6 before summarizing our results and conclusions in §5.5.

5.2 Theoretical Background and Discretization

In this section, we describe our method for relativistic hydrodynamics in a discontinuous-Galerkin code, starting by reviewing the equations for relativistic hydrodynamics in §5.2.1, including a discussion of the equation of state. Then, in §5.2.3, we give the general discontinuous-Galerkin method for solving the relativistic hydrodynamics equations as a set of hyperbolic equations with computation of fluxes given in §5.2.4. Last, in §5.2.5, we present a new operator that enforces physicality of all basis points within a cell while maintaining the volume average within the cell.

5.2.1 Special Relativistic Hydrodynamics

The special relativistic hydrodynamics equations for a relativistic fluid are given by a set of hyperbolic conservation laws,

$$\partial_t \mathbf{U} + \nabla \cdot \mathcal{F}[\mathbf{W}(\mathbf{U})] = 0 \tag{5.1}$$

where the conserved variables $\mathbf{U} = [D, \mathbf{M}, E]^T$ are the relativistic density, relativistic specific momentum, and the total energy density including energy from the rest mass. The flux is

$$\mathcal{F}[\mathbf{W}(\mathbf{U})] = \begin{bmatrix} \rho\mathbf{u} \\ \frac{\rho h}{c^2}\mathbf{u} \otimes \mathbf{u} + P\mathbf{I} \\ \gamma\rho h\mathbf{u} \end{bmatrix}, \tag{5.2}$$

where the rest mass density ρ, the three spacelike components of the 4-velocity denoted here with $\mathbf{u}$, and the pressure P comprises the primitive state $\mathbf{W}(\mathbf{U}) = [\rho, \mathbf{u}, P]^T$. The specific enthalpy h is given by

$$h = \frac{e + P}{\rho} \tag{5.3}$$

where e is the specific internal energy. The conserved state $\mathbf{U}$ can be determined from the primitive state $\mathbf{W}$ by

$$\mathbf{U} = \begin{bmatrix} \gamma\rho \\ \gamma(e + P)\mathbf{u}/c^2 \\ \gamma^2(e + P) - P \end{bmatrix} = \begin{bmatrix} \gamma\rho \\ \gamma\rho h\mathbf{u}/c^2 \\ \gamma^2\rho h - P \end{bmatrix} \equiv \begin{bmatrix} D \\ \mathbf{M} \\ E \end{bmatrix} \tag{5.4}$$

where $\gamma \equiv \sqrt{1 + |\mathbf{u}|^2/c^2}$ is the Lorentz factor and D, $\mathbf{M}$, and E are the relativistic density, relativistic momentum density, and total energy density respectively. We also find it convenient to use the three-velocity $\mathbf{v}$ at times, which relates to $\mathbf{u}$ following $\mathbf{u} = \gamma\mathbf{v}$ and the Lorentz velocity following $\gamma = 1/\sqrt{1 - |\mathbf{v}|^2/c^2}$.

5.2.2 Equations of State

The relativistic hydrodynamics equations in Eq. 5.1 are not complete; an equation of state is used to close the system. Following Ryu et al. (2006), we express the equation of state by relating h to the primitive variables

$$h \equiv h(\rho, P). \tag{5.5}$$

The equation of state also determines the sound speed c_s, which is given by

$$c_s^2 = -\frac{\rho}{nh}\frac{\partial h}{\partial \rho} \quad \text{with} \quad n = \rho\frac{\partial h}{\partial P} - 1 \tag{5.6}$$

where n is the polytropic index. In this work, we explore two choices of equation of state: the equation of state of an ideal gas and the Taub-Matthews approximation to the Synge equation of state described in Mathews (1971).

In a relativistic perfect gas, the adiabatic index decreases with temperature, starting with $\Gamma = 5/3$ for non-relativistic temperatures when $P/\rho \ll c^2$ and decreasing to $\Gamma = 4/3$ for relativistic temperatures when $P/\rho \gg c^2$. The equation of state of the perfect gas is given by the Synge gas (Synge, 1957) :

$$h = c^2 \frac{K_3\left(c^2/\Theta\right)}{K_2\left(c^2/\Theta\right)} \tag{5.7}$$

where K_2 and K_3 are modified Bessel functions of the second kind and $\Theta \equiv P/\rho$ is a temperature-like variable. From a computational standpoint, however, there are significant drawbacks, as these Bessel functions are both expensive to compute and can introduce inaccuracy due to limited machine precision. Even worse, the Bessel functions need to be inverted to recover the primitive variables from conserved variables, which greatly increases computational costs. Consequently, approximations to the equation of state are usually used in simulations.

The simplest approximation to the relativistic perfect gas is the ideal equation of state, which assumes a constant adiabatic index. The enthalpy for the ideal gas is given by

$$h = c^2 + \frac{\Gamma}{\Gamma - 1}\Theta \tag{5.8}$$

where the constant Γ is the adiabatic index (ratio of specific heats.) The corresponding speed of sound is then:

$$\frac{c_s^2}{c^2} = \Gamma\frac{\Theta}{h}. \tag{5.9}$$

For non-relativistic temperatures when $\Theta \ll c^2$, an adiabatic index of $\Gamma = 5/3$ best approximates the perfect gas (consistent with non-relativistic theory) while for relativistic temperatures when $\Theta \gg c^2$ and adiabatic index of $\Gamma = 4/3$ is appropriate. The ideal equation of state is common for relativistic hydrodynamics simulations. However, relativistic fluid systems can have relativistic and non-relativistic temperatures simultaneously at different locations within the fluid, throwing into question the use of a constant adiabatic index across the simulation. Additionally, Taub (1948) showed that $\Gamma \geq 4/3$ becomes inconsistent with relativistic kinetic theory as $\Theta/c^2 \longrightarrow \infty$, suggesting that adiabatic indices above 4/3 are unphysical for ultra-relativistic temperatures.

A more accurate approximation to the Synge gas that is still computationally efficient is the Taub-Matthews approximation to the Synge gas, which we will refer to as the Taub-Matthews equation of state (Mathews, 1971). In this approximation, the enthalpy is given by:

$$h = \frac{5}{2}\Theta + \frac{3}{2}\sqrt{\Theta^2 + \frac{4}{9}c^4} \tag{5.10}$$

with the corresponding sound speed:

$$\frac{c_s^2}{c^2} = \frac{3\Theta^2 + 5\Theta\sqrt{\Theta^2 + \frac{4}{9}c^4}}{12\Theta^2 + 2c^4 + 12\Theta\sqrt{\Theta^2 + \frac{4}{9}c^4}}. \tag{5.11}$$

The Taub-Matthews equation of state satisfies the conditions for causality at high temperatures while correctly approximating the ideal gas equation of state for a subrelativistic gas at low temperatures (Mathews, 1971). As such, the Taub-Matthews equation of state effectively simulates an ideal gas with an adiabatic index that varies from $\Gamma = 5/3$ as $\Gamma = 4/3$ as Θ is taken from $\Theta \longrightarrow 0$ to $\Theta \longrightarrow \infty$. More formally, this can be seen through defining an equivalent adiabatic index[1] (see, e.g. Mignone

[1] Note that since we have not defined a canonical equation of state for the Taub-Matthews equation of state (i.e. $h(S, P)$ where S is entropy), we have not defined a relationship with temperature T, and we cannot compute specific heat capacities and subsequently Γ. Hence the need for the proxy Γ_{eq}.

& McKinney, 2007):

$$\Gamma_{\text{eq}} = \frac{h - c^2}{h - c^2 - \Theta}, \tag{5.12}$$

This relationship, along with the enthalpy and speed of sound, for ideal gases with $\Gamma = 4/3$ and $\gamma = 5/3$, the Synge gas, and the Taub-Matthews equation of state is shown in Fig. 5.1.

5.2.3 Spatial and Temporal Discretizations

In this work, spatial discretization of the hyperbolic conservation law, Eq. 5.1, is performed using a discontinuous-Galerkin method in a similar fashion as was proposed by Núñez-de la Rosa & Munz (2018), following on the influential sequence Cockburn & Shu (1989); Cockburn et al. (1989, 1990); Cockburn & Shu (1998). The discontinuous-Galerkin method requires a mesh defined as the subdivision of the domain into non-overlapping hexahedral ($3D$) or quadrilateral ($2D$) cells denoted $\Omega_k \subset \Omega \subset \mathbb{R}^d$. The approximation of the conserved variables on cell Ω_k is written

$$\mathbf{U}(\mathbf{x}) \approx \mathbf{U}^h(\mathbf{x}) = \sum_{i=1} \mathbf{U}_i \phi_i(\mathbf{x}) \quad \mathbf{x} \in \Omega_k \tag{5.13}$$

where the set $\{\phi_i(\mathbf{x})\}$ is a linearly independent basis that spans a polynomial space of fixed order on element Ω_k. Lagrange polynomials are employed here, where the nodal points are denoted as $\mathbf{x}_j$ such that

$$\phi_i(\mathbf{x}_j) = \delta_{ij} \tag{5.14}$$

where δ is the Kronecker delta function. Globally, $\mathbf{U}^h$ is defined as a piecewise polynomial function with discontinuities permitted at cell boundaries. The restriction of the numerical solution to a cell Ω_k is denoted $\mathbf{U}^h_k$.

On each cell the approximate solution to Eq. 5.1 is computed by enforcing that the residual is orthogonal to the test space, defined in the Galerkin fashion. Practically, after integration by parts, this implies the satisfaction of the weak form

$$\int_{\Omega_k} \frac{\partial \mathbf{U}^h}{\partial t} \phi(\mathbf{x}) d\mathbf{x} + \oint_{\partial\Omega_k} \overline{\mathcal{F}[\mathbf{W}^h(\mathbf{U})] \cdot \mathbf{n}} \phi(\mathbf{x}) ds - \int_{\Omega_k} \mathcal{F}[\mathbf{W}^h(\mathbf{U})] \cdot \nabla\phi(\mathbf{x}) d\mathbf{x} = 0, \quad \forall \phi \in \{\phi_i\} \tag{5.15}$$

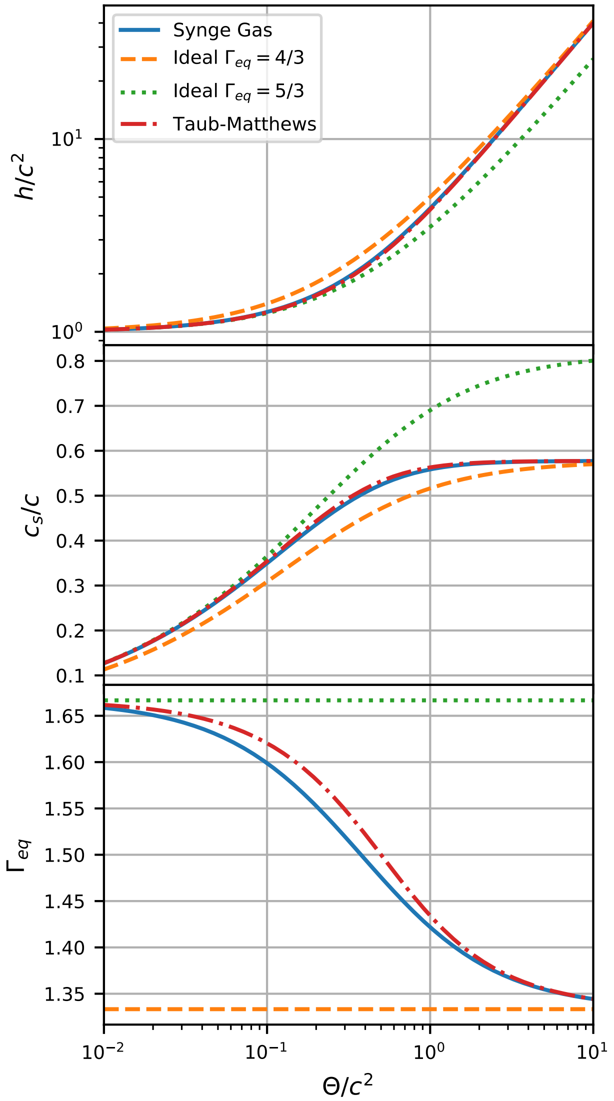


Figure 5.1: Enthalpy (top), sound speed (middle), and equivalent adiabatic index (bottom) as a function of the temperature proxy Θ/c^2 for the Synge gas (solid blue), ideal equation of state with a relativistic $\Gamma = 4/3$ (dashed orange) and a non-relativistic $\Gamma = 5/3$ (finely dashed green), and the Taub-Matthews approximation to the Synge gas (dot-dashed red). With the Synge and Taub-Matthews equations of state, each of the quantities shown here vary smoothly between the two extremes of the ideal equation of state as Θ/c^2 changes from non-relativistic to relativistic. The Taub-Matthews equation of state provides a reasonable approximation to the Synge gas while remaining simple for computation.

on each cell. The second term is the integral of normal flux over the surface of an element. The solution at cell interfaces is double-valued as indicated by the overline; one value corresponding to the data inside the cell, the other from the neighboring cell. As such, the solution is discontinuous and the flux must be computed using a Riemann solver in a fashion similar to the finite volume method. We have implemented two *approximate* Riemann solvers: HLL and HLLC, discussed in §5.2.4. Beyond the choice of Riemann solver, the discrete conservation law, Eq. 5.15 can admit a range of different basis orders. A first order basis (e.g piecewise constant) will eliminate the contribution of $\int_{\Omega_h} \mathcal{F}[\mathbf{W}(\mathbf{U})] \cdot \nabla\phi(\mathbf{x}) d\mathbf{x}$, resulting in a scheme equivalent to a first order finite volume discretization. Moving to higher order bases (e.g. piecewise linear, etc.) will introduce the need to provide additional stabilization (e.g. dissipation) at discontinuities and shocks. For this we use the Moe limiter from Moe et al. (2015) and the minmod limiter (van Leer, 1979) as well as the physicality enforcing operator tailored for relativistic hydrodynamics that we discuss in detail in §5.2.5.

Before the integrals in Eq. 5.15 can be computed, the primitive variables must be calculated for use in the numerical flux. There are different options for computation: interpolate conserved and compute primitives at quadrature points, versus compute primitives at nodal points and interpolate. In Newtonian hydrodynamics, the primitive variables, $\mathbf{W}$, can be recovered algebraically from the conserved state. As such, it is straightforward to interpolate the conserved quantities to the required quadrature point and recover the necessary primitive quantities to construct the flux. In *relativistic* hydrodynamics, such an algebraic recovery of the primitive quantities does not exist; prior work (see e.g. Beckwith & Stone, 2011) has demonstrated that, in the context of finite volume schemes, it is necessary to interpolate *primitive* variables (rather than conserved quantities) in order to ensure that the state remains physical (e.g. $|\mathbf{v}|^2 < c^2$, $\rho > 0$, $P > 0$). Here, we follow a similar procedure: the primitive state is computed from the conserved state at the basis points and then interpolated to quadrature points in order to compute fluxes. In addition to enhanced stability, this minimizes the number of calls to the method that recovers the primitive variables from the conserved state, minimizing the impact that this routine has on overall algorithm performance (see §5.3 for further discussion). Thus, the first step in the assembly is to compute the primitives at nodal points:

$$\mathbf{W}_i = p(\mathbf{U}_i) \tag{5.16}$$

where p computes the primitive variables from the conserved (see Sec. 5.3 for specific details). With this expression, the primitives are easily interpolated to points within the cell using Eq. 5.13, yielding the primitive approximation $\mathbf{W}^h(\mathbf{x}) = \sum_i \mathbf{W_i}\phi_i(\mathbf{x})$. Thus a nonlinear conserved-to-primitive solve is required at each nodal point.

The numerical quadrature for the volumetric contributions of the fluxes are computed as

$$\int_{\Omega_k} \mathcal{F}[\mathbf{W}^h(\mathbf{U})] \cdot \nabla\phi(\mathbf{x})d\mathbf{x} \approx \sum_q w_q \mathcal{F}[\mathbf{W}^h(\mathbf{x}_q)] \cdot \nabla\phi(\mathbf{x}_q) \tag{5.17}$$

and the surface fluxes on the interface shared by Ω_k and $\Omega_{k'}$ are

$$\int_{\partial\Omega_k \cap \partial\Omega k'} \overline{\mathcal{F}[\mathbf{W}^h(\mathbf{U})] \cdot \mathbf{n}}\phi(\mathbf{x})ds \approx \sum_q \omega_q \overline{\mathcal{F}(\mathbf{W}^h_k(\mathbf{x}_q), \mathbf{W}^h_{k'}(\mathbf{x}_q)) \cdot \mathbf{n}}\phi(\mathbf{x}_q). \tag{5.18}$$

Here it is understood that the quadrature rules are defined with respect to the domain of integration. The volumetric term (Eq. 5.17) requires evaluation of the flux at each quadrature point while the surface term (Eq. 5.18) requires evaluation of the numerical flux from cell k and the neighbor k' at each quadrature point.

The temporal discretization we employ uses a multi-stage strong-stability preserving (SSP) Runge-Kutta time integrator similar to that described in Cockburn & Shu (1989); Cockburn et al. (1989, 1990); Cockburn & Shu (1998). SSP time discretization methods were designed to ensure nonlinear stability properties in the numerical solution of spatially discretized hyperbolic partial differential equations, such as Eq. 5.15. These methods assume that there is a time-step, Δt_{FE} such that forward-Euler condition:

$$||\mathbf{U} + \Delta t\mathcal{F}[\mathbf{W}(\mathbf{U})]|| \leq ||\mathbf{U}|| \text{ for } 0 \leq \Delta t \leq \Delta t_{FE} \tag{5.19}$$

is satisfied for all $\mathbf{U}$. An explicit Runge-Kutta (ERK) method is called SSP if the methods can be rewritten as a convex combination of forward Euler methods and the estimate $||\mathbf{U}^{n+1}|| < ||\mathbf{U}^n||$ holds for the numerical solution of Eq. 5.15 whenever the condition given in Eq. 5.19 holds and

$\Delta t \leq C_{SSP} \Delta t_{FE}$, where C_{SSP} is known as the SSP-coefficient. The convex combination above ensures that the strong stability property is also satisfied by the intermediate stages in a Runge-Kutta method (see Gottlieb et al., 2011; Gottlieb, 2015). This may be desirable in many applications, notably in simulations that require positivity (Ferracina & Spijker, 2005, 2004; Higueras, 2004, 2005). In this work, we make use of the second and third order schemes found in Shu & Osher (1989), which were proved to be optimal in Gottlieb & Shu (1998).

5.2.4 Computation of the Surface Flux

The surface flux contributions on the interface shared by Ω_k and $\Omega_{k'}$ require the evaluation of (Eq. 5.18):

$$\sum_q \omega_q \overline{\mathcal{F}(\mathbf{W}_k^h(\mathbf{x}_q), \mathbf{W}_{k'}^h(\mathbf{x}_q)) \cdot \mathbf{n}} \phi(\mathbf{x}_q) \tag{5.20}$$

In the method presented here, this is accomplished by use of an approximate Riemann solver, of which we have implemented the relativistic HLL and HLLC variants due to Schneider et al. (1993) and Mignone & Bodo (2005). Both of these approximate Riemann solvers require an estimate of the maximum and minimum wavespeeds on either side of the interface, which we compute through the maximum and minimum eigenvalues of $\partial \mathbf{F}/\partial \mathbf{U}$ (Mignone & Bodo, 2005):

$$\lambda_{\pm}(\mathbf{W}) = \frac{v_x \pm \sqrt{\sigma_s \left(c^2 - v_x^2 + c^2 \sigma_s\right)}}{1 + \sigma_s} \tag{5.21}$$

where

$$\sigma_s = c_s^2 / \left[\gamma^2 \left(c^2 - c_s^2\right)\right]. \tag{5.22}$$

We compute $\lambda_{\pm}(\mathbf{W})$ for every $\mathbf{W}_k^h(\mathbf{x}_q)$ and $\mathbf{W}_{k'}^h(\mathbf{x}_q))$ to find the maximum and minimum wavespeeds at each surface quadrature point across interface:

$$\lambda_L = \min\left(\lambda_- \left(\mathbf{W}_k^h(\mathbf{x}_q)\right), \lambda_- \left(\mathbf{W}_{k'}^h(\mathbf{x}_q)\right)\right) \tag{5.23}$$

$$\lambda_R = \max\left(\lambda_+ \left(\mathbf{W}_k^h(\mathbf{x}_q)\right), \lambda_+ \left(\mathbf{W}_{k'}^h(\mathbf{x}_q)\right)\right). \tag{5.24}$$

5.2.5 Physicality Enforcing Operator

While using 0^{th} order polynomials for a relativistic hydrodynamics discontinuous-Galerkin method is guaranteed to produce a physical conserved state after every flux update even with shocks when using a local-extremum-diminishing numerical fluxes such as HLL, higher order bases can introduce spurious oscillations and non-physical conserved states within cells around shocks (see Wu & Tang (2016)). To resolve this issue, an operator is needed to smooth the solution within a cell. Taking inspiration from the limiter presented in Moe et al. (2015), we present here a smoothing procedure that enforces physical conserved states within a cell with a physical volume average.

Following Riccardi & Durante (2008) and Wu & Tang (2016), a conserved state that satisfies

$$D > 0, \quad q\left(\mathbf{U}\right) \equiv E/c^2 - \sqrt{D^2 - |\mathbf{M}/c|^2} > 0, \tag{5.25}$$

is a physically admissible state as long as the specific energy $e(\rho, p)$ is continuously differentiable under the chosen equation of state. If a conserved state satisfies Eq. 5.25, the state can be inverted for a primitive state with positive density and pressure with a velocity less than c. Since the space of permissible conserved states under Eq. 5.25 is convex (i.e. any conserved state interpolated between two physically permissible conserved states is also physically permissible (Wu & Tang, 2016)), we can use the same strategies from Moe et al. (2015) in a simple smoothing procedure to enforce physicality within a discontinuous-Galerkin cell. From a high level, we apply an operator to average nodal points within a cell towards a physical volume average.

Before enforcing physicality within cells, we first screen for cells with non-physical nodal points by checking that all conserved states at the nodal points – $\mathbf{U}_i$ – satisfy Eq. 5.25. If any point fails, we flag the cell as needing smoothing to ensure that all points are physical. We then check that the cell volume average $\overline{\mathbf{U}}$ of the conserved state satisfies Eq. 5.25. As long as the cell volume average is physical, a smoothing factor can be found that ensures physicality without changing the global conserved quantities. If the cell volume average is not physical, then the nodal points cannot be made physical through the physicality-enforcing operator without changing the volume average.

To enforce physicality within a cell, we first seek a smoothing factor $s \in [0,1]$ such that the smoothed states

$$\tilde{\mathbf{U}}_i = s\mathbf{U}_i + (1-s)\,\overline{\mathbf{U}} \tag{5.26}$$

at all nodal points in the cell satisfy Eq. 5.25. At each point in the cell, we find the largest smoothing factor such that

$$\tilde{D}_i > 0 \quad \tilde{q}_i \equiv \tilde{E}_i/c^2 - \sqrt{\tilde{D}_i^2 + \left(|\tilde{\mathbf{M}}_i|/c\right)^2} > 0. \tag{5.27}$$

If we assume that $\overline{\mathbf{U}}$ is physical, then $s := 0$ would lead to a physical $\tilde{\mathbf{U}}$, so we can assume that such a smoothing factor $s_i \geq 0$ exists. We find this factor in two stages.

In the first stage, we compute an intermediate stage smoothing factor $s_i^{(1)}$ for each nodal point that ensures a positive D and E. We solve

$$\tilde{D}_i^{(1)} = s_{i,D}^{(1)} D_i + \left(1 - s_{i,D}^{(1)}\right)\overline{D} > 0 \tag{5.28}$$

$$\tilde{E}_i^{(1)} = s_{i,E}^{(1)} E_i + \left(1 - s_{i,E}^{(1)}\right)\overline{E} > 0 \tag{5.29}$$

for the largest $s_{i,D}^{(1)}, s_{i,E}^{(1)} \in [0,1]$ that satisfies the constraints and compute an intermediate smoothing factor $s_i^{(1)} = \min\left(s_{i,D}^{(1)}, s_{i,E}^{(1)}\right)$. We use $s_i^{(1)}$ to compute an intermediate smoothed state

$$\tilde{\mathbf{U}}_i^{(1)} = s_i^{(1)}\mathbf{U}_i + \left(1 - s_i^{(1)}\right)\overline{\mathbf{U}} \tag{5.30}$$

so that we ensure that $\tilde{D}$ and $\tilde{E}$ are positive.

In the second stage, we compute a second stage smoothing factor $s_i^{(2)} \in [0,1]$ such that

$$\tilde{q}_i^{(2)} = \tilde{E}_i^{(2)}/c^2 - \sqrt{\left(\tilde{D}_i^{(2)}\right)^2 + \left(|\tilde{\mathbf{M}}_i^{(2)}|/c\right)^2} > 0. \tag{5.31}$$

where $\tilde{\mathbf{U}}_i^{(2)} = s_i^{(2)}\mathbf{U}_i^{(1)} + \left(1 - s_i^{(2)}\right)\overline{\mathbf{U}}$ is the second smoothed state. Note that since $s^{(2)} := 0$ leads to $\tilde{\mathbf{U}}^{(2)} := \overline{\mathbf{U}}$, we know that an acceptable smoothing factor exists. Solving Eq. 5.31 can be simplified by noting that $\tilde{E}^{(2)}$ is positive for any choice of $s_i^{(2)} \in [0,1]$ since $\tilde{E}^{(1)}$ and $\overline{E}$ are both positive (for the same reasons, $\tilde{D}^{(2)}$ is also always positive). We can rewrite Eq. 5.31 as

$$\left(\tilde{E}_i^{(2)}/c^2\right)^2 > \left(\tilde{D}_i^{(2)}\right)^2 + \left(|\tilde{\mathbf{M}}_i^{(2)}|/c\right)^2 \tag{5.32}$$

$$a\left(s_i^{(2)}\right)^2 + b s_i^{(2)} + c > 0 \tag{5.33}$$

where

$$a = \frac{1}{c^4}\left(\tilde{E}_i^{(1)} - \overline{E}\right)^2 - \left(\tilde{D}_i^{(1)} - \overline{D}\right)^2 - \frac{1}{c^2}\left|\tilde{\mathbf{M}}_i^{(1)} - \overline{\mathbf{M}}\right|^2 \tag{5.34}$$

$$b = \frac{2}{c^4}\overline{E}\left(\tilde{E}_i^{(1)} - \overline{E}\right) - 2\overline{D}\left(\tilde{D}_i^{(1)} - \overline{D}\right) - \frac{2}{c^2}\overline{\mathbf{M}}\cdot\left(\tilde{\mathbf{M}}_i^{(1)} - \overline{\mathbf{M}}\right)^2 \tag{5.35}$$

$$c = \frac{1}{c^4}\overline{E}^2 - \overline{D}^2 - \frac{1}{c^2}\left|\overline{\mathbf{M}}\right|^2 . \tag{5.36}$$

Since $\overline{\mathbf{U}}$ is physical, $s_i^{(2)} := 0$ must satisfy the inequality. Note that the quadratic can only have at most one root within [0,1]; if it had two roots, then either $s_i^{(2)} := 0$ and $s^{(2)}$ do not satisfy the inequality, implying that $\overline{\mathbf{U}}$ is unphysical, or that both satisfy the inequality and that some interior $s_i^{(2)} \in [0, 1]$ do not satisfy the inequality, implying that the space of physical conserved states is not convex, both of which are contradictions. If there are no roots within $[0, 1]$, since $s_i^{(2)} := 0$ satisfies the inequality, $s_i^{(2)} := 1$ must as well, so 1 would be the largest acceptable second stage smoothing factor.

In the case that there is just one root, then since $s_i^{(2)} := 0$ satisfies the inequality, the coefficient a must be negative or 0 (which is the simple linear case), and only the root

$$s_i^{(2)} = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{5.37}$$

can fall within $[0, 1]$, and so we only need to compute this root to find the largest smoothing factor for this nodal point. The final smoothing factor for this nodal point is $s_i = s_i^{(1)} s_i^{(2)}$, which ensures that any $s \leq s_i$ chosen will satisfy Eq. 5.27. After computing s_i for each nodal point in the cell, we compute the final smoothing factor for the cell using $s = \min s_i$, which we use to compute $\tilde{\mathbf{u}}$ using Eq. 5.26.

The procedure for our physicality-enforcing operator goes as follows

1. We flag cells with nodal points with conserved states that violate Eq. 5.25 as cells with non-physical nodal points.

2. We check that the volume average within a flagged cell satisfies equation 5.25, which guarantees that the smoothing procedure will enforce physicality within the cell.

3. For each point in a flagged cell, we compute the largest smoothing factor s_i that will guarantee that the new smoothed state will satisfy Eq. 5.27. For each nodal point, the procedure goes as:

   a) We compute the first stage smoothing factor $s_{i,D}^{(1)}$ and $s_{i,E}^{(1)}$ to ensure positivity of D and E by solving for them in Eq. 5.28.

   b) We compute the first stage smoothing factor $s_i^{(1)} = \min s_{i,D}^{(1)}, s_{i,E}^{(1)}$ and use this to compute the intermediate smoothed state $\tilde{\mathbf{U}}^{(1)}$ using Eq. 5.30.

   c) We then check whether $\tilde{\mathbf{U}}^{(1)}$ satisfies equation 5.25, in which case we use $s_i = s_i^{(1)}$.

   d) If not, we compute $s_i^{(2)}$ by solving the quadratic described in Eq. 5.32 and Eq. 5.34 using the root for $s_I^{(2)}$ in Eq. 5.37. The smoothing factor for this nodal point is then $s_i = s_i^{(1)} s_i^{(2)}$.

4. We compute a final smoothing factor for each cell using $s = \min s_i$, which allows us to compute the smoothed state $\mathbf{U}_i$ at each nodal point using Eq. 5.26.

As long as the volume average conserved state $\overline{\mathbf{U}}$ is physical, this procedure will produce the physical conserved state $\tilde{\mathbf{U}}_i$.

5.3 Recovery of Primitive Variables

Although the conservation laws in relativistic hydrodynamics are similar to those in Newtonian hydrodynamics, the inclusion of the Lorentz factor in conservation of mass, momentum, and energy adds complexity to the equation set in several ways that complicate recovery of primitive variables from conserved variables. Primarily, the Lorentz factor couples every conserved variable with the velocity in all directions. While adding a transverse velocity to a non-relativistic flow will not affect longitudinal evolution, in demonstration of Galilean invariance, a transverse velocity in a relativistic flow contributes to the apparent density, momentum, and energy, fundamentally modifying the dynamics. Additionally, the inclusion of the Lorentz factor leads to a non-linear relationship between the primitive and conserved variables. For even simple choices of equation of state, recovering the primitive state from the conserved state (i.e. inverting Eq. 5.4) requires finding the roots of cubic or higher order polynomials. Last, the relativistic hydrodynamics equations (and causality) require the three-velocity to be bounded by the speed of light, with superluminal velocities leading to complex Lorentz factors. For highly relativistic flows close to the speed of light, we are often limited by machine precision when representing small changes in the three-velocity that equate to large changes in the Lorentz factor. For these reasons, the stability and fidelity of any scheme for relativistic hydrodynamics is fundamentally tied to that of the scheme used to compute primitive variables from conserved quantities. As a result, a wide variety of schemes, including but not limited to those presented in Schneider et al. (1993); Ryu et al. (2006); Riccardi & Durante (2008), have been described in the literature. Each of these options has its advantages and disadvantages from a physical fidelity, stability, and robustness standpoint; however, as far as we are aware, the performance of these different formulations has not previously been examined from a performance portability perspective, as we do here.

We consider two different approaches to recovering the primitive variables from conserved quantities: an analytical approach and an iterative approach. We then develop both of these methods for the ideal gas and Taub-Matthews equations of state to give four algorithms in all. In formulating these, we use the dimensionless variables

$$\xi = \frac{M}{Dc} \quad \text{and} \quad \eta = \frac{E}{Dc^2}. \tag{5.38}$$

This rescaling aids with reducing issues due to large differences in numbers, although this does not eliminate issues of near-speed-of-light velocities.

5.3.1 Ideal Gas Equation of State

In the case of the ideal gas equations of state, the primitive variables can be recovered from the conserved quantities by solving the roots of a quartic equation. One approach demonstrated by Ryu et al. (2006) computes the analytic solution to a quartic polynominal in $\beta = v/c$. For completeness, we restate this method here in terms of the dimensionless parameters ξ and η, which allows us to keep c throughout the set of equations.

