DERIVING THE INITIAL CONDITIONS OF THE ELECTROWEAK AND QCD PHASE TRANSITIONS AND TESTING A NUMERICAL RELATIVITY CODE

by

Joshua E Barrera, BS

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Physics

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

JULY 2021

DERIVING THE INITIAL CONDITIONS OF THE
ELECTROWEAK AND QCD PHASE
TRANSITIONS AND TESTING A
NUMERICAL RELATIVITY CODE

by

Joshua E Barrera

APPROVED BY

David Garrison, Ph.D., Chair

Paul Withey, Ph.D., Committee Member

Rafael de la Torre, M.S., Committee Member

RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

David Garrison, Ph.D., Interim Associate Dean

Miguel Gonzalez, Ph.D., Dean

ABSTRACT

DERIVING THE INITIAL CONDITIONS OF THE
ELECTROWEAK AND QCD PHASE
TRANSITIONS AND TESTING A
NUMERICAL RELATIVITY CODE

Joshua E Barrera
University of Houston-Clear Lake, 2021

Thesis Chair: David Garrison, Ph.D.

The field of Numerical Relativity (NR) has been primarily driven by the study of large scale dynamics involving binary systems of black holes and neutron stars. Due to the nature of the underlying theory, NR also has the ability to simulate relativistic fluids. Being radiation dominated, the early universe can be modeled as a relativistic plasma, and the proper Stress-Energy tensor can be utilized with Einstein's Field Equations to evolve the conditions of the Early Universe over time. These simulations can give us key insights pertaining to the development of the universe and the formation of large-scale systems. Magnetogenesis is of particular interest, as characterizing this phenomenon could shed light on the seeding and formation of galaxies. Additionally, these techniques can be used to derive gravitational-wave spectra from the events that took place during these time periods. This thesis aims to derive the conditions present in the early universe during the Electroweak (EW) and Quantum Chromodynamic (QCD) phase transitions. These conditions are prerequisites for SpecCosmo, a NR code being developed at the University of Houston-Clear Lake, which utilizes the

techniques of NR to model the evolution of the early universe. This thesis will also investigate the ability of SpecCosmo to handle relativistic shocks in its current state. The shock capturing ability of the code will be gauged using a suite of tests from work by Komissarov [1, 2]. Once the shock capturing ability of SpecCosmo has been analyzed, SpecCosmo will be ready to accept the initial conditions calculated here that can then be used in simulations of the early universe to study its development and characteristics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1:
INTRODUCTION

Baumgarte and Shapiro describe Numerical Relativity as "the art and science of developing computer algorithms to solve Einstein's equations for astrophysically realistic, high-velocity, strong-field systems" [3]. Only a handful of analytical solutions to Einstein's field equations exist. In order to find these solutions, several simplifying assumptions must be made which limit our ability to analyze a fully dynamic system. Numerical relativity circumvents this issue by using numerical methods to approximate solutions to a high degree of accuracy. The field has largely been driven by the study of the dynamics of large bodies such as black holes. Here, we take a look at how NR can be used to study the evolution of the early universe.

Statement of the Problem

Modern cosmology bases conclusions about the evolution of our universe by observing the interaction of large scale structures, and looking as far back as the Cosmic Microwave Background (CMB). Theories of modern cosmology suggest a dynamic, turbulent, and violent early universe. Many cosmological models characterize the CMB according to perturbations in the Friedman-Robertson-Walker (FRW) model. The FRW metric is an exact solution to the Einstein field equations. Perturbations of the FRW metric, which grow according to a power-law with time, are thought to be the most accurate cosmological models of the universe [4]. The FRW model, combined with the fact that magnetic fields are a key factor in the evolution of relativistic objects, sets the stage for the application of General Relativity and Magnetohydrodynamics to study the evolution of the early universe. Using results from modern particle physics research and the FRW model as the cosmological model, this thesis will develop the mathematical models and calculate the initial conditions necessary in order to simulate the physics of the early universe at specific epochs of interest. The two epochs this thesis will look at are the Electroweak (EW) phase transition and the Quantum Chromodynamic (QCD) phase transition. These two phase transitions have well known energy expectation values that will be used as starting points for calculations.

Current State of Research

The ongoing goal of this research has been studying the evolution of the Early Universe in order to gain a better understanding of its development. Results from [4] regarding the order of phase transitions have shown that these simulations have the potential to better our understanding of how the early universe formed. Potential phenomena to study are the development of large scale magnetic fields, characterizing the EW and QCD phase transitions, deriving gravitational wave spectra for these and other phenomena, and detailing how these events influenced the evolution of the universe. Using parameters derived from modern Cosmology, and the techniques of Numerical Relativity, we can evolve the state of the universe from roughly $t_0 = 10^{-11}\,\mathrm{s}$ and beyond. Through this mathematical simulation, critical parameters of the early universe can be calculated, and limits on unknown parameters such as magnetic field strength and velocity perturbations can be determined.

Cosmological Model

Einstein Field Equations

The Einstein equations have the familiar compact form of

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \frac{8\pi G}{c^4} T_{\mu\nu} \tag{1.1}$$

where $R_{\mu\nu}$ is the Ricci tensor, $g_{\mu\nu}$ is the spacetime metric, R is the Ricci scalar (the trace of the Ricci tensor), G is Newton's gravitational constant, c is the speed of light, and $T_{\mu\nu}$ is the energy-momentum tensor. For a perfect fluid, $T_{\mu\nu}$ takes the form

$$T^{\mu\nu} = (\epsilon + p) u^\mu u^\nu - p g^{\mu\nu} \tag{1.2}$$

where ϵ is mass-energy density, p is pressure, and u^μ is the four-velocity of matter given by

$$u^\mu = \frac{dx^\mu}{ds} \tag{1.3}$$

and $x^\mu(s)$ describes the worldline of matter in terms of proper time $\tau = c^{-1}s$ [5]. Note from [5] that the Ricci tensor is given by

$$R_{\mu\nu} = \Gamma^\lambda_{\mu\nu,\lambda} - \Gamma^\lambda_{\mu\lambda,\nu} + \Gamma^\lambda_{\mu\nu}\Gamma^\sigma_{\lambda\sigma} - \Gamma^\sigma_{\mu\lambda}\Gamma^\lambda_{\nu\sigma} \tag{1.4}$$

and the Christoffel symbols Γ are given in terms of the metric tensor by

$$\Gamma^{\mu}_{\nu\lambda} = \frac{1}{2} g^{\mu\sigma}(g_{\sigma\nu,\lambda} + g_{\sigma\lambda,\nu} - g_{\nu\lambda,\sigma}). \tag{1.5}$$

By choosing a form of the spacetime metric $g_{\mu\nu}$, one can derive all permutations of $\Gamma^{\mu}_{\nu\lambda}$. It would then be possible to determine all permutations of $R_{\mu\nu}$ and $T_{\mu\nu}$, and therefore build all of the possible permutations of the Einstein equations given by 1.1.

Friedmann Model

The Friedmann model is considered to be the standard of modern cosmology. To define the Friedmann model of cosmology, we will first take a look at the Friedmann-Roberston-Walker (FRW) metric, and then define and discuss the Friedmann equations that can be derived from the Einstein equations using this particular choice of spacetime metric. The spacetime metric in its well-known general form is

$$ds^2 = g_{\mu\nu} dx^{\mu} dx^{\nu}. \tag{1.6}$$

The FRW metric, taken from equation 3.35 in [5], has the form

$$ds^2 = (cdt)^2 - a^2(t)\Big[\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + sin^2(\theta)d\phi^2)\Big]. \tag{1.7}$$

Note several things here: the coordinates of choice $(x^0, x^1, x^2, x^3) = (ct, r, \theta, \phi)$ are spherical, the value $k = -1, 0, 1$ can be selected to define curvature as negative, zero, or positive, respectively, and a(t) is the scale factor of the universe, usually defined as a $=$ 1 at present time, and a(0) $=$ 0 at the beginning of the universe. It is important to take a moment here to distinguish between the scale factor $a(t)$, which shows up in the elements of $R_{\mu\nu}$, and the Ricci scalar $R = R_{00} + R_{11} + R_{22} + R_{33}$. Sometimes the scale factor $a(t)$ is instead represented by $R(t)$, which can be confusing. It has been decided to make the change to $a(t)$ here in hopes of avoiding that confusion. From this point forward, any $R(t)$ which denotes scale factor in equations taken from sources has been changed to $a(t)$.

