

LA-UR- 09-00559

Approved for public release;  
distribution is unlimited.

Title: Cosmological AMR MHD with Enzo

Author(s): Hao Xu  
Hui Li  
Shengtai Li

Intended for: The Astrophysical Journal  
Supplement Series



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Cosmological AMR MHD with Enzo

D. C. Collins<sup>1</sup>, H. Xu<sup>1,2</sup>, M.L. Norman<sup>1</sup>, H. Li<sup>2</sup>, S. Li<sup>2</sup>

## ABSTRACT

In this work, we present EnzoMHD, the extension of the cosmological code Enzo to include magnetic fields. We use the hyperbolic solver of Li et al. (2008) for the computation of interface fluxes. We use constrained transport methods of Balsara & Spicer (1999) and Gardiner & Stone (2005) to advance the induction equation, the reconstruction technique of Balsara (2001) to extend the Adaptive Mesh Refinement of Berger & Colella (1989) already used in Enzo, though formulated in a slightly different way for ease of implementation. This combination of methods preserves the divergence of the magnetic field to machine precision. We use operator splitting to include gravity and cosmological expansion. We then present a series of cosmological and non cosmological tests problems to demonstrate the quality of solution resulting from this combination of solvers.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Numerics</b>	<b>5</b>
2.1	Cosmological MHD Equations . . . . .	5
2.2	Data Structure . . . . .	6
2.3	Consistency . . . . .	7
2.4	Time Stepping . . . . .	8
2.5	Boundary Conditions and Ghost Zones . . . . .	9
2.6	Left Hand Side: Hyperbolic terms . . . . .	10
2.7	Constrained Transport and the Divergence of $\mathbf{B}$ . . . . .	11
2.8	Right Hand Side: Gravitational Acceleration . . . . .	12

---

<sup>1</sup>Center for Astrophysics and Space Sciences, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093

<sup>2</sup>Theoretical division, Los Alamos National Lab, Los Alamos, NM 87545

2.9	Right Hand Side: Expansion Source Terms . . . . .	13
2.10	Dual Energy Formalism . . . . .	13
2.11	Adaptive Mesh Refinement . . . . .	14
<b>3</b>	<b>Numerical Experiments</b>	<b>18</b>
3.1	MHD Tests without AMR . . . . .	18
3.1.1	Brio and Wu shock tube . . . . .	18
3.1.2	Isothermal Tests . . . . .	18
3.1.3	One-dimension MHD Caustics . . . . .	20
3.1.4	The Zel'Dovich Pancake . . . . .	20
3.1.5	Orszag-Tang . . . . .	24
3.1.6	3D Adiabatic Universe with MHD . . . . .	24
3.2	MHD Tests with AMR . . . . .	28
3.2.1	Three-dimension MHD Adiabatic Expansion . . . . .	28
3.2.2	One-dimensional MHD Caustics with AMR . . . . .	28
3.2.3	Zel'Dovich Pancake with AMR . . . . .	28
3.2.4	MHD Galaxy Cluster Formations . . . . .	34
<b>4</b>	<b>Conclusion</b>	<b>48</b>
<b>5</b>	<b>Discussion</b>	<b>49</b>
<b>A</b>	<b>Schematic for the Cosmological MHD Code</b>	<b>49</b>
<b>B</b>	<b>AMR MHD Reconstruction</b>	<b>52</b>
B.1	Hydrodynamic Reconstruction . . . . .	52
B.2	MHD Reconstruction . . . . .	52
B.3	Implementation in Enzo . . . . .	54
<b>C</b>	<b>Flux Correction</b>	<b>55</b>
C.1	Conservation Form . . . . .	55

C.2 Conservation Form and AMR: Enter Flux Correction. . . . .	56
C.3 Flux Correction and MHD . . . . .	57



## 1. Introduction

Enzo is an adaptive mesh refinement (AMR), grid-based hybrid code (hydro + N-Body) which is designed to do simulations of cosmological structure formation. It uses the block-structured AMR algorithm of Berger & Colella (1989) to improve spatial resolution where required, such as in gravitationally collapsing objects. The method is attractive for cosmological applications because it: 1) is spatially- and time-adaptive, 2) uses accurate and well-tested grid-based methods for solving the hydrodynamics equations and 3) can be well optimized and parallelized. The central idea behind AMR is to solve the evolution equations on a fixed resolution grid, adding finer grids in regions that require enhanced resolution. Mesh refinement can be continued to an arbitrary level, based on criteria involving any combination of (dark-matter and/or baryon) over density, Jeans length, cooling time, etc, enabling users to tailor the adaptivity to the problem of interest. Enzo solves the following physics models: collisionless dark-matter and star particles, using the particle-mesh N-body technique (Hockney & Eastwood 1985); gravity, using FFTs on the root grid and multigrid relaxation on the subgrids; cosmic expansion; gas dynamics, using the piecewise parabolic method (PPM) (Colella & Woodward 1984) as extended to cosmology by (Bryan et al. 1995); multi-species non-equilibrium ionization and  $H_2$  chemistry, using backward Euler time differencing (Anninos et al. 1997); radiative heating and cooling, using subcycled forward Euler time differencing (?); and a parameterized star formation/feedback recipe (Cen & Ostriker 1993). Enzo has been successfully used in many cosmological applications, including star formation (Abel et al. 2000, 2002; O'Shea et al. 2005; O'Shea & Norman 2007), Lyman-alpha forest ((Bryan et al. 1999), (Jena et al. 2005)), interstellar medium (Kritsuk & Norman 2002, 2004) and galaxy clusters (Bryan & Norman 1998; Loken et al. 2002; Motl et al. 2004; Hallman et al. 2006). More informations about Enzo are available at <http://cosmos.ucsd.edu/enzo>.

One important piece of physics that is missing from this list is a proper treatment of magnetic fields. Magnetic fields have a broad range of impacts in a broad range of physical situations, from galaxy clusters to protostellar core formation. In this paper we will describe the numerical approximations both with and without cosmology in section 2 and present test problems in section 3. Creating a functional cosmological MHD code takes more than a single algorithm. The purpose of this paper is to document the construction and performance of the algorithms that will be used in MHD simulations with Enzo in the future, as well as simulations that have already been done ((Xu et al. 2008))

EnzoMHD is also a multi purpose code. In this version of this paper, we will discuss it as a cosmological code, but all the same machinery applies in non-cosmological mode. To treat this paper as a method for the non-cosmological MHD code, simply set  $a = 1$  and  $\dot{a} = 0$ , and ignore anything that says cosmology.

## 2. Numerics

### 2.1. Cosmological MHD Equations

Enzo solves the following comoving MHD equations:

$$\frac{\partial \rho}{\partial t} + \frac{1}{a} \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \frac{1}{a} \nabla \cdot (\rho \mathbf{v} \mathbf{v} + \bar{p} \mathbf{I} - \mathbf{B} \mathbf{B}) = -\frac{\dot{a}}{a} \rho \mathbf{v} - \frac{1}{a} \rho \nabla \Phi \quad (2)$$

$$\frac{\partial E}{\partial t} + \frac{1}{a} \nabla \cdot [\mathbf{v}(\bar{p} + E) - \mathbf{B}(\mathbf{B} \cdot \mathbf{v})] = -\frac{\dot{a}}{a} (\rho v^2 + 3p + \frac{B^2}{2}) - \frac{\rho}{a} \mathbf{v} \cdot \nabla \Phi \quad (3)$$

$$\frac{\partial \mathbf{B}}{\partial t} - \frac{1}{a} \nabla \times (\mathbf{v} \times \mathbf{B}) = -\frac{\dot{a}}{2a} \mathbf{B} \quad (4)$$

with the equation of state

$$E = \frac{1}{2} \rho v^2 + \frac{p}{\gamma - 1} + \frac{1}{2} B^2 \quad (5)$$

$$\bar{p} = p + \frac{1}{2} B^2 \quad (6)$$

Here,  $\rho$  is the comoving density,  $p$  is the comoving gas pressure,  $\mathbf{v}$  is the proper peculiar velocity,  $\mathbf{B}$  is the comoving magnetic field,  $E$  is the total peculiar energy per unit comoving volume,  $\bar{p}$  is the total comoving pressure,  $\gamma$  is the ratio of the specific heats,  $\Phi$  is the proper peculiar gravitational potential from both dark-matter and baryons,  $a = (1 + z_i)/(1 + z)$  is the expansion factor and  $t$  is time.

For non-cosmological simulations, the same equations hold, with  $a = 1$  and  $\dot{a} = 0$ . This effectively removed each appearance of  $a$  from the left hand side, and eliminates the terms involving  $\dot{a}$  from the right.

In this formulation, the comoving quantities that are evolved by the solver are related to the proper observable quantities by the following equations:

$$\rho_{proper} = \rho * a(t)^3 \quad (7)$$

$$\Phi_{proper} = \Phi - \frac{1}{2} a \ddot{a} \bar{x}^2 \quad (8)$$

$$\mathbf{B}_{proper} = \mathbf{B} a^{-\frac{3}{2}} \quad (9)$$

To solve these equations, we operator split eqns (1)-(4) into four parts: the left hand side of equations (1)-(3), the left hand side of equation (4), the gravitational acceleration (the two terms involving  $\nabla \Phi$ ), and the expansion terms (the two terms involving  $\dot{a}$ ) These will be discussed in sections 2.6 - 2.7. In section 2.10, we will discuss the Dual Energy formulation in Enzo for

hypersonic flows, and in section 2.11 we will discuss the Adaptive Mesh Refinement algorithm. We first discuss the data structures used to carry all this data in section 2.2

In the following, we will often have cause to separate the purely fluid dynamical quantities  $\rho, \vec{v}, E$  from the magnetic field  $\vec{B}$ . Unless otherwise noted, 'fluid quantities' will refer to the former only.

For ease of reference, we have supplied a schematic summary of the steps involved in appendix A.

## 2.2. Data Structure

In Enzo, both parallelism and AMR are done in block decomposed manner. Each patch of space, called a **grid**, is treated as a unique computational problem with Dirichlet boundary conditions which are stored in a number of Ghost Zones. The number of ghost zones depends on the method used. The pure-hydro methods in Enzo, ZEUS and PPM, use 3 ghost zones. The method we describe here uses 5 ghost zones.

Grids are arranged in a strictly nested hierarchy, with each grid having a cell width half that of its parent (pure hydro Enzo can take any integer refinement, but the interpolation for MHD is restricted to factors of 2.) See figure 1. Each processor keeps a copy of the entire hierarchy, while only one of the processors actually stores the data.

For all physics modules described in this paper, an individual grid cares not for where it sits in space or the hierarchy, and communicates with other grids only through boundary condition fills (section 2.5) and the AMR cycle (section 2.11).

EnzoMHD in its default mode tracks 14 fields, stored at 3 different points of the cell. The 5 hydrodynamic quantities,  $\rho, \vec{v}, E_{total}$  are stored at the center of the cell, denoted  $(i, j, k)$ , and represent the volume average of the respective quantities. These are the same quantities stored in non-MHD Enzo.

EnzoMHD tracks 2 copies of the magnetic field and the electric field. One copy of the magnetic field is stored in the face of the cell perpendicular to that field component, and represents the area average of that field component over that face. This is the primary representation of the magnetic field. So  $B_{f,x}$  is stored in the center of the  $x$  face, denoted  $(i - \frac{1}{2}, j, k)$ ,  $B_{f,y}$  in the  $y$  face at  $(i, j - \frac{1}{2}, k)$ , and  $B_{f,z}$  in the  $z$  face at  $(i, j, k - \frac{1}{2})$ . It is this field that remains divergence free under



the cell centered divergence operator:

$$\begin{aligned}\nabla \cdot \mathbf{B}_f = & \frac{1}{\Delta x} (B_{f,x,i+\frac{1}{2},j,k} - B_{f,x,i-\frac{1}{2},j,k}) + \\ & \frac{1}{\Delta y} (B_{f,y,i,j+\frac{1}{2},k} - B_{f,y,i,j-\frac{1}{2},k}) + \\ & \frac{1}{\Delta z} (B_{f,z,i,j,k+\frac{1}{2}} - B_{f,z,i,j,k-\frac{1}{2}})\end{aligned}\quad (10)$$

This data structure is one element longer in each longitudinal direction, so for an  $nx \times ny \times nz$  grid patch, the  $B_{f,x}$  structure is  $(nx + 1) \times ny \times nz$ .

The second representation of the magnetic field is centered with the fluid quantities at the center of the cell. This field is used wherever a cell centered magnetic quantity is needed, most notably in the hyperbolic solver in section 2.6. It's equal to the first order average of the face centered magnetic field:

$$\begin{aligned}B_{c,x,i,j,k}^{n+1} &= 0.5 * (B_{f,x,i+\frac{1}{2},j,k} + B_{f,x,i-\frac{1}{2},j,k}) \\ B_{c,y,i,j,k}^{n+1} &= 0.5 * (B_{f,y,i,j+\frac{1}{2},k} + B_{f,y,i,j-\frac{1}{2},k}) \\ B_{c,z,i,j,k}^{n+1} &= 0.5 * (B_{f,z,i,j,k+\frac{1}{2}} + B_{f,z,i,j,k-\frac{1}{2}})\end{aligned}\quad (11)$$

The final data structure used in EnzoMHD is the Electric Field, which is stored along the edges of the computational cell. This represents a linear average of the electric field along that line element. Each component is centered along the edge its parallel to, so  $E_x$  lies along the  $x$  edge of the cell at  $(i, j - \frac{1}{2}, k - \frac{1}{2})$ , etc. It is longer than the fluid fields by one in each transverse direction, so  $E_x$  would be  $nx \times (ny + 1) \times (nz + 1)$ .

Each grid also stores one copy of each of the above mentioned fields for use in assigning ghost zones to subgrids. This is described further in 2.5. A temporary field for fluxes is also stored, which exists only while the hyperbolic terms are being updated. This data structure is also stored on the faces of the zone. There are three fluxes for all 7 MHD quantities.

For other configurations of EnzoMHD, more or fewer zones may be used. In purely isothermal mode (which is at present an option only in EnzoMHD, not in Enzo) the total energy field is not tracked, and the isothermal sound speed is taken as a global scalar quantity. This reduces the number of fields tracked everywhere the total energy shows up. With dual energy formalism on (see section 2.10) an additional field corresponding to either gas energy or entropy is stored, giving an additional field where needed.

### 2.3. Consistency

In several places throughout the flow of Enzo, there may be more than one data structure using and writing to a given variable at a given point in space. Ghost zones and face centered fields

(fluxes and magnetic fields) are examples of this. In EnzoMHD, it is imperative that all data at a given point is identical, regardless of the data structure describing it. This may seem like an unnecessary comment, but it isn't; in pure hydro simulations, numerical viscosity will damp out small perturbations caused by slight inconsistencies in data description. Thus in practice, especially in large, stochastic simulations, errors can (and have) go (and gone) unnoticed. By construction EnzoMHD preserves  $\nabla \cdot \mathbf{B}$  to machine precision, but it never *forces*  $\nabla \cdot \mathbf{B} = 0$ ; so if it's not zero at the beginning of a time step, it's not going to be at the end, either. It is also worth mentioning that inconsistencies in and quantity will cause inconsistencies in the flow, which will in turn cause  $\nabla \cdot \mathbf{B}$  issues. Thus any improper handling of *any* fluid quantity will cause errors in  $\nabla \cdot \mathbf{B}$  that will persist and usually grow to catastrophic proportions in a relatively short period of time.