As shown in Schneider et al. (1993), the solution for the special relativistic velocity β can be found from the roots of the quartic polynomial

$$a_3\beta^4 + a_2\beta^2 + a_1\beta + a_0 = 0 \tag{5.39}$$

where the coefficients are given by

$$a_3 = \frac{-2\Gamma(\Gamma-1)\xi\eta}{(\Gamma-1)^2(\xi^2+1)} \tag{5.40}$$
$$a_2 = \frac{\Gamma^2\eta^2 + 2(\Gamma-1)\xi^2 - (\Gamma-1)^2}{(\Gamma-1)^2(\xi^2+1)} \tag{5.41}$$
$$a_1 = \frac{-2\Gamma\xi\eta}{(\Gamma-1)^2(\xi^2+1)} \tag{5.42}$$
$$a_0 = \frac{\xi^2}{(\Gamma-1)^2(\xi^2+1)}. \tag{5.43}$$

Only one root of the polynomial provides a physical $\beta \in [0, 1)$. The root can be found using a root-finding method or analytically (Ryu et al., 2006) through:

$$\beta = \frac{-B + \sqrt{B^2 - 4C}}{2} \tag{5.44}$$

where

$$B = \frac{1}{2}\left(a_3 + \sqrt{a_3^2 - 4a_2 + 4x}\right) \tag{5.45}$$
$$C = \frac{1}{2}\left(x - \sqrt{x^2 - 4a_0}\right) \tag{5.46}$$

We then have that:

$$x = \begin{cases} 2\left(R^2+T\right)^{2/3}\cos\left[\frac{1}{3}\tan^{-1}\left(\frac{\sqrt{-T}}{R}\right)\right] - i_1/3 & \text{if } T < 0 \\ \left(R+\sqrt{T}\right)^{1/3} + \left(R-\sqrt{T}\right)^{1/3} - i_1/3 & \text{otherwise} \end{cases} \tag{5.47}$$

where R, S, and T are found from

$$R = \frac{1}{54}\left(9i_2i_2 - 27i_3 - 2i_1^3\right) \tag{5.48}$$
$$S = \frac{1}{9}\left(3i_2 - a_2^2\right) \tag{5.49}$$
$$T = R^2 + S^3 \tag{5.50}$$

where

$$i_1 = -a_2 \tag{5.51}$$

$$i_2 = a_3 a_1 - 4a_0 \tag{5.52}$$

$$i_3 = 4a_2 a_0 - a_1^2 - a_3^2 a_0. \tag{5.53}$$

$$\tag{5.54}$$

With a solution for β, the rest of the primitive variables can be recovered using

$$\rho = D\sqrt{1-\beta^2} \tag{5.55}$$

$$\mathbf{v} = \frac{\beta}{\xi D}\mathbf{M} \tag{5.56}$$

$$P = (\Gamma - 1)\left(E - \mathbf{M}\cdot\mathbf{v} - \rho c^2\right). \tag{5.57}$$

An alternative strategy for recovering the primitive variables from conserved quantities is to utilize an iterative solver to find the roots. Exploring the iterative approach, we used an iterative solver following the recovery method presented in Riccardi & Durante (2008). This solver has two main advantages. First it uses a proxy for the velocity that scales more evenly from weakly to highly relativistic flows. Second, the resulting quartic polynomial can be solved using the Newton-Raphson method, which it typically more robust, accurate, and faster even using several iterations due to avoiding the slow and imprecise square roots and inverse tangents in the analytic solver.

Instead of recovering the primitives by solving for velocity, Lorentz factor, or pressure, we instead solve for a proxy of the velocity, w, where

$$u = \frac{2w}{1+w^2}. \tag{5.58}$$

We solve for $w \in (0,1)$ by finding the root within $(0,1)$ of the quartic polynomial

$$P(w) = (\alpha - 1)\,\xi w^4 - 2\,(\alpha\eta + 1)\,w^3 + 2\,(\alpha + 1)\,\xi w^2 - 2\,(\alpha\eta - 1)\,w + (\alpha - 1)\,\xi, \tag{5.59}$$

where $\alpha = \Gamma/(\Gamma - 1)$. Within the range $w \in (0,1)$, the equation $P(w) = 0$ has only one root. While $P(w) = 0$ could be solved analytically using the same method for our analytical solver, the Newton-Raphson method is simpler and often quicker, since it only requires addition and multiplication and coefficients of the polynomial can be reused across iterations. We also find that the Newton-Raphson method always converges to the root in $(0, 1)$ as long as the initial guess is in $(0, 1)$, which is consistent with Riccardi & Durante (2008). This obviates the need for a bounded root solver. For reasonably relativistic flows with $\gamma < 10$, this may only take 5 iterations to recover w to within double floating point machine precision ($\Delta w \sim 10^{-16}$).

When ξ is very small, a cubic approximation for a solution for w can be used

$$w = \frac{\alpha - 1}{2\left(\alpha\eta - 1\right)}\xi + \frac{(\alpha - 1)^2}{8\left(\alpha\eta - 1\right)^4}\left[\left(\alpha + 3\right)\left(\alpha\eta + 1\right) - 4\left(\alpha + 1\right)\right]\xi^3 + O(\xi^5). \tag{5.60}$$

Generally, the iterative solver for the ideal equation of state is more accurate than the analytical solver. Often, the iterative solver is also faster. Comparison between the solvers for the ideal equation of state and the solvers for the Taub-Matthews equation of state are explored in section 5.3.3.

5.3.2 Taub-Matthews Equation of State

For the Taub-Matthews equation of state, the primitive state can be recovered from the conserved state by solving a cubic equation for $W = \gamma^2 - 1$. Following Ryu et al. (2006), we solve for W from

$$W^3 + c_1 W^2 + c_2 W + c_3 = 0 \tag{5.61}$$

where

$$c_1 = \frac{\left(\eta^2 + \xi^2\right)\left[4\left(\eta^2 + \xi^2\right) - \left(\xi^2 + 1\right)\right] - 14\xi^2\eta^2}{2\left(\eta^2 - \xi^2\right)^2} \tag{5.62}$$

$$c_2 = \frac{\left[4\left(\eta^2 + \xi^2\right) - \left(\xi^2 + 1\right)\right]^2 - 57\xi^2\eta^2}{16\left(\eta^2 - \xi^2\right)^2} \tag{5.63}$$

$$c_3 = -\frac{9\xi^2\eta^2}{16\left(\eta^2 - \xi^2\right)^2}. \tag{5.64}$$

Eq. 5.61 can be solved analytically and iteratively. Analytically solving the cubic polynomial is straightforward compared to solving the quartic polynomial for the ideal equation of state. The solution for W depends on the discriminant of the cubic equation

$$d = Q^3 + R^2 \tag{5.65}$$

with

$$Q = \frac{1}{9}\left(3c_2 - c_1^2\right) \tag{5.66}$$

$$R = \frac{1}{54}\left(9c_1c_2 - 27c_3 - 2c_1^3\right). \tag{5.67}$$

$$\tag{5.68}$$

If $d < 0$, then Eq. 5.61 has the solution

$$W = 2\sqrt{-Q}\cos\left(\frac{\iota}{3}\right) - \frac{c_1}{3} \tag{5.69}$$

with

$$\iota = \cos^{-1}\left(\frac{R}{\sqrt{-Q^3}}\right). \tag{5.70}$$

Otherwise if $d \geq 0$, then Eq. 5.61 has the solution

$$W = -\frac{c_1}{3} + S + T \tag{5.71}$$

with

$$S = \left(R + \sqrt{d}\right)^{1/3} \tag{5.72}$$

$$T = \left(R - \sqrt{d}\right)^{1/3}. \tag{5.73}$$

A root-finding method can also be used to recover W from Eq. 5.61. As an alternative option to the analytic solution, we use the bracketed root solver Brent's method (Brent, 1973) to recover W. For the Taub-Matthews equation of state, we use Brent's method instead of the Newton-Raphson since Brent's method allows us to bracket the one non-negative root. Unlike for the quartic polynomial solved for the ideal equation of state, the Newton-Raphson method is not guaranteed to converge to the positive root when using a positive initial guess, which leads to an incorrect and unphysical recovered velocity. We first bracket the root W with the region corresponding to

$\gamma \in [1, 200]$, then iteratively expand the upper range if the root is not found. For the tests explored here $\gamma = 200$ is a sufficiently high upper bound that this rebracketing is not needed.

With W recovered, the Lorentz factor and relativistic velocity can be recovered via

$$\gamma = \sqrt{W+1} \qquad \beta = \sqrt{\frac{W}{W+1}}. \tag{5.74}$$

The lab frame density ρ and velocity $\mathbf{v}$ can be recovered via the same method as the ideal equation of state. The pressure with the Taub-Matthews equation of state is recovered via

$$P = \frac{(E - \mathbf{M}\cdot\mathbf{v})^2 - \rho^2}{3\,(E - \mathbf{M}\cdot\mathbf{v})}. \tag{5.75}$$

5.3.3 Conserved to Primitive Solver Comparisons

Fig. 5.2 shows the relative error in the recovered velocity in the ideal gas equation of state and Taub-Matthews equations of state using the analytical method and iterative methods using varying number of iterations. The plots are created by applying the methods on a grid of 25^2 primitive states with $D = 1$ kg m^{-3} and 25 logarithmically spaced pressures from 10^5 to 10^{10} N m^{-2} and 25 logarithmically spaced Lorentz factors from 1 to 100, using $c = 3 \times 10^8$m s^{-1}. Each pair of pressure and Lorentz factor is converted to a conserved state using Eq. 5.4 that is converted back to a primitive state using the specified recovery method. We then compute the relative error of the velocity in the recovered primitive state to the original velocity determined by the Lorentz factor.

For the ideal gas using 64 bits of floating precision, the analytical solver recovers the velocity to 10^{-15} for Lorentz factors below 3 and in some cases recovering it exactly due to machine precision (10^{-16} in this regime). The accuracy of the analytical method decreases roughly as a power law with increasing Lorentz factor, reaching about 10^{-10} at $\gamma = 100$. At this high Lorentz factor, the relative error in recovered Lorentz factor is 10^{-6}, which propogates into other recovered primitives, highlighting the need to accurate recovery of velocity for ultrarelativistic flows. In contrast, the iterative method for the ideal gas recovers the velocity exactly or near machine precision for Lorentz factors below 10 in only 6 iterations, past which the error increases rapidly with Lorentz factor. Owing to the flexibility of the accuracy of the iterative method, increasing the iteration count to

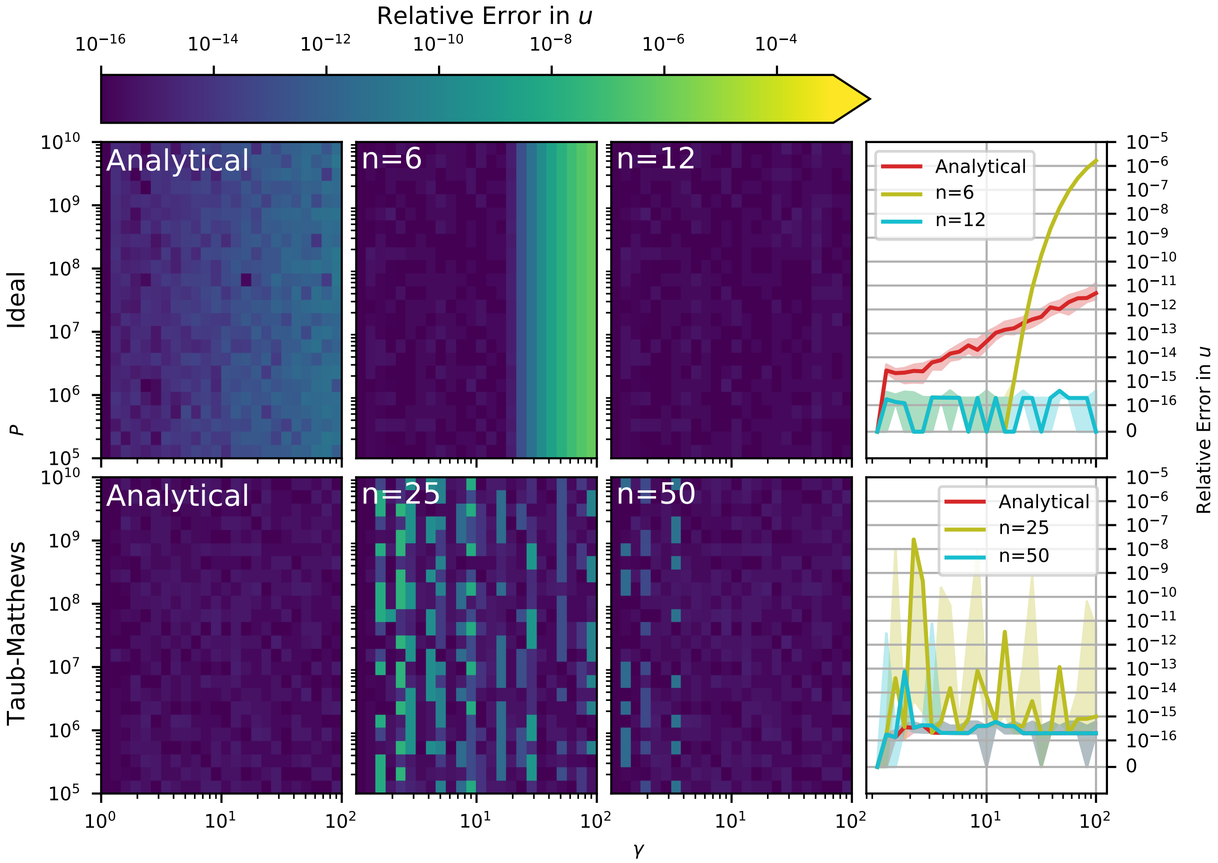


Figure 5.2: Map of the error of the conserved-to-primitive solvers with the error using the analytical method in the left column and using varying numbers of iterations in the middle two columns and error of these configurations versus Lorentz factor in the right column. The top row shows results for the ideal gas, testing the iterative solver with 6 and 12 iterations, and the bottom row shows results for the Taub-Matthews equation of state, testing the iterative solver using 25 and 50 iterations. In all panels, 25 × 25 primitive states are tested with Lorentz factors varying from 1 to 100 on the x-axis and pressures varying from 10^5 to 10^{10} N m^{-2}, using $c = 3 \times 10^8$ m s^{-1} and fixing $D = 1$ kg m^{-3}, these primitive states are first converted to conserved states and then converted back to a primitive state using the specified analytical or iterative solver. In the left three columns, the relative error is shown in color with the y-axis showing the pressure. In the rightmost column, the median (solid line) and first to third quartile (shared region) of the error sampled using different pressures given a specific Lorentz factor. All results in this figure are using the Intel compiler on CPUs. The iterative solver for the ideal equation of state is more accurate than the analytic solver using just 12 iterations for high Lorentz factors and just 6 iterations for low Lorentz factors. For the Taub-Matthews equation of state, the analytical solver is almost always at least or more accurate than the iterative solver.

12 leads to recovering the velocity near machine precision for all Lorentz factors tested. At higher Lorentz factors, the iterative solver has relatively more difficulty in recovering the velocity due to

the method recovering the velocity from a proxy of the velocity and the slow variation of velocity at high Lorentz factors. Small errors in the recovered velocity at high Lorentz factors amplify to large errors in other recovered primitives. We also note that for very high pressures at and above $10^{20}\rho c^2$, analytical method for the ideal gas encounters imaginary numbers and fails to recover the velocity at all, whereas the iterative solver does not fail with very high pressures.

In comparison, the cubic analytic solver for the Taub-Matthews equation of state performs closer to machine precision across the domain of primitive states tested. The iterative solver for the Taub-Matthews equation of state requires many more iterations than for the ideal gas equation of state. We attribute this to the construction of the polynomial for the iterative solver for the ideal equation of state, which is designed to converge in a few iterations. The Taub-Matthews equation of state iterative solver performs worse at lower Lorentz factors since it recovers the velocity from a proxy of the Lorentz factor, and the Lorentz factor varies slowly at low velocities. Small errors in the recovered Lorentz factor at sub-relativistic velocities amplify to large errors in other recovered primitives. Generally, the iterative solver for the Taub-Matthews equation of state is less accurate than the analytical solver, and the high iteration counts required lead to slower performance.

We next investigate the number of iterations required for the iterative solver to reach accuracy parity with the analytic solver in Fig. 5.3. In this figure, we test the same grid of primitive states used in Fig. 5.2, running the iterative solver with increasing number of iterations until it achieves greater accuracy than the analytic solver. For some cases with the ideal gas, the analytic solver recovers the velocity exactly, which we mark with yellow.

The number of iterations required for the iterative solvers to reach accuracy parity depends mostly on the Lorentz factor with some variation in pressure. The iterative solver for the ideal gas requires more iterations at higher Lorentz factors. We attribute this to the iterative solver recovering the primitive state by first recovering a proxy for the velocity instead of Lorentz factor, which requires less precision to recover at low Lorentz factors. For the primitives states tested here that the analytical solver does not recover exactly, the ideal iterative solver requires fewer than 10 iterations to achieve parity. We attribute the low iteration count to the one physical root of the

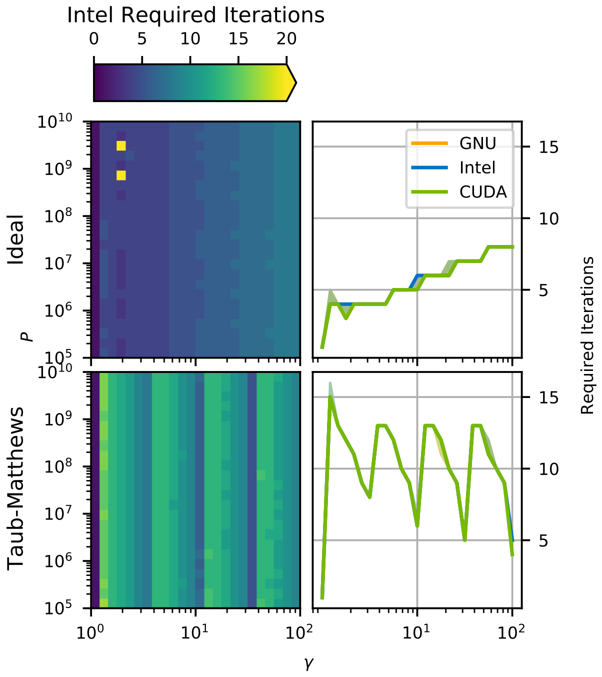


Figure 5.3: Required iterations for the iterative solver to reach the same accuracy as the analytical solver using the same primitive states as Fig. 5.2, with results for the ideal gas in the top row and the Taub-Matthews equation of state in the bottom row. The left column shows the required iterations when compiling with the Intel compiler in color with Lorentz factor on the x axis and pressure on the y axis. For two primitive states the ideal analytic solver recovers the velocity exactly, leading the iterative solver being unable to reach the same accuracy, which we show in yellow. The right column shows the median (solid line) and first to third quartile (shared region) of the error sampled using different pressures given a specific Lorentz factor, Results with the GNU compiler on CPUs are shown in orange, with the Intel compiler on CPUs with the Kokkos OpenMP backend in blue, and with the Kokkos CUDA backend on GPUs in green.

quartic always being the same root.

The iterative solver required comparatively more iterations, almost always more than 5 and upwards of 15 for low Lorentz factors. Generally more iterations are required for lower Lorentz factors, possibly due to the solver recovering a proxy of the Lorentz factor first, from which recovering the velocity is sensitive to precision. The required iterations form a sawtooth with Lorentz factors due to the physical root switching positions.

Depending on the architecture and compiler, the iterative solver for the ideal gas is usually faster than the analytic solver, while for the Taub-Matthews equation of state the iterative solver is almost always slower. We investigate the performance of the recovery methods in Fig. 5.4. Using the same

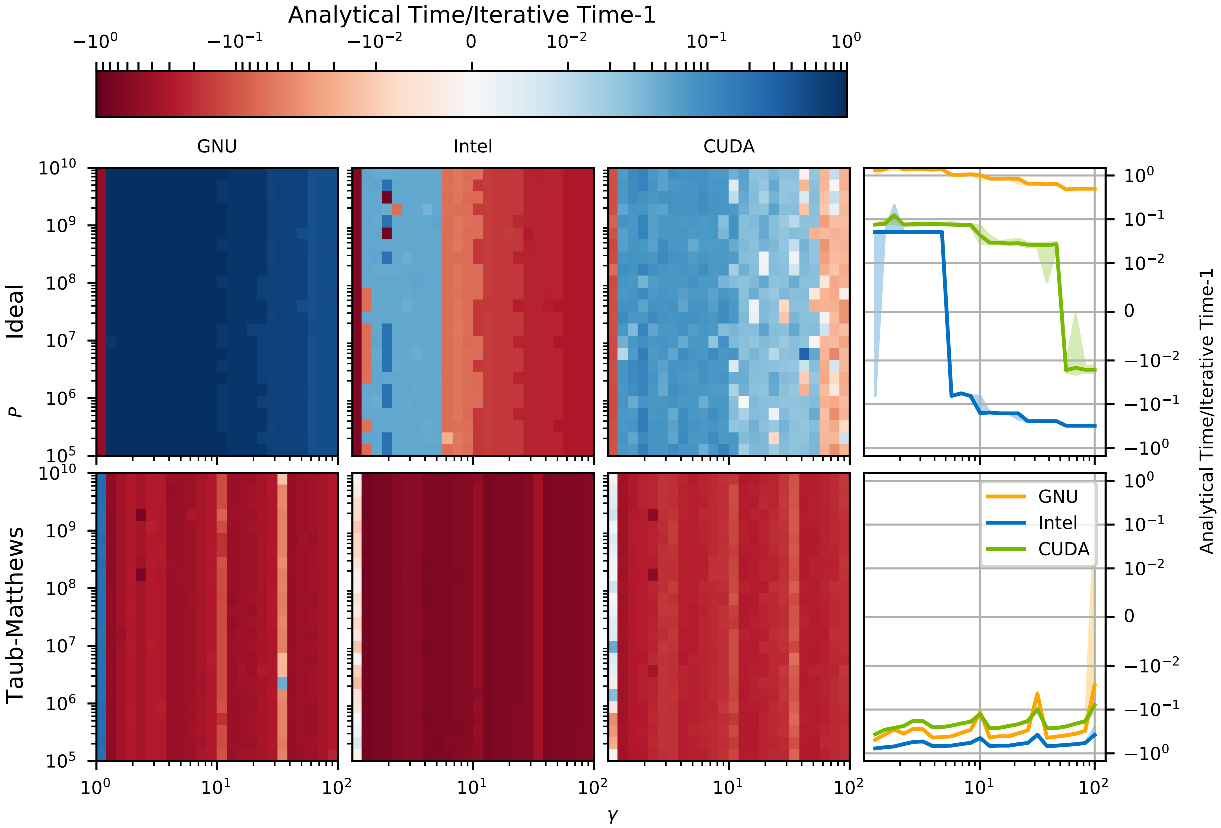


Figure 5.4: Timing comparisons for the iterative solver to reach the same accuracy as the analytic solver, with comparisons as a color map in the left three panels and versus Lorentz factor in the rightmost panel, using the same primitive states as Fig. 5.2 with results for the ideal gas in the top row and the Taub-Matthews equation of state in the bottom row. In all panels we compare results using the metric Analytical Time/Iterative Time − 1, where a positive value shows how much slower the analytical solver is as a fraction of the time the iterative solver takes and a negative value shows the fraction by which the analytical solver is faster. The left three columns show the timing metric in color (blue shows where the iterative method is faster) with the Lorentz factor on the x and the pressure on the y axis, showing comparisons for the GNU and Intel compilers on CPUs with the Kokkos OpenMP backend and on GPUs with the Kokkos CUDA backend across the three columns. The rightmost column shows the median (solid line) and first to third quartile (shared region) of the error sampled using different pressures given a specific Lorentz factor, showing results for all compilers tested (note that this does not compare timings between compilers, only the analytic against the iterative solver for each compiler). For the ideal equation of state, the iterative solver is faster than the analytic solver under a certain threshold of Lorentz factor that is compiler and architecture dependent. The iterative solver for the Taub-Matthews equation of state is almost always slower than the analytic method.

grid of primitive states that we used in Fig. 5.2, we compare the run times of the analytical solvers and iterative solvers with the number of iterations required to achieve accuracy parity, running each of the primitive states from Fig. 5.2 on 10^3 cells with 27 points per cell, taking an average runtime over 100 runs each. We compare timings using the metric Analytical Time/Iterative Time − 1, where the iterative time is with the number of iterations required to match the analytical accuracy, in order to highlight where the iterative solver is faster. Negative values show the fraction by which the analytical method is faster than the iterative method while positive values show the fraction by which the analytical solver is slower.

For the ideal gas on CPUs using the Intel compiler, the iterative solver is about 10% faster than the analytical solver at Lorentz factors below 10 and about 10% slower at Lorentz factors above 10. For higher iteration counts reaching to 10 iterations, the analytical solver begins to be faster than the iterative solver by several percent. However, it should be noted from Fig. 5.2 that in this regime the analytical method introduces more inaccuracy to the primitive state, while the iterative solver can recover the primitive state with much better accuracy at the cost of performance. A red line on the right hand side shows that the analytical solver more quickly identifies the zero velocity case, whereas the iterative solver takes longer due the layout of the code and using the cubic approximation from Eq. 5.60 for near-zero momenta.

Using the GNU compiler on CPUs, the iterative solver is always faster than the analytical solver except for trivial cases. We attribute this slowdown with GNU to the slower math functions required in the analytic solver.

For GPUs, the iterative solver for the ideal gas is faster than the analytical solver by several percent for all but the trivial case and Lorentz factors above 60. This is despite the potential for the kernel to branch at every point if different points require different numbers of iterations, although these timing tests do not exercise this possibility. The timing disparity may be due to the ‘sqrt‘ operation in the analytical solver, which is more optimized on CPUs compared to GPUs.

Considering the Taub-Matthews equation of state, the iterative solver is almost always slower than the analytical solver. This is expected from the larger number of iterations needed for the iterative solver to reach parity with the analytical solver. The performance difference is largest on the Intel compiler, where the optimized math functions allow good performance for the analytical solver.

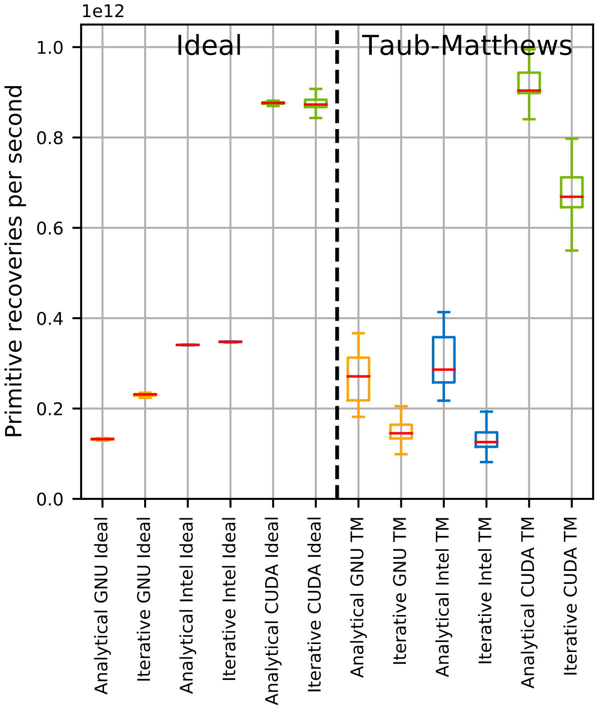


Figure 5.5: Aggregate performance of all methods and compilers tested shown as box and whiskers of the primitive recoveries per second (higher is better) across the grid of primitive states used in Fig. 5.2. Red lines show medians, boxes show the interquartile range, and whiskers show the maximum and minimum values inside of 1.5 times the length of the interquartile range above the 3rd quartile and below the 1st quartile, described by Tukey (1977). We exclude outlier timings from the figure, which range from 10^{11} to 1.2×10^{12} primitive recoveries per second for all methods and compilers. We show results for GNU on CPUs in orange, Intel on CPUs in blue, and CUDA on GPUs in green, for the ideal gas on the left and the Taub-Matthews equation of state on the right. Generally, on CPUs using the Intel compiler allows more primitive recoveries per second than the GNU compiler. The performance for recovery with the Taub-Matthews gas has a much larger spread than recovery with the ideal equation of state. Between the two equations of state, the solvers achieve roughly the same number of recoveries per second on each architecture, indicating that equation of state can have a mitigated impact on the full code's performance.

In Fig. 5.5 we show performance of all methods on all architectures and compilers tested as a box and whisker plot of the attained primitive recoveries per second. Runs on CPUs with GNU and Intel and the Kokkos OpenMP backend were performed on 2-socket node with Intel Xeon Platinum 8268 CPUs on a total of 48 OpenMP threads compiled with AVX512 vectorization. Runs with the Kokos CUDA backend were performed on an NVidia V100 SXM2 Tesla GPU. For the ideal gas, the analytic method is slower than the iterative method on GNU, slightly faster on Intel, and nearly the same performance on GPUs. For the Taub-Matthews approximation to the Taub-Matthews equation of state, the analytical method is generally faster on all architectures, with the performance difference being the greatest on Intel and the smallest on GNU. Between the two equations of states, the analytical solver for both gases performs at about the same speed for each architecture. This suggests that just considering conserved-to-primitive updates, using a Taub-Matthews equation of state is about as fast as using an ideal equation of state, although the more complex computation of wavespeeds and enthalpies in the Taub-Matthews equation of state will lead to slowdowns elsewhere.

Overall, these results demonstrate that, for the ideal gas equation of state, the iterative method to recover the primitive variables from the conserved variables is more flexible, robust, accurate, and in some cases faster than the analytical method. By contrast, for the Taub-Matthews equation of state, the characteristics of the analytic and iterative solver are nearly the opposite, with the iterative solver performing generally worse. Nevertheless, the comparable speed and robustness of the analytical solver for the Taub-Matthews equation of state suggest that the higher fidelity of the Taub-Matthews equation of state comes at little cost to execution time and stability.

5.4 Tests of the Relativistic Hydrodynamics Scheme

To verify the accuracy of the relativistic hydrodynamics scheme, we investigate several standard test problems in 1D and 2D with and without shocks. First, in §5.4.1, we demonstrate convergence of a set of relativistic linear waves in three-dimensions. We then demonstrate the accuracy of the method for discontinuous solutions in §5.4.2 by demonstrating convergence for five different 1D Riemann problems to high resolution reference solutions generated from a publicly available finite volume code Athena++ (Stone et al., 2020a). Next, we demonstrate the scheme's ability to handle multi-dimensional shocks through a series of 2D Riemann problems previously established in the literature. Then, we measure the growth rate of the relativistic Kelvin-Helmholtz instability in 2D in §5.4.5, comparing to results using the finite volume code PLUTO(Mignone et al., 2011). Last, in §5.4.6, we show timing tests of the code evolving the Kelvin-Helmholtz instability.

5.4.1 Linear Waves

Prior work in the literature (see, e.g. Stone et al., 2008a) has demonstrated that the convergence of linear waves in multi-dimensions is a sensitive test of algorithmic fidelity. As far as we are aware, however, linear wave convergence has not been utilized as a test of algorithms for relativistic hydrodynamics. Here, we elucidate how such a test can be established and demonstrate the performance of the algorithm presented here for such a test problem. To generate the linear waves, a perturbation is made to the initial primitive state, $\mathbf{W}_0 = [\rho_0, \mathbf{v}_0, P_0]^T$ (using rest mass density, three-velocity, and pressure), in the form of

$$\mathbf{W}[i] = \mathbf{W}_0[i] + A\mathbf{r}^j[i]\sin(kx - \omega t) \tag{5.76}$$

where $\mathbf{W}$ is the perturbed primitive state, A is the perturbation amplitude (typically $10^{-6} - 10^{-4}$), $\mathbf{r}^j[i]$ is the jth right eigenvector, the wavelength is equal to 1, $k = 2\pi$ and $\omega = k\lambda^j$. Here, we have defined λ is the wavelength and λ^j is the eigenvalue corresponding to the jth right eigenvector of the Jacobian, $A(\mathbf{V})$, given in Mignone et al. (2005). Each eigenvalue/vector pair corresponds to a different set of physics for linear wave testing, giving a total of 5 physically different linear wave tests, which we denote with $j \in \{-, 0^{(1,2,3)}, +\}$. Once we have the perturbed primitives, we need to translate these to a perturbed conserved quantities state, $\mathbf{U}$. This is done using the Jacobian $\partial\mathbf{U}/\partial\mathbf{W}$ in the following equation:

$$\mathbf{U}(t=0) = \left.\frac{\partial\mathbf{U}}{\partial\mathbf{W}}\right|_{\mathbf{W}} \mathbf{W}(t=0) \tag{5.77}$$

The Jacobian, $\partial\mathbf{U}/\partial\mathbf{W}$ must be constructed around a state, $\mathbf{W}$ such that the solution to the non-linear relationship $\mathbf{W}[i]](\mathbf{U}) = \mathbf{W}_0[i] + A\mathbf{r}^j[i]\sin(kx - \omega t)$ at $t = 0$. If this condition is not fulfilled, then a *different* problem is initialized and the evolution of the system will depart from the linear dispersion relation. To fulfill this criteria, we have found that it is necessary to compute the Jacobian using the unperturbed state, $\mathbf{W}_0$, but including the perturbation to the velocity in the Lorentz factor, in order to ensure that coupling between different components of the velocity is accurately captured. While we emphasize that this is done *only* to establish the initial condition in the conserved quantities, this reinforces a fundamental difference between relativistic and non-relativistic hydrodynamics; in the relativistic case the primitive variables are *always* a non-linear function of the conserved quantities due to the presence of the Lorentz factor.