Using this choice of $g_{\mu\nu}$, a detailed calculation of the Christoffel symbols can be found on page 51 of [5]. Because of spherical symmetry, the θ and ϕ componenets can be ignored, and we can reduce the rest of the calculations to the following 00- and 11-components of the spacetime metric and Ricci tensor elements, which are the time and radius components, respectively:

$$g_{00} = 1, \tag{1.8}$$

$$g_{11} = \frac{-a^2(t)}{1 - kr^2}, \tag{1.9}$$

$$R_{00} = \frac{-3\ddot{a}(t)}{a(t)}, \tag{1.10}$$

$$R_{11} = \frac{R\ddot{a}(t) + 2\dot{a}^2(t) + 2c^2 k}{1 - kr^2}. \tag{1.11}$$

Noting that the four-velocity vector $u_\mu = (1, 0, 0, 0)$, we can gather the components of $T_{\mu\nu}$:

$$T_{00} = \epsilon, \tag{1.12}$$

$$T_{11} = \frac{pa^2(t)}{1 - kr^2}. \tag{1.13}$$

Now that we have the 00- and 11-components of $g_{\mu\nu}$, $R_{\mu\nu}$, and $T_{\mu\nu}$, we are able to build the 00- and 11-components of the Einstein equations as follows:

$$3(\dot{a}^2(t) + c^2 k) = \frac{8\pi G\epsilon a^2(t)}{c^2}, \tag{1.14}$$

$$2a(t)\ddot{a}(t) + \dot{a}^2(t) + kc^2 = -\frac{8\pi G p a^2(t)}{c^2}. \tag{1.15}$$

Using a similar process found in [5], eliminating $\dot{a}^2(t)$ from equation 1.15 yields

$$T_{11} = \frac{pa^2(t)}{1 - kr^2}, \tag{1.16}$$

and by choosing $k = -1, 0, 1$ in equation 1.14, we get three new equations:

$$\dot{a}^2(t) = c^2 + \frac{8\pi G\epsilon a^2(t)}{3c^2} \tag{1.17}$$

$$\dot{a}^2(t) = \frac{8\pi G\epsilon a^2(t)}{3c^2} \tag{1.18}$$

$$\dot{a}^2(t) = -c^2 + \frac{8\pi G\epsilon a^2(t)}{3c^2} \tag{1.19}$$

With a given equation of state $p = p(\epsilon)$, we have three equations and three unknowns for any of the three values of k. For the early universe, which is what we are intersted in, the physics is dominated by either pure radiation, or by radiation and highly relativistic particles. For this situation, the equation of state would be $p = \frac{1}{3}\epsilon$, so that the mass-energy density ϵ behaves like $a^{-4}(t)$. Models of the universe determined by this method are referred to as Friedmann models [5].

The last piece of the cosmological model worth discussing is the Hubble Parameter. Starting with the Hubble law (equation 29.7 from [6]):

$$v(t) = H(t)r(t) = H(t)a(t)\varpi \tag{1.20}$$

where ϖ is a constant that identifies a particular thin spherical shell whose radius at time t is defined by

$$r(t) = a(t)\varpi, \tag{1.21}$$

$a(t)$ being the scale factor at time t. Realizing that $v(t) = \frac{dr(t)}{dt}$, we can change 1.20 into

$$\frac{dr(t)}{dt} = H(t)r(t), \tag{1.22}$$

and solving for $H(t)$ gives the relationship

$$H(t) = \frac{\dot{a}(t)}{a(t)}. \tag{1.23}$$

During the radiation dominated era, it is known that $a(t) \sim t^{1/2}$ from [5]. This relationship will be revisited in chapter 2. By taking the derivative with respect to time of this equation and using 1.23, we can arrive at the following relationship between Hubble parameter and time:

$$H(t) = \frac{1}{2t}. \tag{1.24}$$

This equation will be utilized in chapter 2 as a way to double-check that the critical density, Hubble parameter, and initial time are all in agreement. The current value of the Hubble constant is $H_0 \approx 50\ kms^{-1}Mpc^{-1}$, although there are uncertainties in the exact determination of this value [5].

Computational Methods

3+1 Decomposition

Only a handful of analytic solutions to Einstein's Field Equations exist. These solutions require simplifying assumptions that limit the description of physical systems. Calculating the dynamics of a physical system governed by Einstein's equations of general relativity could be impossible to do analytically, but high performance computing and numerical techniques have opened doors for this kind of research. To construct algorithms that will accomplish this task, we first have to recast Einstein's 4-dimensional field equations into a form that is suitable for numerical integration [3]. 3+1 decomposition is a technique used in numerical relativity to carry out computational simulations while still complying with Einstein's requirement of having four dimensions of space-time. The four space-time dimensions are separated into the three spatial dimensions and one time dimension. The physics and underlying mathematics are calculated in the three dimensions at one instant in time. This is where the critical parameters within the stress-energy tensor are calculated, such as densities or pressures. Time is then stepped forward by a small increment, and the physics is recalculated at this new time.

The standard 3+1 or ADM equations offer a general framework for the physics we are interested in [3]. The most notable place to begin would be with the spacetime interval, which in standard 3+1 is written as

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt), \tag{1.25}$$

where α is the lapse, β is the shift, and γ_{ij} is the spatial metric that is induced on the 3D hypersurfaces Σ by the familiar metric g_{ab} through the equation

$$\gamma_{ab} = g_{ab} + n_a n_b. \tag{1.26}$$

BSSN Formulation

The BSSN formulation, or Baumgarte-Shapiro-Shibata-Nakamura formulation, is a special version of the 3+1 decomposition that essentially simplifies the spatial Ricci tensor. In short, the BSSN formulation separates the transverse from longitudinal, or the radiative from nonradiative, degrees of freedom [3]. In order to accomplish this, the conformal factor e^{ϕ} and the trace of the extrinsic curvature K are evolved separately. The spatial metric is decomposed into a conformally related metric $\bar{\gamma}_{ij}$ with determinant $\bar{\gamma} = 1$ and the conformal factor as follows:

$$\gamma_{ij} = e^{4\phi}\bar{\gamma}_{ij}. \tag{1.27}$$

The extrinsic curvature is decomposed into its trace and traceless parts,

$$K_{ij} = e^4\phi\tilde{A}_{ij} + \frac{1}{3}\gamma_{ij}K. \tag{1.28}$$

The Hamiltonian constraint becomes

$$0 = \mathcal{H} = \bar{\gamma}^{ij}\bar{D}_i\bar{D}_j e^{\phi} - \frac{e^{\phi}}{8}\bar{R} + \frac{e^5\phi}{8}\tilde{A}_{ij}\tilde{A}^{ij} - \frac{e^5\phi}{12}K^2 + 2\pi e^{5\phi}\rho, \tag{1.29}$$

while the momentum constraint becomes

$$0 = \mathcal{M}^i = \bar{D}_j(e^6\phi\tilde{A}^{ji}) - \frac{2}{3}e^{6\phi}\bar{D}^i K - 8\pi e^{6\phi}S^i. \tag{1.30}$$

The BSSN evolution equations are as follows:

$$\partial_t\phi = -\frac{1}{6}\alpha K + \beta^i\partial_i\phi + \frac{1}{6}\partial_i\beta^i, \tag{1.31}$$

$$\partial_t \bar{\gamma}_{ij} = -2\alpha \tilde{A}_{ij} + \beta^k \partial_k \bar{\gamma}_{ij} + \bar{\gamma}_{ik} \partial_j \beta^k + \bar{\gamma}_{kj} \partial_i \beta^k - \frac{2}{3} \bar{\gamma}_{ij} \partial_k \beta^k, \tag{1.32}$$

$$\partial_t K = -\gamma^{ij} D_j D_i \alpha + \alpha (\tilde{A}_{ij} \tilde{A}^{ij} + \frac{1}{3} K^2) + 4\pi\alpha(\rho + S) + \beta^i \partial_i K, \tag{1.33}$$

$$\partial_t \tilde{A}_{ij} = e^{-4\phi} \big(- (D_i D_j \alpha)^{TF} + \alpha (R_{ij}^{TF} - 8\pi S_{ij}^{TF}) \big) + \alpha (K \tilde{A}_{ij} - 2\tilde{A}_{il} \tilde{A}^l{}_j) \tag{1.34}$$

$$+\beta^k \partial_k \tilde{A}_{ij} + \tilde{A}_{ik} \partial_j \beta^K + \tilde{A}_{kj} \partial_i \beta^k - \frac{2}{3} \tilde{A}_{ij} \partial_k \beta^k. \tag{1.35}$$

Cactus

The numerical simulations carried out in this research use Cactus as a tool for handling computation. Cactus is an open source collection of computer code that consists of two types of code: flesh and thorns [7]. The flesh is the central core of the code, which connects to modules (thorns) of the user's choosing. Thorns implement the specific mathematics to be evolved during the simulation. The Cactus community maintains multiple "toolkits" for specific fields of research, one being the Einstein Toolkit [8]. This toolkit comes with specific thorns that are equipped to handle the popular problems in numerical relativity, namely the dynamics of large-body binary systems. The research conducted here uses a unique code that will be discussed in greater detain in the next section.