There is a prominent redundancy in the magnetic field, namely the field on the surface of the active zones of grids. See figure 2. Care is taken to include enough ghost zones, and frequent enough ghost zone exchange between grids, that after a time step, two neighboring grids have reached exactly the same answer on the surface between the two grids completely independently.

For the record, and for a few pages of my thesis, these data inconsistencies have come about in the gravity and flux correction modules in Enzo, as well as necessitating several additional calls to the boundary condition routines. These have been fixed or kludged. More descriptions can be found in section ?? . This appendix will likely not appear in the paper, but will in my thesis so is included in this draft.

## 2.4. Time Stepping

Enzo uses hierarchical time stepping to determine it's time step. For each level, the harmonic mean of the 3 Courant conditions is used. This was demonstrated to be the most robust time stepping criterion possible for multi dimensional flows

$$\begin{aligned} \Delta t &= \frac{1}{1/t_x + 1/t_y + 1/t_z} \\ t_x &= \min\left(\frac{\Delta x}{c_{f,x}}\right) \\ t_y &= \min\left(\frac{\Delta y}{c_{f,y}}\right) \\ t_z &= \min\left(\frac{\Delta z}{c_{f,z}}\right) \end{aligned} \tag{12}$$

where the *min* is taken over the computational domain, and  $c_{f,x}$ ,  $c_{f,y}$  and  $c_{f,z}$  are the fast MHD shock speeds along each axis:

$$c_{f,x}^2 = \frac{1}{2} \left( a^2 + \frac{\mathbf{B} \cdot \mathbf{B}}{\rho} + \sqrt{\left( a^2 + \frac{\mathbf{B} \cdot \mathbf{B}}{\rho} \right)^2 - 4a^2 B_x^2 / \rho} \right) \tag{13}$$

and similar definition for the other two.

Thus, if given level has a cell size  $\Delta x$  and the next level of refinement has cell size  $\frac{\Delta x}{r}$ , the more refined grid will have, in principle, time step size  $\frac{\Delta t}{r}$ . In Enzo, the step size is chosen for each level and each subgrid time step. In practice, owing to more finely resolved structures having slightly higher fast shock speeds, fine grids may in fact take more than  $r$  time steps for each parent grid step.

Timesteps are taken in order of coarsest to finest, in a 'W' cycle. See figure 3 So given 3 levels, level 0 takes the first step of  $\Delta t$ . Then level 1 takes a single step of  $\Delta t/2$ . Then level 2 takes one step of  $\Delta t/4$ . Then, given that there are only three grids, it takes another. The last three steps repeat: level 1 then takes its second and final step of  $\Delta t/2$  so it is now at the same time as level 0, followed by two steps on level 2.

## 2.5. Boundary Conditions and Ghost Zones

Ghost Zones are filled in one of three means.

1. **Copying.** The dominant mechanism for filling ghost zones copying from active zones that occupy the same physical space. This also takes into account periodic boundary conditions. For EnzoMHD, face centered fields are copied from the faces of all cells, including those that boarder on active cells. this is somewhat redundant for reasons described in 2.3.
2. **External Root Grids** that lie along the domain wall filled with the External Boundary routine. If the external boundary condition is not periodic, the grids zones are filled by a predetermined algorithm; for instance, outflow boundary conditions set ghost zones to be equal to the outermost active zone, akin to a Neumann condition of zero slope. These involve outflow, reflecting, and a completely general 'inflow'. Note that this is called only on the root grid, and not on subgrids that happen to lie on the edge. This can cause spurious waves at reflecting or outflow boundaries with AMR. Also note for EnzoMHD, the only external boundary conditions that have been tested are periodic and outflow.
3. **Interpolation** The third mechanism is used on refined grids whose ghost zones do not occupy the active space of another grid; these grids have their ghost zones filled by interpolation from the parent grid. Since Enzo uses hierarchical time stepping, subgrid steps that begin in the middle of a parent grid step fill their ghost zones from a linear combination of the parent grid time steps at  $t^n$  and  $t^{n+1}$ .

## 2.6. Left Hand Side: Hyperbolic terms

With the exception of the  $1/a$  term that appears in front of each  $\nabla \cdot$  operator, the left hand side of equations 1-4 are the familiar Ideal MHD equations. A form of equations (1) - (4) more relevant for this treatment is the following:

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (14)$$

where

$$\mathbf{V} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ B_y \\ B_z \\ E \end{pmatrix} \quad (15)$$

$$\mathbf{F} = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p + B^2/2 - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ B_y v_x - B_x v_y = -E_z \\ B_z v_x - B_x v_z = E_y \\ (E + p + B^2/2)v_x - B_x(\mathbf{B} \cdot \mathbf{v}) \end{pmatrix} \quad (16)$$

$$p = (E - \frac{1}{2}\rho \mathbf{v}^2 + \frac{1}{2}\mathbf{B}^2(\gamma - 1)) \quad (17)$$

These form a hyperbolic system of equations, which have been studied extensively in the literature. To take advantage of the work already done on this type of system of equations, we first multiply the cell width  $dx$  by  $a$ . We can then apply the hyperbolic solver of our choice as if nothing were wrong. Upon exiting the solver,  $dx$  is divided by  $a$  to restore it to the original proper value.

Equation 14 is solved by first re-writing it in conservation form, that is taking suitable integrals in time and space. The resulting update is, in one dimension,

$$\hat{V}_{i,j,k}^{n+1} = \hat{V}_{i,j,k}^n - \frac{\Delta t}{\Delta x} (\hat{F}_{x,i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \hat{F}_{x,i-\frac{1}{2},j,k}^{n+\frac{1}{2}}) \quad (18)$$

where  $\hat{V}$  represents the appropriate spatial average of the conserved quantities, and  $\hat{F}$  represents an approximate average to the flux, centered in time at  $t = t + \Delta t/2$ .  $\hat{V}$  is the quantity we store in the cells, and  $\hat{F}$  comes from the hyperbolic solver.

The solver we use to solve the hyperbolic equations is that of Li et al. (2008), which comes in three parts: spatial reconstruction, time centering, and the solution of the Riemann problem. Spatial reconstruction is done using piecewise linear monotonized slopes on the primitive variables



$(\rho, \mathbf{v}, p, \mathbf{B})$ . Time centering of the interface states by  $\Delta t/2$  is performed using either the MUSCL-Hancock or Piecewise Linear Method integration. The Riemann problem is then solved using either the HLLC Riemann solver Li (2005), HLLD solver of Miyoshi & Kusano (2005), or the isothermal HLLD solver of Mignone (2007). These fluxes are computed for the conserved variables  $(\rho, \rho \mathbf{v}, E, \mathbf{B})$ . These fluxes are then differenced to obtain the update values of the fluid quantities only. The fluxes for the magnetic field are stored for use in the Constrained Transport algorithm, discussed in section 2.7. This is done in one dimension on successive sweeps along the  $x, y$ , and  $z$  directions. To reduce operator splitting error, the order of the sweeps is permuted. For more details, see Li et al. (2008).

In isothermal mode, the same method is used, but the energy terms in  $V$  and  $F$  are removed.

In this step, the centered magnetic field  $B_c$  is used.

## 2.7. Constrained Transport and the Divergence of B

One of the biggest challenges for an MHD code is to maintain the divergence free constraint on the magnetic field ( $\nabla \cdot \mathbf{B} = 0$ ). Brackbill & Barnes (1980) found that non-zero divergence can grow exponentially during the computation and cause the Lorentz force to be non-orthogonal to the magnetic field. There are three major ways to assure the divergence remains zero. The first is a divergence-cleaning (or Hodge Projection) approach by Brackbill & Barnes (1980), which solves an extra Poisson's equation to recover  $\nabla \cdot \mathbf{B} = 0$  at each time step. But Balsara & Kim (2004) found that non-locality of the Poisson solver introduces substantial spurious small scale structures in the solution. Additionally, solving Poisson's equation on an AMR mesh is computationally expensive. The second method involves extending the MHD equations to include a divergence wave, as done by Powell et al. (1999), Dedner et al. (2002), which then advects the divergence out of the domain. As most of our solutions are done on periodic domains, this is also an undesirable solution. The third method, and the one we have employed in Enzo, is the constrained transport (CT) method of Evans & Hawley (1988). This method centers the magnetic field on the faces of the computational cells and the electric field on the edges. Once the electric field is computed (more on this later) it's curl is taken to update the magnetic field. This ensures  $\nabla \cdot \mathbf{B} = 0$  for all time, provided it's true initially.

$$\begin{aligned} \hat{B}_{f,x,i-\frac{1}{2},j,k}^{n+1} = \hat{B}_{x,i-\frac{1}{2},j,k}^n - \Delta t \left( \frac{1}{\Delta y} (\hat{E}_{z,i-\frac{1}{2},j+\frac{1}{2},k} - \hat{E}_{z,i-\frac{1}{2},j-\frac{1}{2},k}) + \right. \\ \left. \frac{1}{\Delta z} (\hat{E}_{y,i-\frac{1}{2},j,k+\frac{1}{2}} - \hat{E}_{y,i-\frac{1}{2},j,k-\frac{1}{2}}) \right) \end{aligned} \quad (19)$$

Plugging equation C13 into the divergence operator 10 to find  $\nabla \cdot \mathbf{B}_f^{n+1}$ , one finds all terms are eliminated except the initial divergence  $\nabla \cdot \mathbf{B}_f^n$ .

The CT algorithm of Evans & Hawley (1988) was extended to work with finite volume methods by Balsara & Spicer (1999). This method uses the fact that the MHD Flux has the electromotive



force as two of its components (see the 5<sup>th</sup> and 6<sup>th</sup> components of eqn. 16), so using these components then incorporates all the higher order and shock capturing properties of the Godunov solver into the evolution of the electric field. These components, which are centered at the face the computational cell, are then averaged to obtain an electric field at the edges of the cell. This was the first CT method applied to Enzo, so unless otherwise noted, the simulations presented here were done with this method. The reader is encouraged to read Balsara & Spicer (1999) for the full details.

Gardiner & Stone (2005) extended this idea to include higher order spatial averaging, which eliminates a number of numerical artifacts present in Balsara & Spicer (1999) and increases the accuracy of the method. This method uses the fluxes from the Riemann solver, plus additional information from the data in the cell to construct a linear interpolation from the cell face to the cell edge. The reader is encouraged to see that paper for the details.

After the curl is taken and the face centered field  $\mathbf{B}_f$  is updated, it is then averaged to obtain  $\mathbf{B}_c$ , via equation 11.

## 2.8. Right Hand Side: Gravitational Acceleration

In cosmological simulations, Enzo tracks the proper peculiar gravitational potential.

$$\nabla^2 \Phi = \frac{4\pi G}{a} (\rho_b + \rho_d - \rho_0) \quad (20)$$

where  $\rho_b$  and  $\rho_d$  are baryonic and dark matter comoving density respectively, and  $\rho_0$  is the comoving background density. For non-cosmological simulations, the dark matter and background density are ignored.

The gravitational potential  $\Phi$  is solved in Enzo using a combination of methods. First, the root grid potential (which covers the entire computational domain) is solved for using a fast Fourier transform. Then the subgrids (which hopefully do not cover the computational domain) are solved using a multigrid relaxation technique. This resulting potential  $\Phi$  is then differenced to obtain the acceleration  $\mathbf{g} = \nabla \Phi$ . Specifically,

$$\mathbf{g}_i = \frac{1}{2} (\Phi_{i+1} - \Phi_{i-1}) \quad (21)$$

As mentioned before, the fluxes are computed at the half time point  $t + 1/2\Delta t$ . In order to keep the velocity and consistent with this time centering, they are first advanced by a half time step:

$$\mathbf{v} = \mathbf{v} + \frac{\Delta t}{2} \mathbf{g} \quad (22)$$

After the fluxes are differenced to obtain the new state  $v_x^{n+1}$ , these states are then updated with the accelerations. For the velocity update, a density field centered in time is used. We follow

the same formulation used by Colella & Woodward (1984)

$$v_x^{n+1} = v_x'^{n+1} + \Delta t \frac{\frac{1}{2}(\rho^{n+1} + \rho^n)A_x}{\rho^{n+1}} \quad (23)$$

$$E^{n+1} = E'^{n+1} - \frac{1}{2}\rho^{n+1}(v_x'^{n+1})^2 + \frac{1}{2}\rho^{n+1}(v_x^{n+1})^2 \quad (24)$$

## 2.9. Right Hand Side: Expansion Source Terms

The cosmological expansion source terms are treated in much the same the same way as the gravitational source terms. First, a half time step is added to the values before the flux is computed.

$$\mathbf{v}'^n = \mathbf{v}^n - \frac{1}{2}\Delta t \frac{\dot{a}}{a} \rho^n \quad (25)$$

$$p'^n = p^n - \frac{1}{2}\Delta t \frac{\dot{a}}{a} 3(\gamma - 1)p^n \quad (26)$$

$$\mathbf{B}_c'^n = \mathbf{B}_c^n - \frac{1}{2}\Delta t \frac{\dot{a}}{2a} \mathbf{B}_c^n \quad (27)$$

The quantities  $\mathbf{v}'^n$ ,  $p'^n$  and  $\mathbf{B}'^n$  are then used in the rest of the solver described in section 2.6. After the fluxes are differenced, the source terms are then added to the fluid quantities in full. This is done in a semi-implicit manner, by averaging the quantities to be updated in time. For instance, the expansion contribution to the magnetic field is

$$\frac{\partial \mathbf{B}}{\partial t} = -\frac{\dot{a}}{2a} \mathbf{B} \quad (28)$$

which is discretized

$$\mathbf{B}_{exp}^{n+1} - \mathbf{B}^{n+1} = -\frac{\dot{a}}{2a} \left( \frac{\mathbf{B}_{exp}^{n+1} + \mathbf{B}^{n+1}}{2} \right) \quad (29)$$

and solving for  $\mathbf{B}_{exp}^{n+1}$  we have

$$x = \frac{\dot{a}}{4a} \quad (30)$$

$$\mathbf{B}_{exp}^{n+1} = \frac{(1-x)}{(1+x)} \mathbf{B}^{n+1} \quad (31)$$

Pressure and velocity are updated in manner. See appendix A for the full update.

## 2.10. Dual Energy Formalism

Hypersonic flows are quite common in cosmological simulations. Due to the extremely large gravitational forces, the ratio of kinetic energy  $E_k$  to gas internal energy  $e$  can be as high as  $10^8$ .