Now that the 1D perturbed states **U** and **W** have been determined, we can rotate these for 2D and 3D non-grid-aligned cases. To do this, we first start with a desired number of wavelengths, N, and find the n^{th} acceptable angle, θ, by Eq. 5.78, where $n < N$. The values for N and n for the linear waves tests are shown in Tab. 5.1.

$$\theta = \tan^{-1}\left(\sqrt{\frac{N}{N-n} - 1}\right) \tag{5.78}$$

Table 5.1: Values of N (no. of wavelengths) and n (n^{th} acceptable wavelength) for linear waves tests (see Eq. 5.78)

Test Type	N	n
1D	1	0
2D Grid-Aligned	1	0
2D Non-Grid-Aligned	2	1
3D Grid-Aligned	1	0
3D Non-Grid-Aligned	3	2

From here, the base equations in the 1D form of Eq. 5.76 are rotated by the angle θ. Which is done either about the y axis, $a = (0, 1, 0)$, for 2D or about the $a = (0, -1, 1)$ axis for 3D. The rotation matrix, **R**, is generated via

$$\mathbf{r}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{r}_2 = \begin{bmatrix} a_x a_x & a_x a_y & a_x a_z \\ a_y a_x & a_y a_y & a_y a_z \\ a_z a_x & a_z a_y & a_z a_z \end{bmatrix}, \mathbf{r}_3 = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{5.79}$$

$$\mathbf{R} = \cos(\theta)\mathbf{r}_1 + (1 - \cos(\theta))\mathbf{r}_2 + \sin(\theta)\mathbf{r}_3. \tag{5.80}$$

R is then used to rotate the three-velocity vector, **v**, and the momentum vector, **M**, by left multiplying them by **R**. Next, the (x, y, z) coordinates in each equation are substituted with rotated coordinates

(x', y', z'), where

$$x' = \mathbf{R}\begin{bmatrix}1\\0\\0\end{bmatrix}, \quad y' = \mathbf{R}\begin{bmatrix}0\\1\\0\end{bmatrix}, \quad z' = \mathbf{R}\begin{bmatrix}0\\0\\1\end{bmatrix}. \tag{5.81}$$

Once these values have been substituted, the final, non-grid-aligned equations for $\mathbf{U}$ and $\mathbf{W}$ have been obtained.

For all eigenvalue/eigenvector cases, $j = \{-, 0^{(1,2,3)}, +\}$, tests are run for the rotation configurations in Table 5.1 with basis order and time integrator combinations of (0, RK1), (1, SSPRK2), and (2, SSPRK3). The domain, $\mathbf{L}$, and number of elements in each direction, $\mathbf{N}$, is calculated based on the rotation matrix, $\mathbf{R}$:

$$\mathbf{L} = N\mathbf{R}\left(\frac{\mathbf{e}}{|\mathbf{e}|}\right) \tag{5.82}$$

$$\mathbf{N} = Nn_{\text{elem}}x_\sigma^r\mathbf{R}\left(\frac{\mathbf{e}}{|\mathbf{e}|}\right) \tag{5.83}$$

where N is the number of wavelengths, $\mathbf{e}$ is the direction vector for the default orientation of the wave $\left(\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T\right)$, x_σ is the refinement multiplier per refinement increment (default $x_\sigma = 2$), r is the refinement level, and n_{elem} is the base number of elements, which varies for 1D, 2D, and 3D.

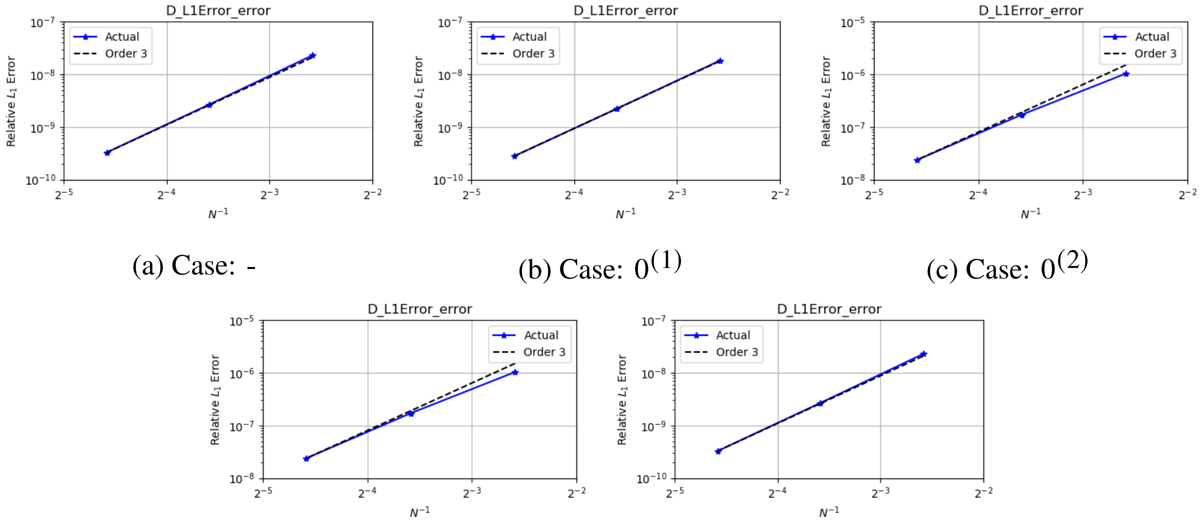
For these tests, the velocity was either set to $\mathbf{v} = \mathbf{0}$ or $\mathbf{v} = \begin{bmatrix}0.5v_{\text{max}} & -0.3v_{\text{max}} & 0.4v_{\text{max}}\end{bmatrix}^T$, where $v_{\text{max}} = 0.05c_s$. The base time step is determined by running the test with adaptive time stepping, which adjusts the time step to maintain a certain CFL during the test (0.2 in this case). The test is then run again 3 times, each time increasing the refinement in both space and time by a factor of 2 to maintain a constant CFL. The L1Error and L2Error are gathered for each test and are fitted against the results using the following equation:

$$\text{L1Error}(dx) = p_0 + p_1(dx)^{p_2} \tag{5.84}$$

where p_0, p_1, and p_2 are fitting constants. The exponent p_2 is the convergence order, which is expected to be 1, 2, and 3 for the time integrators RK1, SSPRK2, and SSPRK3 respectively. Results for the 3D, non-grid-aligned, zero velocity, basis order 2, SSPRK3, test case are shown in Tab. 5.2, while the L1Error is plotted against the expected values for the conserved quantity D in Fig. 5.6.

Table 5.2: Order of convergence for both primitive and conserved variables along the rows for each of the 5 eigenvalue/eigenvector pairs $j \in \{-, 0^{(1,2,3)}, +\}$ along the columns, all tested in 3D with non-grid-aligned waves, using a 2nd order basis with the SSPRK3 integrator. For all cases we expect a 3.0 rate of convergence. Entries with '-' denote variables where the eignvector used for that test does not affect that variable.

Quantity	Eigenvalue/eigenvector Test Case				
	-	$0^{(1)}$	$0^{(2)}$	$0^{(3)}$	+
D	3.099989	3.036570	2.561624	2.561624	3.099989
M_x	3.079648	-	2.838988	2.838988	3.079648
M_y	3.079648	-	2.879077	2.824568	3.079648
M_z	3.079648	-	2.824568	2.879077	3.079648
E	3.099989	3.036570	2.561652	2.561652	3.099989
ρ	3.099989	3.036570	-	-	3.099989
u_x	3.079655	-	2.838988	2.838988	3.079655
u_y	3.079655	-	2.879077	2.824568	3.079655
u_z	3.079655	-	2.824568	2.879077	3.079655
P	3.099989	-	-	-	3.099989



(a) Case: -

(b) Case: $0^{(1)}$

(c) Case: $0^{(2)}$

(d) Case: $0^{(3)}$

(e) Case: +

Figure 5.6: Order of convergence for the relativistic mass density (in solid blue) for three resolutions along the x-axis the 5 eigenvalue/eigenvector pairs $j \in \{-, 0^{(1,2,3)}, +\}$ in different panel. For all tests here we test in 3D with non-grid-aligned waves, using a 2nd order basis with the SSPRK3 integrator. For all cases we expect a 3.0 rate of convergence, which we denote with a dashed black line.

5.4.2 1D Riemann Problems

We now investigate the accuracy of the relativistic hydrodynamics method through considering the evolution of a set of standard 1D Riemann problems in order to characterize how well the code handles shocks. For initial conditions, we use three standard blast waves and a reflecting wall test from Martí & Müller (2003, 2015) and one Sod shock tube, and a reflecting wall test for a total of five different 1D Riemann problems.

For the first four 1D Riemann problems, we use a $[0, 1]$ grid with Dirichlet boundary conditions. These four tests begin divided into a primitive state on the left $\mathbf{W}_L = (\rho, v_x, v_y, p)_L$ for $x \in [0, 0.5)$ and right $\mathbf{W}_R = (\rho, v_x, v_y, p)_R$ for $x \in [0.5, 1]$. In the fifth test, we replace the boundary condition at $x = 1$ with a reflecting boundary and use a uniform initial primitive state through the domain. In all cases, we set $v_z = 0$ and use the ideal equation of state with $\gamma = 5/3$ for the first four tests and $\gamma = 4/3$ for the fifth test.

For each of the five 1D Riemann problems, we use a $[0, 1]$ grid with Dirichlet boundary conditions except for test 5, which uses a reflecting boundary condition on the right wall. The tests begin divided into a primitive state on the left $\mathbf{W}_L = (\rho, v_x, v_y, p)_L$ for $x \in [0, 0.5)$ and right $\mathbf{W}_R = (\rho, v_x, v_y, p)_R$ for $x \in [0.5, 1]$ except for test 5, which begins with a constant primitive state throughout the volume. In all cases, $v_z = 0$.

For reference data, we compute a $n_x = 2^{14}$ cell solution using a HLLC Riemann solver, a second order Van-Leer integrator due to Stone et al. (2020a) for each of the tested Riemann problems. We run the each 1D Riemann problem with five resolutions in powers of two from $n_x = 256$ to $n_x = 4096$ cells with polynomial basis orders 0, 1, and 2 using the HLLC Riemann solver and the iterative primitives recovery method for the ideal gas. For basis orders 1 and 2, we use the limiter from Moe et al. (2015) in addition to the physicality-enforcing operator from § 5.2.5. The physicality-enforcing operator was necessary for all tests with basis orders over 0. Fig. 5.7 shows the density, longitudinal velocity, pressure, and Lorentz factor from the five 1D Riemann problems using $n_x = 128$ with the three polynomial basis orders and the reference solution. Fig. 5.8 shows a log-log plot of the L1 error of the relativistic density, longitudinal relativistic momentum density, and total energy density compared to the reference solution along with power fits to the convergence rate and the expected rate of convergence.

1D Riemann problem 1 is a mildy relativistic blast wave with initial conditions

$$\mathbf{W}_L = \Big(10, 0, 0, (40/3)c^2\Big)_L \qquad \mathbf{W}_R = \Big(1, 0, 0, (2/3 \times 10^{-6})c^2\Big)_R \tag{5.85}$$

where we have followed Núñez-de la Rosa & Munz (2018) and used a pressure close to zero for the right side primitive state for numerical reasons. For this test, we use an adiabatic index $\Gamma = 5/3$. We evolve the shock until $t = 0.4/c$. For this first test we achieved the expected convergence rate in all variables except for the density for basis order 0, which suffers from slow converging dissipation around the blast wave. We also see a small cusp in velocity and oscillations in basis order 2 at the trailing edge of the blast wave which are more apparent in the Lorentz factor. L1 error of basis orders 1 and 2 are comparable, highlighting the difficultly in achieving high-order convergence with higher order methods when the problem contains shocks. However, since the basis order 2 test has more degrees of freedom than the basis order 1 test, the L1 error per degree of freedom is still lower for basis order 2, indicating that higher order bases can still be more efficient.

1D Riemann problem 2 is a highly relativistic blast wave with initial conditions

$$\mathbf{W}_L = \Big(1, 0, 0, (10^3)c^2\Big)_L \qquad \mathbf{W}_R = \Big(1, 0, 0, (10^{-2})c^2\Big)_R, \tag{5.86}$$

using an adiabatic index $\Gamma = 5/3$ and evolved until $t = 0.4/c$. In this test, we see that the sharpness of the resolved density of the blast wave changes with resolution. We see it the sharpest with basis order 1, second with basis order 0, and most diffuse with basis order 2, although for each basis the sharpness improves with resolution. We see a slight cusp in the Lorentz factor for all basis orders just behind the blastwave where the velocity approaches c but in the high resolution finite volume method the region has a flat Lorentz factor. The sharp blast wave in density causes problems for convergence at basis order 0 while higher order bases achieve the expected convergence.

1D Riemann problem 3 is also a highly relativistic blast wave but with a transverse velocity with initial conditions

$$\mathbf{W}_L = \Big(1, 0, 0, (10^3)c^2\Big)_L \qquad \mathbf{W}_R = \Big(1, 0, 0.99, (10^{-2})c^2\Big)_R, \tag{5.87}$$

with an adiabatic index $\Gamma = 5/3$ and evolved until $t = 0.4/c$. With the addition of a relativistic transverse velocity, the blast wave widens into a square plateau in density, somewhat similar to problem 1. Like in problem 2, we find that basis order 1 best captures the blast wave, although resolution improves accuracy for all basis orders. In the Lorentz factor we see a small cusp at the rightmost edge of the rarefaction and some smearing across the blastwave. The wider blast wave allows basis order 0 to achieve the expected convergence rate. L1 error for basis order 2 is greater than the L1 error for basis order 1, although this is mostly due to more degrees of freedom in the summation of the L1 error for basis order 1.

1D Riemann problem 4 is a Sod shock with initial conditions

$$\mathbf{W}_L = \left(1, 0.01c, 0, 1.0c^2\right)_L \qquad \mathbf{W}_R = \left(0.125, 0.01c, 0, 0.1c^2\right)_R, \tag{5.88}$$

using an adiabatic index $\Gamma = 4/3$ and evolving until $t = 0.4/c$. We see some diffusivity across the contact discontinuity and at the leftmost edge of the rarefaction.

For the fifth 1D Riemann problem we study a highly relativistic flow moving to the right and reflecting against the right wall. We use the initial conditions

$$\mathbf{W} = \left(1, 0.99999c, 0, 0.01c^2\right), \tag{5.89}$$

with an adiabatic index $\Gamma = 4/3$ and evolved until $t = 1.5/c$. We see a small cusp in the Lorentz factor at the left edge of the piled up stationary mass. For higher order bases, we see wall heating causing spurious oscillations in the reflected fluid. These leads to slow rates of convergence for basis order 2.

5.4.3 1D Taub-Matthews Equation of State Test

We test the Taub-Matthews approximation to the Synge equation of state against the ideal equation of state using the fifth blast wave problem from Ryu et al. (2006), which highlights the differences between the Synge gas and ideal gas. The initial conditions for the test, using the same notation and domain as §5.4.2, are

$$\mathbf{W}_L = \left(1, 0, 0.9c, (10^3)c^2\right)_L \qquad \mathbf{W}_R = \left(1, 0, 0.99c, (10^{-2})c^2\right)_R, \tag{5.90}$$

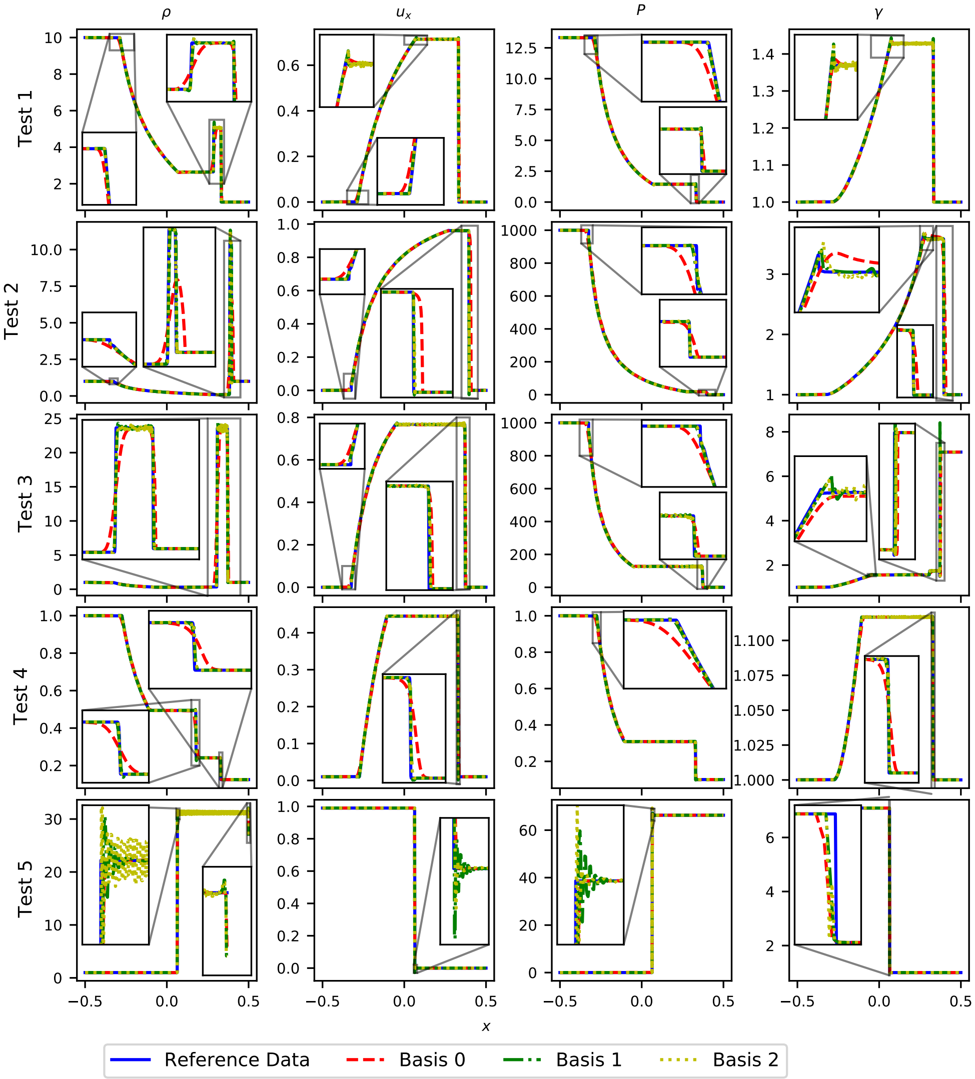


Figure 5.7: Plots of the five 1D Riemann problems tested using the ideal equation of state. Each row shows end state of a different Riemann problem. From top to bottom, the first row shows a mildly relativistic blast wave, the second a highly relativistic blast wave, the third a blast wave with transverse velocity, the fourth a Sod shock tube, and the fifth a planar shock reflection. The columns show from left to right the rest-mass density, the pressure, the velocity, and the Lorentz factor. In each panel we show the reference solution computed with a finite volume scheme (Stone et al., 2020a) with a solid line and the basis 0, 1, and 2 solutions with our method with a red dashed, green dot-dashed, and yellow finely dash line respectively. Although the method can evolve these shocks with the help of the physicality-enforcing operator, small oscillations appear around shocks for higher order bases. These oscillations can be damped out by widening the limiting thresholds for the Moe limiter or by changing the minmod limiter but this results in more diffusion and lower order convergence for basis order 2.

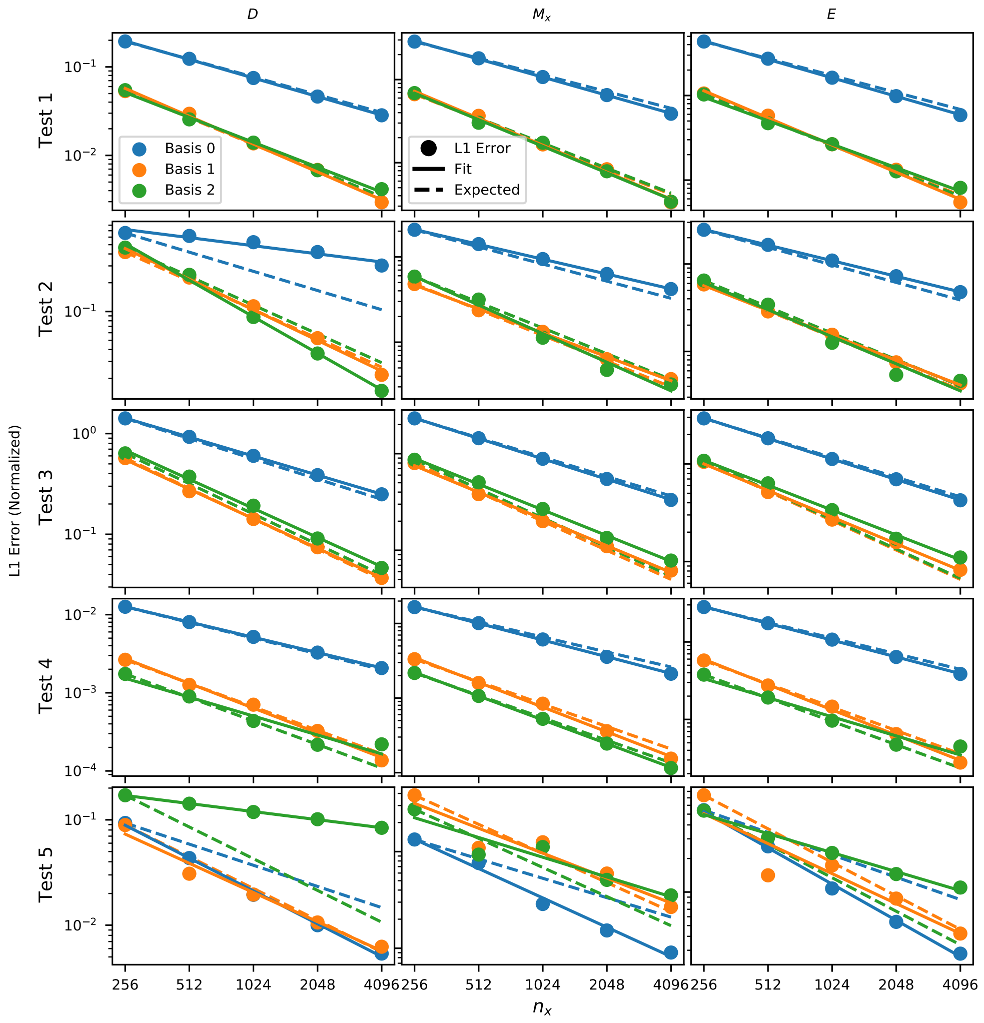


Figure 5.8: Convergence of the L1 error of the method presented here to a high resolution reference solution of the same Riemann problems from Fig. 5.7 computed with a finite volume scheme (Stone et al., 2020a). From top to bottom, the first row shows a mildly relativistic blast wave, the second a highly relativistic blast wave, the third a blast wave with transverse velocity, the fourth a planar shock reflection, and the fifth a Sod shock tube. The columns show from left to right the rest-mass density, the pressure, the velocity, and the Lorentz factor. In each panel we show the L1 error of our method with dots, a fitted convergence rate using logarithmically weighted least squares with a solid line, and a 2/3 convergence rate for basis order 0 and a first order convergence rate for bases 1 and 2 with dashed lines. We use different colors to denote different basis orders, using blue for basis order 0, orange for basis order 1, and green for basis order 2. Due to the presence of shocks, we expect the L1 error of higher order bases to converge to first order at best, although sharp blasts prove difficult for convergence.

which evolves into a blast wave. In the initial state, the temperature stand-in $\Theta = P/\rho$ on the left-hand side is relativistic while Θ on the right-hand side is non-relativistic. As such, for an ideal equation of state, an adiabatic index of $\Gamma = 4/3$ is appropriate for the left-hand side while $\Gamma = 5/3$ is appropriate for the right-hand side. The Taub-Matthew equation of state approximation allows accurate modeling of both sides with a single equation of state.

We show results for the blast wave with the three different equation of state in Fig. 5.9. The Synge gas as approximated by the Taub-Matthews equation of state behaves like the relativistic $\Gamma = 4/3$ ideal gas on the left side of the blast wave (which is contained within $[0.3, 0.4]$ at $t = 0.7$ as shown) and like the non-relativistic $\Gamma = 5/3$ ideal gas on the right side. This is most evident in the velocity profiles and pressure profiles in the relativistic region that occupies most of the domain at this time. The equivalent adiabatic index Γ_{eq} of the Taub-Matthews equation of state is expectedly $4/3$ in the relativistic region and $5/3$ in the non-relativistic region, and varies between these values across the blast wave. In this region within the blast wave, the peak density with the Taub-Matthews equation of state falls between the extremes of the two ideal gases. Notably, the blast wave with the Taub-Matthew equation of state travels slightly faster than either ideal gases, and the minimum transverse velocity is also lower. These results are consistent with the blast waves evolved with the Taub-Matthews equation of state in Ryu et al. (2006).

5.4.4 2D Riemann Problems

Next, we test the robustness of the method evolving intersecting shocks in 2D using the three 2D Riemann problems used in Zanna & Bucciantini (2002); Núñez-de la Rosa & Munz (2018). In each of the three problems, the problem is defined with a $[-1, 1] \times [-1, 1]$ domain divided into four quadrants with different initial states. Following Núñez-de la Rosa & Munz (2018), we denote these states using

$$Q_1 := [0, 1] \times [0, 1] \tag{5.91}$$

$$Q_2 := [-1, 0] \times [0, 1] \tag{5.92}$$

$$Q_3 := [-1, 0] \times [-1, 0] \tag{5.93}$$

$$Q_4 := [0, 1] \times [-1, 0] \tag{5.94}$$

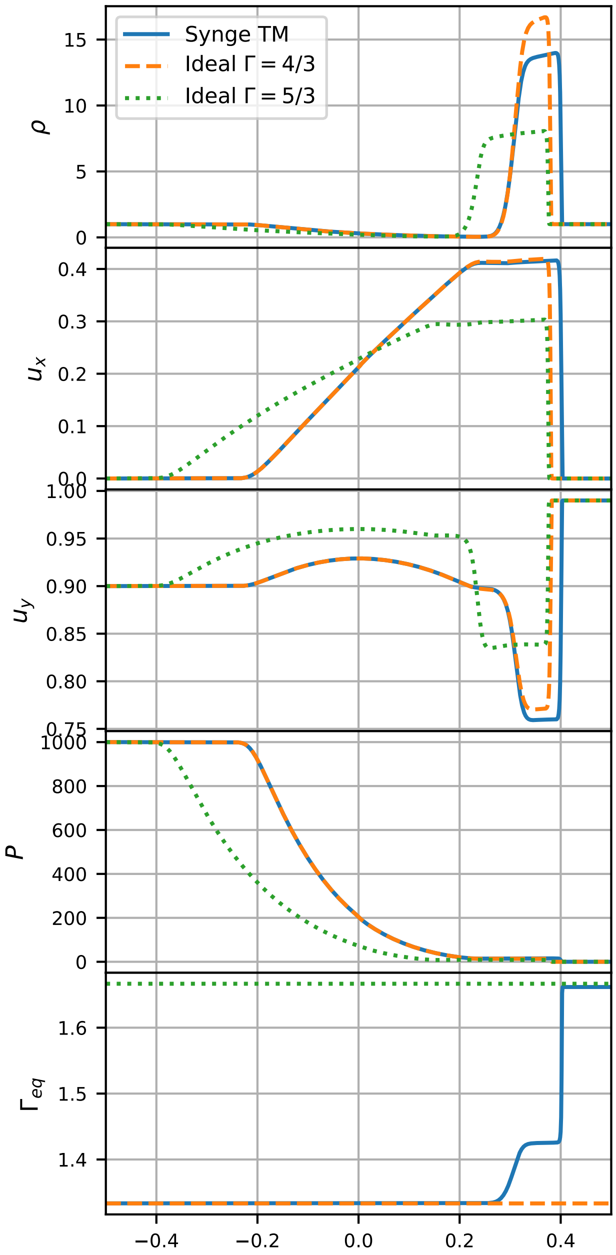


Figure 5.9: Blast wave with relativistic temperatures on the left and non-relativistic temperature on the right, evolved to $t = 0.7$ using the Taub-Matthews equation of state (solid blue), ideal equation of state with adiabatic index $\Gamma = 4/3$ (dashed orange), and ideal equation of state with $\Gamma = 5/3$ (finely dashed green). In order of rows, we show the density ρ, longitudinal velocity u_x, transverse velocity u_y, pressure P, and equivalent adiabatic index $\Gamma_{\text{eq}} = \left(h - c^2\right) / \left(h - c^4 - P/\rho\right)$. The Taub-Matthews equation of state, as an approximation to the Synge gas, behaves apart from both the $\Gamma = 5/3$ and $\Gamma = 4/3$ ideal gases depending on the effective adiabatic index.

and denote the initial primitive states in each of these quadrants by $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{W}_4$ respectively. For all of these Riemann problems, we use an adiabatic index of $\Gamma = 5/3$, use $v_z = 0$ everywhere, and use transmissive boundary conditions on all sides. We evolve each Riemann problem to $t = 0.8/c$. For all 2D shock tests we use the Moe limiter (Moe et al., 2015) and HLLC Riemann solver.