SpecCosmo

SpecCosmo was developed within the Cactus framework, and utilizes GRMHD equations and a modified version of the BSSN formulation of the Einstein Field Equations shown in [4]. We will begin by defining the spacetime interval as

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt), \tag{1.36}$$

where α is the lapse, β is the shift, and γ_{ij} is the spatial metric. The next entity we wish to consider is the stress-energy tensor and the accompanying equations. We will make use of the GRMHD model found in [3] and [4], which is derived from constraints like the continuity equation and conservation of energy-momentum. The MHD stress-energy tensor is given by

$$T^{ab} = (\rho_0 h + b^2)u^a u^b + (P + \frac{b^2}{2})g^{ab} - b^a b^b, \tag{1.37}$$

$$h = 1 + \epsilon + \frac{P}{\rho_0}, \tag{1.38}$$

$$b^a = \frac{1}{\sqrt{4\pi}} B^a_{(u)}, \tag{1.39}$$

$$B^0_{(u)} = \frac{1}{\alpha} u_i B^i, \tag{1.40}$$

$$B^i_{(u)} = \frac{1}{u^0}\left(\frac{B^i}{\alpha} + B^0_{(u)} u^i\right), \tag{1.41}$$

where P is the fluid pressure, ρ_0 is density, B^i is magnetic field, u^a is four-velocity, h is the enthalpy, ϵ is specific internal energy, and b is the magnitude of the magnetic vector field [4].

We are also interested in the MHD evolution equations. These are coupled equations that involve terms from the stress-energy tensor, matter source terms, and magnetic field. The flux-conservative equations of relativistic hydrodynamics take on the general form

$$\partial_t \mathcal{U} + \partial_i \mathcal{F}^i = \mathcal{S} \tag{1.42}$$

where $\mathcal{U}$ is the state vector of *conserved* variables built out of the so-called *primitive* fluid variables $\mathcal{P} = (\rho_0, v^i, P)$, the $\mathcal{F}^i$ are the flux vectors (one for each spatial dimension, i), and where the source vector $\mathcal{S}$ does not contain any derivatives of the primitive fluid variables [9]. Note that the time index is separated from the three spatial indices, as is consistent with 3+1 decomposition. The set of coupled MHD evolution equations are as follows:

$$\partial_t \rho_* + \partial_j(\rho_* v^j) = 0 \tag{1.43}$$

$$\partial_t \tilde{S}_i + \partial_j(\alpha\sqrt{\gamma}\, T^j{}_i) = \frac{1}{2}\alpha\sqrt{\gamma}\, T^{ab} g_{ab,i} \tag{1.44}$$

$$\partial_t \tilde{\tau} + \partial_i(\alpha^2\sqrt{\gamma}\, T^{0i} - \rho_* v^i) = s_{\tilde{\tau}} \tag{1.45}$$

$$\partial_t \tilde{B}^i + \partial_j (v^j \tilde{B}^i - v^i \tilde{B}^j) = 0 \tag{1.46}$$

Here, ρ_* is the total mass-energy density as measure by an observer co-moving with the fluid, v is the fluid velocity, $\tilde{S}_i$ is the momentum density, T is the stress-energy tensor, α is the lapse, $\tilde{\tau}$ is energy, $s_{\tilde{\tau}}$ is the source term, and $\tilde{B}$ is the magnetic field. In all cases, terms with a tilde contain a factor of $\sqrt{\gamma}$, making them the relativistic terms.

Initial Magnetic Field

These simulations make use of the standard MHD equations, which require an initial magnetic field as a form of input in order to evolve forward in time. This warrants some kind of initial B-field value that came from Inflation. The exact strength of this initial magnetic field has not yet been determined. This input parameter can be adjusted each time a simulation is performed, and the resulting background magnetic field data can be compared with observation to verify the accuracy of initial B-field estimations. This initial magnetic field is of interest to the research, as studying this phenomenon could improve our understanding of the origins of these seed fields, their development, and how they influenced the evolution of the universe. Here we will make use of the Biermann Battery equation taken from equation 7 in [10], which yields an evolution equation for the B-field:

$$\partial_t \tilde{B}_i = \frac{1}{q_m \rho_*^2} \nabla p \times \nabla \rho_* \tag{1.47}$$

where $\rho_* = \alpha\sqrt{\gamma}\rho_0 u^0$ is a conserved mass density, p is pressure of the fluid, and q_m is charge per unit mass [10]. This will be incorporated into the MHD evolution equations as a modification to equation 1.12. This equation meets our needs for several reasons. Perhaps most importantly, the standard MHD equations do not produce the initial B-field required at the initial time our simulations begin, so invoking this equations fills that void. The fact that we have already defined the pressure and density terms seen in equation (2.4) as initial conditions means that this initial B-field partial time-derivative term allows us to adjust the evolution equations based on the pressure and density of the fluid that is being modeled.

CHAPTER 2:

INITIAL CONDITIONS

This chapter outlines the process for calculating the initial conditions of a simulation for both the Electroweak and QCD phase transitions. Each phase transition has a well known vacuum expectation value associated with it, which will be the starting point for each calculation. Energy, temperature, scale factor, time, thermal degrees of freedom, and critical density are the six parameters that will be found for each of these phase transitions. Some of the parameters are needed to calculate others, and some of the parameters are critical inputs required by the computer code. Once these parameters are determined, they can be used as inputs before a simulation begins. The data generated by these inputs combined with the evolution equations can be analyzed to determine if the simulation conforms to the FRW model and whether or not the hypothesized values are accurate.

Electroweak Phase Transition

Energy and Temperature

The expression for the total available thermal energy is

$$E_1 = k_B T_1 \tag{2.1}$$

where E is the energy in Joules, k_B is Boltzmann's constant (8.617×10^{-5} eV/K), and T is temperature in Kelvin. For the EW phase transition, the vacuum expectation value of the Higgs field is known to be 246 GeV. Using this value as the average energy available at the time, we get

$$T_1 = \frac{E_1}{k_B} = \frac{246 \times 10^{11}\ eV}{8.617 \times 10^{-5}\ eV/K} \tag{2.2}$$

$$T_1 \approx 2.85 \times 10^{15}\ K \tag{2.3}$$

Scale Factor

Now that we have the average temperature that corresponds to the total available energy at the time, we can use the age of the universe today ($t_2 = 4.35 \times 10^{17}$ s), the average temperature of the universe today ($T_2 = 2.7$ K), and the scale factor today (a_2=1), together with equation 8.2 from [5] to get the scale factor and the initial time of the EW phase transition. The process for deriving 8.2 from [5] goes back to equation 3.76 from [5] (1.14 and 1.15 above), which can be found using the following process: Rearrange equation 1.14 from above for ϵ, and calculate $\dot{\epsilon}$. This new equation for $\dot{\epsilon}$ will yield a $\ddot{a}$ term, which can be eliminated using equation 1.15 from above. This will yield equation 3.79 in [5], which can then be rearranged to give equation 4.6 from [5]. Section 4.3 in [5] shows how to integrate equation 4.6 to give equation 4.40. The Stefan-Boltzmann law gives the relationship $\epsilon \sim T^4$, which then gives the following:

$$\frac{a_1}{a_2} = \frac{T_2}{T_1}, \tag{2.4}$$

$$\frac{a_1}{1} = \frac{2.7\ K}{2.85 \times 10^{15}\ K}, \tag{2.5}$$

$$a_1 \approx 9.58 \times 10^{-16}. \tag{2.6}$$

This result indicates that at the time we are interested in, the universe was around one quadrillion times smaller than it is today. Note that the scale factor has no units as it is a ratio comparing the size of the universe at different times.

Thermal Degrees of Freedom

In order to calculate the critical density parameter in the next section, we first need to know what the thermal degrees of freedom of the system are at the time of interest. The process for calculating thermal degrees of freedom is outlined in section 8.4 of [5], and is essentially dependent on three conditions: whether the particle has an antiparticle, the number of spin states of a particle, and whether the particle is a boson or a fermion. The calculation is as follows:

$$N = N_1 N_2 N_3 \tag{2.7}$$

where N_1 is 1 if the particle has a distinct antiparticle, and 2 if it does not; N_2 is the number of spin states of the particle; N_3 is a statistical mechanical factor which is $\frac{7}{8}$ for fermions and 1 for bosons [5]. This number N needs to be calculated for every individual particle that exists during the particular epoch that is being considered, and then each N value for each particle is added together for a total effective number.

In order for a particle to be counted, the particle's mass must be less than or equal to the vacuum expectation value at the time, otherwise there is not enough available energy for the particle to manifest. All seventeen particles currently know to exist according to the standard model have a mass less than 246 GeV/c^2, so the full standard model is considered for the EW phase transition. The following table outlines how this calculation is performed, and uses a similar format outlined in [11].