This leads to problems when computing the internal energy in this type of flow, as the universe does math with infinite accuracy, but computers do not. Higher order Godunov code typically track only the Total Energy (equation 5). Thus finding the internal energy

$$E_{internal} = E_{total} - E_{kinetic} - E_{magnetic}$$

involves the small difference of two (or three) large numbers, which causes problems when the small number ( $E_{internal}$ ) is near the roundoff noise of the original numbers ( $E_{total}$  and  $E_{kinetic} + E_{magnetic}$ ).

To overcome this, we have implemented two algorithms that solve an additional equation to track the small numbers; the modified entropy equation given in Ryu et al. (1993) and the internal energy equation given in Bryan et al. (1995). These two equations are:

$$\frac{\partial S}{\partial t} + \frac{1}{a} \nabla \cdot (S \mathbf{v}) = -\frac{3(\gamma - 1)\dot{a}}{a} S \quad (32)$$

$$\frac{\partial \rho e}{\partial t} + \frac{1}{a} \nabla \cdot (\rho e \mathbf{v}) = -\frac{3(\gamma - 1)\dot{a}}{a} \rho e + \frac{p}{a} \nabla \cdot \mathbf{v} \quad (33)$$

where  $S \equiv p/\rho^{\gamma-1}$  is the comoving modified entropy and  $e$  is the internal energy. The modified entropy equation is valid only outside the shocks where the entropy is conserved. Use of either (not both) of these equations is at the discretion of the simulator.

Through the course of the simulation, the ratio of Internal Energy to Total Energy is monitored. When this ratio is less than some pre-set value  $\eta$ , one of the modified equations is used. As in Li et al. (2008), we use  $\eta = 0.008$ . Li et al. (2008) compared this two approaches and found almost identical results.

## 2.11. Adaptive Mesh Refinement

Structured AMR, initially devised by Berger & Colella (1989), is a technique for increasing resolution of a simulation in parts of a simulation that “need it most,” while conserving memory and CPU cycles in areas that don’t. Refinement criteria will not be described here, as they vary from simulation to simulation. AMR has four basic necessary parts:

1. **Patch Solver** This is the algorithm that actually solves the finite volume PDEs in question, as described by sections 2.6 - 2.10.
2. **Refinement Operator** This is the routine that creates fine resolution elements from coarse ones. In *Enzo*, we use conservative, volume weighted interpolation for the fluid quantities  $\rho, E, \vec{v}$ . For the Magnetic Fields, we use the method described by Balsara (Balsara 2001), with some slight modifications in implementation. This method constructs a quadratic divergence free polynomial, and area-weighted averages are used for the fine grid quantities. This is described in more detail in appendix B.

3. **Projection Operator** This is the routine that projects the fine grid data back to the parent coarse grid. For Enzo, the parent grid is simply replaced by a volume-weighted average of the fine cells. For the face centered Magnetic Field, this is an area weighted average, though in practice we don't explicitly average the magnetic field, as discussed in below and in appendix B.2
4. **Correction Operator** Once the projection operator replaces the solution on the coarse grids, the evolution on the coarse grids is no longer consistent with the underlying equations in the manner they were discretized. That is to say, the total change of any conserved quantity inside the region is no longer equal to the flux across its surface. For the Enzo hydro fields, this is corrected with the Flux Correction mechanism.

This needs a picture. Its coming.

EnzoMHD does all of these steps for the fluid quantities, but for the magnetic field it slightly alters this procedure. In order to overcome a shortcoming in the original data structures used in Enzo, we combined the projection and correction operations for the magnetic fields in one step. The net effect of the correction operator is to ensure that all zones are updated by finest resolution fluxes available, even if they were updated by coarse data initially. For the magnetic field update, we don't project the actual magnetic field that is of interest, but rather the electric field (effectively the 'flux' for  $B_f$ ), then take the curl of the newly projected electric field. Thus the coarse magnetic data co-located with the fine grids get updated with the fine data, and the bounding zones don't need correction at all.

More detail on this process can be found in appendix C



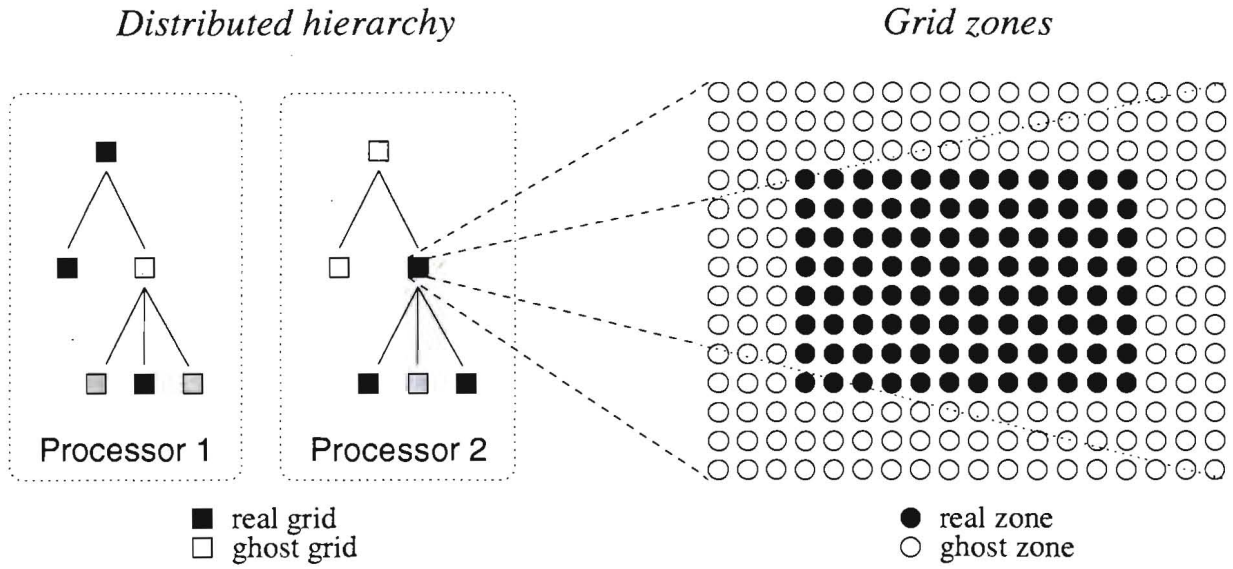


Fig. 1.— A schematic of an AMR hierarchy and a grid patch with ghost zones

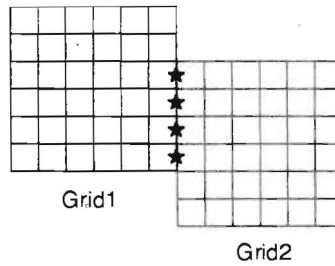


Fig. 2.— Data redundancy of the face centered magnetic fields: the face centered field denoted by the stars are updated by both grid 1 and grid 2. Enough ghost zones are exchanged to ensure that the entire stencil for the update of these fields is the same in both data structures.

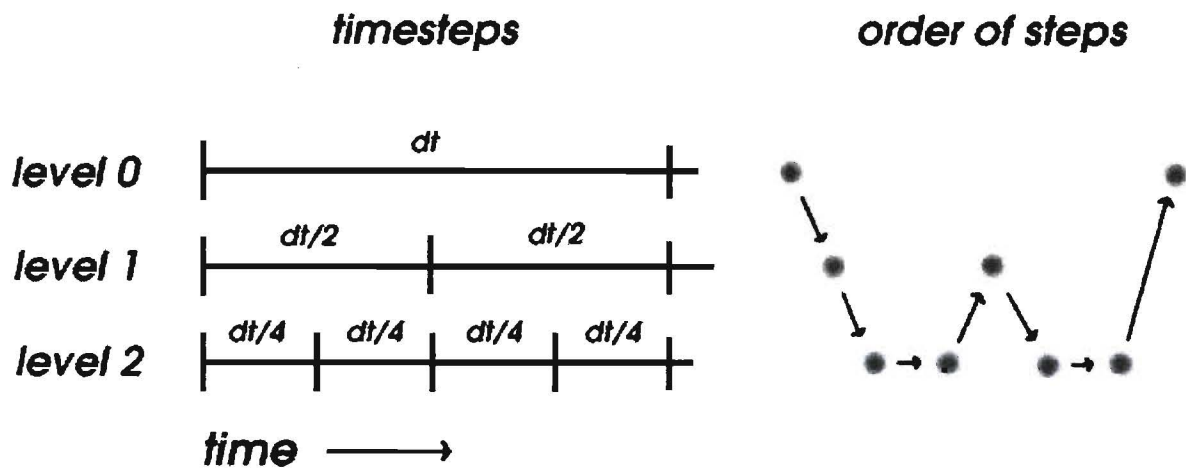


Fig. 3.— A depiction of the timestep strategy in Enzo

### 3. Numerical Experiments

EnzoMHD has many configurations available. Here, we test some of the possible configurations, to indicate the quality of solution possible with EnzoMHD.

#### 3.1. MHD Tests without AMR

We first test our code in unigrid (fixed resolution) mode, in order to ensure consistency of the patch solver with the algorithm described in Li et al. (2008). We do two one dimensional cosmology tests (Caustics and Zel’dovich Pancake), two one dimensional non-cosmological tests (Brio and Wu and the Kim Isothermal), one 2d non-cosmological test (Orszag Tang) and one 3d cosmological test.

##### 3.1.1. Brio and Wu shock tube

The shock tube defined by Brio & Wu (1988) is a standard test of any MHD solver, as it displays a number of the important MHD waves, including a compound wave. Compound waves are not a property of pure hydrodynamics, because the system is convex. However, due to the more complex nature of the MHD equations, certain initial conditions can cause flows in which at one point the shock speed in a given family is higher than the wave speed for that family, causing a shock, but lower in the post shock region, causing a rarefaction immediately following the shock.

This can be seen in figure 4. The problem was run with 800 zones to a time  $t = 0.2$ , using the HLLD solver in Enzo. This shock tube shows, from left to right, a fast rarefaction, slow compound (shock–rarefaction), contact, slow shock, and fast rarefaction. It can be seen that this solver captures this shock tube problem quite well.

##### 3.1.2. Isothermal Tests

One of the primary application areas of Enzo MHD will be in simulating turbulence and star formation in cold molecular clouds. Due to the fast cooling time of these environments, an isothermal equation of state is a good approximation a large portion of these processes. In simulations done by Kritsuk et al. (2007) using Enzo and other works by the same authors an isothermal equation of state is approximated by using an adiabatic solver and setting  $\gamma = 1.001$ .

To test if this approximation is appropriate for this code, we ran the isothermal shock tube of Kim et al. (1999). One can see from figure 5 that this approach works well, as shock jumps and positions are all correct, and features are reasonably sharp. This test was run with 256 zones to a time of 0.1.

However, in turbulent simulations with gravitational collapse, the measured value is initially

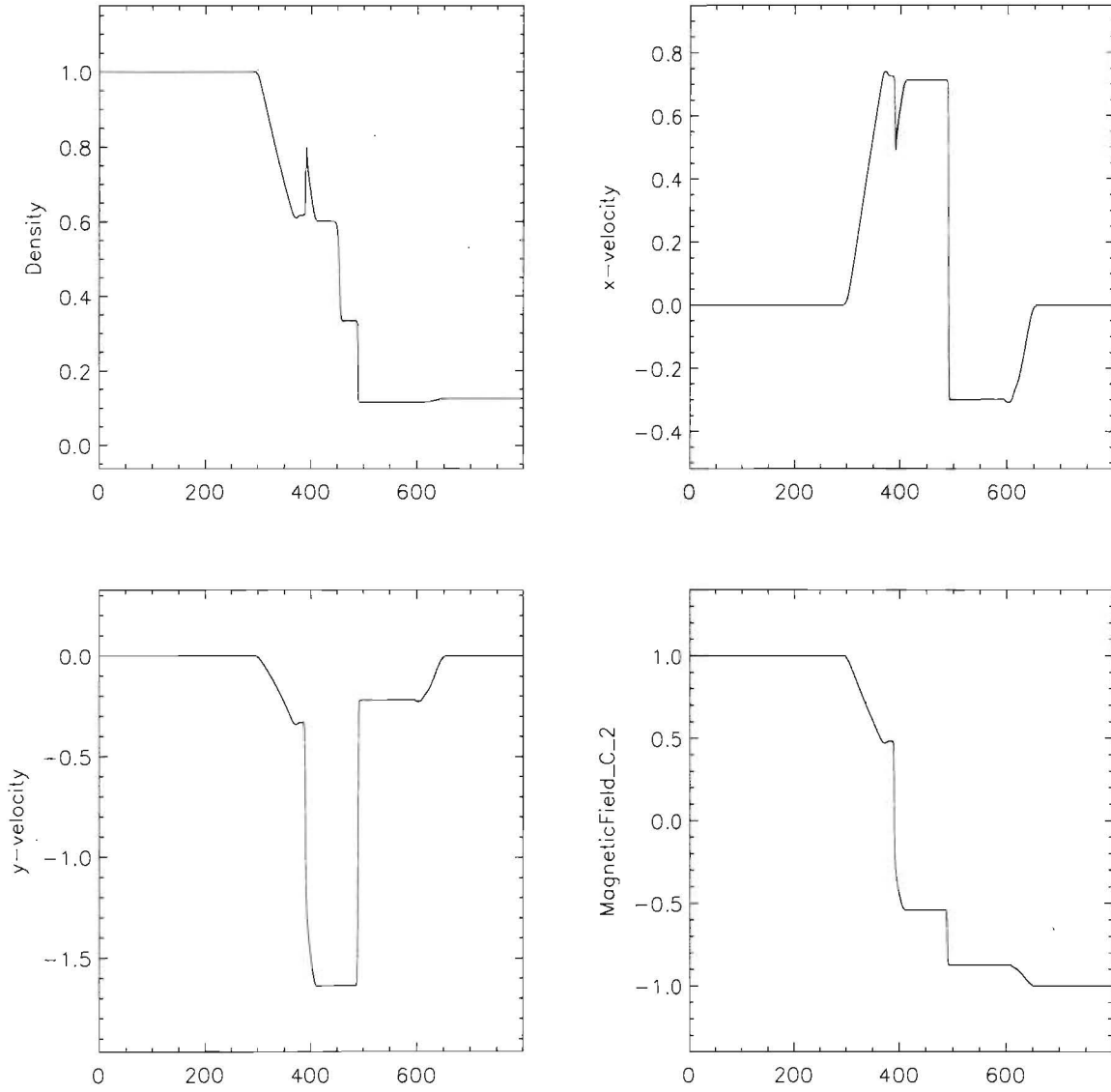


Fig. 4.— The shock tube of Brio & Wu (1988), showing from left to right a fast rarefaction, slow compound (shock + rarefaction), contact, slow shock, and fast rarefaction.  $T=0.08$ , and 800 zones were used.



uniform, but after a few hundred timesteps, spreads in the measured sound speed  $\sqrt{p/\rho}$  can be as large as 1000, which is far from isothermal. It is believed that the difference between this code and what has been done in the past with Enzo stems from the Riemann solver. The HLL family of Riemann solvers assumes a particular wave structure in computing the interface flux. This wave structure, for HLLC and HLLD, contains a contact discontinuity which is not present in the isothermal Riemann fan, and does not reduce appropriately in the  $\gamma \rightarrow 1$  limit. To combat this, we installed the Isothermal variant of HLLD by Mignone (2007). The results of this code on the Kim test are nearly identical to that in figure 5 and not reproduced here. The problem seen are, of course, eliminated as the sound speed is set as an input parameter.