5.4.4.1 2D Riemann Problems: Test 1

In this test, the domain begins with a low density and pressure region in the upper right, a high density and pressure region in the lower left, and intermediate density and high pressure regions in the upper left and lower right with initial velocities moving into the lower density region with $\beta = 0.7$.

$$\mathbf{W}_1 := (0.035145216124503, 0.0, 0.0, 0.162931056509027c^2) \tag{5.95}$$

$$\mathbf{W}_2 := (0.1, 0.7c, 0.0, 1.0c^2) \tag{5.96}$$

$$\mathbf{W}_3 := (0.5, 0.0, 0.0, 1.0c^2) \tag{5.97}$$

$$\mathbf{W}_4 := (0.1, 0.0, 0.7c, 1.0c^2) \tag{5.98}$$

Results from the first 2D Riemannn problem is shown in Fig. 5.10 with the 1st and 2nd order bases, the system evolves with stationary contact discontinuities between the high density and moving intermediate density regions, planar shocks moving from the intermediate density regions into the low density regions, and curved shocks bowing into the intermediate density regions from the diagonal. A jet-like, low density structure forms into the high density region with gentle density and pressure gradients forming ahead and behind it. Our method evolves the curved shocks with symmetric shock fronts using both low order and high-order bases. When using bases over 0th order, the physicality-enforcing operator described in §5.2.5 is necessary to avoid negative densities, pressures, and otherwise unphysical states. With the 2nd order basis, we see subtle boundary effects where the shocks traveling transverse to the boundary into the first quadrant intersect with the

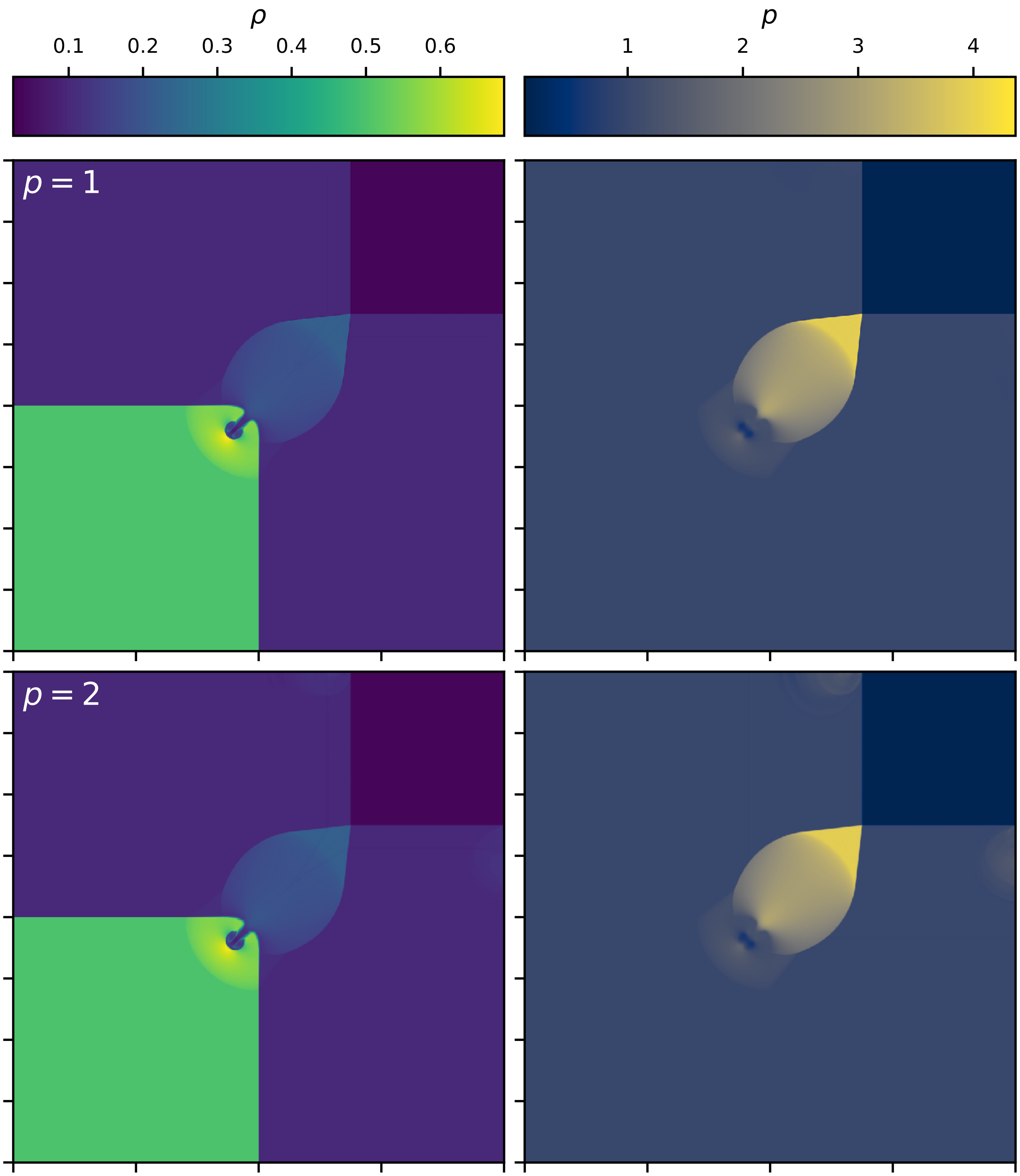


Figure 5.10: Plots of the 2D Riemann problem test 1 with two colliding shocks using the initial conditions in eq. 5.95, using a 1^{st} order basis in the top row and a 2^{nd} order basis in the bottom row. We show the rest-mass density in the left column and the pressure in the right column at $t = 0.8/c$ on a grid with 1024 elements. Note the boundary effects where shocks traveling into the first quadrant intersect with the outflow boundaries when using the 2^{nd} order basis.

outflow boundary conditions. Boundary effects with the 2nd order basis are seen again in § 5.4.4.2 and § 5.4.5.2.

5.4.4.2 2D Riemann Problems: Test 2

In this test, all four quadrants begin with different densities, equal pressures, and each move diagonally clockwise around the origin.

$$\mathbf{W}_1 := (0.5, 0.5c, -0.5c, 5.0c^2) \tag{5.99}$$

$$\mathbf{W}_2 := (1.0, 0.5c, 0.5c, 5.0c^2) \tag{5.100}$$

$$\mathbf{W}_3 := (3.0, -0.5c, 0.5c, 5.0c^2) \tag{5.101}$$

$$\mathbf{W}_4 := (1.5, -0.5c, -0.5c, 5.0c^2) \tag{5.102}$$

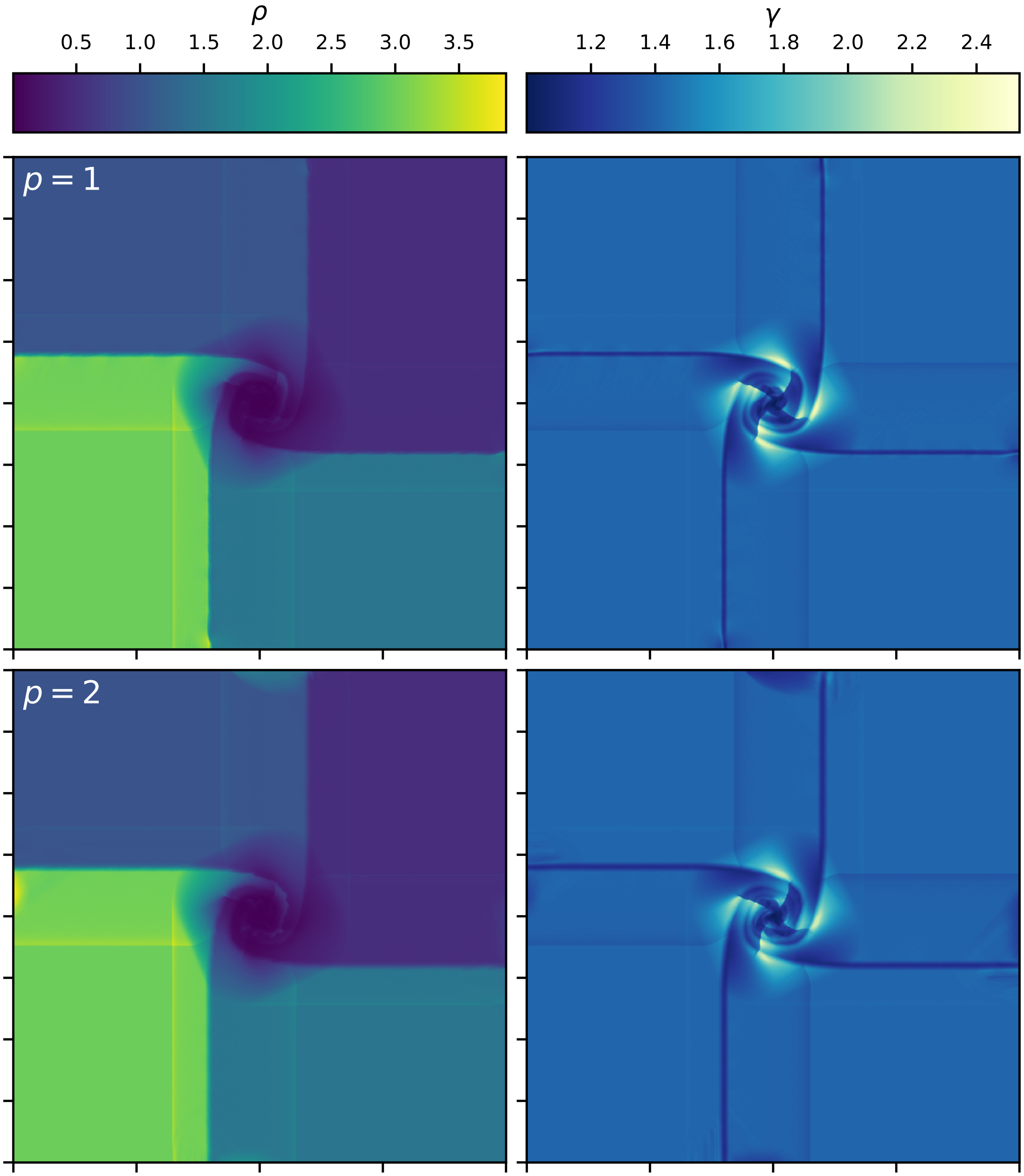


Figure 5.11: Plots of the 2D Riemann problem test 2 with four vortex sheets using the initial conditions in eq. 5.99, using 1^{st} order basis in the top row and a 2^{nd} order basis in the bottom row. We show the rest-mass density in the left column and the pressure in the right column at $t = 0.8/c$ using a grid with 1024 elements. Note the boundary effects where the vortex sheets intersect with the outflow boundaries which are subtle using the 1^{st} order basis and more apparent when using the 2^{nd} order basis, especially along the top boundary. Like the 1D test of a shock reflecting against a wall, this test highlights unresolved difficulties of higher order bases leading to boundary effects.

Results from the second 2D Riemannn problem are shown in Fig. 5.11 with the 1st and 2nd order bases, the system develops into four vortex sheets that expand from the origin. A low rest mass region forms at the center of the vortex sheets at the origin. The physicality-enforcing operator ensures positive densities and pressures in this region. With the 2nd order basis, we see subtle boundary effects where the shocks traveling transverse to the boundary into the first quadrant intersect with the outflow boundary conditions. These boundary effects are not apparent with the 1st order basis.

5.4.4.3 2D Riemann Problems: Test 3

This tests begins with overdense first and third quadrants following

$$\mathbf{W}_1 := (1.0, 0.0, 0.0, 1.0c^2) \tag{5.103}$$

$$\mathbf{W}_2 := (0.5771, -0.3529c, 0.0, 0.4c^2) \tag{5.104}$$

$$\mathbf{W}_3 := (1.0, -0.3529c, -0.3529c, 1.0c^2) \tag{5.105}$$

$$\mathbf{W}_4 := (0.5771, 0.0, -0.3529c, 0.4c^2). \tag{5.106}$$

Rarefactions move from the second and fourth quadrants into the first and third quadrants, producing curved shocks where the rarefactions intersect.

Results from the third 2D Riemann problem are shown in Fig. 5.12 with the 2nd order basis. The method evolves the curved shocks and rarefactions without issue. No boundary effects are apparent in this test.

5.4.5 Kelvin-Helmholtz Instability

The relativistic Kelvin-Helmholtz instability provides a useful benchmark with which to explore the performance of the scheme presented here for shear-flow type problems. Previous work, e.g. Mignone et al. (2009); Beckwith & Stone (2011) has revealed significant differences in the performance of different numerical schemes for this classic fluid flow problem and subsequent work Lecoanet et al. (2016) has further elucidated the issues raised in prior works through the comparison

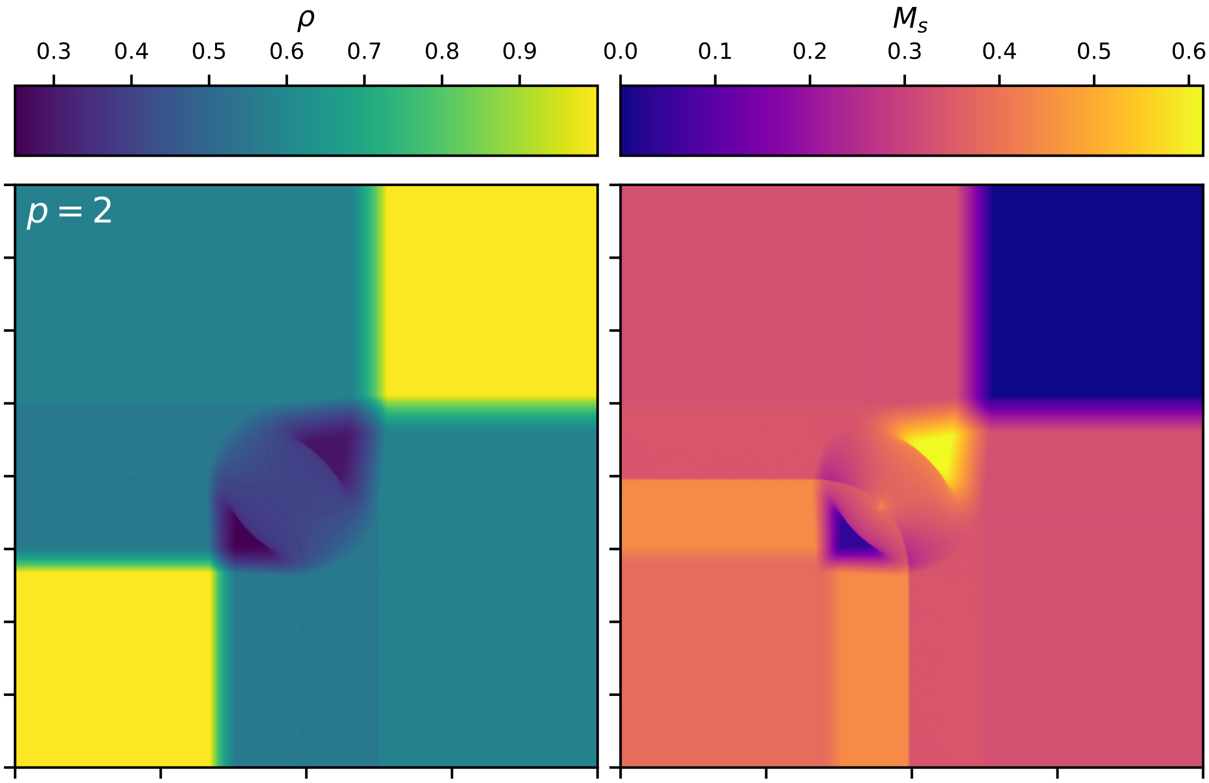


Figure 5.12: Plots of the 2D Riemann problem test 3 with intersecting rarefactions using the initial conditions in eq. 5.103. We show the rest-mass density left column and the pressure right column at $t = 0.8/c$ using a 2^{nd} order basis on a grid with 1024 elements.

of finite volume and spectral methods. Here, we compare the discontinuous-Galerkin scheme presented here with a finite volume method previously presented in the literature Mignone et al. (2011), explore both the linear and non-linear regime of the instability and examine performance metrics for the scheme.

We simulate the Kelvin-Helmholtz instability on a $[-0.5, 0.5] \times [-1.0, 1.0]$ domain with a single interface along $y = 0$, specified with a smoothly varying profile using a mesh of square cells with twice as many cells in y than x, testing mesh sizes in powers of 2 from 256×512 to 4096×8192 for a total of 6 different mesh sizes. We tested using basis orders 0, 1, and 2, however due to memory constraints and increasing execution time, we forgo the highest resolution mesh using basis order 1 and the two highest resolutions using basis order 2. We conduct separate tests using the HLLC and HLL Riemann solvers and using a shear velocity $v_{x,0} = 0.25c$. We run a total of 60 simulations exploring growth rates of the Kelvin-Helmholtz instability. In all these calculations, we use an ideal equation of state with adiabatic index $\gamma = 4/3$ using the iterative conserved-to-primitive solver, an initial density $\rho_0 = 1$, an initial pressure $P_0 = c^2$, a perturbation amplitude $A = 0.05$, and a shearing layer thickness $a = 0.01$. We use $k = 2\pi$ so that the wavelength of the perturbations in x is 1 and for each test run until $t = 5$ to verify from the growth rate that the transverse velocity perturbations have saturated past the linear growth phase.

5.4.5.1 Linear Growth Phase

We explore the growth of the instability by examining the spatial average

$$\langle v_y^2 \rangle = \frac{1}{|\Omega|} \int_\Omega v_y^2 \, dV \tag{5.107}$$

where Ω is the domain and $|\Omega|$ is the volume of the domain. Fig. 5.13 shows $\langle v_y^2 \rangle$ as a function of time in the left column for the Kelvin-Helmholtz instability simulations explored in this work, where Riemann solvers are grouped by column and basis order and reconstruction method grouped by rows. Except for the lowest resolution simulations, all simulations with the HLLC solver enter a linear growth phase by $t = 2.0$ and display non-linear features by $t = 4.0$. By contrast, simulations that utilize the HLL Riemann solver, especially with the 0th order basis, exhibit large levels of numerical diffusion and substantially reduced growth rates for all but the largest number of degrees of freedom. However, for basis order greater than zero, the HLL Riemann solver exhibits rapid convergence to a well-defined growth rate, while the reference finite volume schemes that utilize this same Riemann solver exhibit changing growth rates over this same range of degrees of freedom.

We quantify this result by measuring the growth rate, r of $\langle v_y^2 \rangle$ by fitting $\log\langle v_y^2 \rangle(t) = A + rt$ to the measured $\langle v_y^2 \rangle$ using a least squares curve fit in log space over $t = 1.5$ to $t = 3.0$. We measure the growth rate early in the linear growth phase from $t = 1.5$ to $t = 3.0$ before non-linear modes dominate. We perform the fit in log space so as to not favor the larger changes in $\langle v_y^2 \rangle$ at later times. The growth rate of $\langle v_y^2 \rangle$ for all simulations and methods versus the degrees of freedom is shown in Fig. 5.14. Here, the degrees of freedom for a given resolution $n_x \times n_y$ and basis order p is DOF $= n_x \times n_y \times (p + 1)^2$. Except for the discontinuous-Galerkin methods using the 0th order basis, the growth rates using different methods converge to approximately the same value with higher resolutions. Generally, using higher order bases, using the HLL Riemann solver over the HLLC Riemann solver, and using the discontinuous-Galerkin method over the finite volume method lead to faster convergence of growth rate. Notably, the overall second order accurate discontinuous-Galerkin scheme (first order basis, second order time integration scheme) achieves a converged growth rate at lower numbers of degrees of freedom than a overall second order accurate finite volume scheme, using either the HLLC or HLL Riemann solver.

This result is explored in more detail in Fig. 5.15. The data of this figure shows the difference in growth rate between the highest resolution simulation with a certain method and the lower resolution simulations with the same methods versus the degrees of freedom. The discontinuous-Galerkin simulations with a 1st order basis show the most effective convergence of the simulations explored here, with HLLC converging slightly faster at the highest resolutions and HLL converging faster at lower resolutions. By contrast, the overall second order accurate finite volume schemes exhibit slower convergence than this scheme, despite the equivalent order of accuracy, while the first order accurate discontinuous-Galerkin scheme exhibits similar convergence rates as the finite volume schemes when combined with the HLLC Riemann solver, but low convergence rates with the HLL solver. We also note that the discontinuous-Galerkin simulations with a 2nd order basis do not converge below a 10^{-1} difference even with high resolutions, which we attribute to interaction of the flow with outflow boundary conditions used here, highlighting the need for improved fidelity boundary conditions in order to realize the promise of higher order discontinuous-Galerkin methods.

5.4.5.2 Non-linear Evolution

Fig. 5.16, 5.17, and 5.18 show the state of the Kelvin Helmholtz instability at $t = 3.0$ using the method presented in this work and the reference finite volume scheme Mignone et al. (2011) with the 4 highest resolutions explored in this study. The different figures show results using 0th, 1st, and 2nd order bases or 1st, 2nd, and 3rd order methods respectively, where a 1st method is only available for our code. In Fig. 5.16 using our method with a 0th order basis or a 1st order method,

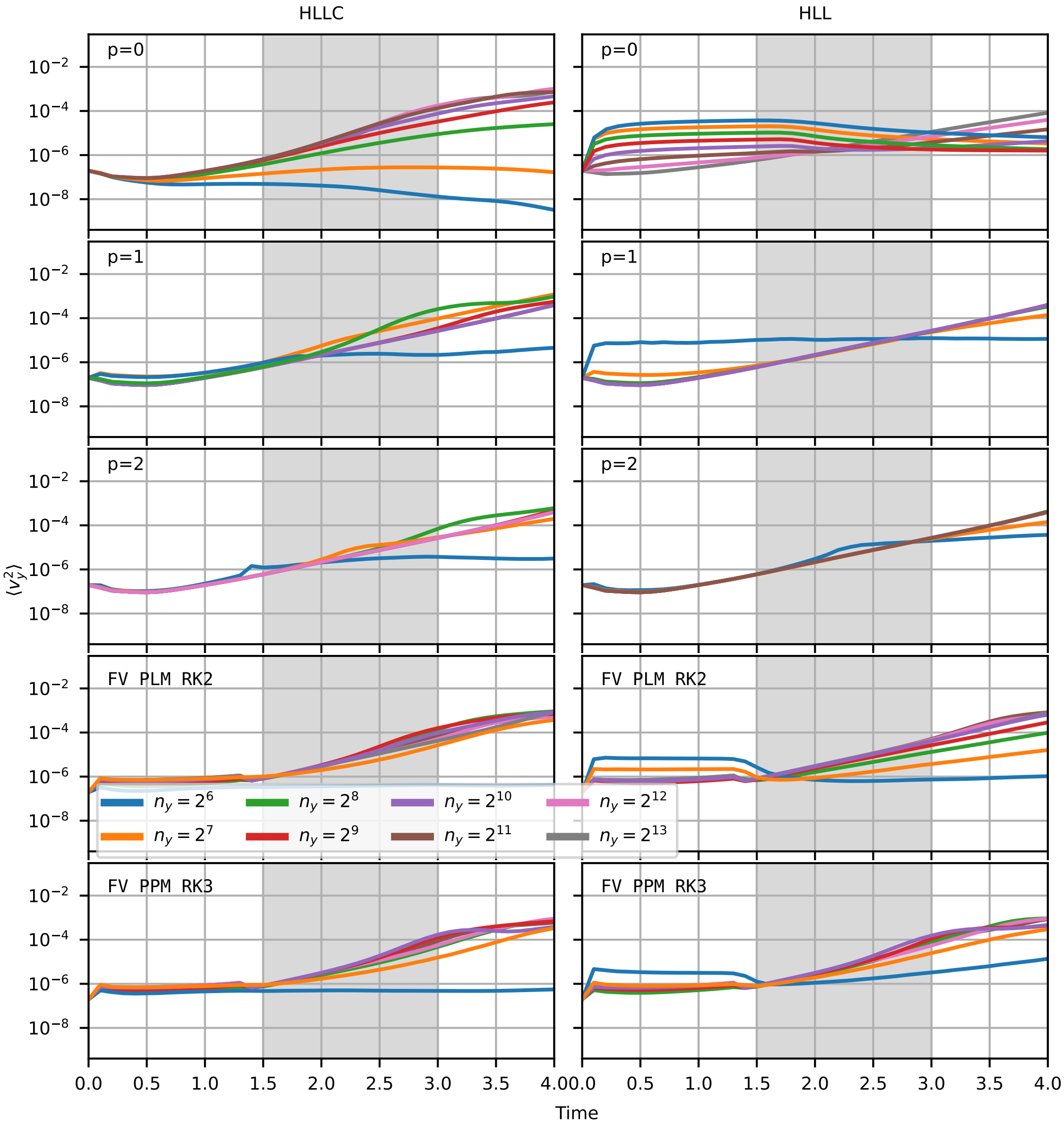


Figure 5.13: Mean square of the transverse velocity v_y over time of the relativistic 2D Kelvin Helmholtz instability using our DG method using a 0th, 1st, and 2nd order bases respectively in the top three rows and using the finite volume code PLUTO with PLM and PPM reconstruction respectively in the bottom two rows. In the left column we show results including the contact discontinuity in the Riemann solver (using HLLC with our method and HLLD with PLUTO) and without the contact discontinuity using the HLL Riemann solver in the right column. The gray band from $t = 1.5$ to $t = 3.0$ shows the region over which we measure the growth rate shown in other plots. Higher resolutions generally lead to faster growth rates while the more diffusive HLL Riemann solver leads to steadier growth rates due to diminished secondary instabilities.

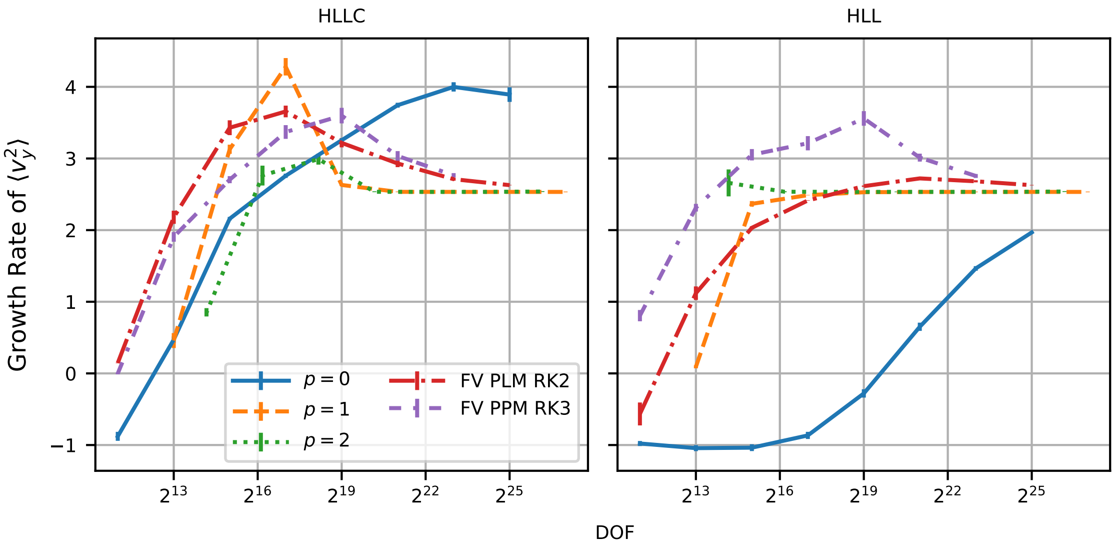


Figure 5.14: Growth rates of $\langle v_y^2 \rangle$ versus degrees of freedom from $t = 1.5$ to $t = 3.0$ of the relativistic 2D Kelvin Helmholtz instability using our DG method using the finite volume code PLUTO. In the left column we show results including the contact discontinuity in the Riemann solver (using HLLC with our method and HLLD with PLUTO) and without the contact discontinuity using the HLL Riemann solver in the right column. Growth rates are measured by computing least squares fit of a $\langle v_y^2 \rangle \propto t^\omega$ model to the data shown in Fig. 5.13, with error bars showing the standard deviation of the least squares fit.

we see significant differences between the HLL and HLLC solutions; the HLL Riemann solver struggles to grow the instability, although the structure of the perturbation resembles results with simple structures when using higher orders. Secondary instabilities appear to be nonexistent. By contrast, the HLLC Riemann solver generates secondary vortices that increasing in amplitude with higher resolutions. Looking at Fig. 5.17 and 5.18, the 2nd and 3rd order methods from this work quickly converge to simple structures. The finite volume method also converges to a similar simple structure, although it requires more resolution compared to the discontinuous-Galerkin method presented here.

Figs. 5.19, 5.20, and 5.21 show the state of the Kelvin Helmholtz instability at $t = 5.0$, which is well into the non-linear phase, using the method presented in this work and with the reference finite volume scheme with the 4 highest resolutions explored in this study. The different figures show results using 1st, 2nd, and 3rd order methods respectively, where a 1st is only available for our code. In Fig. 5.19 using our method with a 0th order basis or a 1st order method, we again

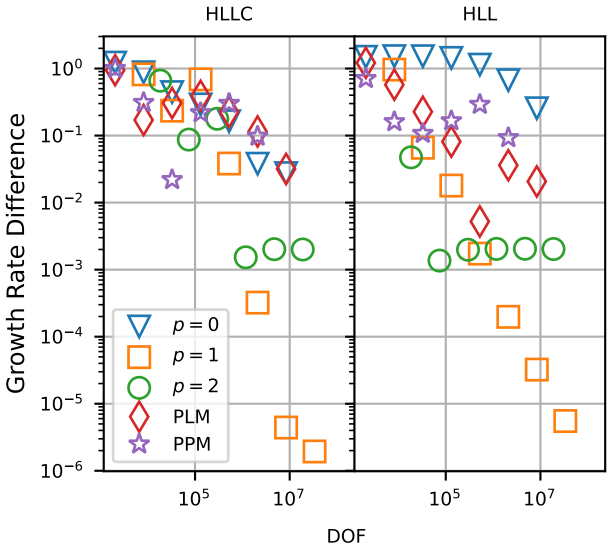


Figure 5.15: The absolute difference in growth rate between the highest resolution simulation for each method and each of the lower resolution simulations which serves as rough measure of the error of the growth rate, plotted versus the degrees of freedom. The discontinuous-Galerkin simulations with a 1st order basis show the most effective convergence of the simulations explored here, with HLLC converging slightly faster at the highest resolutions and HLL converging faster at lower resolutions. The discontinuous-Galerkin simulations with a 2nd order basis do not converge below a 10^{-1} difference even with high resolutions, which we attribute to the boundary effects that worsen with higher resolution. Otherwise, the other methods converge at varying rates, the 0th order basis discontinuous-Galerkin methods converging the slowest.

see significant differences between the HLL and HLLC solutions. The HLLC solution grows faster than the HLL solution but neither resemble the structures seen with higher order bases. Using the HLLC Riemann solver, secondary vortices are apparent during the non-linear phase, which become more defined with higher resolution. Examining Figs. 5.20 and 5.21, the 2nd and 3rd order methods from this work quickly converge with higher resolution to simple structures during the non-linear phase. Results with HLL over HLLC and with a 2nd order basis over a 1st order basis are generally smoother with fewer secondary vortices. The solution generated by the reference finite volume scheme also converges to roughly the same structures as the discontinuous-Galerkin method, although secondary instabilities are obvious along the interface between the primary vortices. Note that the mode of these secondary instabilities increased with resolution, with smaller but more numerous instabilities at higher resolutions.

Our interpretation of these results is that the secondary structures found in the finite volume method at the end of the linear growth phase serve to seed non-linear structures that are observed at late times; a result somewhat consistent with that reported by Lecoanet et al. (2016). What is notable is that these structures vanish in the second order accurate (first order basis) discontinuous-Galerkin scheme presented here at lower resolution than in the finite volume scheme for the HLLC Riemann solver and are *absent* in the HLL Riemann solver based scheme, indicating a role played by the dissipation of the HLLC Riemann solver in the formation of these structures. In addition, the presence of these structures in the finite volume scheme utilizing the HLL solver and the clear dependency of the properties of these structures on the reconstruction method (PLM vs. PPM) is another point of contrast between discontinuous-Galerkin methods and finite volume schemes. This is strongly reminiscent of the results presented by Lecoanet et al. (2016), where finite volume schemes were demonstrated to exhibit similar secondary vortices at moderate resolutions (similar to these presented here), which then disappeared at higher resolutions; the higher resolution simulations being comparable to spectral methods. The absence of such secondary vortices for combinations of the discontinuous-Galerkin algorithms presented here suggest that these methods may be less susceptible to such considerations. However, we stress that this is a single application on both methods and that the performance of either method may depend on details of the set up of the instability. Over the development of the method, we also explored the analytic growth rate of perturbations given the initial conditions from Bodo et al. (2004), where we found that the growth rate of the instability generally did not match the analytically predicted growth rate, and that an initial transient outgoing wave from the initial perturbation caused significant boundary effects with the 2nd order basis. Although our discontinuous-Galerkin method provides apparently better results in this case, more development especially around boundary conditions is required.

5.4.6 Performance

To test the performance of the method on multiple architectures, we timed simulations of the Kelvin Helmholtz instability on CPUs and GPUs, using the perturbations described in §5.4.5. For

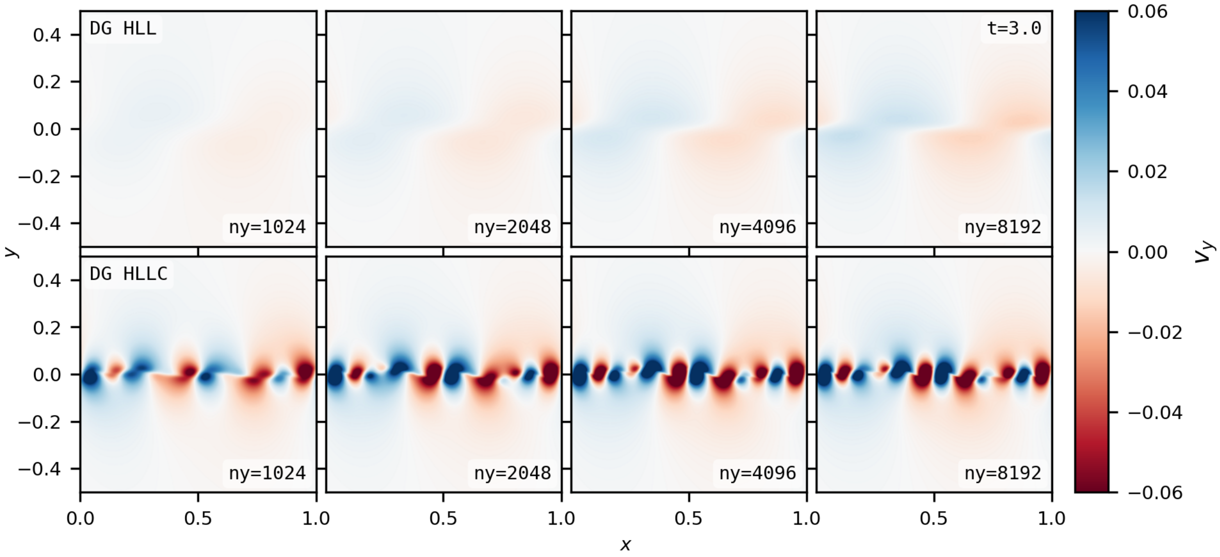


Figure 5.16: Snapshots of the transverse velocity at $t = 3.0$ from simulations of the relativistic Kelvin-Helmholtz instability using the method presented in this work using a 0th order basis. We show results using the HLL Riemann solver in the top row and with HLLC in the bottow row. We show the four highest resolution simulations across the columns, ranging from 512 × 1024 to 4096 × 8192 cells from left to right. With basis order zero, at this stage, using the HLL Riemann solver the method has difficulty growing the Kelvin Helmholtz instability, although the structure of the perturbation resembles results with simple structures when using higher orders. The HLLC Riemann solver generates secondary vortices that get worse with high resolutions, which leads to a climbing growth rate.

both architectures, we time the performance of the code with $v_{x,0} = 0.25c$ using basis orders 0, 1, and 2 and resolutions of 256 × 512, 512 × 1024, and 1024 × 2048 with each basis order testing both HLLC and HLL for a total of 18 simulations for both architectures. We conduct CPU testing on 1024 cores spread across 22 dual socket nodes with Intel Xeon Platinum 8268 CPUs, comprising approximately ∼ 88TFLOPS in total. For GPU runs we use 32 NVidia Tesla V100-SXM2 GPUs spread across 8 nodes, comprising approximately ∼ 250TFLOPS in total. These computational resources were chosen to accommodate the memory needed for the largest simulation in the performance profiling suite.