Particle	*Anti*	*Colors*	*Spins*	*States*	*Fermion/Boson*	*Effective no.*
u	2	3	2	12	0.875	10.5
d	2	3	2	12	0.875	10.5
c	2	3	2	12	0.875	10.5
s	2	3	2	12	0.875	10.5
t	2	3	2	12	0.875	10.5
b	2	3	2	12	0.875	10.5
e	2	1	2	4	0.875	3.5
μ	2	1	2	4	0.875	3.5
τ	2	1	2	4	0.875	3.5
ν_e	2	1	1	2	0.875	1.75
ν_μ	2	1	1	2	0.875	1.75
ν_τ	2	1	1	2	0.875	1.75
g	1	8	2	16	1	16
γ	1	1	2	2	1	2
W	2	1	3	6	1	6
Z	1	1	3	3	1	3
H	1	1	1	1	1	1
g_*						106.75

Table 2.1: Particle States for EW Phase Transition

Here, the number of states for each particle is the product of columns 2, 3, and 4. The number of states is multiplied by the fermion/boson factor to give the effective number of each particle, and those effective numbers are added together for a total effective number.

Critical Denisity and Hubble Parameter

The process outlined in chapter 29 section 2 of [6] shows how to develop the equation that will be used here to calculate the critical density. We will also need the previously calculated values for temperature and thermal degrees of freedom. From equation 29.77 in [6], we have

$$\rho_{rel} = \frac{u_{rel}}{c^2} = \frac{g_* a T^4}{2c^2} \tag{2.8}$$

where g_* is the effective number of degrees of freedom for the EW phase transition as calculated above, and T is the temperature from 2.3 above. Putting the values for g_* and T into this equation yields

$$\rho_{rel} = \frac{(106.75)(7.5641 \times 10^{-16}\ J\ m^{-3}\ K^{-4})(2.85 \times 10^{15}\ K)^4}{2c^2} \tag{2.9}$$

$$\rho_{rel} \approx 2.96 \times 10^{31}\ kg\ m^{-3} \tag{2.10}$$

We will then set this value equal to the critical density found in equation 29.12 from [6]. If the relativistic density is equal to critical density, this would have several consequences, one being that the density contribution from matter particles is near zero ($\rho_m \approx 0$). This makes sense in our situation since we are assuming a radiation dominated universe, and all particles are considered relativistic. This would in turn make the matter density parameter approximately equal to zero, and the relativistic density parameter approximately equal to 1, which from equation 29.79 and 29.80 in

[6] would result in

$$\Omega_m = \frac{\rho_m}{\rho_c} \approx \frac{0}{\rho_c} \approx 0 \tag{2.11}$$

$$\Omega_{rel} = \frac{\rho_{rel}}{\rho_c} \approx 1. \tag{2.12}$$

This result would also satisfy $\Omega_m + \Omega_{rel} = 1$ and therefore $k = 0$ for the Friedmann model, and suggest that the universe is flat [6]. By using equation 29.12 from [6] for ρ_c and setting $\rho_{rel} = \rho_c$, we can then calculate the Hubble parameter as follows:

$$\rho_c = \frac{3H^2(t)}{8\pi G} = \rho_{rel} \tag{2.13}$$

$$H(t) = \sqrt{\frac{8\pi G \rho_{rel}}{3}} \tag{2.14}$$

$$H(t) = \sqrt{\frac{8\pi(6.67 \times 10^{-11}\ m^3\ kg^{-1}\ s^{-2})(2.96 \times 10^{31}\ kg\ m^{-3})}{3}} \tag{2.15}$$

$$H(t) \approx 1.29 \times 10^{11}\ s^{-1} \tag{2.16}$$

Time

For initial time, we can refer directly to equation 8.34 from [5]. Rewriting $\frac{c}{\sqrt{\varepsilon}} = \frac{1}{\sqrt{\rho_c}}$ gives

$$t_1 = \sqrt{\frac{3}{32\pi G\rho_c}} \tag{2.17}$$

where ρ_c is the critical density, G is Newton's gravitational constant, and c is the speed of light in vacuum. Putting these constants into the equation above, and using the previously calculated value for ρ_c, we get

$$t_1 = \sqrt{\frac{3}{32\pi(6.67 \times 10^{-11}\ m^3\ kg^{-1}\ s^{-2})(2.96 \times 10^{31}\ kg\ m^{-3})}} \tag{2.18}$$

$$t_1 \approx 3.90 \times 10^{-12}\ s. \tag{2.19}$$

We can also refer to equation 53 from [11] to cross-check this result and confirm some agreement between different source. Using table A1 from [11], we can infer $g_* \approx 106$ based on the work in this paper, and we get

$$t = \sqrt{\frac{90\hbar^3 c^5}{32\pi^3 G g_{*\epsilon}(T)}}(k_B T)^{-2} \tag{2.20}$$

$$t = \frac{2.4}{\sqrt{g_{*\epsilon}(T)}} T_{MeV}^{-2} \tag{2.21}$$

$$t = \frac{2.4}{\sqrt{106}}(246000)^{-2} \tag{2.22}$$

$$t \approx 3.85 \times 10^{-12}\ s \tag{2.23}$$

where temperature T has been converted back into units of energy ($246\ GeV = 246000\ MeV$).

As a third check, we can use equation 1.24 from above:

$$t = \frac{1}{2H(t)} \tag{2.24}$$

$$t = \frac{1}{2(1.29 \times 10^{11}\ s^{-1})} \tag{2.25}$$

$$t \approx 3.88 \times 10^{-12}\ s \tag{2.26}$$

The level of agreement between these three methods of calculating initial time gives confidence in the results. The differences in the three results are likely due to the choice of significant digits carried over in calculations and digits of precision used for the known constants. This thesis will choose $3.88 \times 10^{-12}\ s$ as the declared result since it is the median of the three different calculations for time.

QCD Phase Transition

Energy and Temperature

Using the same approach as in 2.1.1, the expression for the total available thermal energy is

$$E_1 = k_B T_1 \tag{2.27}$$

where E is the energy in Joules, k_B is Boltzmann's constant, and T is temperature in Kelvin. For the QCD phase transition, we will use a vacuum expectation value of 170 MeV. Using this value as the average energy available at the time, we get

$$T_1 = \frac{E_1}{k_B} = \frac{246 \times 10^{11}\ eV}{8.617 \times 10^{-5}\ eV/K} \tag{2.28}$$

$$T_1 \approx 1.97 \times 10^{12}\ K \tag{2.29}$$

Scale Factor

Using the same approach as in 2.1.2, we can use the age of the universe today, the average temperature of the universe today, and the scale factor today with equation 8.2 from [5] to get the scale factor and the initial time of the QCD phase transition. Starting with equation 8.2,

$$\frac{a_1}{a_2} = \frac{T_2}{T_1} \tag{2.30}$$

$$\frac{a_1}{1} = \frac{2.7\ K}{1.97 \times 10^{12}\ K} \tag{2.31}$$

$$a_1 \approx 1.38 \times 10^{-12} \tag{2.32}$$

which indicates that at the time we are interested in, the universe was around one trillion times smaller than it is today, and about 1000 times larger than the value calculated for the EW phase transition.

Thermal Degrees of Freedom

For the QCD phase transition, the thermal degrees of freedom are calculated in the same way as the EW phase transition in section 2.1.4. With the vacuum expectation value of 170 MeV, we ignore all particles with a mass greater than 170 MeV/c^2, and table 2.1 reduces to

Particle	*Anti?*	*Colors*	*Spins*	*States*	*Fermion/Boson*	*Effective no.*
u	2	3	2	12	0.875	10.5
d	2	3	2	12	0.875	10.5
s	2	3	2	12	0.875	10.5
e	2	1	2	4	0.875	3.5
μ	2	1	2	4	0.875	3.5
ν_e	2	1	1	2	0.875	1.75
ν_μ	2	1	1	2	0.875	1.75
ν_τ	2	1	1	2	0.875	1.75
g	1	8	2	16	1	16
γ	1	1	2	2	1	2
g_*						61.75