### 3.1.3. One-dimension MHD Caustics

This test is taken from Li et al. (2008), which initially derived from a pure hydro version from Ryu et al. (1993). This problem is used to test the ability of the code to capture shocks and to deal with hypersonic flows. Initially,  $v_x = -\frac{\pi}{2}\sin(2\pi x)$ ,  $\rho = 1$  and  $p = 10^{-10}$ . Caustics are formed because of the compression by the velocity field. The Mach number of the initial peak velocity is  $1.2 \times 10^4$ . The pressure can easily become negative for such high Mach number flow.

We performed the test with same magnetic field settings as in Li et al. (2008). The magnetic field in the x and z directions are always zero while  $B_y = 0, 10^{-3}, 0.02$  and  $0.05$ . The calculation was done with 1024 cells and the results at  $t = 3$  are shown in figure 6. Our results match the results from CosmoMHD (Li et al. 2008) quite well, as expected.

### 3.1.4. The Zel'Dovich Pancake

The Zel'Dovich pancake is a popular test problem for codes that include gravity in comoving coordinates. The problem setups are taken from Li et al. (2008). This takes place in a purely baryonic universe with  $\Omega = 1$  and  $h = \frac{1}{2}$ . The initial scale factor  $a_i = 1$  corresponds to  $z_i = 20$ . The initial velocity field is sinusoidal with the peak value  $0.65/(1 + z_i)$ , and  $v = 0$  at the center of the box. The initial comoving box size is  $64h^{-1}Mpc$ . The shocks forms at  $z = 1$ . The initial baryonic density and pressure are uniform with  $\rho = 1$  and  $p = 6.2 \times 10^{-8}$ . The tests were run with 1024 cells, both with and without magnetic fields. When the magnetic fields are present, Our results are almost identical to the results from CosmoMHD (Li et al. 2008), as expected. Results can be seen in figure 7.

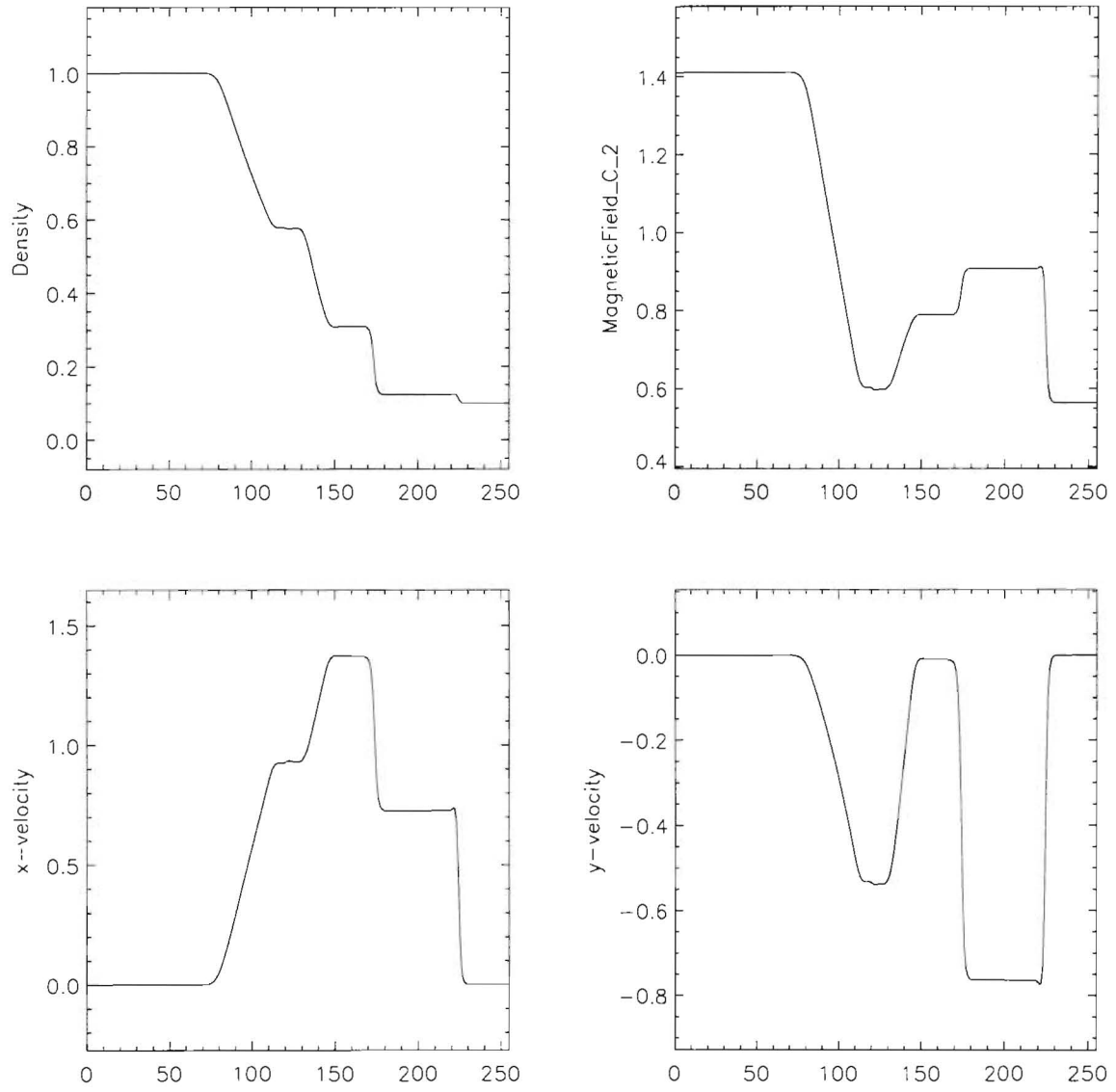


Fig. 5.— The shock tube of Kim et al. (1999).

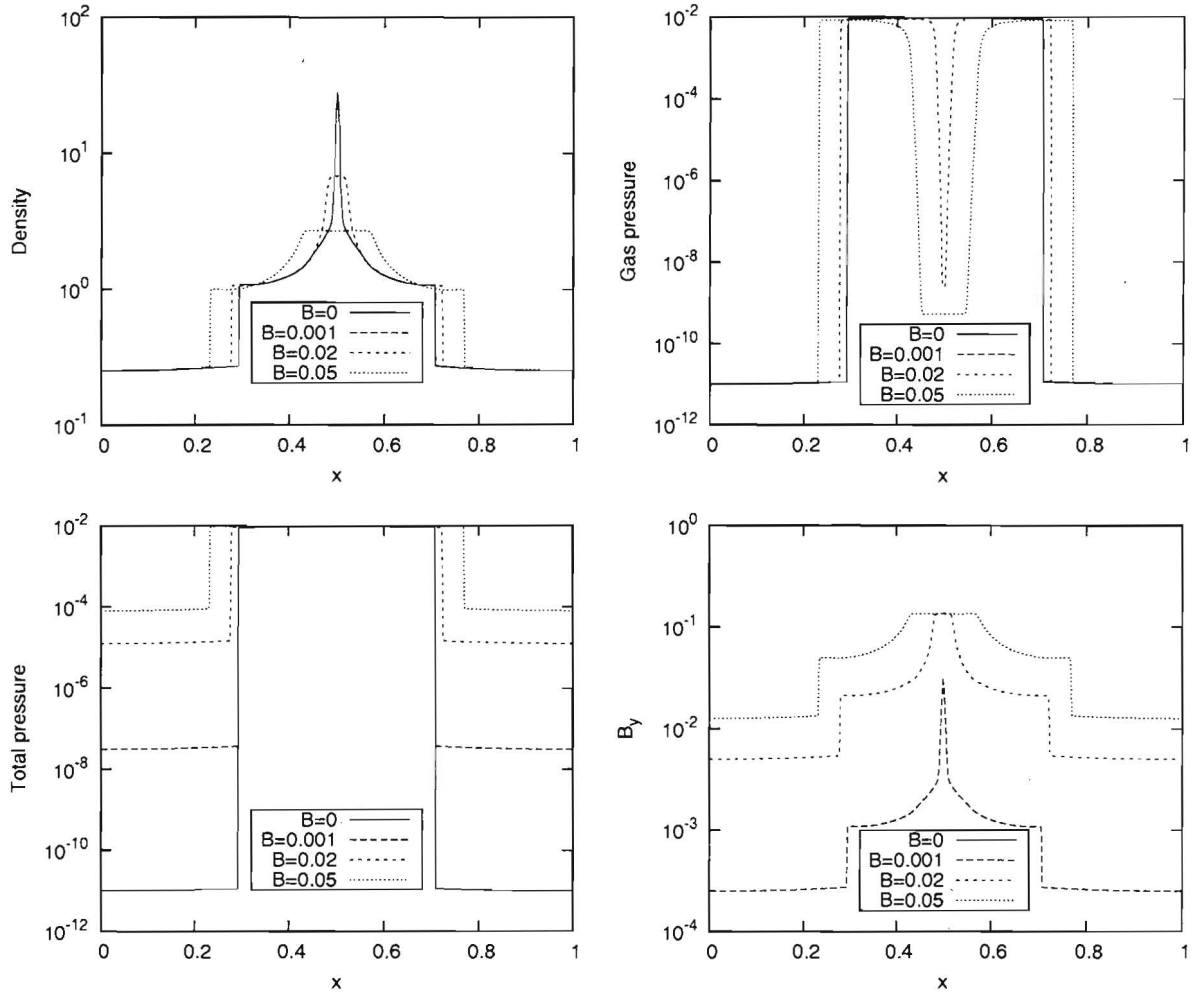


Fig. 6.— 1-D MHD caustics at  $t = 3$ . Density, gas pressure, total pressure and  $B_y$  are plotted. For the small field runs, almost no change can be seen, while larger field runs decrease the peak of the density considerably due to the increased pressure.

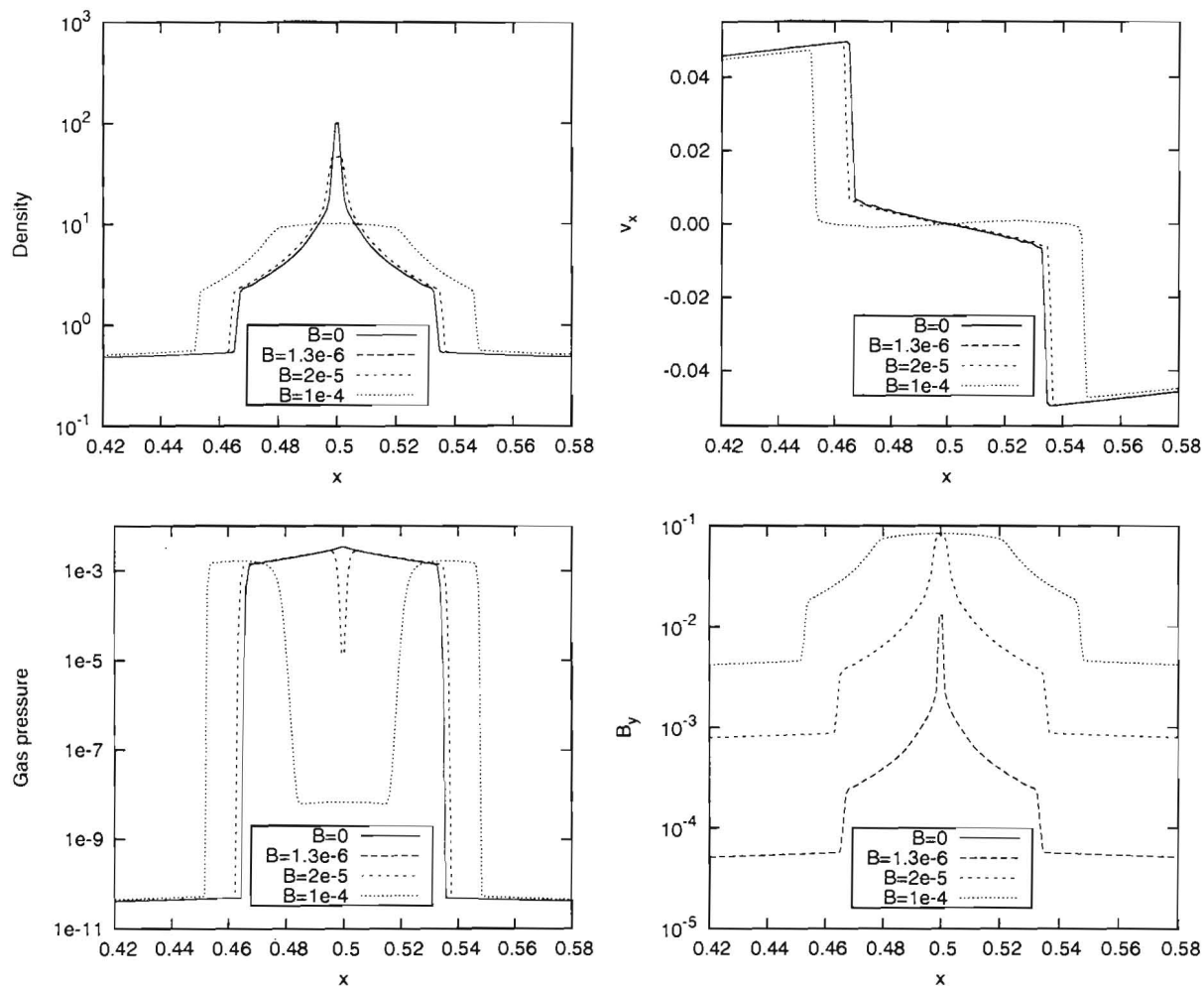


Fig. 7.— The Zel'Dovich Pancake problem with various values of the magnetic field, at  $t = 0$ . Increasing the magnetic field strength increases the central magnetic pressure, reducing the density and changing the overall solution structure. Results match those of Li et al. (2008).

### 3.1.5. Orszag-Tang

The Orszag-Tang Vortex was originally developed by Orszag & Tang (1979) to demonstrate that small scale structure can be generated by the nonlinearities in the MHD equations. It now serves as a standard test problem to demonstrate the accuracy and diffusivity of MHD codes.