We show profiling results with the HLLC and HLL Riemann solvers and with the 0th, 1st, and 2nd order bases in Fig. 5.22. The degree of freedom updates per second is computed with

$$\text{DOF per second} = \frac{\text{DOF} \times \text{steps} \times \text{stages per step}}{\text{time to solution in seconds}}, \tag{5.108}$$

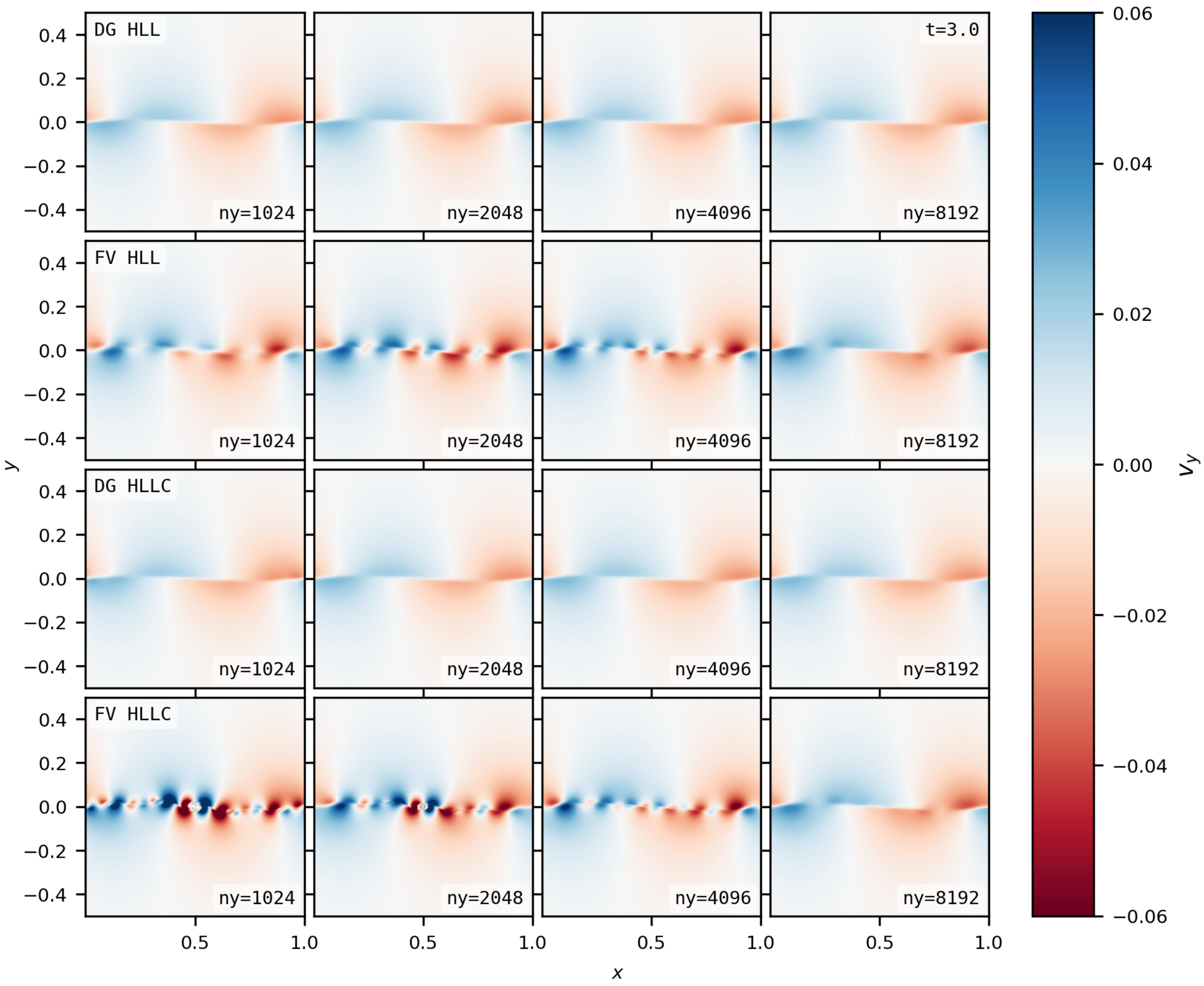


Figure 5.17: Snapshots of the transverse velocity at t = 3.0 from simulations of the relativistic Kelvin-Helmholtz instability using the method presented in this work using a 1st order basis in the first and third row and with the PLUTO finite volume MHD code with a first order method. We show results using the HLL Riemann solver in the top two rows and with HLLC for our code and with HLLD for PLUTO in the bottow two rows. We show the four highest resolution simulations across the columns, ranging from 512 × 1024 to 4096 × 8192 cells from left to right. Note that DG method has 4 times as many degrees of freedom with the 1st order basis, meaning that our 512 × 1024 simulation is comparable in degrees of freedom to the 1024 × 2048 simulation using PLUTO. At this times and these resolutions, the results with our DG method have converged to a similar solution with a simple structure. Results with PLUTO converge towards the DG method results, with secondary vortices present at lower resolutions that are more pronounced with HLLC.

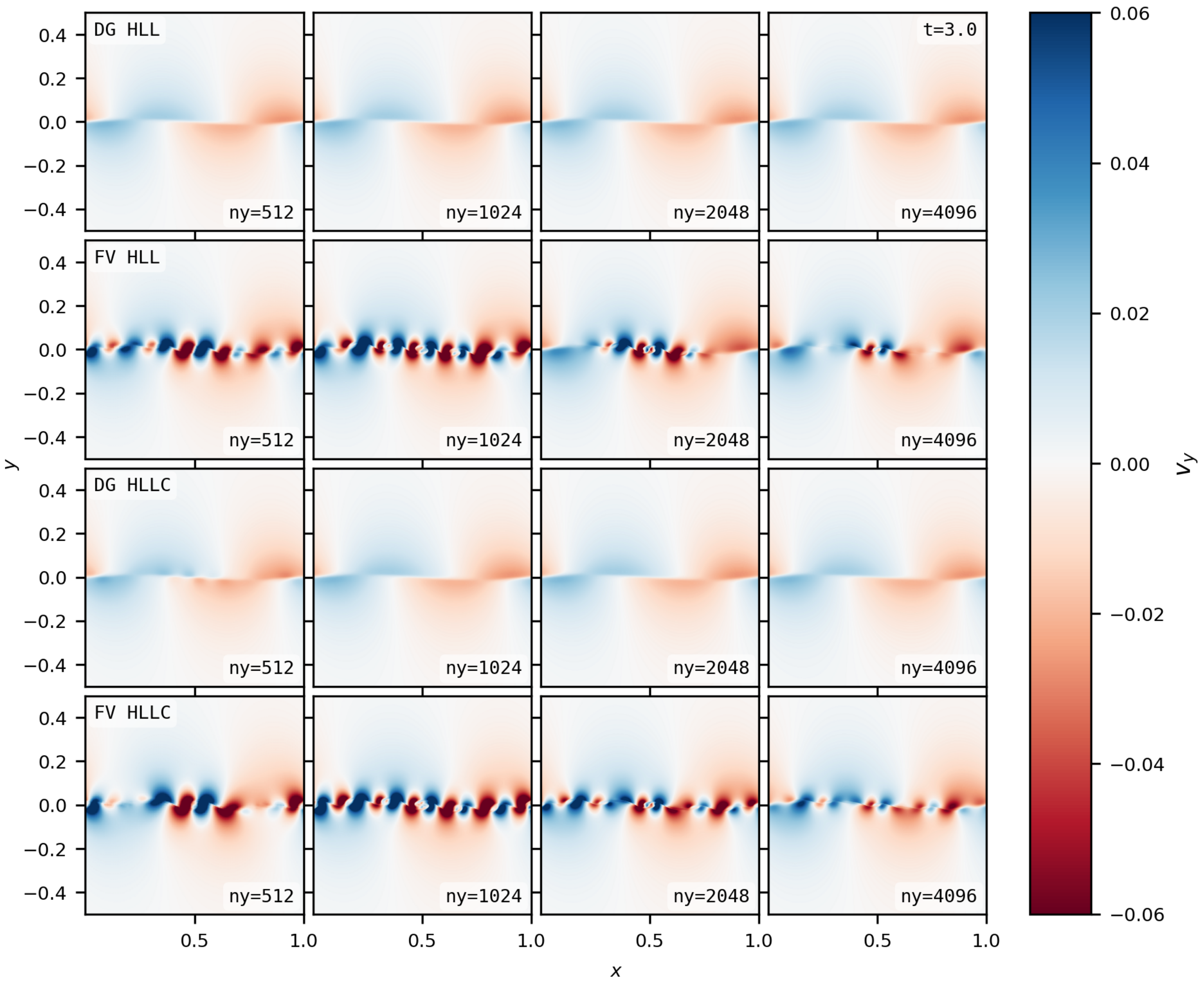


Figure 5.18: Snapshots of the transverse velocity at $t = 3.0$ from simulations of the relativistic Kelvin-Helmholtz instability using the method presented in this work using a 2nd order basis in the first and third row and with the PLUTO finite volume MHD code with a second order method. We show results using the HLL Riemann solver in the top two rows and with HLLC for our code and with HLLD for PLUTO in the bottom two rows. We show the four highest resolution simulations across the columns, ranging from 512 × 1024 to 4096 × 8192 cells from left to right. Note that DG method has 4 times as many degrees of freedom with the 1st order basis, meaning that our 512×1024 simulation has degrees of freedom between the 1024×2048 simulation and 2048×4096 simulation using PLUTO. With this higher order basis at $t = 3.0$, we also see the results with our DG method converge quickly to simple structures while the results with PLUTO require more resolution to suppress secondary vortices. However, in our results using 4096 × 8912 cells with basis order 2, we see anomalously high transverse velocities away from the interface, which is caused by boundary effects at high resolutions that will be addressed in future improvements to the method.

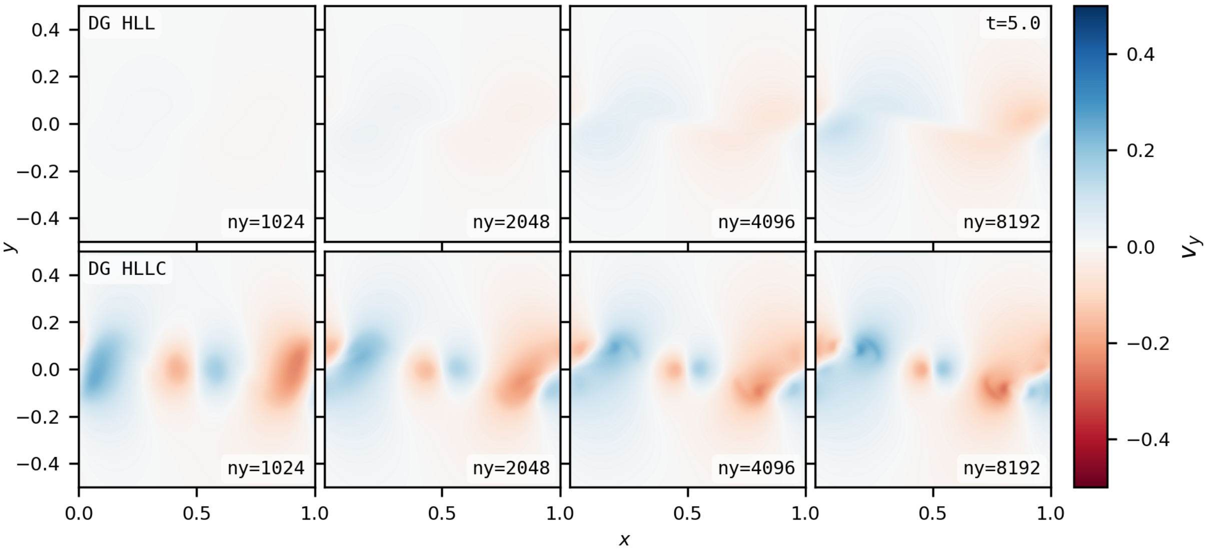


Figure 5.19: Snapshots of the transverse velocity at $t = 5.0$ from simulations of the relativistic Kelvin-Helmholtz instability using the method presented in this work using a 0th order basis. We show results using the HLL Riemann solver in the top row and with HLLC in the bottom row. We show the four highest resolution simulations across the columns, ranging from 512 × 1024 to 4096 × 8192 cells from left to right. At late times into what should be the linear growth phase, our DG method with the HLL solver struggles to growth the instability at low resolutions. The HLLC method has developed some structures but they do not resemble results at higher orders.

which serves as a measure of computational efficiency. With the `RK1`, `SSPRK2`, and `SSPRK3` integrators used for basis orders 0, 1, and 2 we use 1, 2, and 3 stages per step for the respective basis orders.

We show profiling results with the HLLC and HLL Riemann solvers and with the 0th, 1st, and 2nd order bases, between which we see little difference in performance. Comparing between the CPU and GPU runs, we see that the CPU performance becomes saturated at around 10^6DOF while the GPUs have not saturated the performance, even with simulations using more than 10 times the degrees of freedom. Simulations with more degrees of freedom would not fit within GPU memory here, indicating that our present implementation is unable to fully saturate GPU performance. Note that the theoretical peak throughput of the GPU resources using here is approximately three times the throughput for the CPU resources. Memory bandwidth resources between RAM and the registers on CPUs and HBM memory and the registers on GPUs is similarly greater on GPUs Since the CPUs and GPUs achieve roughly the same updates per second, this indicates underutilization

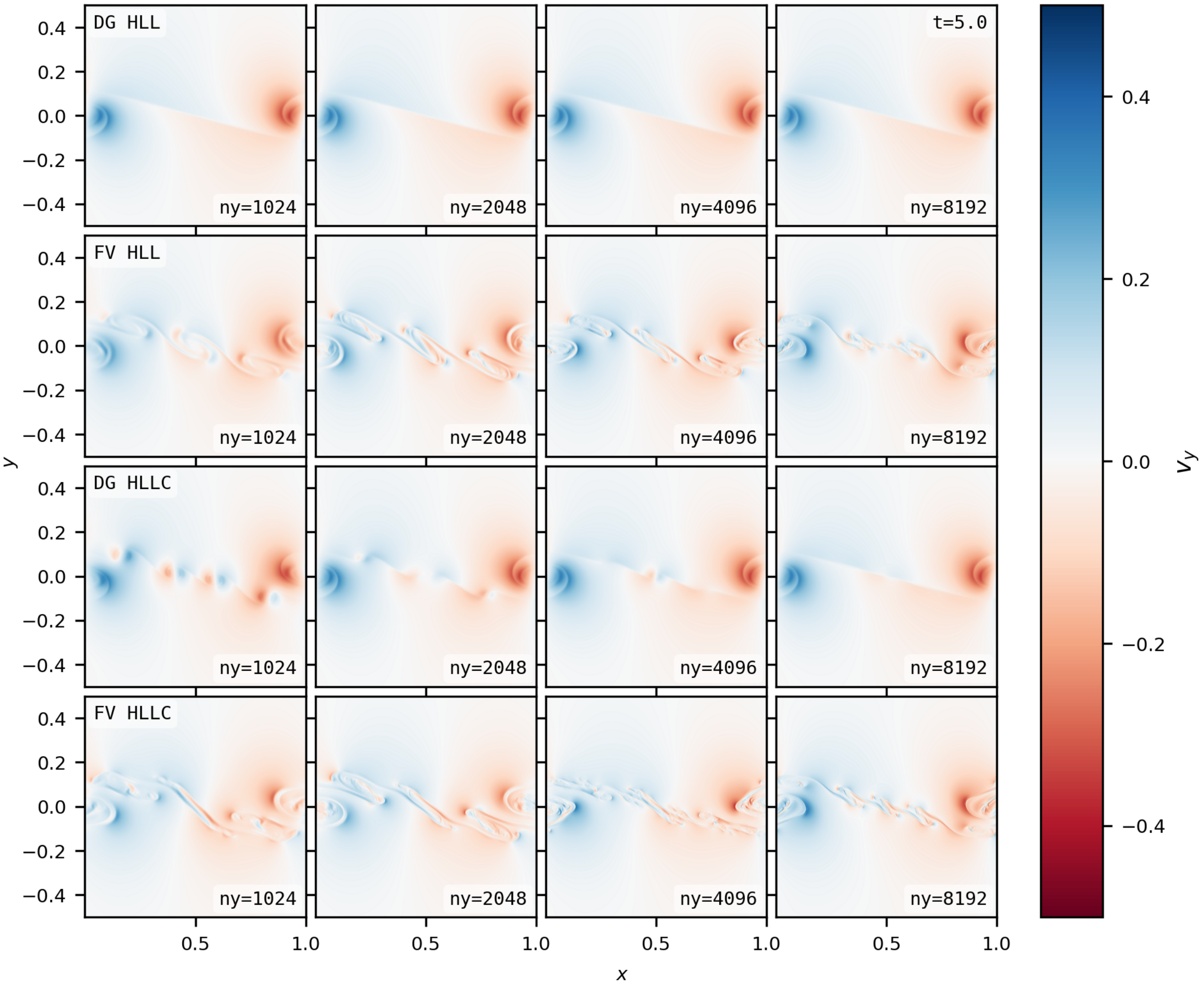


Figure 5.20: Snapshots of the transverse velocity at $t = 5.0$ from simulations of the relativistic Kelvin-Helmholtz instability using the method presented in this work using a 1st order basis in the first and third row and with the PLUTO finite volume MHD code with PLM reconstruction. We show results using the HLL Riemann solver in the top two rows and with HLLC for our code and with HLLD for PLUTO in the bottom two rows. We show the four highest resolution simulations across the columns, ranging from 512×1024 to 4096×8192 cells from left to right. Note that DG method has 4 times as many degrees of freedom with the 1st order basis, meaning that our 512×1024 simulation is comparable in degrees of freedom to the 1024×2048 simulation using PLUTO. At this later time once the instability has entered into the nonlinear growth phase, the DG method shows clear roll ups at all resolutions. Secondary vortices are suppress with higher resolutions and by the more diffusive HLL solver. In contrast, the PLUTO results show secondary instabilities through out the perturbation, although these diminish with resolution. Notably, the structure of the instabilities with the DG method versus the finite method are very different.

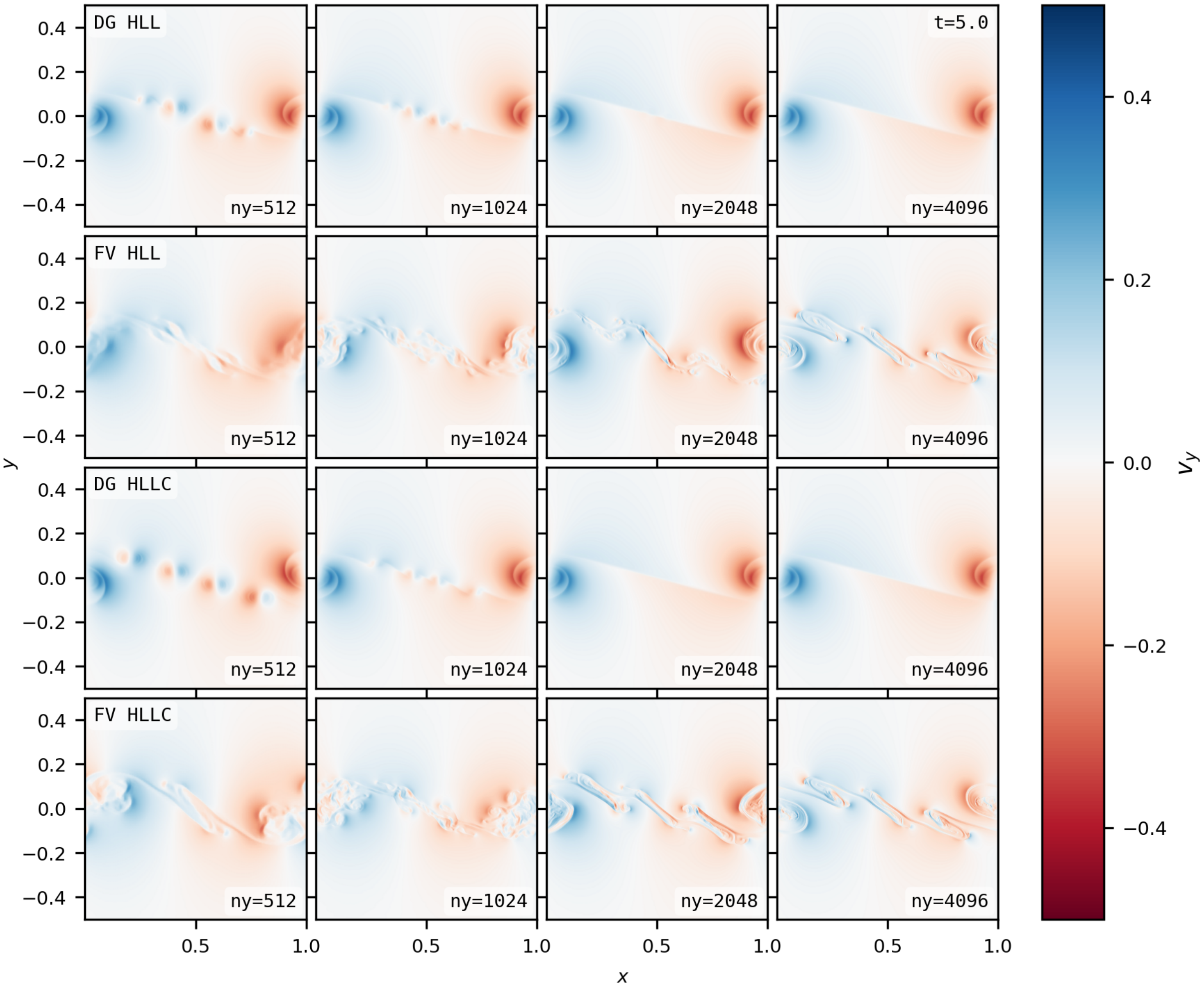


Figure 5.21: Snapshots of the transverse velocity at $t = 5.0$ from simulations of the relativistic Kelvin-Helmholtz instability using the method presented in this work using a 2$^{\mathrm{nd}}$ order basis in the first and third row and with the PLUTO finite volume MHD code with PPM reconstruction. We show results using the HLL Riemann solver in the top two rows and with HLLC for our code and with HLLD for PLUTO in the bottom two rows. We show the four highest resolution simulations across the columns, ranging from 512×1024 to 2048×4096 cells from left to right. Note that DG method has 4 times as many degrees of freedom with the 1$^{\mathrm{st}}$ order basis, meaning that our 512×1024 simulation has degrees of freedom between the 1024×2048 simulation and 2048×4096 simulation using PLUTO. The suppression of secondary vortices with our DG method is enhanced with basis order 2 compared to basis order 1, requiring fewer cells and degrees of freedom. Secondary instabilities still appear with the finite volume method, largely unaffected by the increase in method order.

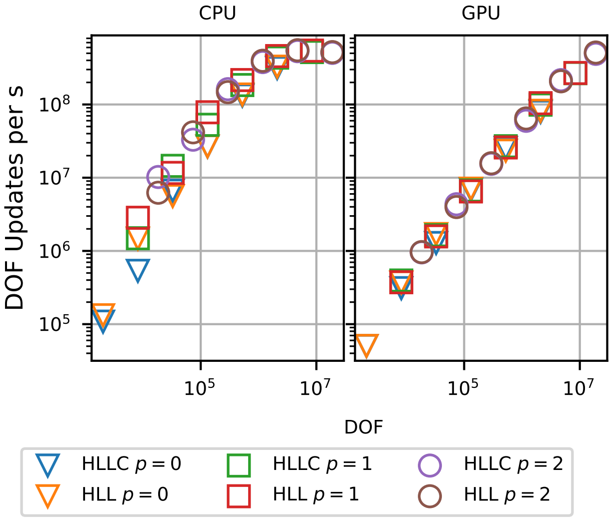


Figure 5.22: Performance of the code modeling the Kelvin Helmholtz instability from section §5.4.5, plotting updates to degrees of freedom per second versus degrees of freedom, using 1024 cores spread across 22 dual socket nodes with Intel Xeon Platinum 8268 CPUs (comprising approximately ~ 88TFLOPS in total) in the left column and using 32 NVidia Tesla V100-SXM2 GPUs (comprising approximately ~ 250TFLOPS in total) spread across 8 nodes on the right, where the peak computational throughput of the GPUs used are roughly three times the peak computational throughput of the CPUs. The computational resources for both tests was chosen to accommodate the memory needed for the largest simulation in the suite. We show profiling results with the HLLC and HLL Riemann solvers and with the 0^{th}, 1^{st}, and 2^{nd} order bases, between which we see little difference in performance. Comparing between the CPU and GPU runs, however, we see that the CPU performance becomes saturated at around 10^6 DOFs while the GPUs have not saturated the performance, even with simulations using more than 10 times the degrees of freedom.

of GPU FLOPS. i.e. our implementation is failing to meet computation or memory bounds, where the arithmetic-intensity of discontinuous-Galerkin methods lead to typically memory bound algorithms.

These performance characteristics are consistent with insufficient work within individual kernels to offset kernel launch overhead, as was the case in the K-Athena magnetohydrodynamics code presented in Grete et al. (2021a) and was resolved in the Parthenon adaptive-mesh refinement framework and AthenaPK magnetohydrodynamics code presented in Grete et al. (2022). We performed an informal profiling of our method evolving the Kelvin-Helmholtz instability on a single V100 GPU using `nvprof`. With a timeline trace, we verified for problem sizes that occupied the entirety of the HBM memory of a single GPU that a large percentage of compute time on the

GPU, $> 70\%$, was dominated by short duration 4μs kernel calls. These kernel durations would be consumed by kernel launch overhead from within the CUDA API.

With the launch of each kernel, between the APIs, drivers, and hardware a few microseconds are spent launching the kernel on the GPU. Unless sufficient work is done within each kernel, this launch overhead will dominate runtime. For our implementation, the work done within individual kernels can be increased with more degrees of freedom. However, the GPU has insufficient memory to allow enough work to hide kernel launch overhead, hence the underutilization of the GPU. In Parthenon and AthenaPK, this kernel launch overhead was hidden by fusing together the work from multiple kernels into fewer, larger kernelsGrete et al. (2021a). Similar improvements would be needed for our implementation in order to saturate GPU performance.

5.5 Summary

We have presented a scheme to evolve the relativistic hydrodynamics equations using a discontinuous-Galerkin method. Within our scheme, we have developed a robust method for enforcing physicality of the conserved state via a operator. Our presentation of the method includes relativistic HLL and HLLC Riemann solvers, multiple methods for recovering the primitive variables from conserved variables with the ideal equation of state, and the Taub-Matthews approximation to the Synge equation of state, using physical units that keep factors of c. We implement the method using the Kokkos performance portability library, which allows us to run CPUs and GPUs supported by Kokkos.

The novel physicality-enforcing operator in the work allows evolution of shocks with high-order basis methods. The operator strictly enforces positive density and pressure and subluminal velocities on all basis points within a cell by smoothing nonphysical points towards the physical volume average. Additionally, the method conserves volume averages of conserved variables.

In our exploration of methods to recover primitive variables from conserved variables when using an ideal equation of state, we found that the iterative method from Riccardi & Durante (2008) was faster, more robust, and more accurate than the analytical method from Ryu et al. (2006), consistent with findings from Riccardi & Durante (2008). The iterative method for ideal gases presented here recovers the primitive variables by solving a quartic as described in Eq. 5.58, which provides more digits of precision in simultaneously in sub-relativistic and ultra-relativistic regimes compared to solving in terms of the velocity or Lorentz factor. Additionally, the Newton-Raphson method as applied to Eq. 5.59 gives comparable accuracy to the analytic method in under 10 iterations, as is explored in Fig. 5.2. More iterations allow a more accurate recovery with the iterative method compared to the analytic method. In the case of our implementation, the iterative method is faster to compute for $\gamma < 10$ on CPUs and always faster on GPUs except in trivial cases.

Conversely, in our exploration of methods to recover primitives variables from conserved variables with the Taub-Mathews equation of state, the analytical method detailed in Ryu et al. (2006) was faster than the iterative method implemented in this work. With the Taub-Mathews equation of state, recovering the primitives requires solving a cubic equation, which has a much simpler analytical solution compared to the quartic equation for the ideal gas. Solving this cubic equation iteratively requires a bounded root solver, where we use Brent's method in this work. The iterative method we implemented for the Taub-Matthews equation of state requires many more iterations to achieve acceptable accuracy than the iterative solver for the ideal gas. As such, we found the analytic method for the Taub-Matthews equation of state to outperform the iterative method in terms of time to solution and accuracy on both CPUs and GPUs.

With this method, we ran several standard test problems, including linear waves, 1D and 2D Riemann problems, and the relativistic Kelvin-Helmholtz problem. The iterative conserved-to-primitive solver facilitated more relativistic problems and the physicality-enforcing operator allowed stable evolution with higher order bases for problems with shocks. In some test problems with a shock moving transverse to an outflow boundary conditions, we saw some non-physical boundary effects when using a 2nd order basis.

In our tests of the Kelvin-Helmholtz instability, comparing to results using a finite volume reference scheme (Mignone et al., 2011), the discontinuous-Galerkin method presented in this work can better suppress secondary vortices and instabilities compared to the finite volume method. Our method works best with a 1st order basis, which is a 2nd order method in space and time, since the

0th order basis is slow to grow the instability with low resolution while with the 2nd order basis boundary effects enter in at the outflow boundaries with high resolution.

In the tests of the Kelvin-Helmholtz instability and some of the 2D Riemann problems, we saw numerical boundary effects enter at the outflow boundary conditions, which increased with higher resolutions. Further development of the outflow boundaries with higher order bases is required.

Finally, in the exploration of the performance of our implementation evolving the Kelvin-Helmholtz instability, we found that our implementation is unable to saturate performance on GPUs before the problem size grows too large for the GPU memory. From these performance results and profiling using `nvprof`, we suspect that insufficient work inside individual kernels, leading to kernel launch overhead dominating runtime, is responsible for the lack of performance on GPUs. Combining the work from multiple kernels – as was done in the Parthenon framework presented in Grete et al. (2021a) – would be needed for our implementation in order to saturate GPU performance.

Acknowledgments

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This work was supported in part by LDRD project #209240. SAND # SAND2022-1601 O.