Table 2.2: Particle States for QCD Phase Transition

Critical Denisity and Hubble Parameter

Using the same approach as in 2.1.4, and inserting the values for temperature and thermal degrees of freedom that correspond to the QCD phase transition:

$$\rho_{rel} = \frac{g_* a T^4}{2c^2} \tag{2.33}$$

$$\rho_{rel} = \frac{(61.75)(7.5641 \times 10^{-16}\ J\ m^{-3}\ K^{-4})(1.97 \times 10^{12})^4}{2c^2} \tag{2.34}$$

$$\rho_{rel} = \rho_c \approx 3.91 \times 10^{18}\ kg\ m^{-3} \tag{2.35}$$

Jumping to equation 2.14 above for Hubble Parameter, we get

$$H(t) = \sqrt{\frac{8\pi G \rho_{rel}}{3}} \tag{2.36}$$

$$H(t) = \sqrt{\frac{8\pi(6.67 \times 10^{-11}\ m^3\ kg^{-1}\ s^{-2})(3.91 \times 10^{18}\ kg\ m^{-3})}{3}} \tag{2.37}$$

$$H(t) \approx 46700\ s^{-1} \tag{2.38}$$

and from the relationship between Hubble Parameter and time, we get

$$t = \frac{1}{2H(t)} \tag{2.39}$$

$$t = \frac{1}{2(46700\ s^{-1})} \tag{2.40}$$

$$t \approx 1.07 \times 10^{-5}\ s. \tag{2.41}$$

Time

Using the same approach as in 2.1.5, we begin with equation 2.11, and insert the critical density value for the QCD phase transition calculated above to get

$$t_1 = \sqrt{\frac{3}{32\pi G\rho_c}} \tag{2.42}$$

$$t_1 = \sqrt{\frac{3}{32\pi(6.67 \times 10^{-11}\ m^3\ kg^{-1}\ s^{-2})(3.91 \times 10^{18}\ kg\ m^{-3})}} \tag{2.43}$$

$$t_1 \approx 1.07 \times 10^{-5}\ s. \tag{2.44}$$

Again, double checking this result using equation 53 from [11], the value $g_* = 61.72$ from table A1, and the vacuum expectation value of 170 MeV, we have

$$t = \sqrt{\frac{90\hbar^3 c^5}{32\pi^3 G g_{*\epsilon}(T)}}(k_B T)^{-2} \tag{2.45}$$

$$t = \frac{2.4}{\sqrt{g_{*\epsilon}(T)}} T_{MeV}^{-2} \tag{2.46}$$

$$t = \frac{2.4}{\sqrt{61.72}}(170)^{-2} \tag{2.47}$$

$$t \approx 1.06 \times 10^{-5}\ s. \tag{2.48}$$

All three methods for calculating initial time during the QCD phase transition have shown to be almost exactly the same, so this thesis will choose $1.07 \times 10^{-5}\ s$ for the declared result.

Summary of Key Parameters

In the previous two sections, five different parameters were calculated for two different epochs based on the vacuum expectation values being the available energies. The following table shows these twelve total parameters and the corresponding epoch that they belong to. These parameters are used as initial conditions that the computer code needs in order to begin calculating new values and evolving forward in time.

Initial Condition	*Electroweak*	*QCD*
Energy	$246\,\mathrm{GeV}$	$170\,\mathrm{MeV}$
Temperature	$2.85 \times 10^{15}\,\mathrm{K}$	$1.97 \times 10^{12}\,\mathrm{K}$
Thermal Degrees of Freedom	106.75	61.75
Scale Factor	9.58×10^{-16}	1.38×10^{-12}
Initial Time	$3.88 \times 10^{-12}\,\mathrm{s}$	$1.07 \times 10^{-5}\,\mathrm{s}$
Critical Density	$2.96 \times 10^{31}\,\mathrm{kg\,m^{-3}}$	$3.91 \times 10^{18}\,\mathrm{kg\,m^{-3}}$
Hubble Parameter	$1.29 \times 10^{11}\,\mathrm{s^{-1}}$	$46\,700\,\mathrm{s^{-1}}$

Table 2.3: Initial Conditions

CHAPTER 3:

SHOCK TESTING

In Numerical Relativity, the term "shock" refers to a sharp discontinuity (or jump) between any two values calculated by a numerical relativity code during a simulation. A codes ability to handle shocks during a simulation are dependent on the shock-capturing scheme implemented in the code, the resolution chosen at run-time, and the computational capability of the system being used. In general, the higher the resolution of the shock-capturing scheme is, the more computational power and/or time is required for running the simulation.

SCR3 and Previous Testing

SCR3 is a fourth-order weighted essentially non-oscillaroty scheme built by following the method outlined in work by Liu, Osher, and Chan [12]. It is an additional finite differencing method available to be chosen at runtime, along with 2nd-order finite, 4th-order finite, spectral methods, and spectral methods (one-dimensional) for a total of five different shock capturing schemes. Work done in [12] has shown SCR3 to be a more effective shock capturing routine than 2nd-order finite differencing with artificial viscosity. According to [12], the next step should be refining the SCR3 scheme by adding artificial viscosity. However, it is noted that work in [12] was based on the use of FixedCosmo, whereas the work presented in this thesis utilizes SpecCosmo. Since the code has been changed, the goal of the work in this series of shock tests was to determine how the results using SpecCosmo compare to the previous results using FixedCosmo with the same parameters. All shock tests were ran without the use of artificial viscosity except for the Alfven wave shock test.

Shock Test Types

There are eight different shock test types considered, first introduced in work by Komissarov, and later in work by Duez, Liu, Shapiro, and Stephens [9]. These tests are one-dimensional (z-axis chosen in this work), starting with a discontinuity at z = 0, and extending in the positive and negative directions along the axis by the same amount. For each test type, the density and velocity of the relativistic fluid are compared with results from [9] and [12].

The eight different test types are Fast Shock, Slow Shock, Switch-on Rarefaction, Switch-off Rarefaction, Shock Tube 1, Shock Tube 2, Collision, and Non-linear Alfven Wave. In fast and slow shock, initial data along the axis satisfies the special relativistic Rankine-Hugonoit jump conditions for MHD shocks, and thus the discontinuity travels with a speed μ without changing its pattern [9]. In switch-on/off rarefaction, positive and negative directions of the axis are connected by a rarefaction wave at t = 0 [9]. The challenge with these tests arises when the tangential component of the magnetic field (B^y) is switch on or off when transitioning from one side of the axis to another [9]. In Shock Tubes 1 and 2, left and right sides of the axis are connected by a rarefaction wave, a contact discontinuity, and a shock wave [9]. In Collision, both sides of the axis travel with equal speeds in opposite directions. In Non-linear Alfven Wave, the setup differs from the other 7 tests in that there are no discontinuities initially [9]. The left and right states are separated by a width (W = 0.5), and are joined by continuous functions at t = 0 [9].

The following table taken from Komissarov [2] details the initial conditions and test parameters for each shock test type. These initial conditions are scripted in the code for each test type, and can be chosen at the start of a simulation run in the parameter file.

Within the thorn mhd_init, the variable initial_mhd can be changed to any one of

Problem	Left state	Right state	Grid	Time
Fast Shock ($\mu = 0.2$)	$u^i = (25.0, 0.0, 0.0)$ $B^i = (20.0, 25.02, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$u^i = (1.091, 0.3923, 0.00)$ $B^i = (20.0, 49.0, 0.0)$ $P = 367.5,\ \ \rho = 25.48$	$n = 40$	$t = 2.5$
Slow Shock ($\mu = 0.5$)	$u^i = (1.53, 0.0, 0.0)$ $B^i = (10.0, 18.28, 0.0)$ $P = 10.(!),\ \ \rho = 1.0$	$u^i = (.9571, -0.6822, 0.00)$ $B^i = (10.0, 14.49, 0.0)$ $P = 55.36,\ \ \rho = 3.323$	$n = 200$	$t = 2.0$
Switch-off Fast Rarefaction	$u^i = (-2.0, 0.0, 0.0)$ $B^i = (2.0, 0.0, 0.0)$ $P = 1.0,\ \ \rho = 0.1$	$u^i = (-0.212, -0.590, 0.0)$ $B^i = (2.0, 4.710, 0.0)$ $P = 10.0,\ \ \rho = 0.562$	$n = 150$	$t = 1.0$
Switch-on Slow Rarefaction	$u^i = (-0.765, -1.386, 0.0)$ $B^i = (1.0, 1.022, 0.0)$ $P = 0.1,\ \ \rho = 1.78 \times 10^{-3}$ (!)	$u^i = (0.0, 0.0, 0.0)$ $B^i = (1.0, 0.0, 0.0)$ $P = 1.0,\ \ \rho = 0.01$	$n = 150$	$t = 2.0$
Alfvén wave ($\mu = 0.626$)	$u^i = (0.0, 0.0, 0.0)$ $B^i = (3.0, 3.0, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$u^i = (3.70, 5.76, 0.00)$ $B^i = (3.0, -6.857, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$n = 200$	$t = 2.0$
Compound wave	$u^i = (0.0, 0.0, 0.0)$ $B^i = (3.0, 3.0, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$u^i = (3.70, 5.76, 0.00)$ $B^i = (3.0, -6.857, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$n = 200$	$t = 0.1, 0.75, 1.5$
Shock Tube 1	$u^i = (0.0, 0.0, 0.0)$ $B^i = (1.0, 0.0, 0.0)$ $P = 1000.,\ \ \rho = 1.0$	$u^i = (0.0, 0.0, 0.0)$ $B^i = (1.0, 0.0, 0.0)$ $P = 1.0,\ \ \rho = 0.1$ (!)	$n = 400$	$t = 1.0$
Shock Tube 2	$u^i = (0.0, 0.0, 0.0)$ $B^i = (0.0, 20.0, 0.0)$ $P = 30.,\ \ \rho = 1.0$	$u^i = (0.0, 0.0, 0.0)$ $B^i = (0.0, 0.0, 0.0)$ $P = 1.0,\ \ \rho = 0.1$ (!)	$n = 500$	$t = 1.0$
Collision	$u^i = (5.0, 0.0, 0.0)$ $B^i = (10.0, 10.0, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$u^i = (-5.0, 0.0, 0.0)$ $B^i = (10.0, -10.0, 0.0)$ $P = 1.0,\ \ \rho = 1.0$	$n = 200$	$t = 1.22$

Table 3.1: Shock Test Initial Data

the following: fast_shock, slow_shock, fast_rare, slow_rare, shock1, shock2, collision, or alfven. Specifying one of these values for mhd_init chooses which values for the initial velocity, initial magnetic field, density, and pressure are chosen to be the initial data at runtime.