The initial conditions are on a 2 dimensional periodic box, 256 zones on a side.  $\mathbf{v} = v_0(-\sin(2\pi y)\hat{x} + \sin(2\pi x)\hat{y})$ ,  $\mathbf{B} = B_0(-\sin(2\pi y)\hat{x} + \sin(4\pi x)\hat{y})$ ,  $v_0 = 1$ ,  $B_0 = 1/\sqrt{4\pi}$ ,  $\rho_0 = 25/(36\pi)$ ,  $p_0 = 5/(12\pi)$ , and  $\gamma = 5/3$  which gives a peak mach number of 1 and peak plasma beta  $\beta = p_0/(B_0^2/2) = 10/3$ . Figure 8 shows the density at  $t = 0.48$ , from which one can see that the solution agrees with other solutions to the problem in the literature.

### 3.1.6. 3D Adiabatic Universe with MHD

We have also performed the 3D adiabatic CDM Universe test described by Li et al. (2008) both with and without magnetic fields. We also compared the non-magnetized results with the results run using the PPM solver (Colella & Woodward 1984). Adiabatic evolution of a purely baryonic Universe was computed with an initial CDM power spectrum with the following parameters:  $\Omega = \Omega_b = 1$ ,  $h = 0.5$ ,  $n = 1$  and  $\sigma_8 = 1$  in a computational volume with side length  $L = 64h^{-1}Mpc$ . The transfer function from Bardeen et al. (1986) was used to calculate the power spectrum of the initial density fluctuations. Evolution was done from  $z = 30$  to  $z = 0$ . We used  $256^3$  cells for each simulation. The comparisons are made at the final epoch,  $z = 0$ . Though this test is identical to that of Li et al. (2008), our results can't be compared with theirs directly since different random seeds were used for the realization of the initial density and velocity.

Figure 9 shows a comparison of the mass-weighted temperature distribution, figure 10 is a comparison of the volume-weighted density distribution, and figure 11 shows a comparison of the baryonic density of a slice of  $32h^{-1}Mpc \times 32h^{-1}Mpc$ . The contour lines are  $0.1 \times 10^k$ , where  $k = 0, 0.1, 0.2, \dots, 3$ . The discrepancies between PPM and MHD solvers are small, indicating the two codes perform roughly the same. The nature of the differences is expected, since PPM solver has third order accuracy while the MHD solver has second order accuracy and larger numerical diffusion. This allows PPM to capture shocks in fewer zones, which causes the dense shocked gas to not only have a smaller volume fraction, but also be hotter and slightly less dense than in the MHD solver.

We have done a run with an initial magnetic field,  $B_x = B_z = 0$ ,  $B_y = 2.5 \times 10^{-9}$  Gauss, which is  $4.3176 \times 10^{-7}$  in code units. Figure 12 shows the density and magnetic energy density contours of a slice. The density contour levels are  $0.1 \times 10^k$ , where  $k = 0, 0.2, 0.4, \dots, 6$ , and the magnetic energy density contours are  $10^{-14} \times 10^k$ , where  $k = 0, 0.2, 0.4, \dots, 6$  in code units. As expected, they are well correlated. Figure 13 shows the scaled divergence of the magnetic fields, averaged over the entire box, as a function of redshift. The divergence of the magnetic fields is close to the round-off





Fig. 8.— Density from the Orszag-Tang vortex, at  $t=0.48$ .

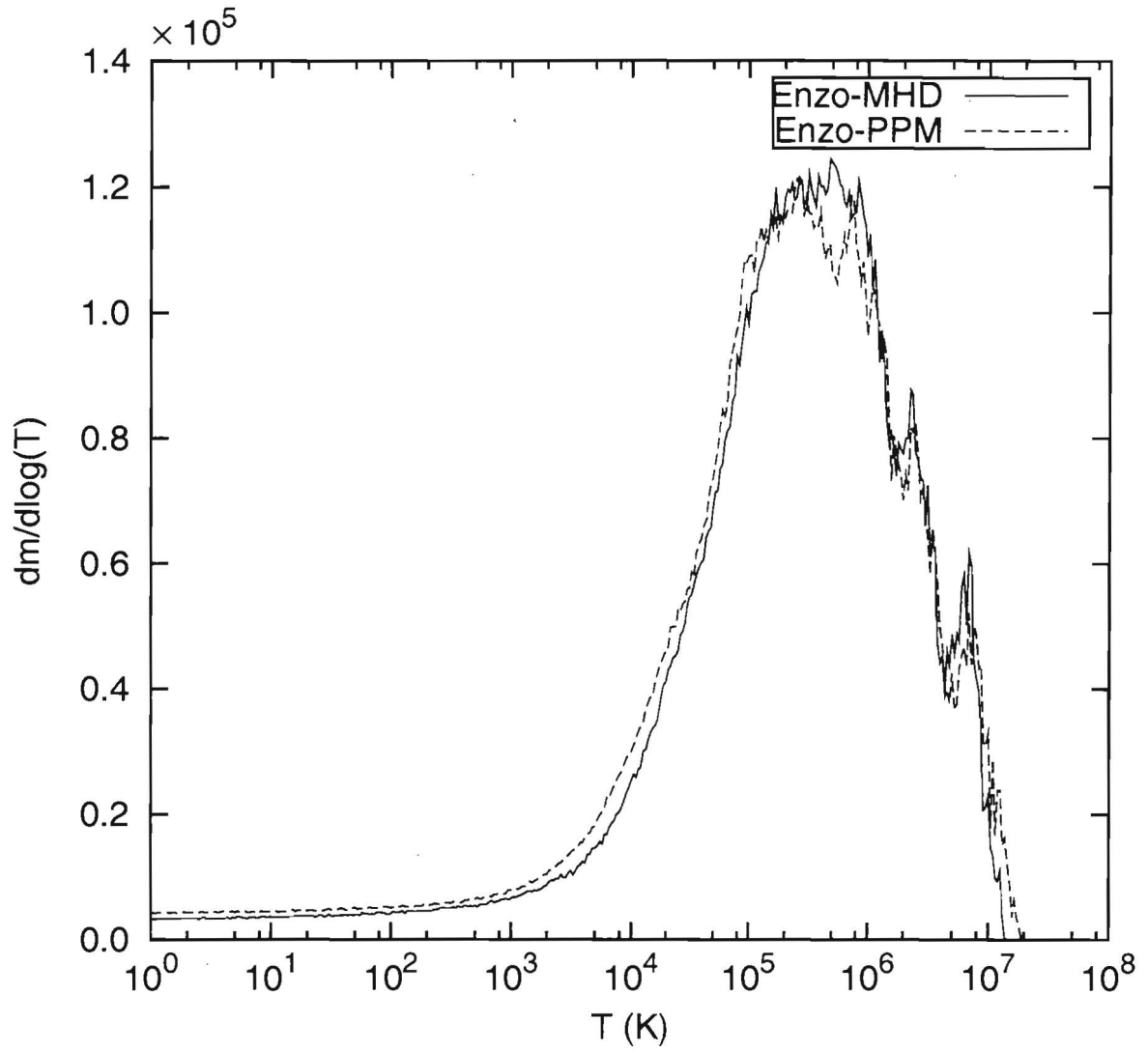


Fig. 9.— Comparison of mass-weighted temperature histogram at  $z = 0$  for the 3D purely baryonic adiabatic Universe simulation. The solid line is from the MHD code and the dashed line is from Enzo-PPM.

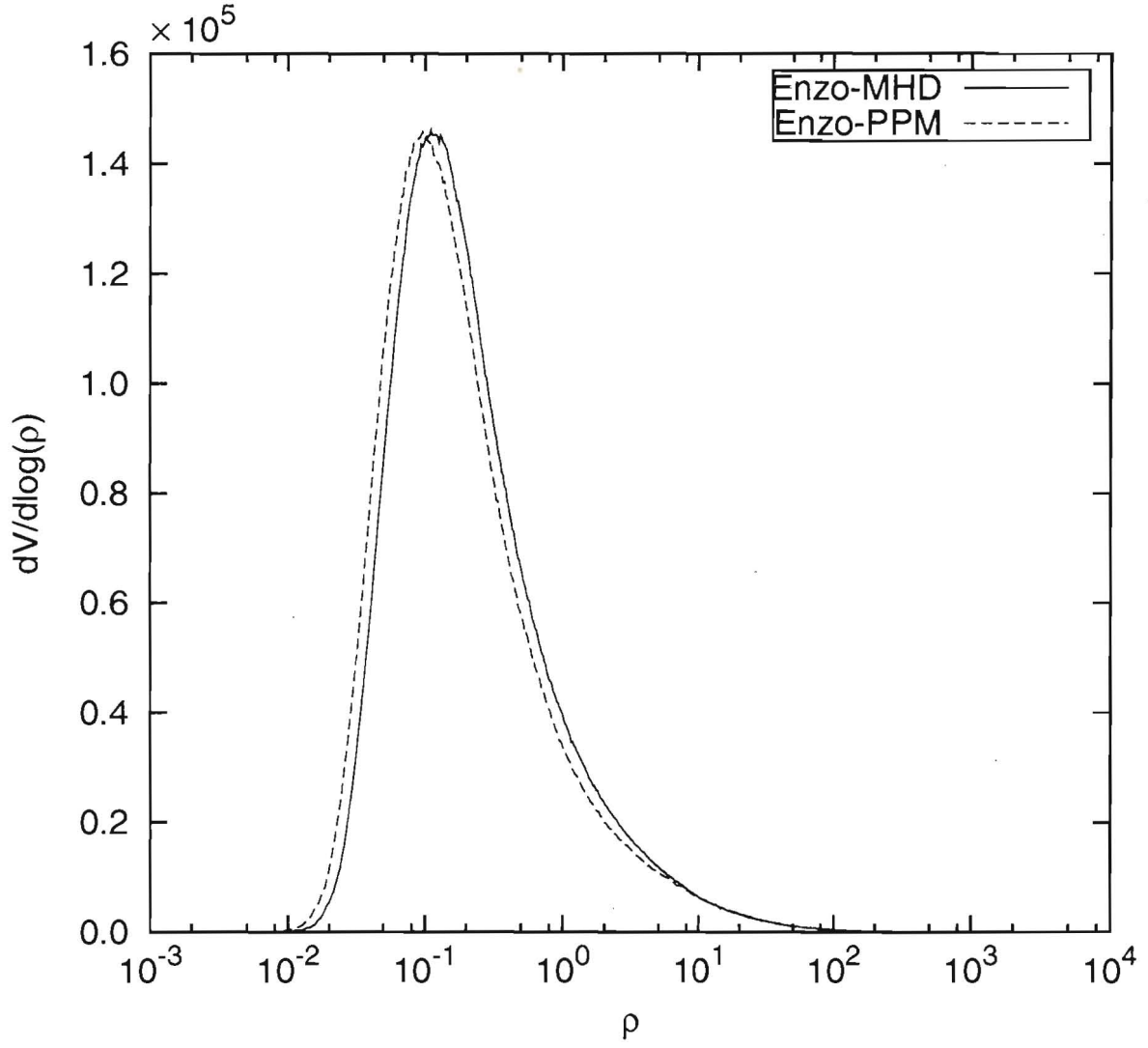


Fig. 10.— Comparison of volume-weighted density histogram at  $z = 0$  for the 3D purely baryonic adiabatic Universe simulation. The solid line is from the MHD code and the dashed line is from Enzo-PPM.



error.

### 3.2. MHD Tests with AMR

#### 3.2.1. Three-dimension MHD Adiabatic Expansion

This test is taken from Bryan et al. (1995). This test uses a completely homogenous universe with initial  $T_i = 200K$  and  $v_i = 100km/s$  in the x-direction at an initial redshift of  $z_i = 20$ . In the code units, the initial density is 1.0 and initial velocity is 0.0027833915 and the initial pressure is  $1.2367812 \times 10^{-9}$ . Additionally we have a uniform magnetic field  $B_x = B_y = B_z = 0.0001$  in code units, which is  $2.65710 \times 10^{-7}G$  in cgs.

The simulation used a  $16^3$  root grids with 2 levels of refinement in the center region and ran to  $z = 0$ .

The expansion terms in eqns (1) - (4) operate like drag terms, so that in the absence of a source, the velocity decreases as  $v = v_i a^{-1}$ , the temperature as  $T = T_i a^{-2}$  and the magnetic field should decrease as  $a^{-1/2}$ .

The temperature at  $z = 0$  is 0.453406K, 0.024% below the analytic result of 0.453515K. The velocity at  $z = 0$  is 4.76176km/s, compared to the analytic result 4.7619km/s, a 0.0029% discrepancy. The final magnetic field strength is  $6.02513 \times 10^{-10}G$  ( $2.1821651 \times 10^{-5}$  in the code units) compared to the analytic result  $6.02517e-10G$ , a 0.0006% difference. Figure 14 shows the  $B_y$  as a function of redshift, the solid line shows the theoretical value.

#### 3.2.2. One-dimensional MHD Caustics with AMR

We also ran the the 1d MHD Caustic test with AMR, using 256 root grid zones with 2 levels of refinement, again by a factor of 2, giving an effective resolution is 1024 cells. Figure 15 shows comparisons of density and gas pressure of non-AMR and AMR runs with different initial magnetic field strengths, as described before. Figure 16 shows the comparisons of  $B_y$  for runs with different initial values of  $B_y$ . In both plots, the AMR result is sampled to the finest resolution. The AMR runs give almost identical results to the unigrid runs, while the CPU time and memory were greatly saved in the AMR runs.

#### 3.2.3. Zel'Dovich Pancake with AMR

We also ran the pancake problem with AMR. The problem was set up with the same initial conditions as the unigrid run, but with a root grid of 256 root cells and 2 levels of refinement by 2. We compared these results having effectively 1024 cells to the results of our previous high resolution

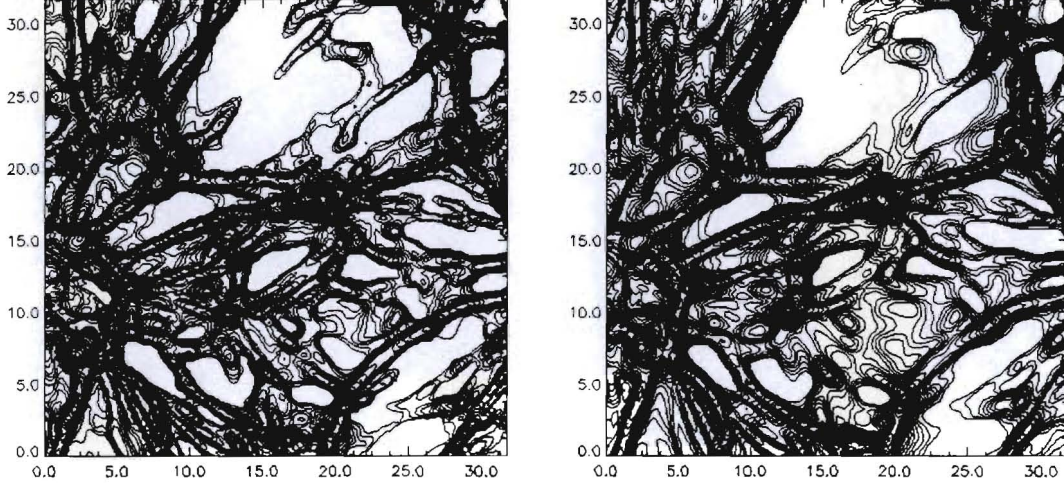


Fig. 11.— Comparison of density contours of a slice of  $32h^{-1}Mpc \times 32h^{-1}Mpc$  at  $z = 0$ . The left panel shows the result from the PPM solver, while the right panel shows the results from the MHD solver. The contour lines are  $0.1 \times 10^k$  and  $k = 0, 0.1, 0.2, \dots, 3$  in code units.