CHAPTER 6

SIMULATIONS OF GALAXY CLUSTERS WITH MAGNETIC AGN JET FEEDBACK

6.1 Motivation

The hot, diffuse plasmas called the intracluster medium (ICM) comprising the majority of baryonic mass in galaxy clusters is known to maintain significant magnetic fields (Carilli & Taylor, 2002; Govoni & Feretti, 2004; Donnert et al., 2018). These magnetic fields have been observed via a number of techniques which include: inference from synchrotron emitting radio relics; the magnetic Sunyaev-Zeldovich (SZ) effect, where magnetic fields lead to modified electron energy distributions which leads to steeper radial variance the X-rays from inverse Compton scattering of photons from the cosmic microwave background (CMB) off electrons in the ICM (Hu & Lou, 2004); Faraday rotation, where the magnetic fields rotate the polarization of photons passing through the magnetic fields of the galaxy cluster (Clarke et al., 2001; Carilli & Taylor, 2002; Clarke, 2004); and cold fronts where magnetic fields suppress a Kelvin Helmholtz instability, preserving a sharp discontinuity in the gas (Vikhlinin et al., 2001a,b; Ghizzardi et al., 2010). These measurements of the magnetic fields in galaxy clusters, however, do not directly give the magnetic field strengths or geometry but instead inform inferences of these properties using assumptions of magnetic length scales and models. Although the magnetic fields aren't dominant over gravitational forces or gas pressure in the ICM, they are nevertheless believed to be dynamically important, maintaining field strengths from $\sim 1 - 50\mu$m (Vacca et al., 2018; Donnert et al., 2018). The amplification of the cluster magnetic fields to their present values is likewise an open question, where shocks, turbulence, and jets launched by active galactic nuclei (AGN) are likely to play large roles (Donnert et al., 2018). Computational modeling is a cornerstone for inferring magnetic fields, understanding their dynamical role in the ICM, and how magnetic fields are created and amplified in galaxy clusters.

One aspect of specific aspect that can be addressed by global galaxy cluster simulations is how the magnetized jets launched by AGN can affect magnetic fields and energy balance within the ICM (Li et al., 2006; Wang et al., 2020). The jets emitted by AGN are collimated by the magnetic fields generated by the AGN accretion disk and are thus inherently defined by their magnetic fields. Observations of these jets have inferred a helical *magnetic tower* structure of the AGN jet (Gabuzda, 2021). Simulations of these magnetic AGN jets in isolation and with their impact on the galaxy cluster has been performed in the past (Li et al., 2006; Gan et al., 2017; Martí, 2019; Barniol Duran et al., 2017) although their role in the self-regulation of AGN feedback and cooling has been under-investigated. Kinetic jet models are able to self-regulate (Meece Jr, 2016; Meece et al., 2017) while thermal-only heating models have difficulty self-regulating while also maintaining a realistic galaxy cluster, as was explored in Chapter 2. To rectify this gap in exploration, my current work is focused on simulations comparing magnetized AGN feedback to kinetic jet and thermal feedback to ascertain how well magnetized AGN feedback triggered by cold gas accretion can self-regulate in galaxy clusters.

To best explore this question, we intend to perform the highest resolution simulations of galaxy clusters to date, using world-class supercomputers. Said supercomputers, however, use GPUs from a number of different manufacturers for the majority of their computational throughput. Thus, a performance portable magnetohydrodynamics code such as the K-Athena code presented in Chapter 4 is needed to utilize these GPU supercomputers. K-Athena only supports uniform grids, however, and simulating galaxy clusters with high resolution near the galaxy cluster core requires adaptive mesh refinement (AMR) – where the resolution of the simulation grid is increased near fine features in the system and decreased where the flow is smooth. Resolving the entire ~ 4 Mpc box of a galaxy cluster simulation down to 10 pc size grid cells would require ~ 10 EB (exabytes) of memory disk space to store one output whereas the current largest supercomputer provides ~ 100 PB (petabytes). AMR allows effectively the same accuracy of resolution for a fraction of the data volume, allowing us to resolve a central box of ~ 40 kpc around the AGN down to 10 pc.

Although implementing AMR into K-Athena would be possible, such a project would be challenging for a small university team. Thus, we collaborated with Los Alamos National Laboratory and researchers at the Institute of Advanced Study to develop Parthenon, a performance portable AMR framework. Using this new framework we developed the performance portable AMR MHD code AthenaPK, a successor to K-Athena that can perform these AMR simulations of magnetized galaxy cluster with magnetized AGN feedback.

6.2 Methodology

6.2.1 Simulation Setup

The exascale galaxy cluster simulations we intend to run will use a Cartesian grid in a cubic volume with side length of 3.2 Mpc, with 128^3 cells in the base grid of a static mesh refinement hierarchy. We enforce 3 levels of refinement with $[-400, 400]^3$kpc (where the root grid is the 0th level), 5 levels of refinement on $[-100, 100]^3$kpc, and 11 levels of refinement on $[-12.5, 12.5]^3$kpc, giving us $\sim$ 12 pc resolution on the finest grid. We are currently testing simulations with the physics discussed below with lower resolutions that fit into local supercomputer resources. These test simulations will inform our upcoming simulation campaign on exascale systems.

Cosmological expansion is neglected in these simulations. We use a ΛCDM model to get the virial mass of the NFW halo and to set its gas temperature, following Meece et al. (2017). We set redshift $z = 0$ at initialization with $\Omega_M = 0.3$, $\Omega_\Lambda = 0.7$, and $H_0 = 70$ km s^{-1}. We note that the precise details of the cosmological model will not impact explorations of the baryonic physics in the halo core, which is our primary interest.

6.2.1.1 Gravitational Potential

The gravitational potential has three components: a dark matter halo profile, a brighest cluster galaxy (BCG) with a mass profile, and a supermassive black hole (SMBH). We chose parameters for each of these to reflect a typical galaxy cluster. The dark matter follows the NFW profile (Navarro et al., 1997), using $M_{NFW} = 1 \times 10^{15}\mathrm{M}_\odot$ for the mass within the virial radius and a concentration parameter $c_{NFW} = 6$. The gravitational field from the NFW profile takes the form

$$g_{\mathrm{NFW}}(r) = \frac{G}{r^2} \frac{M_{NFW}\left[\ln\left(1 + \frac{r}{R_{NFW}}\right) - \frac{r}{r+R_{NFW}}\right]}{\ln\left(1 + c_{NFW}\right) - \frac{c_{NFW}}{1+c_{NFW}}}. \tag{6.1}$$

The scale radius R_{NFW} for the NFW profile is computed from

$$R_{NFW} = \left(\frac{M_{NFW}}{4\pi\rho_{NFW}\left[\ln\left(1 + c_{NFW}\right) - c_{NFW}/\left(1 + c_{NFW}\right)\right]}\right)^{1/3} \tag{6.2}$$

where the scale density ρ_{NFW} is computed from

$$\rho_{NFW} = \frac{200}{3}\rho_{crit}\frac{c_{NFW}^3}{\ln\left(1 + c_{NFW}\right) - c_{NFW}/\left(1 + c_{NFW}\right)}. \tag{6.3}$$

The critical density ρ_{crit} is computed from

$$\rho_{crit} \equiv \frac{3H_0^2}{8\pi G}. \tag{6.4}$$

We use a Hernquist BCG profile

$$g_{BCG}(r) = G\frac{M_{BCG}}{R^2}\frac{1}{\left(1 + \frac{r}{R}\right)^2} \tag{6.5}$$

with $M_{BCG} = 10^{11} M_{\odot}$ and $R_{BCG} = 4$ kpc. We include the gravitational field from a SMBH black hole with $M_{SMBH} = 4 \times 10^8 M_{\odot}$ at the center of the cluster halo.

6.2.1.2 Entropy Profile

Initial entropy profile of the gas follows the form

$$K \equiv \frac{k_b T}{n_e^{2/3}} \tag{6.6}$$

for the specific entropy, where k_b is Boltzmann's constant, T is the temperature, and n_e is the electron density, and is initialized follows a power law

$$K(r) = K_0 + K_{100}\left(r/100\,\mathrm{kpc}\right)^{\alpha_K}, \tag{6.7}$$

as introduced in the ACCEPT database (Cavagnolo et al., 2009). We use $K_0 = 10.0\,\mathrm{keV\ cm}^2$, $K_{100} = 150.0\ \mathrm{keV\ cm}^2$, and $\alpha_K = 1.1$ for the initial entropy profile, which is a typical profile for a CC cluster.

6.2.1.3 Initial Pressure and Density (Hydrostatic Equilibrium)

We compute the initial pressure and density by enforcing the initial cluster to be in hydrostatic equilibrium given the gravitational profile described above and the ACCEPT-like entropy profile, assuming an ideal gas with adiabatic index $\gamma = 5/3$. Additionally, to close the set of equations to define the initial gas profile we fix the density at $r = 2000$ kpc to $\rho = 10^{-28}$ g cm^{-3}.

6.2.1.4 Linearly Interpolated Tabular Cooling

We use a sub-cycling cooling method using a linearly interpolated tabular cooling function. Over each hydrodynamic cycle, we integrate the internal energy with cooling using an RK45 method, where the difference between the fourth order and fifth order estimations is used to adjust the subcycle. When the relative error in the change in internal energy over a subcycle is greater than 10^{-5}, we redo the subcycle. We limit the minimum subcycle time step to be $1/100$ the fluid time step. Additionally, we limit the fluid time step to be no greater than $1/10$ of the cooling time within any cell. We use the cooling table from Schure et al. (2009) using solar metallicity.

We use a helium mass fraction $\chi = 0.25$, with the remaining baryonic mass being hydrogen and electrons, which allows temperature T to be defined from density ρ and pressure P following

$$T = \frac{\mu m_h}{k_B} \frac{P}{\rho}. \tag{6.8}$$

where m_h is the atomic mass of hydrogen, k_B is Boltzmann's constant, and μ is the mean particle mass per m_h, found by

$$\mu = \left[\frac{3}{4}\chi + (1 - \chi)^2 \right]^{-1}. \tag{6.9}$$

6.2.1.5 Precessing Jet Coordinates

For injection of kinetic feedback by the AGN, initialization of the magnetic tower field, and feedback from the magnetic tower, we assume a precessing AGN jet and so employ coordinate transforms to convert Cartesian coordinates relative to the simulation frame to cylindrical coordinates relative to the precessing jet and transform cylindrical vector fields relative to the precessing jet to Cartesian coordinates relative to the simulation frame.

First, we define axes for Cartesian coordinates relative to the jet. Let $\phi_{jet} = \phi_{0,jet} + \omega_{\phi,jet} t$ be the azimuthal angle of the jet (relative to $\hat{x}$), where $\phi_{0,jet}$ is the initial azimuthal angle and $\omega_{\phi,jet}$ is the precession frequency, and let θ_{jet} by the inclination angle of the jet. The axis of the jet points along the vector $\hat{n} \equiv \left(1, \theta_{jet}, \phi_{jet}\right)$ in spherical coordinates relative to the simulation frame.

Using Sympy to generate the coordinate transforms, a position with simulation Cartesian coordinates $(x_{sim}, y_{sim}, z_{sim})$ has the following Cartesian coordinates relative to the jet

$$x_{jet} = x_{sim} \cos\left(\phi_{jet}\right) \cos\left(\theta_{jet}\right) + y_{sim} \sin\left(\phi_{jet}\right) - z_{sim} \sin\left(\theta_{jet}\right) \cos\left(\phi_{jet}\right) \tag{6.10}$$

$$y_{jet} = -x_{sim} \sin\left(\phi_{jet}\right) \cos\left(\theta_{jet}\right) + y_{sim} \cos\left(\phi_{jet}\right) + z_{sim} \sin\left(\phi_{jet}\right) \sin\left(\theta_{jet}\right) \tag{6.11}$$

$$z_{jet} = x_{sim} \sin\left(\theta_{jet}\right) + z_{sim} \cos\left(\theta_{jet}\right). \tag{6.12}$$

and the following cylindrical coordinates relative to the jet

$$r = \sqrt{x_{jet}^2 + y_{jet}^2} \tag{6.13}$$

$$\theta = \arctan \frac{y_{jet}}{x_{jet}} \tag{6.14}$$

$$h = z_{jet}. \tag{6.15}$$

Given a vector in cylindrical coordinates relative to the jet (such as the magnetic vector potential, magnetic field, or kinetic jet velocity) denoted by (v_r, v_θ, v_h) at position (r, θ, h), can be expressed in Cartesian coordinates relative to the jet following

$$v_{x,jet} = v_r \cos(\theta) - v_\theta \sin(\theta) \tag{6.16}$$

$$v_{y,jet} = v_r \sin(\theta) + v_\theta \cos(\theta) \tag{6.17}$$

$$v_{z,jet} = v_h. \tag{6.18}$$

This vector in simulation Cartesian coordinates can then be found from multiplying the vector $(v_{x,jet}, v_{y,jet}, v_{z,jet})$ by the DCM matrix to convert from jet Cartesian coordinates to simulation

Cartesian coordinates:

$$\begin{bmatrix} v_{x,sim} \\ v_{y,sim} \\ v_{z,sim} \end{bmatrix} = \begin{bmatrix} \cos\left(\phi_{jet}\right)\cos\left(\theta_{jet}\right) & -\sin\left(\phi_{jet}\right)\cos\left(\theta_{jet}\right) & \sin\left(\theta_{jet}\right) \\ \sin\left(\phi_{jet}\right) & \cos\left(\phi_{jet}\right) & 0 \\ -\sin\left(\theta_{jet}\right)\cos\left(\phi_{jet}\right) & \sin\left(\phi_{jet}\right)\sin\left(\theta_{jet}\right) & \cos\left(\theta_{jet}\right) \end{bmatrix} \begin{bmatrix} v_{x,jet} \\ v_{y,jet} \\ v_{z,jet} \end{bmatrix} \tag{6.19}$$

6.2.1.6 Magnetic tower

We initialize the galaxy cluster with a pre-existing magnetic tower following the form described in Li et al. (2006). The magnetic fields are described in cylindrical coordinates as

$$B_r = B_0 2\frac{hr}{\ell^2}\exp\left(\frac{-r^2 - h^2}{\ell^2}\right) \tag{6.20}$$

$$B_\theta = B_0 \alpha \frac{r}{\ell}\exp\left(\frac{-r^2 - h^2}{\ell^2}\right) \tag{6.21}$$

$$B_h = B_0 2\left(1 - \frac{r^2}{\ell^2}\right)\exp\left(\frac{-r^2 - h^2}{\ell^2}\right) \tag{6.22}$$

where r is the distance from the jet axis aligned along $\hat{h}$, θ is the polar angle around $\hat{h}$, and h is the height from the assumed accretion disk. The parameter α controls the the relative strength between poloidal and toroidal fields, where $\alpha \sim 2.6$ corresponds to roughly equal poloidal and toroidal flux. Following Li et al. (2006), we use $\alpha = 20$, which corresponds to a strong toroidal flux consistent with a magnetic tower that is highly wound by a rotating accretion disk.

6.2.1.7 AGN Feedback

We include AGN feedback using thermal heating, kinetic jet, and magnetic tower models exploring different relative strengths. We divide the AGN feedback between the three mechanisms following

$$\dot{E}_{AGN} = \dot{E}_T + \dot{E}_K + \dot{E}_B = \left(f_T + f_K + f_B\right)\dot{E}_{AGN}. \tag{6.23}$$

where $\dot{E}_{AGN}$ is the total AGN feedback rate; $\dot{E}_T$, $\dot{E}_K$, and $\dot{E}_B$ are the total thermal, kinetic, and magnetic AGN feedback rates; and f_T, f_K, and f_B are the thermal, kinetic, and magnetic fractions of the total AGN feedback rate.

6.2.1.8 Thermal AGN Feedback

In the thermal feedback model, thermal energy is deposited at a uniform heating rate per volume within a sphere around the center of the halo where the presumed AGN resides.

$$\dot{e}_T\left(r\right) = \begin{cases} \frac{f_T \dot{E}_{AGN}}{(4/3)\pi R_T^3} & r \leq R_T \\ 0 & \text{otherwise} \end{cases} \tag{6.24}$$

where we use $R_T = 0.5$ kpc for the radius of thermal feedback.

6.2.1.9 Kinetic AGN Feedback

In the kinetic feedback model, kinetic energy and mass is injected above and below a presumed accretion disk inside an cylindrical jet. We align the jet along the z axis with a radius $R_K = 1$ kpc and extend it $H_K = 10$ kpc above and below the xy plane. The rate of mass injected by the jet is set proportional to the kinetic jet power $f_K \dot{E}_{AGN}$ divided by a kinetic jet efficiency parameter $\epsilon_K = 10^{-3}$ (Meece et al., 2017):

$$\dot{M}_K = \frac{f_K \dot{E}_{AGN}}{\epsilon_K c^2} \tag{6.25}$$

where c is the speed of light. The jet then injects a mass density

$$\dot{\rho}_K = \frac{\dot{M}_K}{2\pi R_K^2 H_{jet}} \tag{6.26}$$

with a jet speed

$$v_K = \sqrt{2\epsilon_K} c. \tag{6.27}$$

heading away from the xy plane The momentum density injected into the cluster is then

$$\dot{\mathbf{M}}_K\left(\mathbf{r}\right) = \begin{cases} \operatorname{sign}(z)\dot{\rho}_K v_K \hat{h} & \text{when } r \leq R_K \text{ and } |h| \leq H_K \\ 0 & \text{otherwise} \end{cases} \tag{6.28}$$

where r here is the distance from the jet axis and h is the signed height above or below the accretion disk. The injected kinetic energy per volume is

$$\dot{\mathbf{e}}_K\left(\mathbf{r}\right) = \begin{cases} \frac{1}{2}\dot{\rho}_K v_K^2 & \text{when } r \leq R_K \text{ and } |z| \leq H_K \\ 0 & \text{otherwise} \end{cases} \tag{6.29}$$

so that the total kinetic energy injected matches $f_K E_{AGN}$.

6.2.1.10 Magnetic AGN Feedback

In the magnetic feedback model, a magnetic field is deposited proportional to the magnetic tower field proposed in Li et al. (2006)

$$\mathcal{B}_r = \mathcal{B}_0 2\frac{hr}{\ell^2}\exp\left(\frac{-r^2-h^2}{\ell^2}\right) \tag{6.30}$$

$$\mathcal{B}_\theta = \mathcal{B}_0 \alpha\frac{r}{\ell}\exp\left(\frac{-r^2-h^2}{\ell^2}\right) \tag{6.31}$$

$$\mathcal{B}_h = \mathcal{B}_0 2\left(1-\frac{r^2}{\ell^2}\right)\exp\left(\frac{-r^2-h^2}{\ell^2}\right) \tag{6.32}$$

where r is the distance from the jet axis, h is the signed height above or below the accretion disk, ℓ is the length scale, α controls the ratio of polodial to torodial fields, and $\mathcal{B}_0$ is the strength of the magnetic field. A vector potential corresponding to this magnetic field can be written as

$$\mathcal{A}_r = 0 \tag{6.33}$$

$$\mathcal{A}_\theta = \mathcal{B}_0 \ell\frac{r}{\ell}\exp\left(\frac{-r^2-h^2}{\ell^2}\right) \tag{6.34}$$

$$\mathcal{A}_h = \mathcal{B}_0 \ell\frac{\alpha}{2}\exp\left(\frac{-r^2-h^2}{\ell^2}\right) \tag{6.35}$$

so that $\nabla \times \mathcal{A} = \mathcal{B}$. Constructing the magnetic fields from the vector potential is preferred to maintain $\nabla \cdot \mathbf{B}$ as close to zero as possible.

We apply magnetic fields from Equation 6.30 aligned to a precessing jet, and so the coordinate and vector transformations described in §6.2.1.5 as necessary to transform $(x, y, z) \longrightarrow (r, \theta, h)$ and $(\mathcal{B}_r, \mathcal{B}_\theta, \mathcal{B}_h) \longrightarrow (\mathbf{B}_x, \mathbf{B}_y, \mathbf{B}_z)$.

We use the magnetic field from Equation 6.30 to apply the initial magnetic field and to apply magnetic feedback, which feedback can be scaled to a specified field rate or power.

Initializing magnetic fields is simple. To inject a B_0 magnetic field, we set the initial magnetic field to $\mathbf{B} = \mathcal{B}|_{\mathcal{B}_0 = B_0}$. Injecting constant magnetic field increase is also simple. To inject a field rate of $\dot{B}_0$, we add a magnetic field $\dot{\mathbf{B}} = \mathcal{B}|_{\mathcal{B}_0 = \dot{B}_0}$

Injecting magnetic energy at a *specified power* is much more complicated since the existing magnetic field must be considered and both linear and quadratic contributions from the injected magnetic field must also be considered. Given an existing magnetic field $\mathbf{B}_n$ and a timestep Δt, we inject a magnetic field following the magnetic tower model with a strength B_p that must be determined so that the new magnetic field is

$$\mathbf{B}_{n+1} = \mathbf{B}_n + \Delta t \mathcal{B}|_{\mathcal{B}_0 = B_p}. \tag{6.36}$$

The change in total magnetic energy is then

$$\begin{aligned}
\Delta E_B &= \int_\Omega \frac{1}{2}\mathbf{B}_{n+1} \cdot \mathbf{B}_{n+1} - \frac{1}{2}\mathbf{B}_n \cdot \mathbf{B}_n \, dV & (6.37)\\
&= \frac{1}{2}\left[\int_\Omega \mathbf{B}_n \cdot \mathbf{B}_n + 2\Delta t \mathbf{B}_n \cdot \mathcal{B}|_{\mathcal{B}_0 = B_p} + (\Delta t)^2 \mathcal{B}|_{\mathcal{B}_0 = B_p} \cdot \mathcal{B}|_{\mathcal{B}_0 = B_p} - \mathbf{B}_n \cdot \mathbf{B}_n \, dV\right] & (6.38)\\
&= \frac{1}{2}\left[\int_\Omega \mathbf{B}_n \cdot \mathbf{B}_n + 2\Delta t \mathbf{B}_n \cdot \mathcal{B}|_{\mathcal{B}_0 = B_p} + (\Delta t)^2 \mathcal{B}|_{\mathcal{B}_0 = B_p} \cdot \mathcal{B}|_{\mathcal{B}_0 = B_p} - \mathbf{B}_n \cdot \mathbf{B}_n \, dV\right] & (6.39)\\
&= \Delta t B_p \int_\Omega \mathbf{B}_n \cdot \mathcal{B}|_{\mathcal{B}_0 = 1} \, dV \quad + \quad (\Delta t)^2 B_p^2 \int_\omega \frac{1}{2}\mathcal{B}|_{\mathcal{B}_0 = 1} \cdot \mathcal{B}|_{\mathcal{B}_0 = 1} \, dV & (6.40)
\end{aligned}$$

where Ω is the simulation domain. To determine the magnetic field strength B_p to be injected, the two integrals in Equation 6.40 corresponding the linear and quadratic contributions must first be computed (via reduction over the entire domain), then B_p can be determined by the quadratic formula (only one root should be positive).

For the case of magnetic field injection by the AGN, the change in magnetic energy is set to $\Delta E_B = \Delta t f_B \dot{E}_{AGN}$ and B_{AGN} is determined by the reductions above.

Applying a magnetic tower field injects a finite total magnetic energy even when applied over all space due to the exponential decay away from the AGN. The total magnetic energy E_B when applied over all space is given by

$$E_B = \int_0^\infty \int_0^{2\pi} \int_{-\infty}^\infty \frac{1}{2}\mathbf{B} \cdot \mathbf{B} r \, dh \, d\theta \, dr = B_0^2 \frac{\pi^{3/2}\left(\alpha^2 + 5\right)\ell^3}{8\sqrt{2}}. \tag{6.41}$$

6.2.1.11 AGN cold mass triggering

AGN feedback is triggered by cold mass around the presumed AGN. AGN triggering occurs within a $r_{acc} = 10$ kpc radius accretion zone around the AGN. Within the accretion zone, gas with a temperature below the user-defined threshold $T_{cold} = 5 \times 10^4$ K triggers AGN feedback. The mass accretion rate onto the AGN follows

$$\dot{M}_{AGN} = \int_{r<r_{acc}} \rho_{cold}(\mathbf{r})/t_{acc} \mathrm{d}V \tag{6.42}$$

where $\rho_{cold}(\mathbf{r})$ is equal to $\rho(\mathbf{r})$ in cells where $T(\mathbf{r}) \leq T_{cold}$ and 0 otherwise, and $t_{acc} = 100$ Myr is the accretion time scale. The total AGN feedback rate is then set to

$$\dot{E}_{AGN} = \epsilon_{AGN} \dot{M}_{AGN} c^2 \tag{6.43}$$

where $\epsilon_{AGN} = 10^{-3}$ is the cold mass triggering efficiency.

The accreted mass is removed from the simulation. Mass is only removed from cells within the accretion zone with a temperature below the cold gas temperature threshold. The density removed follows the rate

$$\dot{\rho}\ (\mathbf{r}) = \begin{cases} \rho(\mathbf{r})/t_{acc} & T(\mathbf{r}) < T_{cold} \\ 0 & \text{otherwise} \end{cases} \tag{6.44}$$

6.3 Current State of Simulations

Each of the components to initialize the galaxy cluster simulation and evolve the cluster with triggered AGN feedback have been individually tested and verified to work as expected. Integrated tests of all these components are underway. Analysis pipelines using YT (Turk et al., 2011) are also in development. Testing of the full magnetized galaxy cluster simulation set up is expected to be completed by early summer 2022 in time for exascale simulations later in the summer.

CHAPTER 7

SUMMARY AND FUTURE DIRECTIONS

The ultimate goal of this dissertation is to better understand the behavior of diffuse astrophysical plasmas, especially as applied to the intracluster medium (ICM), and to develop better numerical tools and methods to explore these plasmas. In this final chapter in Section 7.1 I first summarize each of the chapters comprising peer-reviewed or near submission work, which includes Chapters 2, 3, 4, and 5. I then describe the ongoing and future work of these projects in Section 7.2, including the many projects spawned by PARTHENON and ATHENAPK, the work at Sandia National Laboratories enabled by the relativistic discontinuous-Galerkin (DG) method I presented in Chapter 5, the ongoing magnetized galaxy cluster simulations and future additions to those simulations, and finally the work on magnetized jets in AGN accretion disks that I plan to explore as a postdoctoral fellow at Los Alamos National Laboratory.

7.1 Summary of Dissertation Work

7.1.1 Chapter 2: Tests of AGN Feedback Kernels in Simulated Galaxy Clusters

In Chapter 2, we explored the energy deposition of active galactic nuclei (AGN) feedback that is necessary to prevent cooling catastrophes within the cluster while maintaining a realistic entropy profile (Glines et al., 2020). To this end, we ran 91 simulations of idealized galaxy clusters with a simplified model of AGN feedback, abstracting the thermalization of AGN jets and magnetic fields as a spherically symmetric heating kernel balanced to the cooling within the cluster, testing a range of heating kernel profiles with varying degrees of central peaking (See Figure 2.2 and Table 2.1). We did not find a spherical heating kernel that produced both a quasi-stable galaxy cluster that did not undergo a cooling catastrophe and also kept an observationally realistic entropy profile. We did find that sharply centrally peaked heating kernels prevented cooling catastrophes by severely exceeding radiative cooling in the cluster core where cooling times are short compared to the lifetime of the cluster. These centrally overpowered heating kernels led to centrally inverted entropy profiles where the high central entropy was resistant to overcooling but was in-congruent with entropy profiles of observed galaxy clusters. We also found that weakly centrally peaked heating kernels kept realistic entropy profiles but failed to offset central cooling, leading to cooling catastrophes well under the observationally expected lifetimes for these clusters (See Figure 2.3).

Although these simulations did not conclusively rule out a thermal-only abstraction for AGN feedback, they do point towards more complex mechanisms than pure heating at play in self-regulating cool-core (CC) clusters. To explore such questions, we would like to explore high fidelity simulations with enough primary and secondary physics to realistically model the magnetized ICM and AGN jet, including at least magnetic fields and potentially non-ideal MHD effects, cosmic ray pressure, and possibly relativistic AGN jet velocities. Such simulations would also need at least an order of magnitude increase in computing resources in order to include the additional physics and increase resolution of the ICM. Enabling such simulations was the goal of the work presented in Chapter 4 developing K-Athena– a performance portable MHD code that can utilize upcoming supercomputers. This goal of high resolution magnetized galaxy cluster simulations is coming to fruition with the in-progress work presented Chapter 6.

7.1.2 Chapter 3: Magnetized Decaying Turbulence in the Weakly Compressible Taylor-Green Vortex

In Chapter 3, we explored the development of magnetized turbulence from the decay of a large scale flow, performing 9 simulations of the magnetized Taylor-Green vortex. The decaying turbulent plasma scenario models the growth of turbulence in the ICM due to large scale infrequent perturbations, such as from a galaxy cluster merger or an AGN outburst. These simulations are distinct from more commonly performed driven turbulence simulations, where a stochastic force is applied to the plasma, continually injecting energy at the injection scale. In this aspect, after turbulence is well developed in our simulations the energy spectrum of the simulations is uncontaminated by injected energy and is purely a result of the extant energy in the turbulence.

In these simulations, magnetic energy came to dominate over kinetic energy even when the initial magnetic field was small. We found that the magnetized turbulence developed from this decaying flow scaled following a $k^{-4/3}$ power law, flatter than the $k^{-5/3}$ power law for hydrodynamical turbulence with comparatively more energy at smaller scales in the magnetized turbulence, confirming results from related driven turbulence simulations (See Figure 3.4, Grete et al., 2018, 2021b). Using the energy transfer analysis developed by Grete et al. (2017), we explored the development of the energy spectrum from the energy transfers between kinetic and magnetic energy at different scales. The buildup of energy at smaller scales was aided by non-local energy transfer from large scale kinetic energy to all scales of magnetic energy via magnetic tension; a mechanism absent in hydrodynamical turbulence. In general, the magnetized turbulence behaved differently from hydrodynamical turbulence and thus should not be ignored in explorations of turbulence in the ICM.

7.1.3 Chapter 4: K-Athena: A Performance Portable Structured Grid Finite Volume Magnetohydrodynamics Code

In Chapter 4, we present the performance portable magnetohydrodynamics (MHD) code K-Athena, which is designed to enable computational astrophysics on the next generation of supercomputers while maintaining performance on traditional supercomputers. These new supercomputers will use graphics processing units (GPUs) from a number of different vendors for the majority of their computational throughput. At the same time, supercomputers using traditional CPUs will persist for the near future. Astrophysics codes capable of efficiently utilizing both GPUs from all manufacturers and traditional central processing units (CPUs) are needed to enable simulations on any given hardware that a computational astrophysicists might have access to. Performance portable codes provide this high performance on multiple architecture with a single code base, eliminating the development cost involved with creating and maintaining multiple versions for different architectures. K-Athena fulfills this need for a performance portable MHD code for uniform grids.

We demonstrated K-Athena running simulations on many of the largest supercomputers available at the time, showing high performance and efficiency on both CPUs and GPUs. K-Athena ran at high performance on the NVidia "Volta" V100 GPUs on Summit, achieving 76% parallel efficiency and attaining at peak a speed of 1.94×10^{12} cell updates/second. We also performed a roofline analysis to compute how efficiently K-Athena was using each level of memory, ultimately demonstrating that K-Athena was limited by the DRAM bandwidth on all architectures. This roofline analysis also allowed us to compute for K-Athena a 62.8% performance portability metric as measuring against theoretical performance limited by the DRAM bandwidth (Pennycook et al., 2016). The K-Athena code has been used for two papers to date (Grete et al., 2021b; Glines et al., 2021), including for the magnetized turbulence work shown in Chapter 3.

7.1.4 Chapter 5: Relativistic Discontinuous-Galerkin Hydrodynamics

In Chapter 5, we presented a robust method for evolving the special relativistic hydrodynamics equations using a DG method. In a DG method, the fluid within individual elements of the simulation domain is represented by linear combinations of a polynomials instead of just a cell average, allowing linear, quadratic, and higher order spatial contributions to be carried in each element. The methods have the dual advantage of being easily raised to arbitrary spatial orders by just increasing the order of the polynomial basis and for non-Cartesian mesh boundary conditions since stencils span a single cell (for which there exists an internal Cartesian-like grid) rather than multiple cells. These traits make them extremely valuable for terrestrial plasma simulations, where non-rectangular apparatuses introduce irregular boundaries that are best handled with unstructured meshes but still benefit from higher order methods.

The relativistic hydrodynamics method we present includes a unique exploration of solvers for the primitive recovery step – the non-trivial inversion of a transcendental equation to get the primitive variables from the conserved variables, which is an essential step to compute fluxes in both FV and DG methods. We show accuracy and speed performance for analytic and iterative solvers for both the ideal equation of state and the Taub-Matthews approximation to the Synge gas equation of state. In this exploration we show that the iterative methods for finding the roots of the quartic polynomial to invert the ideal equation of state is both faster and more accurate than the analytic method while the analytic methods for finding the roots of the cubic polynomial to invert the Taub-Matthews equation of state is conversely faster and more accurate. This reversed result may be a consequence of the high complexity in analytic quartic solvers and the simplicity of the Newton-Raphson method for the ideal equation of state versus the lower complexity of analytic cubic solvers and higher complexity in the bisection method needed for the Taub-Matthews solver.