Shock Test Results

The following figure taken from [9] gives a baseline for what the test results should look like for each shock test type. This figure includes analytical calculations as well as simulated data.

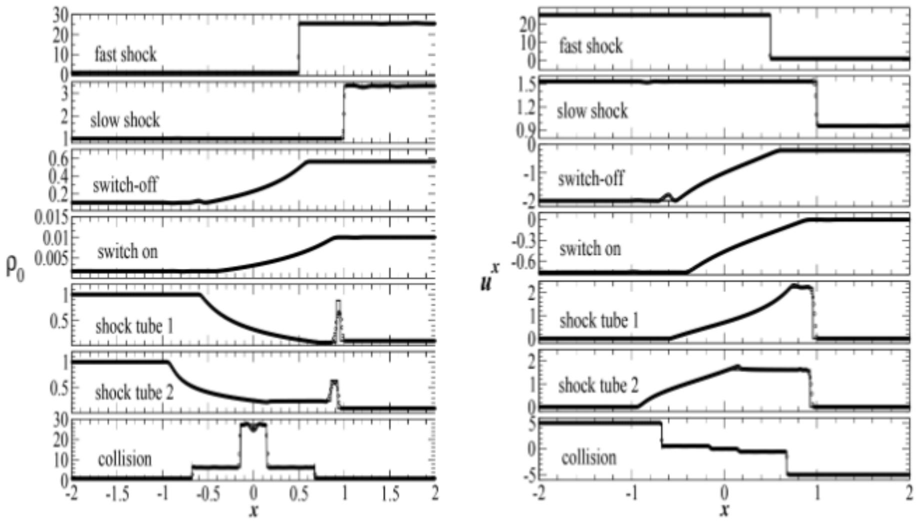

Figure 3.1: Density and Velocity Profiles from Duez et al.

The following figure taken from [12] shows the shock test results using FixedCosmo and 2nd-order finite differencing as the shock capturing method.

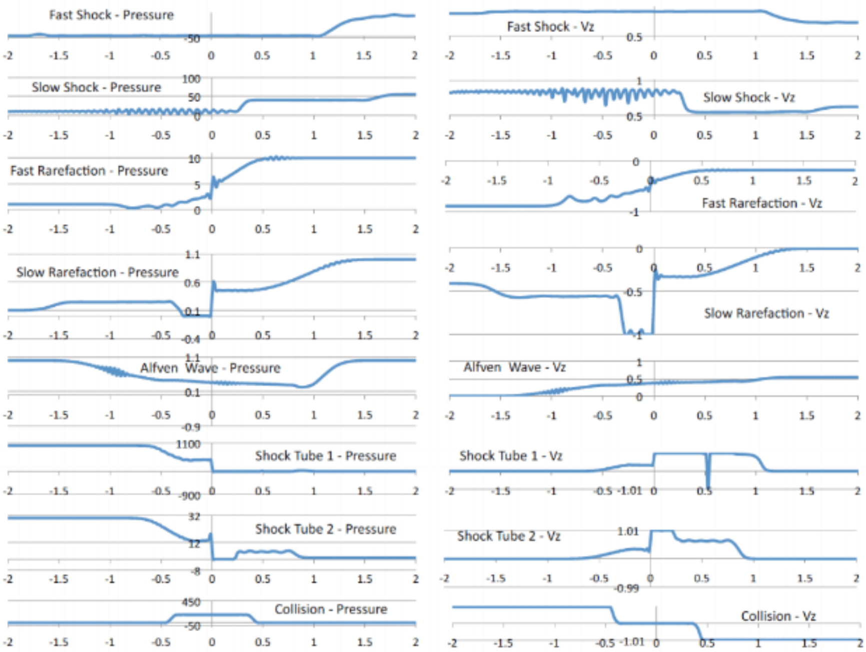

Figure 3.2: FixedCosmo Shock Tests using 2nd-order Finite Differencing

The shock tests presented in work by [12] all utilized a Courant factor (dtfac in .par file and from this point forward) of 0.5. This factor adjusts the ratio between grid resolution and time step, and appears to have been small enough in the work that used FixedCosmo to allow each shock test to run to completion. Using SpecCosmo, the first several shock test runs using a value of 0.5 and 0.05 for dtfac in this work quickly showed large spikes in the data sets near the discontinuity at z=0. After several trials, a value of 0.002 for dtfac was found to be successful for some of the tests. Smaller values for dtfac that allowed the other tests to compete were determined using the same trial-and-error process. The table below details the shock test type, grid resolution, number of iterations needed for completion, and the dtfac that was used for each test type. Other than the different values for dtfac, all parameters were maintained from the work done in [12] in order find a baseline for SpecCosmo as well as to compare how SpecCosmo handles shocks as opposed to FixedCosmo.

Test Type	*Grid Res.*	*Iterations*	*dtfac*
Fast Shock	0.1	250	0.001
Slow Shock	0.02	200	0.002
Switch-off Fast	0.0267	100	0.002
Switch-on Slow	0.008	500	0.0005
Shock Tube 1	0.008	500	0.000 05
Shock Tube 2	0.02	250	0.002
Collision	0.02	122	0.001
Alfven	0.02	240	0.002

Table 3.2: Parameter File Settings

Fast and Slow Shock

Below are the results for density and velocity for the Fast and Slow Shock tests at t_{final} using SpecCosmo.

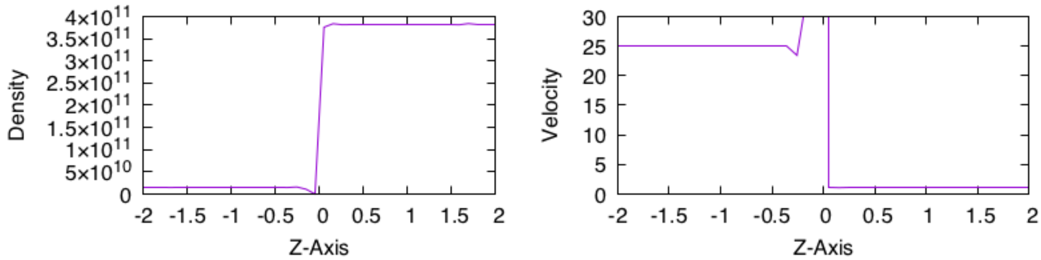

Figure 3.3: Fast Shock, Density and Velocity

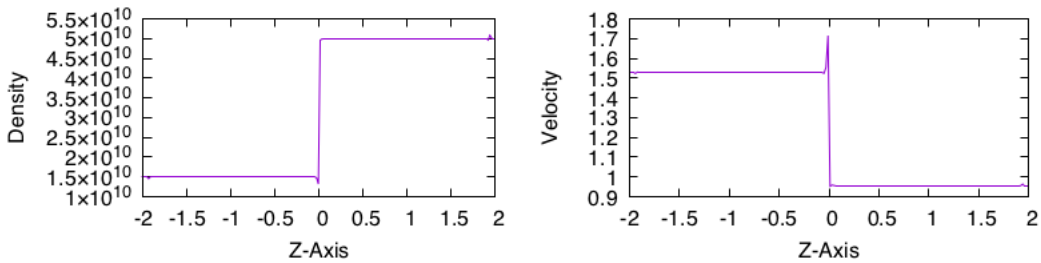

Figure 3.4: Slow Shock, Density and Velocity

Fast and Slow Rarefaction

Below are the results for density and velocity for the Fast and Slow Rarefaction tests at t_{final} using SpecCosmo.