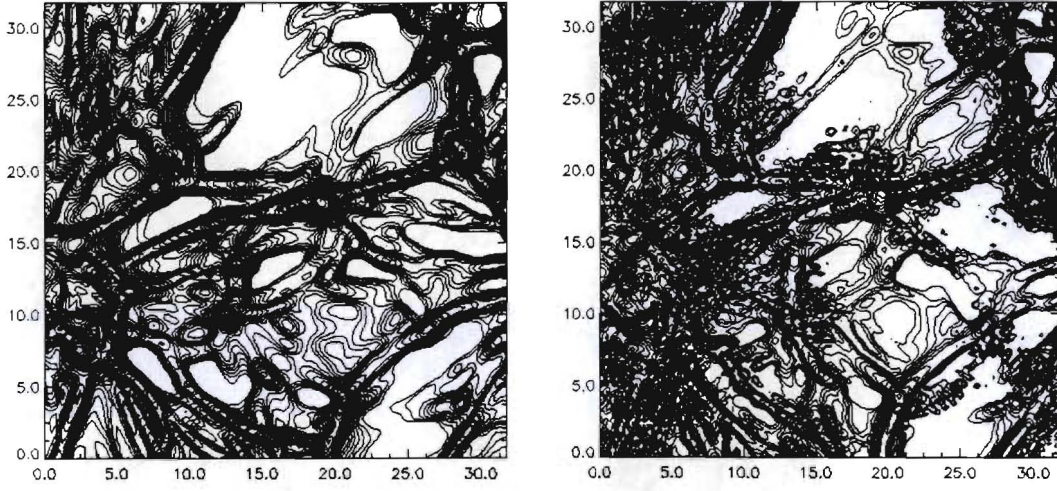


Fig. 12.— Density and magnetic energy density contours of a slice of  $32h^{-1}Mpc \times 32h^{-1}Mpc$  at  $z = 0$ . The left panel shows density, which the contour lines are shown with levels  $0.1 \times 10^k$  and  $k = 0, 0.2, 0.4, \dots, 6$  in code units. The right panel shows magnetic energy density, which the contour lines are shown with levels  $10^{-14} \times 10^k$  and  $k = 0, 0.2, 0.4, \dots, 6$  in code units.

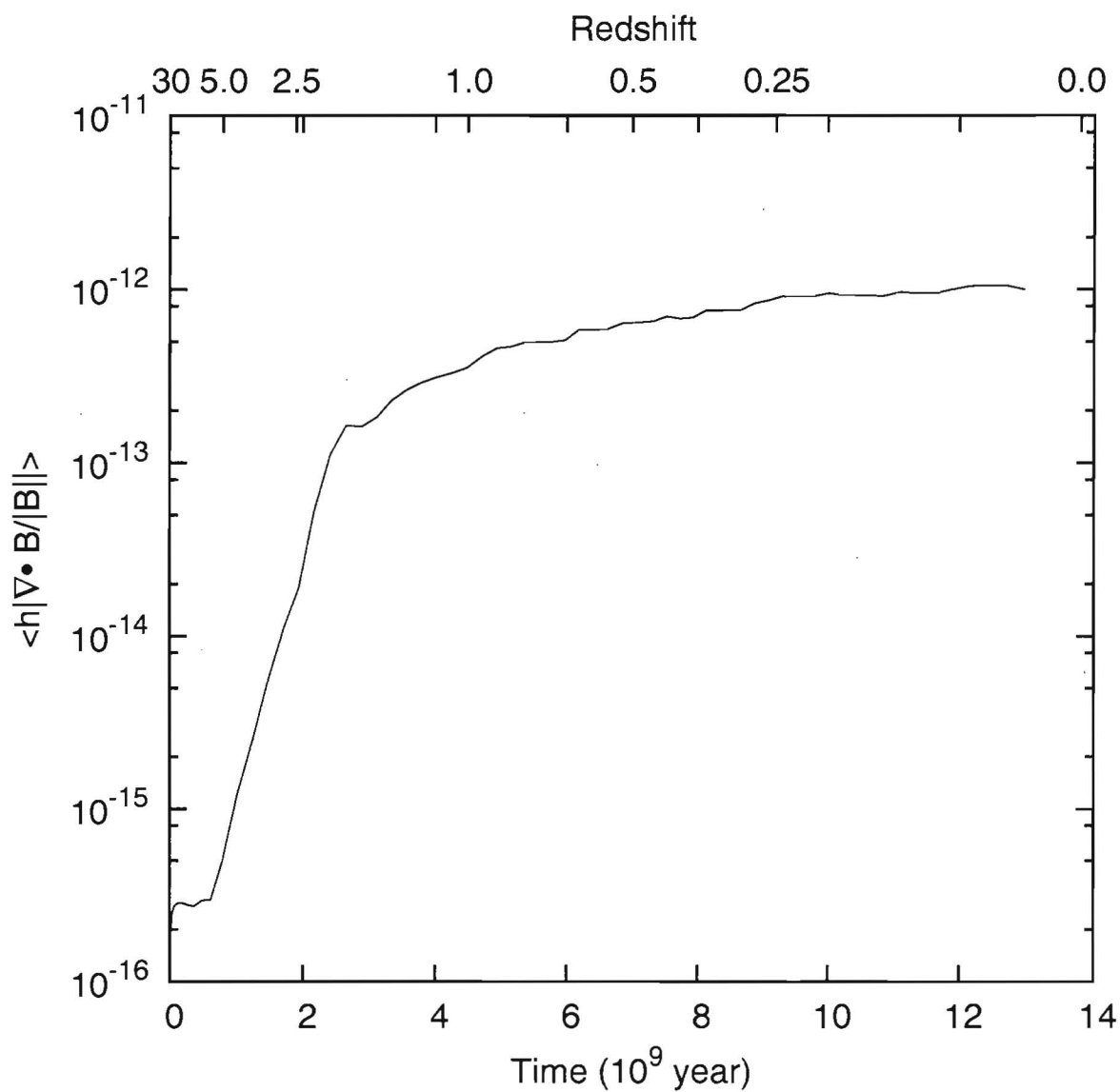


Fig. 13.— The scaled divergence  $\langle |h \nabla \cdot \mathbf{B} / |\mathbf{B}| | \rangle$  of the magnetic fields for the 3-D simulations of a purely baryonic adiabatic Universe

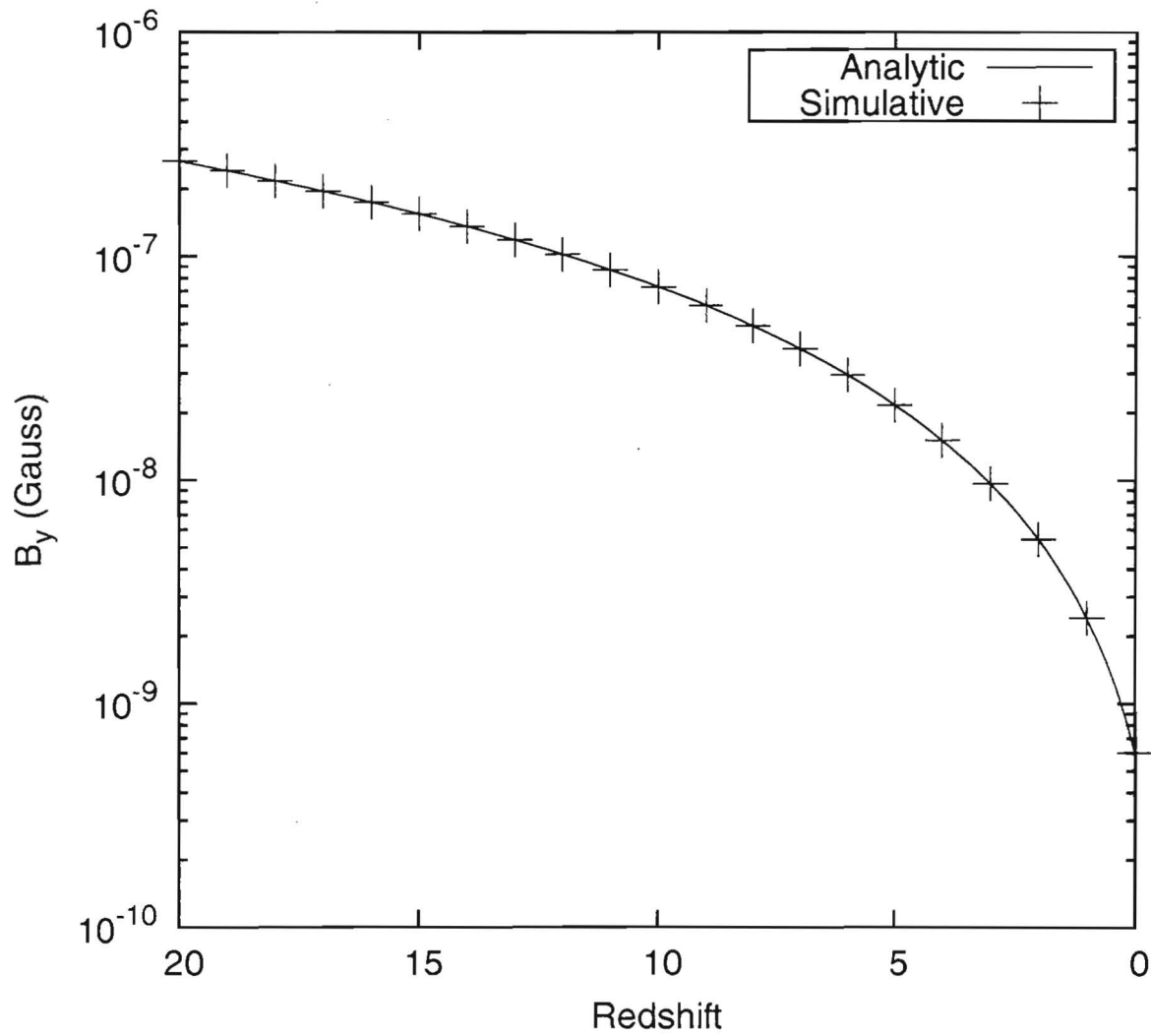


Fig. 14.— Magnetic field in the y direction in the AMR MHD adiabatic expansion test. The pluses show the results of simulation and the solid line is the analytic result.

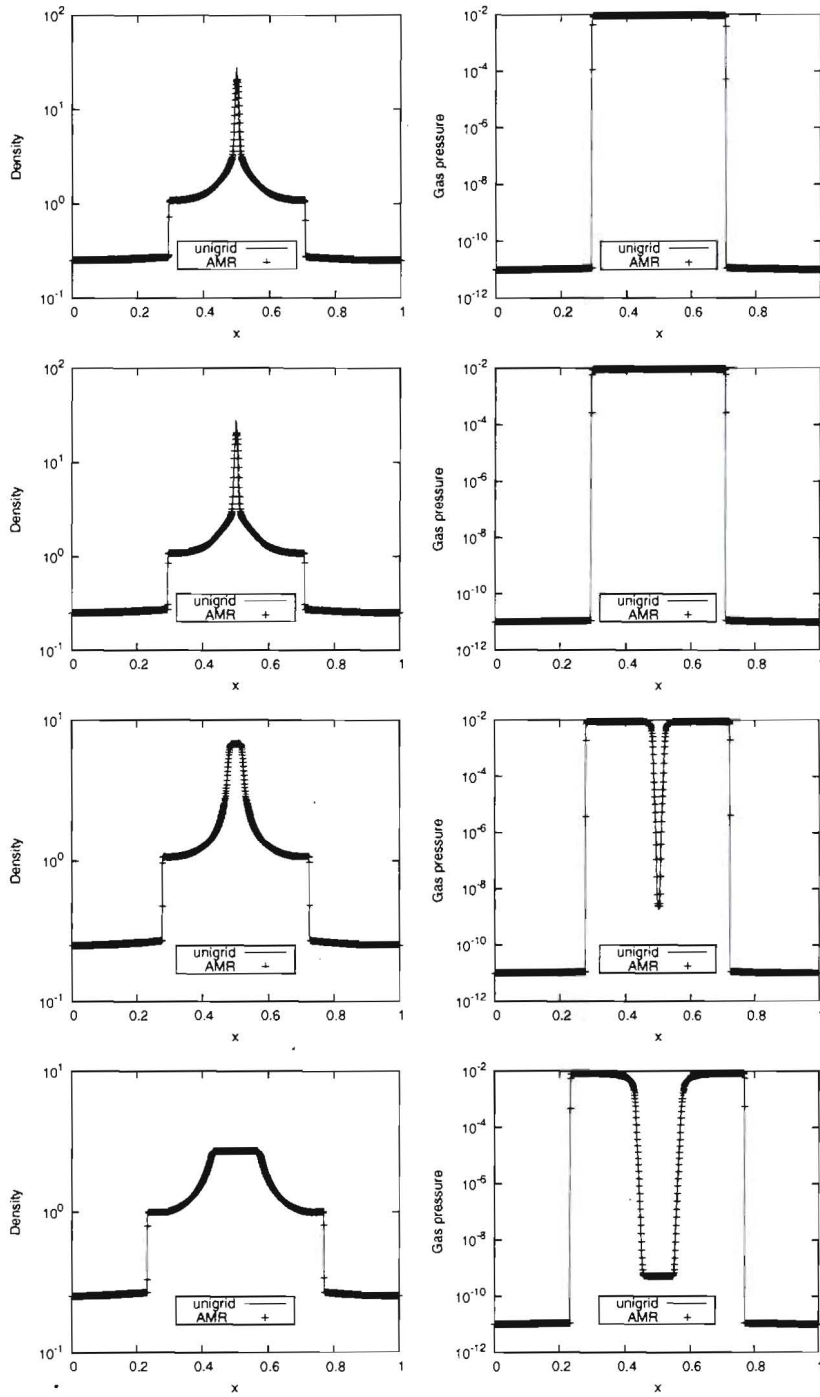


Fig. 15.— Comparisons of density and pressure in the MHD Caustic tests, non-AMR vs AMR. The left column shows density and the right column shows gas pressure. Initial magnetic field of each row from top to bottom is 0, 1e-3, 0.02 and 0.05.