For the method we developed a novel operator for maintaining the physicality of conserved states when running DG with higher orders (see Section 5.2.5). When shocks arise in DG simulations (and in FV simulations), non-physical conserved variables can be introduced after integrating fluxes around the discontinuity, especially when using spatial orders higher than 0th order. In a special relativistic hydrodynamics method these non-physical conserved variables can correspond to negative pressures or densities, superluminal velocities, or may correspond to imaginary or complex variables. These non-physical conserved variables can be screened using Equation 5.25. Within the algorithm we developed, when non-physical conserved variables are detected at interior points in a DG cell the physicality enforcing operator smooths all points within the cell towards the volume average so that all points are made physical as long as the volume average is physical. In practice, this operator was necessary to run any simulation with shocks with a spatial 1st order basis and higher.

We also compared the accuracy of this method relative to a FV scheme at evolving the relativistic Kelvin-Helmholtz instability using both the HLL and HLLC Riemann solvers and with 0th, 1st, and 2nd order bases. We found that the DG method we developed better suppresses non-physical secondary vortices and instabilities in the linear growth phase of the Kelvin-Helmholtz instability when compared to the FV method, especially when using higher order bases (see Figures 5.16, 5.17, 5.18, 5.19, 5.20, and 5.21). Non-physical boundary effects entered into the outflow boundary conditions with increased resolution, however, indicating that more development higher order outflow boundary conditions is needed.

7.2 Ongoing and Future Work

7.2.1 Parthenon and AthenaPK

The K-Athena MHD code, however, was only capable of running performance portable *uniform grid* simulations, where the resolution of the simulation is the same across the domain. This limited the applicability of K-Athena for studying galaxy clusters, a class of systems requiring high resolution near the central AGN and AGN jet but also including an expansive halo that can be simulated with low resolution. Although possible, the design of K-Athena would make implementation of performance portable AMR difficult for a small team. A performance portable MHD code with *adaptive mesh refinement* (AMR) was needed for galaxy cluster simulations. In fact, many simulated systems from both astrophysical and terrestrial plasma physics would benefit from performance portable AMR capabilities. Such an AMR code would impact a large cross-section of the computational plasma community.

To fulfill this need, we founded a collaboration to develop the Parthenon AMR framework (`https://github.com/lanl/parthenon`): a performance portable AMR framework based on the AMR implementation in Athena++ but tuned for performance portability on GPUs and CPUs (Grete et al., 2022). Originally conceived as an AMR capable successor to K-Athena, Parthenon has gone on to be the basis for many codes in the computational plasma astrophysics community. These include AthenaPK (`https://gitlab.com/theias/hpc/jmstone/athena-parthenon/athenapk`, Athena-Parthenon-Kokkos) an AMR-capable successor to K-Athena; Phoebus (`https://github.com/lanl/phoebus`), a general relativistic (GRMHD) code with neutrino radiation for modeling neutron star mergers and intermediate mass black hole candidates; KHARMA, another GRMHD code to be used for interpretation of black hole imaging via the Event Horizon Telescope as part of a 2022 INCITE award; RIOT, a Los Alamos National Laboratory-based multiphysics code (Grete et al., 2022); and likely more codes yet to be developed. This collaboration, which K-Athena inspired, will enable exascale simulations of the ICM, galaxy clusters, magnetized turbulence, the formation of intermediate mass black holes, AGN accretion disks, black hole imaging, planet formation, and many terrestrial plasma systems. The success of K-Athena and later Parthenon has also inspired Kokkos integration into other codes, including Athena-K, a GRMHD code in development at the Institute of Advanced Science, and Enzo-E (Bordner & Norman, 2018).

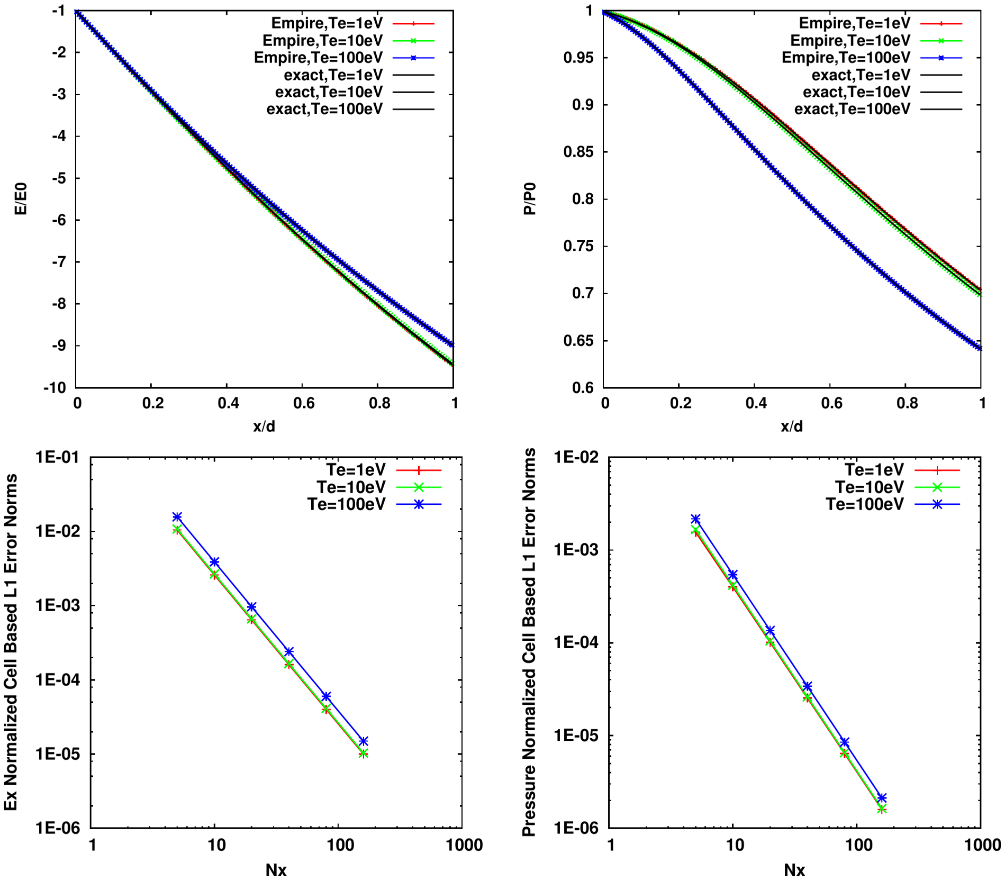
7.2.2 Relativistic DG Methods

The merit of the algorithm we developed has already been demonstrated in two upcoming papers from Sandia National Laboratories on which I am co-author (Roberds et al., 2022; Hamlin et al., 2022), which relied on this method and my implementation for different flavors of extended relativistic MHD methods. Both papers use this relativistic method as a basis for a relativistic two-fluid MHD scheme. In the relativistic two-fluid MHD equations, the electrons and ions of the plasma comprise two relativistic fluids with distinct densities, flow velocities, and pressures. These two fluids are coupled together via Maxwell's equations and Ohm's law (Amano, 2016).

Roberds et al. (2022) uses this two-fluid method to study electron emission across a warm diode – a scenario where the electron species is accelerated across a gap via injected kinetic energy and injected with sufficient thermal energy to be non-negligible to the injected kinetic energy. The solution using the two-fluid MHD method was compared against a semi-analytic model for the 1D warm diode problem and found to converge to 2$^{\text{nd}}$ order accuracy as was expected for this problem (see Figure 7.1). Preliminary results are reported in Laity et al. (2021).

Hamlin et al. (2022) uses the two-fluid method for 2D numerical simulations of a magnetron, a device that converts electrostatic potential in the electron population into microwave energy. That work compares the fluid approach to the PIC method conventionally used for that system. Results are awaiting publication.

Lessons learned from developing the relativistic hydrodynamics method for DG will be applied to relativistic MHD algorithms implemented in AthenaPK for future projects. These methods and implementations yet to be developed will be the basis for studies and simulations of relativistic AGN jet feedback to determine whether the relativistic nature of the jet has an impact on energy



©Sandia National Laboratories
Laity et al. (2021)

Figure 7.1: Electric field (top left) and pressure (top right) along the 1D warm diode with electron temperatures $T_e = 1, 10, 100$ eV using the relativistic two-fluid MHD DG method with my contributions in red, green and blue for each temperature and in black showing the exact solutions with a semi-analytic model. L1 error in electric field (bottom left) and pressure (bottom right) of the relativistic two-fluid MHD DG method to the exact solution, showing 2$^{\text{nd}}$ order convergence as was expected for the second-order accurate fluid solver. Figures taken from Laity et al. (2021).

deposition and thermalization within a magnetized ICM.

7.2.3 Simulations of Magnetized Galaxy Clusters

In Chapter 6 I share my current work developing simulations of magnetized AGN jets within a magnetized galaxy cluster. In order to achieve resolutions higher than previously possible in galaxy cluster simulations with AGN feedback we developed Parthenon, a performance portable AMR framework, and on top of that AthenaPK, a performance portable MHD code with AMR. These code developments in performance portability will allow us to run on more architectures and specifically on the GPU supercomputers comprising the highest echelons of current and near-future computing resources available. The unique scales of these computing resources will enable higher fidelity galaxy cluster simulations than previously explored, giving us better tools to examine the thermalization of AGN feedback within the ICM.

AthenaPK currently implements all the necessary components to explore high resolution simulations studying the magnetic aspect of AGN feedback, which is discussed in Chapter 6. These components include a new precessing magnetic tower injection model for AGN feedback, which allows exploration of whether precessing magnetic towers can self-regulate a CC cluster like a precessing kinetic jet (Meece Jr, 2016). These simulations and their analysis will run in summer 2022, with publication of results expected later this year.

With this performance portable MHD code as a base, we can add a variety of additional physics while keeping the simulations computationally feasible. The first addition will be cosmic rays, which may play an important role in self-regulating AGN feedback. Non-thermal, relativistically moving electrons and especially protons comprising these cosmic rays have long been suspected to play a key role in offsetting cooling and preventing cooling flows in the ICM, proving additional heating and pressure (Loewenstein et al., 1991; Ando & Nagai, 2008). Although the cosmic ray energy density is low compared to the thermal energy density in the ICM (Dunn & Fabian, 2004), they may be a key factor in AGN feedback by elongating and inflating AGN-created bubbles by exerting anisotropic pressure along magnetic field lines (Guo & Oh, 2008; Guo & Mathews, 2011). AGN feedback itself injects cosmic rays into the ICM by creating shocks and turbulence where charged particles can be accelerated to relativistic velocities via the first and second order Fermi processes (Krymskii, 1977; Bell, 1978; Bustamante et al., 2010). Thus, cosmic ray injection may be an important component of a self-regulating AGN.

The implementation of cosmic rays in AthenaPK will follow Jiang & Oh (2018), which has a publicly available implementation in AthenaPK's parent code Athena++. We will then extend the current simulations campaign exploring magnetized AGN feedback to include the injection of cosmic rays to see how they affect self-regulation of the AGN.

Beyond the addition of cosmic ray pressure, we will also investigate how using a non-ideal

MHD that better describes the plasma of the ICM might affect AGN feedback and self-regulation. Specifically, we will explore Braginskii MHD, which includes anisotropic transport of particles not present in ideal MHD, which introduced anisotropic heat conduction and anisotropic pressure along magnetic field lines (Braginskii, 1965). This model of the plasma better reflects the weakly collisional nature of the ICM (Reynolds, 2018). In the ICM, these non-ideal effects may play a role in magnetized turbulence (Kunz et al., 2011; Ruszkowski & Oh, 2011), the amplification of magnetic fields (St-Onge et al., 2020), and may bring a small but non-negligible amount of heating from cluster outskirts into the core (Voigt et al., 2002; Voigt & Fabian, 2004; Ruszkowski & Oh, 2011; Yang & Reynolds, 2016b). This more accurate representation of the ICM has been of keen interest in the last decade (Ruszkowski & Oh, 2011; Berlok et al., 2020) and will be a key feature for high fidelity simulations of galaxy clusters.

7.2.4 AGN Accretion Disk Channel for Intermediate Mass Black Holes

Being one of the only performance portable AMR frameworks available, the PARTHENON library is poised to impact many computational studies in astrophysical and terrestrial plasmas. ATHENAPK is also likely to be applied to many different systems in the near future. With more and more GPU supercomputers coming online, there are ample computational resources available with diverse GPU architectures but few codes that can use them.

One such area of exploration, headed by scientists at Los Alamos National Laboratories and to which I will be contributing, is the formation of intermediate mass black holes via AGN accretion disks.

With the advent of gravitational wave observatories such as the Laser Interferometer Gravitational-Wave Observatory (LIGO), we now have unprecedented access to the masses of previously unobservable black holes via binary black hole (BBH) mergers. In the observation of GW190521 by LIGO, we observed an 85 $M_\odot$ black hole merge with a 66 $M_\odot$ black hole creating a 142 $M_\odot$ black hole, the heaviest BBH merger to date (LIGO Scientific Collaboration and Virgo Collaboration et al., 2020). This merger poses theoretical inconsistencies since black hole masses in the $\sim 60 - 120$ M$_\odot$ mass gap are excluded by conventional theories of black hole formation via pair instability supernovae, despite both progenitors in the BBH falling into this *black hole mass gap* (Woosley, 2017). The mechanism by which these BBHs may have formed is as yet poorly understood (Koliopanos, 2018), although several formation channels have been proposed including primordial black holes (Lacroix & Silk, 2018), Population III stars (massive stars formed from metal poor gas in the early universe; Lacroix & Silk, 2018), mergers of stellar-mass black holes in dense environments (Rose et al., 2021), and super-Eddington accretion (accreting faster than the traditional limit where radiation pressure emitted by accreting gas balances gravitational pull) in dense environments (Ogawa et al., 2017; Toyouchi et al., 2021).

One channel of particular interest is the *AGN channel*, where stellar-mass black holes can accrete at super-Eddington rates in the gas-rich environment of an AGN accretion disk (McKernan et al., 2012, 2014). Such regions in the AGN accretion disk could form multiple > 50 M mass black holes that could produce mergers such as GW190521. Of benefit to observational verification, said super-Eddington accretion and the jets emitted from a mergers in an accretion disk should have a signature as the jet breaks out of the AGN accretion disk (Zhu et al., 2021).

However, there are multiple aspects of the AGN channel that still need to be studied to determine how to identify the combined the gravitational wave and electromagnetic signature on a BBH merger embedded in an AGN accretion disk. Many of these aspects will be studied with performance portable AMR simulations built upon the Parthenon library that I helped enable, including GRMHD simulations of the BBH using Phoebus and jet simulations within the AGN accretion disk using AthenaPK.

As a Metropolis fellow at Los Alamos National Laboratory, I will perform simulations of relativistic magnetized jets emerging from an AGN accretion to better characterize the electromagnetic signature of a BBH merger for heavy black holes formed in the AGN channel. These simulations will use the magnetized AGN jet physics implemented as part of my PhD work and re-contextualize them into an AGN accretion disk using the "shearing box" approximation. This approximation reformulates the MHD equations into a local, Cartesian reference frame co-rotating with a disk

(Hawley et al., 1995; Sharma et al., 2006; Stone & Gardiner, 2010). The flow introduced by the shearing box will approximate the AGN disk environment surrounding jets emanating from mergers. After exploring these simulations comprising a magnetized jet escaping from a shearing box, we will explore including coupling radiative transfer to the magnetohydrodynamics in order to better model the dense environment of the AGN accretion disk. This work will use the same algorithm as the cosmic ray solver we will use in the magnetized galaxy cluster simulations (Jiang et al., 2014; Jiang & Oh, 2018). The inclusion of radiative transfer will enable better predictions of electromagnetic observations of jets escaping AGN accretion disks. The next step will be to implement relativistic MHD jets in order to explore high velocity jets from mergers to see if relativistic effects impact the jet structure and electromagnetic signature. That work will be informed by the relativistic DG method developed in Chapter 5.

BIBLIOGRAPHY

Aarseth, S. J., Gott, III, J. R., & Turner, E. L. 1979, The Astrophysical Journal, 228, 664

Afzal, A., Ansari, Z., Faizabadi, A. R., & Ramis, M. K. 2017, Archives of Computational Methods in Engineering, 24, 337

Alexakis, A., Mininni, P. D., & Pouquet, A. 2005, Phys. Rev. E, 72, 046301

Allen, S. W., Evrard, A. E., & Mantz, A. B. 2011, Annual Review of Astronomy and Astrophysics, 49, 409

Amano, T. 2016, The Astrophysical Journal, 831, 100

Ando, S., & Nagai, D. 2008, Monthly Notices of the Royal Astronomical Society, 385, 2243

Arenas, A., & Chorin, A. J. 2006, Proceedings of the National Academy of Sciences, 103, 4352

Artigues, V., Kormann, K., Rampp, M., & Reuter, K. 2019, arXiv e-prints, arXiv:1911.08394

Aymar, R., Barabaschi, P., & Shimomura, Y. 2002, Plasma Physics and Controlled Fusion, 44, 519

Bambic, C. J., Morsony, B. J., & Reynolds, C. S. 2018, The Astrophysical Journal, 857, 84

Banerjee, N., & Sharma, P. 2014, MNRAS, 443, 687

Barniol Duran, R., Tchekhovskoy, A., & Giannios, D. 2017, Monthly Notices of the Royal Astronomical Society, 469, 4957

Bartelmann, M. 2010, Classical and Quantum Gravity, 27, 233001

Bartelmann, M., & Schneider, P. 2001, Physics Reports, 340, 291

Basilakos, S., Plionis, M., & Lima, J. A. S. 2010, Physical Review D, 82, 083517

Bauer, M., Treichler, S., Slaughter, E., & Aiken, A. 2012, in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '12 (Los Alamitos, CA, USA: IEEE Computer Society Press), 66:1–66:11

Baumjohann, W., & Treumann, R. A. 2012, Basic Space Plasma Physics (Revised Edition) (World Scientific Publishing Company)

Beckingsale, D. A., Burmark, J., Hornung, R., et al. 2019, in 2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 71–81

Beckwith, K., & Stone, J. M. 2011, The Astrophysical Journal Supplement Series, 193, 6

Beg, F. 2019, From Interstellar Cloud to Star to Laboratory: Frontier HEDP Studies of Magnetized Colliding Plasma Flows with Strong Radiative Cooling, Tech. Rep. DOE-UCSD-14493, Univ. of California, San Diego, CA (United States), doi:10.2172/1500122

Bell, A. R. 1978, Monthly Notices of the Royal Astronomical Society, 182, 147

Bellan, P. M. 2008, Fundamentals of Plasma Physics (Cambridge University Press)

Bennett, J. C., Baker, G. M., Bettencourt, M. T., et al. 2015, doi:10.2172/1432926

Beresnyak, A. 2019, Living Reviews in Computational Astrophysics, 5, 2

Beresnyak, A., Giuliani, J. L., Jackson, S. L., et al. 2018, IEEE Transactions on Plasma Science, 46, 3881

Berlok, T., Pakmor, R., & Pfrommer, C. 2020, Monthly Notices of the Royal Astronomical Society, 491, 2919

Berlok, T., & Pessah, M. E. 2015, The Astrophysical Journal, 813, 22

Binney, J., & Tabor, G. 1995, Monthly Notices of the Royal Astronomical Society, 276, 663

Binney, J., & Tremaine, S. 1987, Galactic Dynamics

Bird, R. B., Stewart, W. E., & Lightfoot, E. N. 2006, Transport Phenomena (John Wiley & Sons)

Bittencourt, J. A. 2004, Fundamentals of Plasma Physics (New York, NY: Springer New York), doi:10.1007/978-1-4757-4030-1

Blandford, R., Meier, D., & Readhead, A. 2019, Annual Review of Astronomy and Astrophysics, 57, 467

Blanton, E. L., Clarke, T. E., Sarazin, C. L., Randall, S. W., & McNamara, B. R. 2010, Proceedings of the National Academy of Sciences, 107, 7174

Bodo, G., Mignone, A., & Rosner, R. 2004, PHYSICAL REVIEW E, 4

Böehringer, H., & Morfill, G. E. 1988, The Astrophysical Journal, 330, 609

Bonafede, A., Dolag, K., Stasyszyn, F., Murante, G., & Borgani, S. 2011, Monthly Notices of the Royal Astronomical Society, 418, 2234

Bondi, H. 1952, Monthly Notices of the Royal Astronomical Society, 112, 195

Booth, C. M., & Schaye, J. 2009, MNRAS, 398, 53

Boozer, A. H. 2005, Reviews of Modern Physics, 76, 1071

Bordner, J., & Norman, M. L. 2018, arXiv:1810.01319 [astro-ph, physics:physics], arXiv:1810.01319

Brachet, M. E., Bustamante, M. D., Krstulovic, G., et al. 2013, Physical Review E, 87, 013110

Brachet, M. E., Meiron, D. I., Orszag, S. A., et al. 1983, Journal of Fluid Mechanics, 130, 411

Braginskii, S. I. 1965, Reviews of Plasma Physics, 1, 205

Brandenburg, A., & Dobler, W. 2010, Astrophysics Source Code Library, ascl:1010.060

Bregman, J. N., & David, L. P. 1989, The Astrophysical Journal, 341, 49

Brent, R. P. 1973, Algorithms for Minimization Without Derivatives (Englewood Cliffs, New Jersey: Prentice-Hall)

Britzen, S., Fendt, C., Eckart, A., & Karas, V. 2017, Astronomy & Astrophysics, 601, A52

Brüggen, M. 2003a, The Astrophysical Journal, 593, 700

—. 2003b, The Astrophysical Journal, 592, 839

Brüggen, M., & Vazza, F. 2015, in Magnetic Fields in Diffuse Media, ed. A. Lazarian, E. M. de Gouveia Dal Pino, & C. Melioli (Berlin, Heidelberg: Springer), 599–614

Bryan, G. L., Norman, M. L., O'Shea, B. W., et al. 2014, The Astrophysical Journal Supplement Series, 211, 19

Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D., & Brown, B. P. 2020, Physical Review Research, 2, 023068

Bustamante, M., Jez, P., Monroy Montañez, J. A., et al. 2010, High-Energy Cosmic-Ray Acceleration, https://cds.cern.ch/record/1249755, doi:10.5170/CERN-2010-001.533

Butsky, I. S., & Quinn, T. R. 2018, The Astrophysical Journal, 868, 108

Carilli, C. L., & Taylor, G. B. 2002, Annual Review of Astronomy and Astrophysics, 40, 319

Carlberg, R. G., Yee, H. K. C., & Ellingson, E. 1997, The Astrophysical Journal, 478, 462

Carter Edwards, H., Trott, C. R., & Sunderland, D. 2014, Journal of Parallel and Distributed Computing, 74, 3202

Casner, A. 2021, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379, 20200021

Cavagnolo, K. W., Donahue, M., Voit, G. M., & Sun, M. 2008, ApJL, 683, L107

—. 2009, ApJS, 182, 12

Chandran, B. D. G., & Cowley, S. C. 1998, Physical Review Letters, 80, 3077

Chatterjee, G., Schoeffler, K. M., Kumar Singh, P., et al. 2017, Nature Communications, 8, 15970

Chen, F. F., & Chen, F. F. 1984, Introduction to Plasma Physics and Controlled Fusion, 2nd edn. (New York: Plenum Press)

Chen, J., & Liu, Q. H. 2013, Proceedings of the IEEE, 101, 242

Chiuderi, C., & Velli, M. 2015, Basics of Plasma Astrophysics, UNITEXT for Physics (Milano: Springer Milan), doi:10.1007/978-88-470-5280-2

Choquette, J., Gandhi, W., Giroux, O., Stam, N., & Krashinsky, R. 2021, IEEE Micro, 41, 29

Churazov, E., Sunyaev, R., Forman, W., & Böhringer, H. 2002, Monthly Notices of the Royal Astronomical Society, 332, 729

Ciotti, L., & Ostriker, J. P. 1997, The Astrophysical Journal, 487, L105

Clarke, T. E. 2004, Journal of The Korean Astronomical Society, 37, 337

Clarke, T. E., Kronberg, P. P., & Böhringer, H. 2001, The Astrophysical Journal, 547, L111

Cockburn, B., Hou, S., & Shu, C.-W. 1990, Mathematics of Computation, 54, 545

Cockburn, B., Kanschat, G., & Schötzau, D. 2005, Computers & Fluids, 34, 491

Cockburn, B., Lin, S.-Y., & Shu, C.-W. 1989, Journal of computational Physics, 84, 90

Cockburn, B., & Shu, C.-W. 1989, Mathematics of computation, 52, 411

—. 1998, Journal of Computational Physics, 141, 199

Colafrancesco, S., Dar, A., & De Rújula, A. 2004, Astronomy and Astrophysics, 413, 441

Craxton, R. S., Anderson, K. S., Boehly, T. R., et al. 2015, Physics of Plasmas, 22, 110501

Dagum, L., & Menon, R. 1998, IEEE Comput. Sci. Eng., 5, 46

Dallas, V., & Alexakis, A. 2013a, Physical Review E, 88, 053014

—. 2013b, Physics of Fluids, 25, 105106

—. 2013c, Physical Review E, 88, 063017

Dawson, J. M. 1983, Reviews of Modern Physics, 55, 403

Deakin, T., McIntosh-Smith, S., Price, J., et al. 2019, in 2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 1–13

Deakin, T., Price, J., Martineau, M., & McIntosh-Smith, S. 2018, International Journal of Computational Science and Engineering, 17, 247

Dekel, A., & Birnboim, Y. 2007, Monthly Notices of the Royal Astronomical Society, 383, 119

Dennis, T. J., & Chandran, B. D. G. 2005, The Astrophysical Journal, 622, 205

Domainko, W., Gitti, M., Schindler, S., & Kapferer, W. 2004, Astronomy & Astrophysics, 425, L21

Domaradzki, J. A., Teaca, B., & Carati, D. 2010, Physics of Fluids, 22, 051702

Donnert, J., Vazza, F., Brüggen, M., & ZuHone, J. 2018, Space Science Reviews, 214, doi:10.1007/s11214-018-0556-8

Du, P., Weber, R., Luszczek, P., et al. 2011, From CUDA to OpenCL: Towards a Performance-portable Solution for Multi-platform GPU Programming

Dubois, Y., Devriendt, J., Slyz, A., & Silk, J. 2009, Monthly Notices of the Royal Astronomical Society: Letters, 399, L49

Dubois, Y., Devriendt, J., Slyz, A., & Teyssier, R. 2010, Monthly Notices of the Royal Astronomical Society, 409, 985

Dunn, R. J. H., & Fabian, A. C. 2004, Monthly Notices of the Royal Astronomical Society, 355, 862

Ebisu, T., Ishiyama, T., & Hayashi, K. 2022, Physical Review D, 105, 023016

Edgar, R. 2004, New Astronomy Reviews, 48, 843

Edwards, H. C., Trott, C. R., & Sunderland, D. 2014, Journal of Parallel and Distributed Computing, 74, 3202 , domain-Specific Languages and High-Level Frameworks for High-Performance Computing

Egan, H., O'Shea, B. W., Hallman, E., et al. 2016, arXiv:1601.05083 [astro-ph], arXiv:1601.05083

Fabian, A. 2012, Annual Review of Astronomy and Astrophysics, 50, 455

Fabian, A. C. 1994, Annual Review of Astronomy and Astrophysics, 32, 277

Fabian, A. C., Sanders, J. S., Allen, S. W., et al. 2003, Monthly Notices of the Royal Astronomical Society, 344, L43

Fabian, A. C., Sanders, J. S., Taylor, G. B., et al. 2006, Monthly Notices of the Royal Astronomical Society, 366, 417

Fabian, A. C., Sanders, J. S., Ettori, S., et al. 2000, MNRAS, 318, L65

Fabjan, D., Borgani, S., Tornatore, L., et al. 2010, Monthly Notices of the Royal Astronomical Society, 401, 1670

Federrath, C. 2013, Mon. Not. R. Astron. Soc., 436, 1245

—. 2016, Journal of Plasma Physics, 82, doi:10.1017/S0022377816001069

Ferland, G. J., Porter, R. L., van Hoof, P. A. M., et al. 2013, arXiv:1302.4485 [astro-ph], arXiv:1302.4485

Ferracina, L., & Spijker, M. 2005, Mathematics of Computation, 74, 201

Ferracina, L., & Spijker, M. N. 2004, SIAM Journal on Numerical Analysis, 42, 1073

Feynman, R. P., Hey, J. G., & Allen, R. W. 1998, Feynman Lectures on Computation (USA: Addison-Wesley Longman Publishing Co., Inc.)