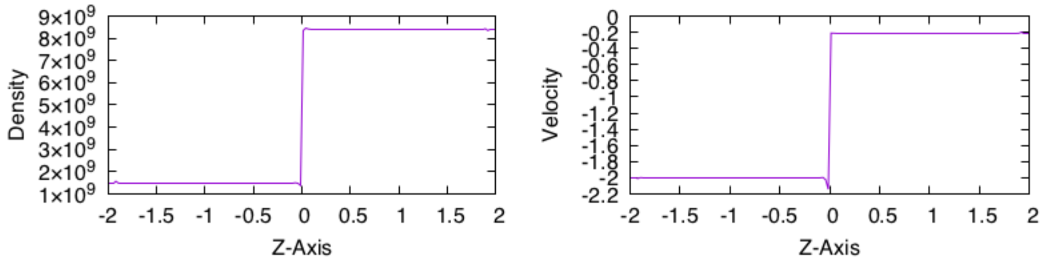

Figure 3.5: Switch-off Fast Rarefaction, Density and Velocity

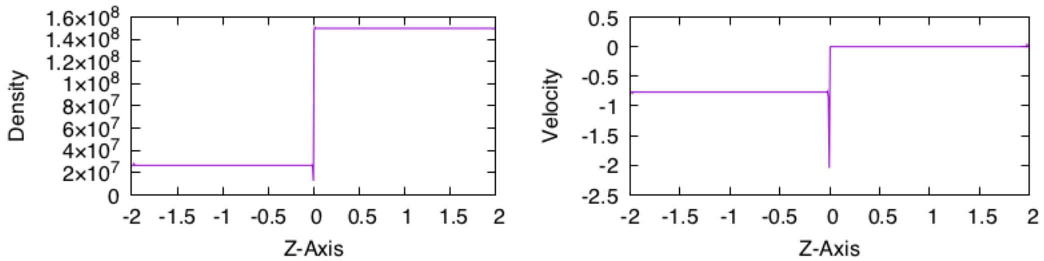

Figure 3.6: Switch-on Slow Rarefaction, Density and Velocity

Shock Tube 1 and 2

Below are the results for density and velocity for the Shock Tube 1 and Shock Tube 2 tests at t_{final} using SpecCosmo.

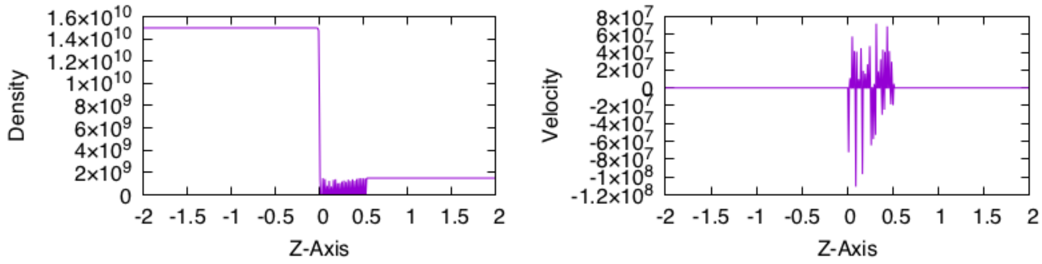

Figure 3.7: Shock Tube 1, Density and Velocity

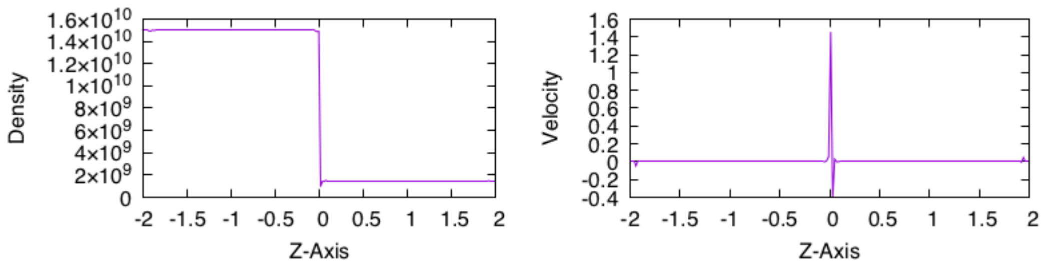

Figure 3.8: Shock Tube 2, Density and Velocity

Collision

Below are the results for density and velocity for the Collision test at t_{final} using SpecCosmo.

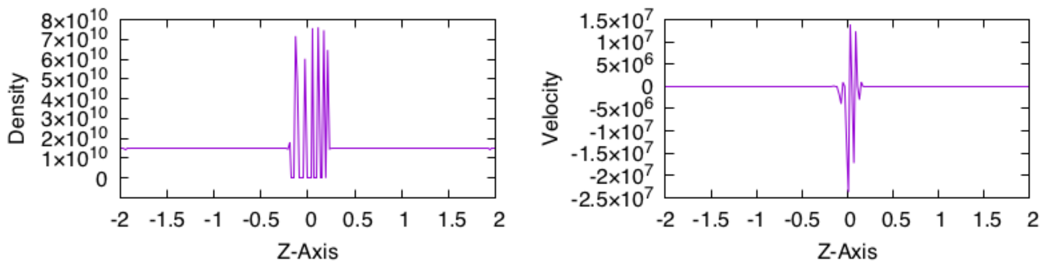

Figure 3.9: Collision, Density and Velocity

Alfven

Below are the results for density and velocity for the Alfven wave test at t_{final} using SpecCosmo.

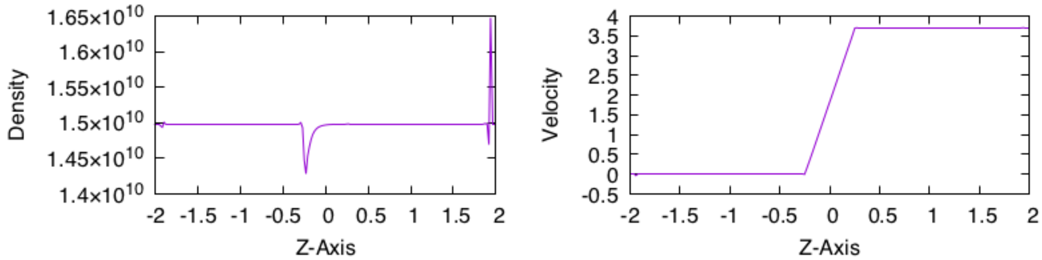

Figure 3.10: Alfven Wave, Density and Velocity

CHAPTER 4:
CONCLUSIONS AND DISCUSSION

Initial Conditions

In chapter 1, the FRW comsological model and resulting Friedmann equations that come from Einstein's field equations were developed. This model leads to the relationships between scale factor, temperature, age of the universe, and other parameters that were then used to calculated the initial conditions. In chapter 2, the initial conditions required to run a simulation using SpecCosmo were calculated at two different epochs in the early universe: the Electroweak phase transition and the QCD phase transition. These two epochs are important milestones in the universes evolution where certain details about the energy and matter content are known from particle physics research. Using what we know about particle physics at these epochs, it is then possible to calculate the energy content, temperature, critical density, size, and age of the universe at these times. All of these parameters are needed by the evolution equations in SpecCosmo as a starting point for a simulation. These initial conditions are supplied to the parameter file as initial input values for the SpecCosmo thorn. An example of a parameter file for the Biermann Electroweak simulation is given in the appendix, with the values from this work for the EW phase transition specified as inputs.

Shock Tests

In chapter 3, the suite of shock tests outlined by Komissarov [2], and reproduced in [4] and [12] for FixedCosmo was performed using SpecCosmo. The results for each shock test characterize SpecCosmo's ability to handle discontinuities that arise in data while simulations are running. Upon initial investigation, it was noted that the previously used value of 0.5 for dtfac was not usable in these tests. In order for the shock capturing scheme to handle the initial discontinuity without introducing numerical error into the data, dtfac needed to be several orders of magnitude smaller than in previous work. After several iterations for each shock test, it was observed that values for dtfac larger than 0.002, and therefore larger timesteps, resulted in infinities appearing in the data sets early on. However, all of the tests were able to finish the prescribed number of time iterations using the above documented values for dtfac in each test, the largest being 0.002. Even with reducing dtfac from what was usable in the shock tests with FixedCosmo, these dtfac values result in a much larger timestep than what would be used in an early universe simulation using SpecCosmo, which suggests that SpecCosmo is in fact handling the shocks quite well. The smallest values used for dtfac in these shock tests was 0.00005. The target value for dtfac in a simulation at the EW phase transition would be on the order of 10^{-13} and is adaptive, which would result in much smaller timesteps that are adjustable, even when taking differences in grid resolution into account. p

Future Work

The next step to build upon this work would be to utilize the initial conditions derived here at the Electroweak phase transition in order to study the development of initial magnetic fields in the early universe. In order to proceed on that front, the shock capturing ability of SpecCosmo has been tested and verified to meet expectations. It was observed in this work that some of the shock tests did not produce data sets that reflected previous work, perhaps due to SpecCosmo differing from FixedCosmo by having dynamically evolving spacetime rather than fixed or static spacetime. This difference in the evolution equations might cause perturbations or discontinuities to have a more significant effect on the data produced by the code. Adding artificial viscosity to the shock tests using SpecCosmo is another path forward. The addition of artificial viscosity should enhance SCR3's shock capturing ability and/or extend run time of the shock tests without numerical error. This may also allow larger values for dtfac to be used, which would in turn allow for larger timesteps. In terms of future work for initial conditions, it may also be of interest to investigate the ratio of dark matter to normal matter.