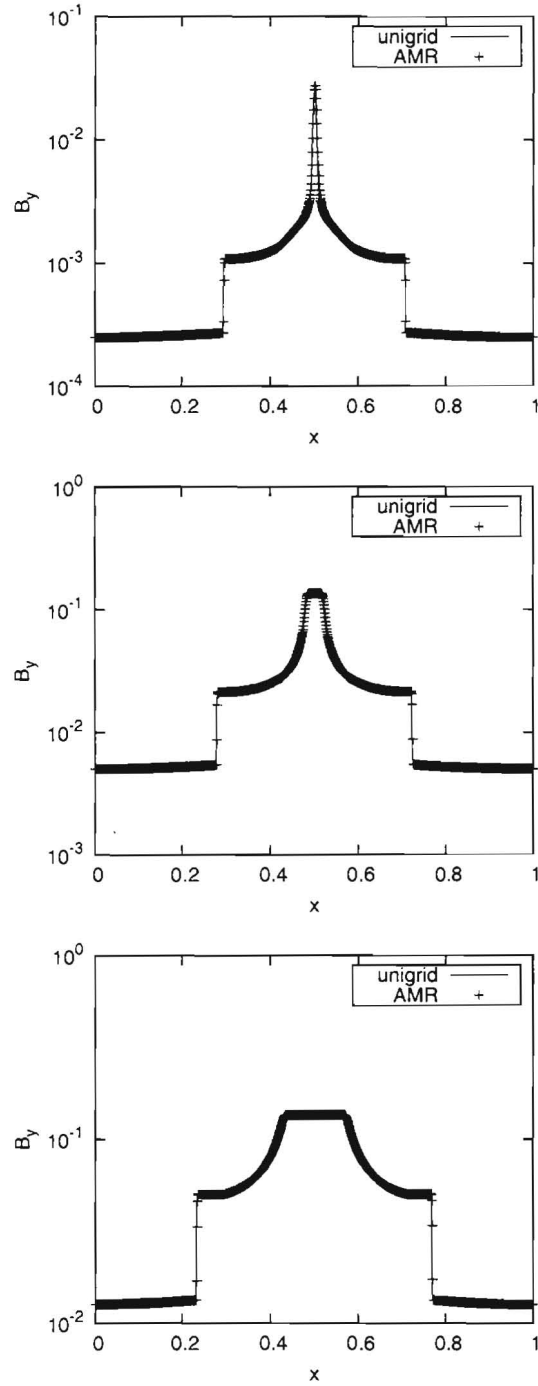


Fig. 16.— Comparison of  $B_y$  in the MHD Caustic tests, non-AMR vs AMR. Initial magnetic field of each panel from left to right is  $1e-3$ ,  $0.02$  and  $0.05$ .

which actually had 1024 cells. Figure 17 shows comparisons of density and gas pressure between the non-AMR and AMR runs, with different initial values for  $B_y$ . Figure 18 shows the comparisons of  $B_y$  with different initial values. Again, the AMR computation got very similar results, while saving CPU and memory resources.

### 3.2.4. MHD Galaxy Cluster Formations

Cluster formation (without MHD) has been studied intensively by researchers using Enzo ((Norman 2005), (Bryan & Norman 1998), (Loken et al. 2002), (Motl et al. 2004), (Hallman et al. 2006)) It is one of the most important applications of Enzo’s high dynamic range. Many cluster simulations have been run with Enzo with a wide variety of physics (i.e. radiative cooling, star formation, etc) and we can compare these results to similar simulations run with MHD. More information about Enzo simulated cluster can be found in Simulated Cluster Archive at <http://lca.ucsd.edu/data/sca/>. Here, we present just one simulation to demonstrate the MHD code.

This simulation uses a Lambda CDM cosmology model with parameters  $h = 0.7$ ,  $\Omega_m = 0.3$ ,  $\Omega_b = 0.026$ ,  $\Omega_\Lambda = 0.7$ ,  $\sigma_8 = 0.928$ . The survey volume is  $256 h^{-1}$  Mpc on a side. The simulations were computed from a  $128^3$  root grid with 2 level nested static grids in the center where the cluster form. This gives an effective resolution of  $512^3$  cells ( $0.5 h^{-1}$  Mpc per cell) and dark matter particle mass mass resolution of  $1.49 \times 10^{10}$  solar masses initially in the central region. Adaptive mesh refinement is allowed only in the region where the galaxy cluster forms, with a total of 8 levels of refinement beyond the root grid, for a maximum spatial resolution of  $7.8125 h^{-1}$  Kpc. While the baryons are resolved at higher and higher spatial and mass resolution at higher levels, the dark matter particles maintain constant mass so as not to add any additional noise. The simulations are evolved from  $z = 30$  to  $z = 0$ , and all results are shown at the redshift  $z = 0$ . We concentrate our study on a cluster of  $M = 1.2 \times 10^{15} M_\odot$ .

In order to isolate the effects of the numerical approximation from the effects of MHD, we first run the simulations adiabatically without additional physics and the magnetic field set to zero, and compare to a PPM run with identical parameters. In table 1, we list the basic parameters for the clusters formed in each solver. The viral radius,  $R_{vir}$  is calculated for an over density  $\frac{\delta\rho}{\rho}$  of 200.  $M_{vir}$ ,  $M_{dm}$  and  $M_{gas}$  are the total mass, mass of the dark matter and mass of the baryon inside the virial radius, respectively.  $T_{vir}$  is the average of the temperature of the ICM inside the virial radius. Evidently, there is very little difference between the results from the two solvers.

Figures 19-21 show the images of the logarithmic projections of the dark matter density, gas density, and X-ray weighted temperature, respectively, at  $z = 0$ . Both PPM and MHD solvers show very similar images in all three quantities, differing only slightly in the small scale details.

Figure 22-24 show the radial profiles of dark matter density, gas density, and x-ray weighted temperature. The profiles match quite well in all three quantities, with only minor differences. There is a slight deviation in the radial profiles of dark matter density near the center of the cluster,

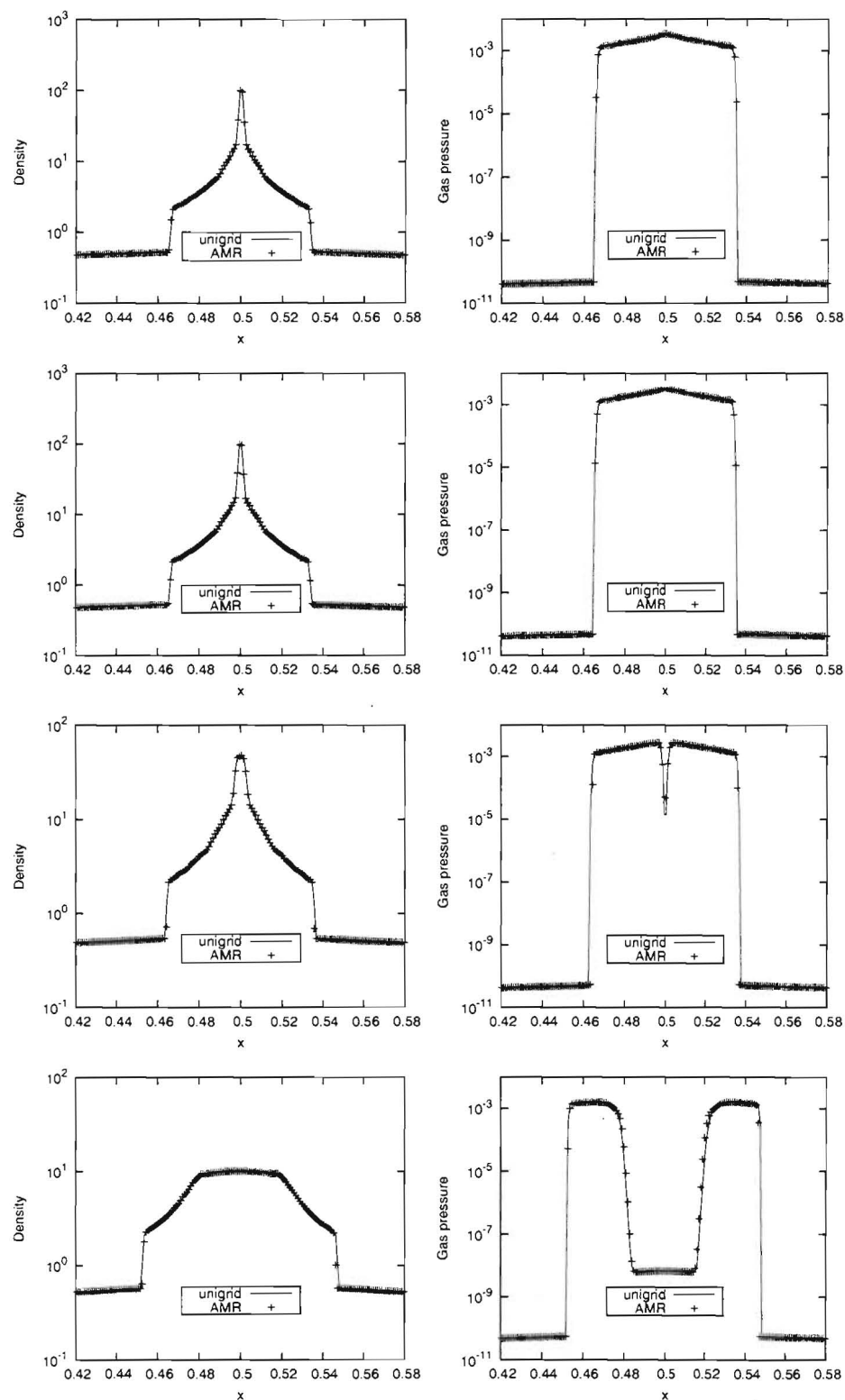


Fig. 17.— Comparisons of density and pressure in non-AMR and AMR runs of the Pancake test. The left column shows density and the right column shows gas pressure. Initial magnetic field of each row from top to bottom is 0,  $1.3 \times 10^{-6}$  G,  $2 \times 10^{-5}$  G and  $1 \times 10^{-4}$  G.



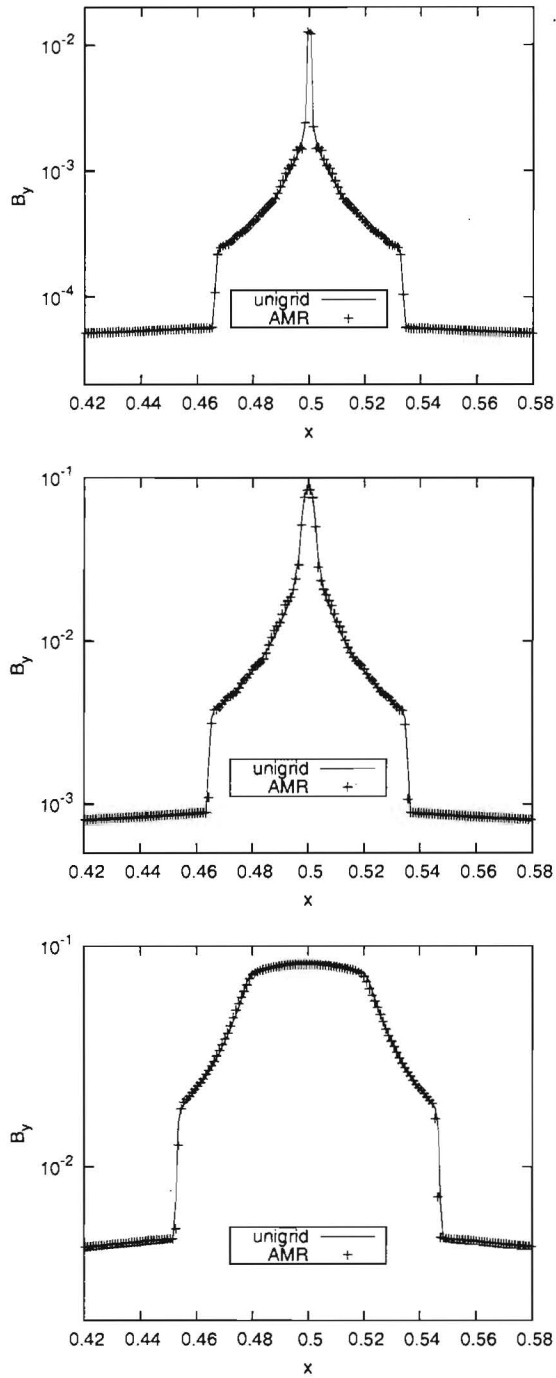


Fig. 18.— Comparisons of magnetic y component in non-AMR and AMR runs of the Pancake test. Initial magnetic field of each panel from left to right is  $1.3\text{e-}6$ ,  $2\text{e-}5$  and  $1\text{e-}4\text{G}$ .

but this is near the resolution limit of the simulation, so not a trustworthy data point. In the density profile, it can be seen that the MHD solver gives a slightly higher average density. The temperature agreement is good enough to not worry about.

We have also run the simulations with non-zero initial magnetic field. A uniform initial magnetic field of  $9.72753 \times 10^{-10} G$  ( $1 \times 10^{-7}$  in code units) in the y direction was added to the system at the start of simulation at  $z = 30$ . Since Dolag et al. (1999) has shown that the initial magnetic fields structures are not important to the final magnetic fields structures in their MHD SPH simulations, no other initial magnetic fields configuration will be used in this paper. Figure 25 shows 4 projections of the cluster center: gas density, temperature, magnetic energy, and synthetic Faraday rotation measurement  $RM = \frac{e^3}{2\pi m^2 c^4} \int_0^d n_e B ds$ . We can see that the gas density and temperature images are almost identical to the MHD run with zero magnetic fields. As expected, the magnetic energy is concentrated in the cluster core. The maximum magnetic fields is  $1.0630270 \times 10^{-8} G$ . The RM is about  $2\text{--}3 \text{ radm}^{-2}$  at the cluster core. Figure 26 shows comparison of the radial profiles of the simulations with and without initial magnetic fields, while figure 27 depicts the volume weighted averaged radial profiles of the magnetic field strength and plasma  $\beta$ . Since  $\beta$  is quite large, these small magnetic fields acts as a passive tracer of the plasma and has little effects on dark matter and gas dynamics.

To further test our code, we also ran a simulation with a relatively large initial magnetic fields. We also included radiative cooling, star formation, and stellar feedback. The radiative cooling models X-ray line and bremsstrahlung emission in a 0.3 solar metallicity plasma. The star formation model turns cold gas into collisionless star particles at a rate  $\rho \dot{S}_F = \eta_{SF} \frac{\rho_b}{\max(\tau_{cool}, \tau_{dyn})}$ , where  $\eta_{SF}$  is the star formation efficiency factor 0.1, and  $\tau_{cool}$  and  $\tau_{dyn}$  are the local cooling time and free fall time, respectively. Stellar feedback returns a fraction of stars' rest energy as thermal energy at a rate  $\Gamma_{SF} = \eta_{SN} \dot{S}_F c^2$  to the gas. We did two runs, one without initial magnetic fields and the other is with a large initial magnetic fields of  $B_y = 1.0 \times 10^{-4}$  in code units ( $9.72753 \times 10^{-7} \text{ Gauss}$ .) Figure 28 shows the radial profiles of gas density and temperature of both runs and the magnetic field strength and the plasma  $\beta$  of the run with magnetic fields.