Fuhry, M., Giuliani, A., & Krivodonova, L. 2014, International Journal for Numerical Methods in Fluids, 76, 982

Gabuzda, D. C. 2021, Galaxies, 9, 58

Gan, Z., Li, H., Li, S., & Yuan, F. 2017, The Astrophysical Journal, 839, 14

Gao, L., Navarro, J. F., Frenk, C. S., et al. 2012, Monthly Notices of the Royal Astronomical Society, 425, 2169

Gaspari, M. 2015, Proceedings of the International Astronomical Union, 11, 17

Gaspari, M., Brighenti, F., & Temi, P. 2012a, Monthly Notices of the Royal Astronomical Society, 424, 190

Gaspari, M., Melioli, C., Brighenti, F., & D'Ercole, A. 2011, Monthly Notices of the Royal Astronomical Society, 411, 349

Gaspari, M., Ruszkowski, M., & Sharma, P. 2012b, The Astrophysical Journal, 746, 94

Gaspari, M., & Sądowski, A. 2017, The Astrophysical Journal, 837, 149

Gaspari, M., Temi, P., & Brighenti, F. 2017, Monthly Notices of the Royal Astronomical Society, 466, 677

Ghizzardi, S., Rossetti, M., & Molendi, S. 2010, Astronomy & Astrophysics, 516, A32

Gitti, M., Brighenti, F., & McNamara, B. R. 2012, Advances in Astronomy, 2012, e950641

Giuliani, J. L., Beg, F. N., Gilgenbach, R. M., et al. 2012, IEEE Transactions on Plasma Science, 40, 3246

Glines, F. W., Anderson, M., & Neilsen, D. 2015, in 2015 IEEE International Conference on Cluster Computing, 611–618, iSSN: 2168-9253

Glines, F. W., Grete, P., & O'Shea, B. W. 2021, Physical Review E, 103, 043203

Glines, F. W., O'Shea, B. W., & Voit, G. M. 2020, The Astrophysical Journal, 901, 117

Glines, F. W., Beckwith, K. R. C., Braun, J. R., et al. 2022, The Astrophysical Journal Supplement Series

Godunov, S. K. 1959, Matematicheskii Sbornik, 89, 271

Gómez, P. L., Loken, C., Roettiger, K., & Burns, J. O. 2002, The Astrophysical Journal, 569, 122

Gottlieb, S. 2015, in Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014, ed. R. M. Kirby, M. Berzins, & J. S. Hesthaven, Lecture Notes in Computational Science and Engineering (Cham: Springer International Publishing), 17–30

Gottlieb, S., Ketcheson, D. I., & Shu, C.-W. 2011, Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations (World Scientific)

Gottlieb, S., & Shu, C.-W. 1998, Mathematics of Computation, 67, 73

Govoni, F., & Feretti, L. 2004, International Journal of Modern Physics D, 13, 1549

Grete, P., Glines, F. W., & O'Shea, B. W. 2021a, IEEE Transactions on Parallel and Distributed Systems, 32, 85

Grete, P., O'Shea, B. W., & Beckwith, K. 2018, The Astrophysical Journal, 858, L19

—. 2021b, The Astrophysical Journal, 909, 148

—. 2021c, The Astrophysical Journal, 909, 148

Grete, P., O'Shea, B. W., Beckwith, K., Schmidt, W., & Christlieb, A. 2017, Physics of Plasmas, 24, 092311

Grete, P., Vlaykov, D. G., Schmidt, W., & Schleicher, D. R. G. 2016, Physics of Plasmas, 23, 062317

Grete, P., Dolence, J. C., Miller, J. M., et al. 2022, arXiv:2202.12309 [astro-ph], arXiv:2202.12309

Griebel, M., & Zaspel, P. 2010, Computer Science - Research and Development, 25, 65

Guo, F., & Mathews, W. G. 2011, The Astrophysical Journal, 728, 121

Guo, F., & Oh, S. P. 2008, Monthly Notices of the Royal Astronomical Society, 384, 251

Hahn, O., Martizzi, D., Wu, H.-Y., et al. 2017, Monthly Notices of the Royal Astronomical Society, 470, 166

HajiRassouliha, A., Taberner, A. J., Nash, M. P., & Nielsen, P. M. F. 2018, Signal Processing: Image Communication, 68, 101

Hamlin, N. D., Smith, T., Roberds, N., Glines, F., & Beckwith, K. 2022, 26

Hammond, J. R., & Mattson, T. G. 2019, in Proceedings of the International Workshop on OpenCL, IWOCL'19 (New York, NY, USA: Association for Computing Machinery)

Harlow, F. H. 1962, The Particle-in-Cell Method for Numerical Solution of Problems in Fluid Dynamics, Tech. Rep. LADC-5288, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), doi:10.2172/4769185

Harlow, F. H., Evans, M., & Richtmyer, R. D. 1955, A Machine Calculation Method for Hydrodynamic Problems (Los Alamos Scientific Laboratory of the University of California)

Hawkins, M. R. S. 2007, Astronomy & Astrophysics, 462, 581

Hawley, J. F., Gammie, C. F., & Balbus, S. A. 1995, The Astrophysical Journal, 440, 742

Heinrich, A. M., Chen, Y.-H., Heinz, S., Zhuravleva, I., & Churazov, E. 2021, Monthly Notices of the Royal Astronomical Society, stab1557

Heroux, M. A., & Willenbring, J. M. 2012, Scientific Programming, 20, doi:10.1155/2012/408130

Heroux, M. A., Bartlett, R. A., Howle, V. E., et al. 2005, ACM Trans. Math. Softw., 31, 397

Heroux, M. A., Doerfler, D. W., Crozier, P. S., et al. 2009, doi:10.2172/993908

Higueras, I. 2004, Journal of Scientific Computing, 21, 193

—. 2005, SIAM Journal on Numerical Analysis, 43, 924

Hillel, S., & Soker, N. 2016, Monthly Notices of the Royal Astronomical Society, 455, 2139

Ho, L. C. 2004, Coevolution of Black Holes and Galaxies: Volume 1, Carnegie Observatories Astrophysics Series (Cambridge University Press)

Hoekstra, H., Bartelmann, M., Dahle, H., et al. 2013, Space Science Reviews, 177, 75

Holmberg, E. 1941, The Astrophysical Journal, 94, 385

Holmen, J. K., Humphrey, A., Sunderland, D., & Berzins, M. 2017, in Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact, PEARC17 (New York, NY, USA: ACM), 27:1–27:8

Holmen, J. K., Peterson, B., & Berzins, M. 2019, in 2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 36–49

Hopkins, P. F. 2014, Astrophysics Source Code Library, ascl:1410.003

Hornung, R., Jones, H., Keasler, J., et al. 2015, ASC Tri-lab Co-design Level 2Milestone Report 2015, Tech Report LLNL-TR-677453, LLNL

Howes, G. G., Dorland, W., Cowley, S. C., et al. 2008, Physical Review Letters, 100, 065004

Hu, J., & Lou, Y.-Q. 2004, The Astrophysical Journal, 606, L1

Huarte-Espinosa, M., Frank, A., Blackman, E. G., et al. 2012, The Astrophysical Journal, 757, 66

Humpherys, J., Jarvis, T. J., & Evans, E. J. 2017, Foundations of Applied Mathematics

Incropera, F. P., & DeWitt, D. P. 1981, Fundamentals of Heat Transfer (Wiley)

Intel. 2021, Xeon Platinum 8280 Specs, https://www.intel.com/content/www/us/en/products/sku/192478/intel-xeon-platinum-8280-processor-38-5m-cache-2-70-ghz.html

Intel Corporation. 2016, Intel 64 and IA-32 Architectures Optimization Reference Manual

Iwai, H. 1999, IEEE Journal of Solid-State Circuits, 34, 357

Jia, Z., Maggioni, M., Smith, J., & Scarpazza, D. P. 2019, arXiv:1903.07486 [cs], arXiv: 1903.07486

Jia, Z., Maggioni, M., Staiger, B., & Scarpazza, D. P. 2018, arXiv:1804.06826 [cs], arXiv: 1804.06826

Jiang, Y.-F., & Oh, S. P. 2018, The Astrophysical Journal, 854, 5

Jiang, Y.-F., Stone, J. M., & Davis, S. W. 2014, The Astrophysical Journal Supplement Series, 213, 7

Jubelgas, M., Springel, V., & Dolag, K. 2004, Monthly Notices of the Royal Astronomical Society, 351, 423

Jubelgas, M., Springel, V., Enßlin, T., & Pfrommer, C. 2008, Astronomy and Astrophysics, 481, 33

Kale, L. V., & Krishnan, S. 1993, CHARM++: A Portable Concurrent Object Oriented System Based on C++, Tech. rep., Champaign, IL, USA

Katz, N., Weinberg, D. H., & Hernquist, L. 1996, The Astrophysical Journal Supplement Series, 105, 19

Khosroshahi, H. G., Jones, L. R., & Ponman, T. J. 2004, Monthly Notices of the Royal Astronomical Society, 349, 1240

Kida, S., & Orszag, S. A. 1990, Journal of Scientific Computing, 5, 85

Klimontovich, Y. L. 1994, Physics-Uspekhi, 37, 737

Klöckner, A., Warburton, T., Bridge, J., & Hesthaven, J. S. 2009, Journal of Computational Physics, 228, 7863

Kochanek, C. S. 2006, in Gravitational Lensing: Strong, Weak and Micro, ed. P. Schneider, C. S. Kochanek, & J. Wambsganss (Berlin, Heidelberg: Springer), 91–268

Koliopanos, F. 2018, arXiv:1801.01095 [astro-ph], arXiv:1801.01095

Kolmogorov, A. 1941, Akademiia Nauk SSSR Doklady, 30, 301

Komissarov, S., & Porth, O. 2021, New Astronomy Reviews, 92, 101610

Konstantinidis, E., & Cotronis, Y. 2017, Journal of Parallel and Distributed Computing, 107, 37

Korpi, M. J., Brandenburg, A., Shukurov, A., Tuominen, I., & Nordlund, A. 1999, The Astrophysical Journal, 514, L99

Kramer, R. M. J., Cyr, E. C., Miller, S. T., et al. 2020, A Plasma Modeling Hierarchy and Verification Approach, Tech. Rep. SAND-2020-3576, Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), doi:10.2172/1608511

Kroupp, E., Stambulchik, E., Starobinets, A., et al. 2018, Physical Review E, 97, 013202

Krymskii, G. F. 1977, Akademiia Nauk SSSR Doklady, 234, 1306

Kumar, P., & Zhang, B. 2015, Physics Reports, 561, 1

Kunz, M. W., Schekochihin, A. A., Cowley, S. C., Binney, J. J., & Sanders, J. S. 2011, Monthly Notices of the Royal Astronomical Society, 410, 2446

Lacroix, T., & Silk, J. 2018, The Astrophysical Journal, 853, L16

Laity, G., Robinson, A., Cuneo, M., et al. 2021, Towards Predictive Plasma Science and Engineering through Revolutionary Multi-Scale Algorithms and Models, Final Report., Tech. Rep. SAND2021-0718, Sandia National Lab. (SNL-NM), Albuquerque, NM (United States); Sandia National Laboratories, SNL California, doi:10.2172/1813907

Landauer, R. 1988, Nature, 335, 779

Larson, R. B. 1981, Monthly Notices of the Royal Astronomical Society, 194, 809

Lecoanet, D., McCourt, M., Quataert, E., et al. 2016, Monthly Notices of the Royal Astronomical Society, 455, 4274

Ledvina, S. A., Ma, Y.-J., & Kallio, E. 2008, Space Science Reviews, 139, 143

Lee, E., Brachet, M. E., Pouquet, A., Mininni, P. D., & Rosenberg, D. 2008, Physical Review E, 78, 066401

—. 2010, Physical Review E, 81, 016318

Leiserson, C. E., Thompson, N. C., Emer, J. S., et al. 2020, Science, 368, eaam9744

LeVeque, R. J. 2002, Finite Volume Methods for Hyperbolic Problems (Cambridge; New York: Cambridge University Press)

Li, H., Lapenta, G., Finn, J. M., Li, S., & Colgate, S. A. 2006, The Astrophysical Journal, 643, 92

Li, Y., & Bryan, G. L. 2012, The Astrophysical Journal, 747, 26

—. 2014a, The Astrophysical Journal, 789, 54

—. 2014b, The Astrophysical Journal, 789, 153

Li, Y., Bryan, G. L., Ruszkowski, M., et al. 2015, ApJ, 811, 73

Li, Y., Gendron-Marsolais, M.-L., Zhuravleva, I., et al. 2020, The Astrophysical Journal Letters, 889, L1

LIGO Scientific Collaboration and Virgo Collaboration, Abbott, R., Abbott, T. D., et al. 2020, Physical Review Letters, 125, 101102

Lima, J. A. S., Cunha, J. V., & Alcaniz, J. S. 2003, Physical Review D, 68, 023510

Lind, S. J., Rogers, B. D., & Stansby, P. K. 2020, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 476, 20190801

Liu, C., Zhou, G., Shyy, W., & Xu, K. 2019, Shock Waves, 29, 1083

Lo, Y. J., Williams, S., Van Straalen, B., et al. 2015, in High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation, ed. S. A. Jarvis, S. A. Wright, & S. D. Hammond (Springer International Publishing), 129–148

Loewenstein, M., Zweibel, E. G., & Begelman, M. C. 1991, The Astrophysical Journal, 377, 392

Longair, M. S. 2008, Galaxy Formation, 2nd edn., Astronomy and Astrophysics Library (Berlin ; New York: Springer)

Luo, W., Li, Y., Wang, H., et al. 2019, Laser and Particle Beams, 37, 301

Lyutikov, M. 2007, The Astrophysical Journal, 668, L1

Malyshkin, L., & Kulsrud, R. 2001, The Astrophysical Journal, 549, 402

Marcowith, A., Ferrand, G., Grech, M., et al. 2020, arXiv:2002.09411 [astro-ph], arXiv:2002.09411

Markevitch, M., Vikhlinin, A., & Mazzotta, P. 2001, The Astrophysical Journal, 562, L153

Marques, D., Duarte, H., Ilic, A., et al. 2017, in 2017 International Conference on High Performance Computing Simulation (HPCS), 898–907, iSSN: null

Martí, J.-M. 2019, Galaxies, 7, 24

Martí, J. M., & Müller, E. 2003, Living Reviews in Relativity, 6, doi:10.12942/lrr-2003-7

—. 2015, Living Reviews in Computational Astrophysics, 1, 3

Martineau, M., McIntosh-Smith, S., & Gaudin, W. 2017, Concurrency and Computation: Practice and Experience, 29, e4117, e4117 cpe.4117

Martizzi, D., Hahn, O., Wu, H.-Y., et al. 2016, Monthly Notices of the Royal Astronomical Society, 459, 4408

Mathews, W. G. 1971, The Astrophysical Journal, 165, 147

May, M. M., & White, R. H. 1966, Physical Review, 141, 1232

McComb, W. D. 1990, The Physics of Fluid Turbulence

McCourt, M., Sharma, P., Quataert, E., & Parrish, I. J. 2012, Monthly Notices of the Royal Astronomical Society, 419, 3319

McDonald, M., Veilleux, S., Rupke, D. S. N., & Mushotzky, R. 2010, ApJ, 721, 1262

McDonald, M., McNamara, B. R., Voit, G. M., et al. 2019, The Astrophysical Journal, 885, 63

McKernan, B., Ford, K. E. S., Kocsis, B., Lyra, W., & Winter, L. M. 2014, Monthly Notices of the Royal Astronomical Society, 441, 900

McKernan, B., Ford, K. E. S., Lyra, W., & Perets, H. B. 2012, Monthly Notices of the Royal Astronomical Society, 425, 460

McNamara, B. R., & Nulsen, P. E. J. 2007, Annual Review of Astronomy and Astrophysics, 45, 117

McNamara, B. R., Wise, M., Nulsen, P. E. J., et al. 2000, ApJL, 534, L135

Medina, D. S., St-Cyr, A., & Warburton, T. 2014, arXiv:1403.0968 [cs], arXiv:1403.0968

Meece, G. R., O'Shea, B. W., & Voit, G. M. 2015, The Astrophysical Journal, 808, 43

Meece, G. R., Voit, G. M., & O'Shea, B. W. 2017, The Astrophysical Journal, 841, 17pp

Meece Jr, G. R. 2016, AGN Feedback and Delivery Methods for Simulations of Cool-Core Galaxy Clusters (Michigan State University)

Meier, D. L. 1999, The Astrophysical Journal, 518, 788

Messina, P. 2017, Computing in Science Engineering, 19, 63

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953, The Journal of Chemical Physics, 21, 1087

Metzler, C. A., & Evrard, A. E. 1994, The Astrophysical Journal, 437, 564

Mignone, A., & Bodo, G. 2005, Monthly Notices of the Royal Astronomical Society, 364, 126

—. 2006, Monthly Notices of the Royal Astronomical Society, 368, 1040

Mignone, A., & McKinney, J. C. 2007, Monthly Notices of the Royal Astronomical Society, 378, 1118

Mignone, A., Plewa, T., & Bodo, G. 2005, The Astrophysical Journal Supplement Series, 160, 199

Mignone, A., Ugliano, M., & Bodo, G. 2009, Monthly Notices of the Royal Astronomical Society, 393, 1141

Mignone, A., Zanni, C., Tzeferacos, P., et al. 2011, The Astrophysical Journal Supplement Series, 198, 7

Miller, G. H., Moses, E. I., & Wuest, C. R. 2004, Optical Engineering, 43, 2841

Miniati, F. 2014, The Astrophysical Journal, 782, 21

—. 2015, The Astrophysical Journal, 800, 60

Mo, H., Van den Bosch, F., & White, S. 2010, Galaxy Formation and Evolution (Cambridge; New York: Cambridge University Press)

Moe, S. A., Rossmanith, J. A., & Seal, D. C. 2015, arXiv:1507.03024 [math], arXiv:1507.03024

Montgomery, D., & Turner, L. 1981, The Physics of Fluids, 24, 825

Morganti, R. 2017, Frontiers in Astronomy and Space Sciences, 4

Myers, A., Colella, P., & Straalen, B. V. 2016, The Astrophysical Journal, 816, 56

Nakamura, M., Li, H., & Li, S. 2006, The Astrophysical Journal, 652, 1059

—. 2007, The Astrophysical Journal, 656, 721

Narayan, R., & Medvedev, M. V. 2001, The Astrophysical Journal, 562, L129

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, The Astrophysical Journal, 462, 563

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, The Astrophysical Journal, 490, 493

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1997, ApJ, 490, 493

Navarro, J. F., Hayashi, E., Power, C., et al. 2004, Monthly Notices of the Royal Astronomical Society, 349, 1039

Nelson, D., Pillepich, A., Springel, V., et al. 2019, Monthly Notices of the Royal Astronomical Society, 490, 3234

Nolte, P. o. P. a. A. D. D., & Nolte, D. D. 2001, Mind at Light Speed: A New Kind of Intelligence (Simon and Schuster)

Norman, M. L., & Bryan, G. L. 1999, in The Radio Galaxy Messier 87, ed. H.-J. Röser & K. Meisenheimer, Lecture Notes in Physics (Berlin, Heidelberg: Springer), 106–115

Núñez-de la Rosa, J., & Munz, C.-D. 2018, Computer Physics Communications, 222, 113

NVIDIA Corporation. 2014, NVIDIA's Next Generation CUDA Compute Architecture: Kepler GK110/210

—. 2016, NVIDIA Tesla P100

—. 2017, NVIDIA Tesla V100 GPU Architecture

Ogawa, T., Mineshige, S., Kawashima, T., Ohsuga, K., & Hashizume, K. 2017, Publications of the Astronomical Society of Japan, 69, 33

Ongena, J., Koch, R., Wolf, R., & Zohm, H. 2016, Nature Physics, 12, 398

Ottinger, P. F., & Schumer, J. W. 2006, Physics of Plasmas, 13, 063109

Panagoulia, E. K., Fabian, A. C., & Sanders, J. S. 2014, Monthly Notices of the Royal Astronomical Society, 438, 2341

Parrish, I. J., Quataert, E., & Sharma, P. 2009, The Astrophysical Journal, 703, 96

Patterson, D. 2010, IEEE Spectrum, 47, 28

Pennycook, S., Sewall, J., & Lee, V. 2019, Future Generation Computer Systems, 92, 947

Pennycook, S. J., Sewall, J. D., & Lee, V. W. 2016, arXiv:1611.07409 [cs], arXiv:1611.07409

Pfrommer, C., Enßlin, T. A., Springel, V., Jubelgas, M., & Dolag, K. 2007, Monthly Notices of the Royal Astronomical Society, 378, 385

Pillepich, A., Nelson, D., Springel, V., et al. 2019, Monthly Notices of the Royal Astronomical Society, 490, 3196

Pouquet, A., Lee, E., Brachet, M. E., Mininni, P. D., & Rosenberg, D. 2010, Geophysical and Astrophysical Fluid Dynamics, 104, 115

Prasad, D., Sharma, P., & Babul, A. 2015, ApJ, 811, 108

—. 2017, Monthly Notices of the Royal Astronomical Society, 471, 1531

—. 2018, The Astrophysical Journal, 863, 62

Pratt, G. W., Arnaud, M., Biviano, A., et al. 2019, Space Science Reviews, 215, 25

Pratt, G. W., Croston, J. H., Arnaud, M., & Böhringer, H. 2009, Astronomy and Astrophysics, 498, 361

Rafferty, D. A., McNamara, B. R., & Nulsen, P. E. J. 2008, ApJ, 687, 899

Reed, W. H., & Hill, T. R. 1973, Triangular Mesh Methods for the Neutron Transport Equation, Tech. Rep. LA-UR-73-479; CONF-730414-2, Los Alamos Scientific Lab., N.Mex. (USA)

Reguly, I. Z., & Mudalige, G. R. 2020, Computers & Fluids, 199, 104425

Rephaeli, Y., & Silk, J. 1995, The Astrophysical Journal, 442, 91

Revaz, Y., Combes, F., & Salomé, P. 2008, A&A, 477, L33

Reynolds, C. 2018, The Micro- and Macro-Physics of Thermal Conduction in the ICM, 49

Reynolds, O. 1883, Philosophical Transactions of the Royal Society of London, 174, 935

Riccardi, G., & Durante, D. 2008, in International Mathematical Forum, Vol. 42, 2081–2111

Richardson, L. F. 1922, Weather Prediction by Numerical Process (Cambridge: Cambridge University Press)

Ritchie, B. W., & Thomas, P. A. 2002, Monthly Notices of the Royal Astronomical Society, 329, 675

Ritos, K., Kokkinakis, I. W., & Drikakis, D. 2018, Computers & Fluids, 173, 307

Roberds, N. A., Cartwright, K. L., Sandoval, A. J., et al. 2022, 9

Roettiger, K., Loken, C., & Burns, J. O. 1997, The Astrophysical Journal Supplement Series, 109, 307

Rogers, K. K., & Peiris, H. V. 2021, Physical Review D, 103, 043526

Roh, S., Ryu, D., Kang, H., Ha, S., & Jang, H. 2019, The Astrophysical Journal, 883, 138

Rose, S. C., Naoz, S., Sari, R., & Linial, I. 2021, arXiv:2201.00022 [astro-ph], arXiv:2201.00022

Rosin, M. S., Schekochihin, A. A., Rincon, F., & Cowley, S. C. 2011, Monthly Notices of the Royal Astronomical Society, 413, 7

Rott, N. 1990, Annual Review of Fluid Mechanics, 22, 1

Rudakov, L. I., & Sudan, R. N. 1997, Physics Reports, 283, 253

Russell, H. R., McNamara, B. R., Fabian, A. C., et al. 2016, MNRAS, 458, 3134

Russell, H. R., McDonald, M., McNamara, B. R., et al. 2017, ApJ, 836, 130

Ruszkowski, M., & Begelman, M. C. 2002, The Astrophysical Journal, 581, 223

Ruszkowski, M., Brüggen, M., & Begelman, M. C. 2004, The Astrophysical Journal, 611, 158

Ruszkowski, M., Lee, D., Brüggen, M., Parrish, I., & Oh, S. P. 2011, The Astrophysical Journal, 740, 81

Ruszkowski, M., & Oh, S. P. 2011, Monthly Notices of the Royal Astronomical Society, 414, 1493

Ryu, D., Chattopadhyay, I., & Choi, E. 2006, The Astrophysical Journal Supplement Series, 166, 410

Ryutov, D. D., & Remington, B. A. 2002, Plasma Physics and Controlled Fusion, 44, B407

Sammak, S., Nouri, A. G., Ansari, N., & Givi, P. 2015, in Mathematical Modeling of Technological Processes, ed. N. Danaev, Y. Shokin, & A.-Z. Darkhan, Communications in Computer and Information Science (Cham: Springer International Publishing), 124–132

Sanchez, R., & Newman, D. E. 2015, Plasma Physics and Controlled Fusion, 57, 123002

Sarazin, C. L. 1988, X-Ray Emission from Clusters of Galaxies

Schekochihin, A. A. 2020, arXiv:2010.00699 [astro-ph, physics:nlin, physics:physics], arXiv:2010.00699

Schekochihin, A. A., Cowley, S. C., Dorland, W., et al. 2009, The Astrophysical Journal Supplement Series, 182, 310

Schekochihin, A. A., Cowley, S. C., Taylor, S. F., Maron, J. L., & McWilliams, J. C. 2004, The Astrophysical Journal, 612, 276

Schmidt, W., & Federrath, C. 2011, Astronomy & Astrophysics, 528, A106

Schneider, V., Katscher, U., Rischke, D. H., et al. 1993, Journal of Computational Physics, 105, 92

Schure, K. M., Kosenko, D., Kaastra, J. S., Keppens, R., & Vink, J. 2009, Astronomy & Astrophysics, 508, 751

Sharma, P., Hammett, G. W., Quataert, E., & Stone, J. M. 2006, The Astrophysical Journal, 637, 952

Shebalin, J. V., Matthaeus, W. H., & Montgomery, D. 1983, Journal of Plasma Physics, 29, 525

Short, C. J., Thomas, P. A., & Young, O. E. 2013, Monthly Notices of the Royal Astronomical Society, 428, 1225

Shu, C.-W., & Osher, S. 1989, Journal of Computational Physics, 83, 32

Shumlak, U. 2015, High Fidelity Physics Using the Multi-Fluid Plasma Model

Sijacki, D., Springel, V., Di Matteo, T., & Hernquist, L. 2007, Monthly Notices of the Royal Astronomical Society, 380, 877

Simionescu, A., ZuHone, J., Zhuravleva, I., et al. 2019, Space Science Reviews, 215, 24

Simon, H. D. 1992, Parallel Computational Fluid Dynamics - Implementations and Results, Tech. rep., Cambridge, MA (United States); MIT Press

Sinars, D. B., Sweeney, M. A., Alexander, C. S., et al. 2020, Physics of Plasmas, 27, 070501

Smith, B., O'Shea, B. W., Voit, G. M., Ventimiglia, D., & Skillman, S. W. 2013, The Astrophysical Journal, 778, 152

Smith, B. D., Bryan, G. L., Glover, S. C. O., et al. 2017, Monthly Notices of the Royal Astronomical Society, 466, 2217

Sommerfeld, A. 1909, Ein Beitrag zur hydrodynamischen Erklärung der turbulenten Flüssigkeitsbewegungen

Spitzer, L. 1956, Physics of Fully Ionized Gases

—. 1978, Physical Processes in the Interstellar Medium, doi:10.1002/9783527617722

Springel, V. 2005, Monthly Notices of the Royal Astronomical Society, 364, 1105

—. 2010, Annual Review of Astronomy and Astrophysics, 48, 391

Springel, V., Yoshida, N., & White, S. D. M. 2001, New Astronomy, 6, 79

St-Onge, D. A., Kunz, M. W., Squire, J., & Schekochihin, A. A. 2020, arXiv e-prints, 2003, arXiv:2003.09760

Steijl, R., & Barakos, G. N. 2018, Computers & Fluids, 173, 22

Steinwandel, U. P., Boess, L. M., Dolag, K., & Lesch, H. 2021, arXiv:2108.07822 [astro-ph], arXiv:2108.07822

Stokes, G. G. 1851, Transactions of the Cambridge Philosophical Society, 9, 8

Stone, J. E., Gohara, D., & Shi, G. 2010, Computing in Science Engineering, 12, 66

Stone, J. M., & Gardiner, T. 2009, New Astronomy, 14, 139

Stone, J. M., & Gardiner, T. A. 2010, The Astrophysical Journal Supplement Series, 189, 142

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008a, The Astrophysical Journal Supplement Series, 178, 137

—. 2008b, The Astrophysical Journal Supplement Series, 178, 137

Stone, J. M., & Norman, M. L. 1992, The Astrophysical Journal Supplement Series, 80, 753

Stone, J. M., Tomida, K., White, C. J., & Felker, K. G. 2020a, The Astrophysical Journal Supplement Series, 249, 4

—. 2020b, arXiv:2005.06651

Straatsma, T. P., Antypas, K. B., & Williams, T. J. 2017, Exascale Scientific Applications: Scalability and Performance Portability, 1st edn. (Chapman & Hall/CRC)

Sunyaev, R. A., & Zel'dovich, Y. B. 1980, Annual Review of Astronomy and Astrophysics, 18, 537

Synge, J. 1957, The Relativistic Gas, Series in Physics (North-Holland Publishing Company)

Tabor, G., & Binney, J. 1993, Monthly Notices of the Royal Astronomical Society, 263, 323

Taub, A. H. 1948, Physical Review, 74, 328

Taylor, G. I. 1938, Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences, 164, 476

Taylor, G. I., & Green, A. E. 1937, Proceedings of the Royal Society of London Series A, 158, 499

Teyssier, R. 2002, Astronomy & Astrophysics, 385, 337

Theis, T. N., & Wong, H.-S. P. 2017, Computing in Science Engineering, 19, 41

Tobias, S. M. 2021, Journal of Fluid Mechanics, 912, doi:10.1017/jfm.2020.1055

Top500. 2000, ASCI Red | TOP500, https://www.top500.org/system/168753/

—. 2010, Tianhe-1A - NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050 | TOP500, https://www.top500.org/system/176929/

—. 2020, Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu Interconnect D | TOP500, https://www.top500.org/system/179807/

—. 2021, Frontera - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR | TOP500, https://www.top500.org/system/179607/

Toro, E. F. 2009, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, 3rd edn. (Dordrecht ; New York: Springer)

Toyouchi, D., Inayoshi, K., Hosokawa, T., & Kuiper, R. 2021, The Astrophysical Journal, 907, 74

Trac, H., & Pen, U.-L. 2003, Publications of the Astronomical Society of the Pacific, 115, 303

Tremmel, M., Karcher, M., Governato, F., et al. 2017, Monthly Notices of the Royal Astronomical Society, 470, 1121

Tremmel, M., Quinn, T. R., Ricarte, A., et al. 2019, Monthly Notices of the Royal Astronomical Society, 483, 3336

Treumann, R. A., & Baumjohann, W. 1997, Advanced Space Plasma Physics (PUBLISHED BY IMPERIAL COLLEGE PRESS AND DISTRIBUTED BY WORLD SCIENTIFIC PUBLISHING CO.), doi:10.1142/p020

Trott, C. R., Lebrun-Grandié, D., Arndt, D., et al. 2022, IEEE Transactions on Parallel and Distributed Systems, 33, 805

Tskhakaya, D., Matyash, K., Schneider, R., & Taccogna, F. 2007, Contributions to Plasma Physics, 47, 563

Tukey, J. W. 1977, Exploratory Data Analysis (Reading, Mass. : Addison-Wesley Pub. Co.)

Tümer, A., Tombesi, F., Bourdin, H., et al. 2019, Astronomy & Astrophysics, 629, A82

Turk, M. J., Smith, B. D., Oishi, J. S., et al. 2011, The Astrophysical Journal Supplement Series, 192, 9

Vacca, V., Murgia, M., Govoni, F., et al. 2018, Galaxies, 6, 142

Vahala, G., Keating, B., Soe, M., et al. 2008, Commun. Comput. Phys., 23

van Dyke, M. 1982, NASA STI/Recon Technical Report A, 82, 36549

van Leer, B. 1979, Journal of Computational Physics, 32, 101

Vidal-García, A., Falgarone, E., Arrigoni Battaia, F., et al. 2021, Monthly Notices of the Royal Astronomical Society, 506, 2551

Vikhlinin, A., Markevitch, M., & Murray, S. S. 2001a, The Astrophysical Journal, 549, L47

—. 2001b, The Astrophysical Journal, 551, 160

Villiers, J.-P. D., Hawley, J. F., & Krolik, J. H. 2003, The Astrophysical Journal, 599, 1238

Vlaykov, D. G., Grete, P., Schmidt, W., & Schleicher, D. R. G. 2016, Physics of Plasmas, 23, 062316

Voigt, L. M., & Fabian, A. C. 2004, \mnras, 347, 1130

Voigt, L. M., Schmidt, R. W., Fabian, A. C., Allen, S. W., & Johnstone, R. M. 2002, Monthly Notices of the Royal Astronomical Society, 335, L7

Voit, G. M. 2005, Reviews of Modern Physics, 77, 207

Voit, G. M., & Bryan, G. L. 2001, Nature, 414, 425

Voit, G. M., Donahue, M., Bryan, G. L., & McDonald, M. 2015, Nature, 519, 203

Voit, G. M., Meece, G., Li, Y., et al. 2017, The Astrophysical Journal, 845, 80

Wadsley, J., Stadel, J., & Quinn, T. 2004, New Astronomy, 9, 137

Wagh, B., Sharma, P., & McCourt, M. 2014, Monthly Notices of the Royal Astronomical Society, 439, 2822

Walker, S., Simionescu, A., Nagai, D., et al. 2019, Space Science Reviews, 215, 7

Wang, C., Ruszkowski, M., Pfrommer, C., Oh, P., & Yang, H. 2020, 236, 124.02

Wang, S., Khoury, J., Haiman, Z., & May, M. 2004, Physical Review D, 70, 123008

Wang, Z. J., Fidkowski, K., Abgrall, R., et al. 2013, International Journal for Numerical Methods in Fluids, 72, 811

Weinberger, R., Springel, V., & Pakmor, R. 2020, The Astrophysical Journal Supplement Series, 248, 32

Weinberger, R., Springel, V., Hernquist, L., et al. 2017, Monthly Notices of the Royal Astronomical Society, 465, 3291

White, C. J., Stone, J. M., & Gammie, C. F. 2016a, The Astrophysical Journal Supplement Series, 225, 22

—. 2016b, The Astrophysical Journal Supplement Series, 225, 22

White, M., Cohn, J. D., & Smit, R. 2010, Monthly Notices of the Royal Astronomical Society, 408, 1818

Williams, S., Waterman, A., & Patterson, D. 2009, Commun. ACM, 52, 65

Wilson, J. R. 1972, The Astrophysical Journal, 173, 431

Woosley, S. E. 2017, The Astrophysical Journal, 836, 244

Wu, H.-Y., Evrard, A. E., Hahn, O., et al. 2015, Monthly Notices of the Royal Astronomical Society, 452, 1982

Wu, K., & Tang, H. 2016, The Astrophysical Journal Supplement Series, 228, 3

Wu, K. K. S., Fabian, A. C., & Nulsen, P. E. J. 1998, Monthly Notices of the Royal Astronomical Society, 301, L20

Xu, Z., Zhao, H., & Zheng, C. 2015, Journal of Computational Physics, 281, 844

Yang, H.-Y. K., & Reynolds, C. S. 2016a, The Astrophysical Journal, 829, 90

—. 2016b, The Astrophysical Journal, 818, 181

Yang, Y., Shi, Y., Wan, M., Matthaeus, W. H., & Chen, S. 2016, Phys. Rev. E, 93, 061102

Young, D. S. D. 2010, The Astrophysical Journal, 710, 743

Zanna, L. D., & Bucciantini, N. 2002, Astronomy & Astrophysics, 390, 1177

Zhang, U.-H., Schive, H.-Y., & Chiueh, T. 2018, The Astrophysical Journal Supplement Series, 236, 50

Zhang, W., Almgren, A., Beckner, V., et al. 2019, Journal of Open Source Software, 4, 1370

Zhao, D., & Aluie, H. 2018, Phys. Rev. Fluids, 3, 054603

Zheng, Y., Kamil, A., Driscoll, M. B., Shan, H., & Yelick, K. 2014, in 2014 IEEE 28th International Parallel and Distributed Processing Symposium, 1105–1114

Zhu, J.-P., Zhang, B., Yu, Y.-W., & Gao, H. 2021, The Astrophysical Journal, 906, L11

Zhuravleva, I., Churazov, E., Schekochihin, A. A., et al. 2019, Nature Astronomy, 3, 832

—. 2014, Nature, 515, 85

ZuHone, J. A., Markevitch, M., & Johnson, R. E. 2010, The Astrophysical Journal, 717, 908

Zylstra, A. B., Hurricane, O. A., Callahan, D. A., et al. 2022, Nature, 601, 542