REFERENCES

[1] S. Komissarov, Monthly Notices of the Royal Astronomical Society **303**, 343 (1999).

[2] S. Komissarov, arXiv preprint astro-ph/0209213 (2002).

[3] T. W. Baumgarte and S. L. Shapiro, *Numerical relativity: solving Einstein's equations on the computer* (Cambridge University Press, 2010).

[4] D. Garrison, Journal of Gravity **2014** (2014).

[5] J. N. Islam, *An introduction to mathematical cosmology* (Cambridge University Press, 2002).

[6] B. W. Carroll and D. A. Ostlie, *An introduction to modern astrophysics* (Cambridge University Press, 2017).

[7] CactusCode, *http://cactuscode.org*.

[8] E. Toolkit, *http://einsteintoolkit.org*.

[9] M. D. Duez, Y. T. Liu, S. L. Shapiro, and B. C. Stephens, Physical Review D **72**, 024028 (2005).

[10] D. Garrison, Astronomische Nachrichten **342**, 75 (2021).

[11] L. Husdal, Galaxies **4**, 78 (2016).

[12] P. R. Smith, *High Resolution Shock Capturing for Use with SpecCosmo a General Relativity Magnetohydrodynamics Code*, Ph.D. thesis (2014).

APPENDIX:

PARAMETER FILES

Fast Sock Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.001

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 250
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 40
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"
```

```
mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0

mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "fast_shock"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"
```

```
#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/fast_shock_dtfac_001/chkpt"
IO::checkpoint_dir = "SCR3/fast_shock_dtfac_001/chkpt"
IO::recover_file = "SCR3/fast_shock_dtfac_001/chkpt"
IO::recover_dir = "SCR3/fast_shock_dtfac_001/chkpt"
IO::out_dir = "SCR3/fast_shock_dtfac_001"
#IO::recover = "autoprobe"
```

Slow Shock Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.002

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 200
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 200
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "slow_shock"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/slow_shock_dtfac_002/chkpt"
IO::checkpoint_dir = "SCR3/slow_shock_dtfac_002/chkpt"
IO::recover_file = "SCR3/slow_shock_dtfac_002/chkpt"
IO::recover_dir = "SCR3/slow_shock_dtfac_002/chkpt"
IO::out_dir = "SCR3/slow_shock_dtfac_002"
#IO::recover = "autoprobe"
```

Fast Rarefaction Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.002

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 100
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 150
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "fast_rare"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/fast_rare_dtfac_002/chkpt"
IO::checkpoint_dir = "SCR3/fast_rare_dtfac_002/chkpt"
IO::recover_file = "SCR3/fast_rare_dtfac_002/chkpt"
IO::recover_dir = "SCR3/fast_rare_dtfac_002/chkpt"
IO::out_dir = "SCR3/fast_rare_dtfac_002"
#IO::recover = "autoprobe"
```

Slow Rarefaction Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.0005

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 500
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 500
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "slow_rare"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/slow_rare_dtfac_0005/chkpt"
IO::checkpoint_dir = "SCR3/slow_rare_dtfac_0005/chkpt"
IO::recover_file = "SCR3/slow_rare_dtfac_0005/chkpt"
IO::recover_dir = "SCR3/slow_rare_dtfac_0005/chkpt"
IO::out_dir = "SCR3/slow_rare_dtfac_0005"
#IO::recover = "autoprobe"
```

Shock Tube 1 Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.00005

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 500
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 500
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "shock1"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/shock1_dtfac_00005/chkpt"
IO::checkpoint_dir = "SCR3/shock1_dtfac_00005/chkpt"
IO::recover_file = "SCR3/shock1_dtfac_00005/chkpt"
IO::recover_dir = "SCR3/shock1_dtfac_00005/chkpt"
IO::out_dir = "SCR3/shock1_dtfac_00005"
#IO::recover = "autoprobe"
```

Shock Tube 2 Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.002

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 250
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 200
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "shock2"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/shock2_dtfac_002/chkpt"
IO::checkpoint_dir = "SCR3/shock2_dtfac_002/chkpt"
IO::recover_file = "SCR3/shock2_dtfac_002/chkpt"
IO::recover_dir = "SCR3/shock2_dtfac_002/chkpt"
IO::out_dir = "SCR3/shock2_dtfac_002"
#IO::recover = "autoprobe"
```

Collision Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.001

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 122
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 200
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "collision"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/collision_dtfac_001/chkpt"
IO::checkpoint_dir = "SCR3/collision_dtfac_001/chkpt"
IO::recover_file = "SCR3/collision_dtfac_001/chkpt"
IO::recover_dir = "SCR3/collision_dtfac_001/chkpt"
IO::out_dir = "SCR3/collision_dtfac_001"
#IO::recover = "autoprobe"
```

Alfven Parameters

```
ActiveThorns = "slab boundary cartgrid3d coordbase ioascii iobasic ioutil localinterp
localreduce mol symbase time iojpeg pugh pughinterp pughreduce pughslab nanchecker
mhd_analysis mhd_init specgrmhd"

#time::timestep_method = "given"
#time::timestep = 0.0005
time::dtfac = 0.002

#Cactus::terminate = "runtime"
#Cactus::max_runtime = 5760
Cactus::cctk_itlast = 240
Cactus::cctk_initial_time = 0.0
cactus::cctk_timer_output = "full"
pugh::timer_output = "yes"

Nanchecker::check_every = 10
Nanchecker::check_after = 0
Nanchecker::check_vars = "all"
Nanchecker::action_if_found = "terminate"

grid::type = "byrange"
grid::xmin = -0.04
grid::xmax = 0.04
grid::ymin = -0.04
grid::ymax = 0.04
grid::zmin = -2.0
grid::zmax = 2.0
grid::domain = "full"

driver::global_nx = 7
driver::global_ny = 7
driver::global_nz = 200
driver::ghost_size = 3

driver::periodic = "no"
driver::periodic_x = "yes"
driver::periodic_y = "yes"
driver::periodic_z = "yes"

mol::ode_method = "ICN"
mol::MoL_Intermediate_Steps = 3
mol::MoL_Num_Scratch_Levels = 0
```

```
mhd_init::initial_data = "flatspace"
mhd_init::initial_gauge = "geodesic"
mhd_init::gauge_condition = "geodesic"
mhd_init::initial_mhd = "alfven"
mhd_init::itime = 0.0
mhd_init::maxvel = 1.00
#mhd_init::update_Hubble = "no"

specgrmhd::ch = 1.0
specgrmhd::cp = 10.0
specgrmhd::bound = "flat"
specgrmhd::diff = "SCR3"
specgrmhd::kphi = 1.0
specgrmhd::kgam = 1.0
specgrmhd::kalpha = 1.0
specgrmhd::kbeta = 1.0
#specgrmhd::fix_lapse = "yes"
#specgrmhd::fix_shift = "yes"
#specgrmhd::slicing = "None"
#specgrmhd::sigma = 0.5
specgrmhd::add_AV_bulk = "no"
specgrmhd::add_AV_shear = "no"
specgrmhd::add_DM = "no"
#specgrmhd::calc_constraint = "no"
specgrmhd::nn = 1.0
specgrmhd::kl = 0.1
specgrmhd::kq = 0.1
specgrmhd::scp = 0.08
specgrmhd::e1 = 0.000000000000001
#specgrmhd::sensitivity = 0.0000000000000001

iobasic::outinfo_every = 1
iobasic::outinfo_vars = "mhd_analysis::rho_out mhd_analysis::temp_out"
IOBasic::outScalar_style = "gnuplot"
IOASCII::out1D_style = "gnuplot f(x)"
IOBasic::outScalar_every = 1
IOBasic::outScalar_vars = "specgrmhd::calcvars mhd_analysis::output"

IOASCII::out1D_every = 1
IOASCII::out1D_vars = "specgrmhd::calcvars mhd_analysis::output"

#IOHDF5::out_every = 100
#IOHDF5::out_vars = "mhd_analysis::output"

#IOHDF5::checkpoint = "yes"
```

```
#IO::checkpoint_every = 1000

#IO::out_mode = "onefile"
#IO::out_unchunked = "yes"
IO::checkpoint_file = "SCR3/alfven_dtfac_001/chkpt"
IO::checkpoint_dir = "SCR3/alfven_dtfac_001/chkpt"
IO::recover_file = "SCR3/alfven_dtfac_001/chkpt"
IO::recover_dir = "SCR3/alfven_dtfac_001/chkpt"
IO::out_dir = "SCR3/alfven_dtfac_001"
#IO::recover = "autoprobe"
```