The magnetic fields reached  $20 \mu G$  in the core region, a few times larger than the observations (Carilli & Taylor 2002). In the center where  $\beta$  reaches a minimum, the kinetic energy is a few

Table 1: Cluster Properties

Parameter	Hydro PPM	MHD
$R_{vir}(Mpc)$	2.22946	2.22674
$M_{vir}(M_s)$	1.26462e-15	1.25999e-15
$M_{dm}(M_s)$	1.09746e-15	1.09683e-15
$M_{gas}(M_s)$	1.67158e-14	1.6316e-14
$T_{vir}(K)$	8.68422e-07	8.66301e-07

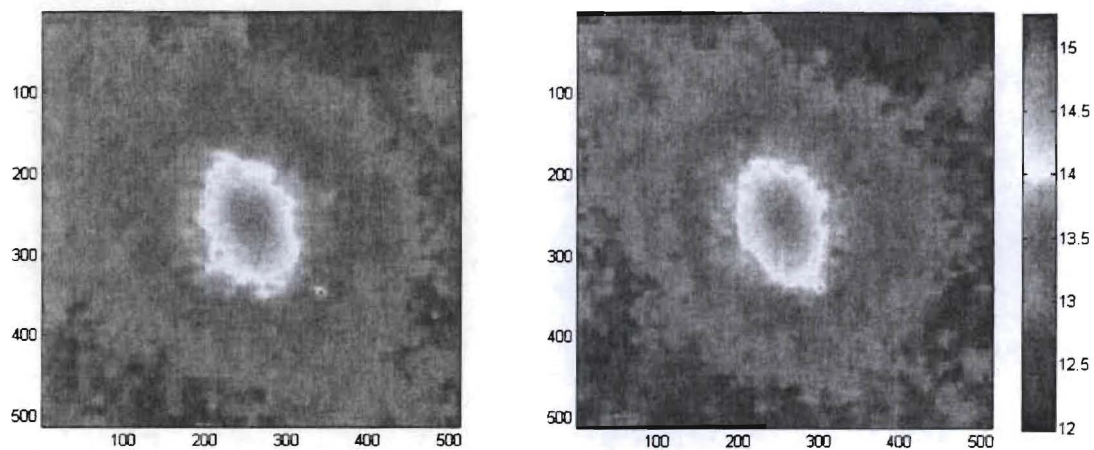


Fig. 19.— Logarithmic projected dark matter density at  $z = 0$ . The images cover the inner 4 Mpc/h of cluster centers. The left panel shows the result from the PPM solver and the right panel shows the result from the MHD solver. The color bar is in  $M_{\odot} \text{Mpc}^{-3}$ .

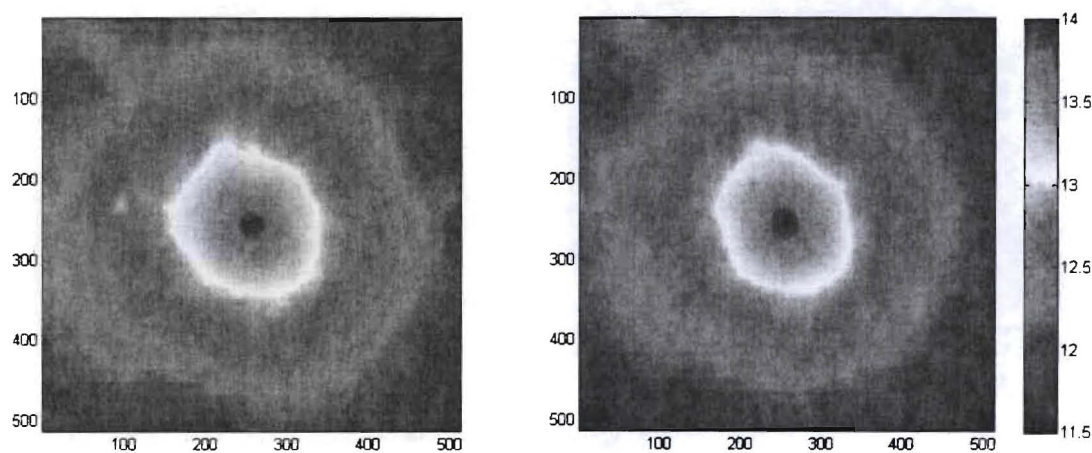


Fig. 20.— Logarithmic projected gas density at  $z = 0$ . The images cover the inner 4 Mpc/h of cluster centers. The left panel shows result from PPM solver and the right panel shows result from MHD solver. The color bar is in  $M_{\odot} \text{Mpc}^{-3}$ .

percent of the thermal energy, as expected from (Iapichino & Niemeyer 2008). The magnetic field has become dynamically important in the cluster center. The effect is not significant in the density, as seen in the upper right plot in figure 28, but definitely noticable in the temperature field, as some of the thermal pressure that was balancing the collapse is replaced by magnetic pressure. In this way, magnetic fields may help to cool cluster cores, giving a better match to observations. Detailed analysis of the magnetic field structure and their influence on the cluster will be presented in forthcoming paper.

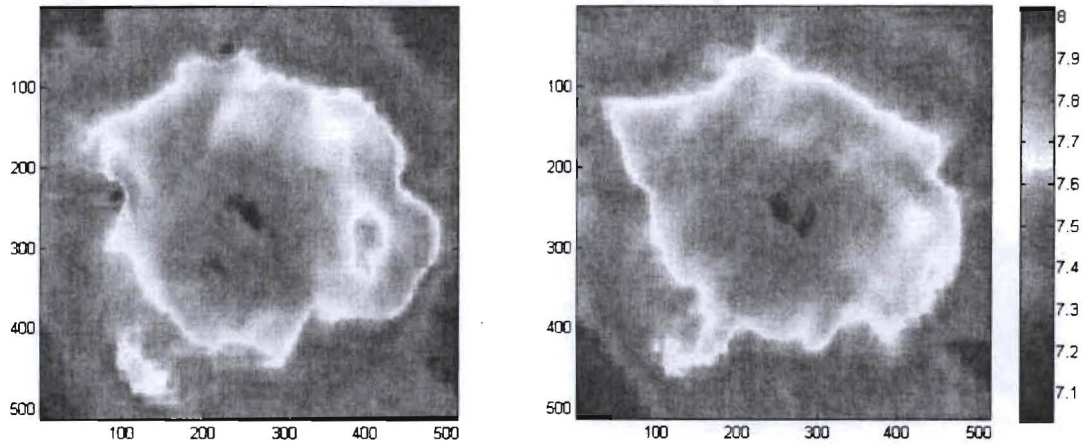


Fig. 21.— Logarithmic projected X-ray weighted temperature at  $z = 0$ . The images cover the inner 4 Mpc/h of cluster centers. The left panel shows result from PPM solver and the right panel shows result from MHD solver. The unit is Kelvin.



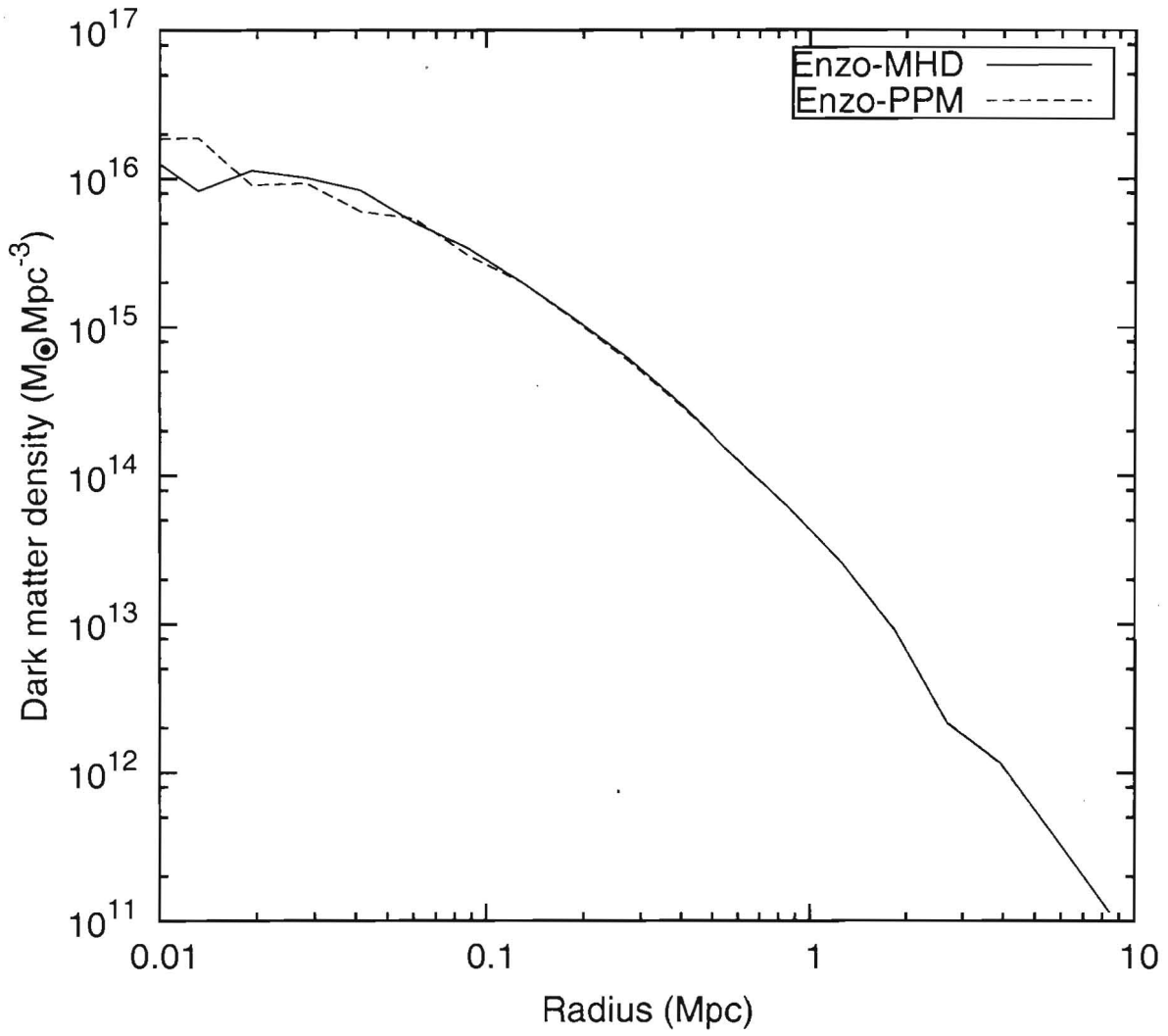


Fig. 22.— Spherically averaged dark matter density radial profile at  $z = 0$  from MHD solver and PPM solver.

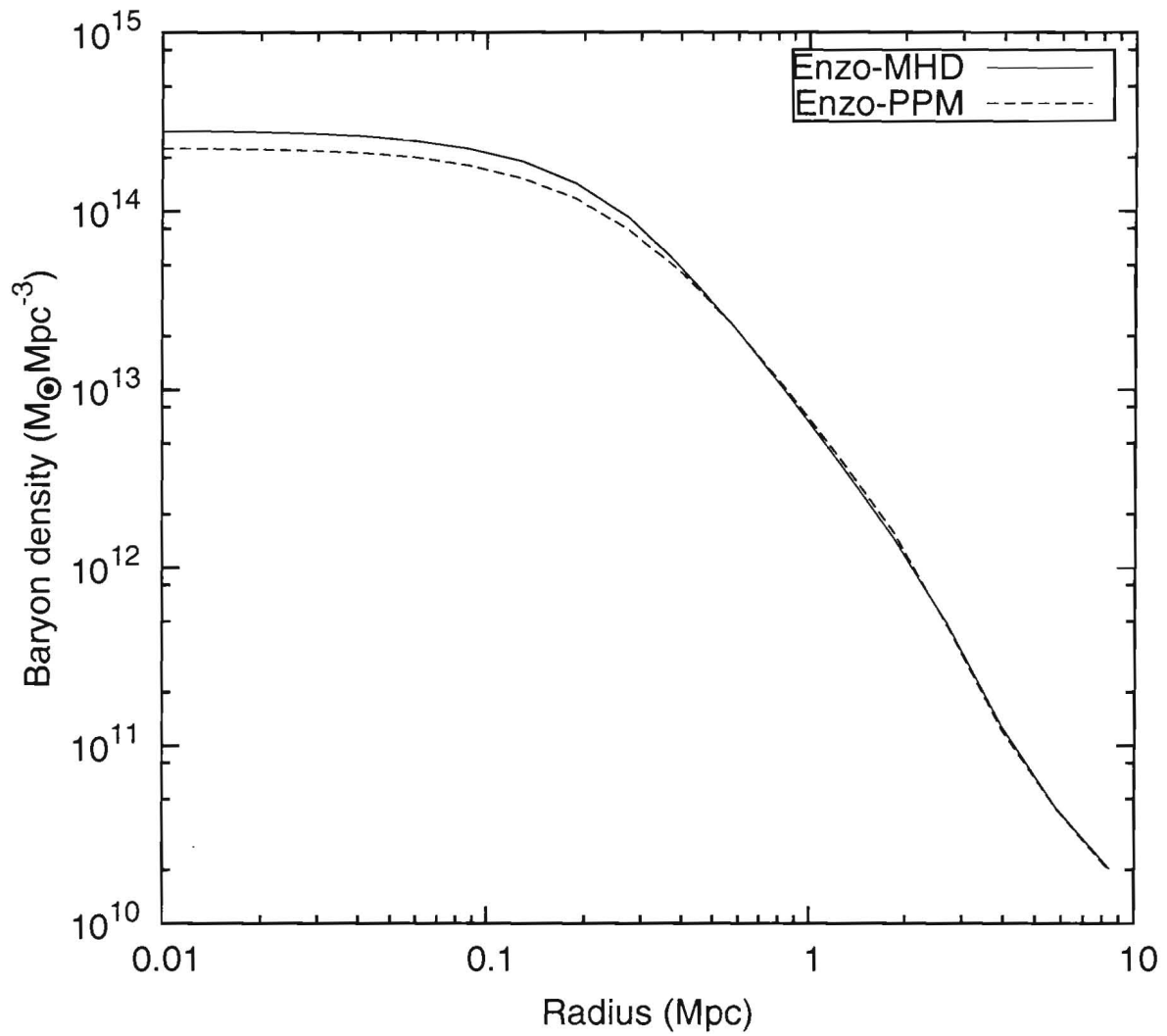


Fig. 23.— Spherically averaged gas density radial profiles at  $z = 0$  from MHD solver and PPM solver.

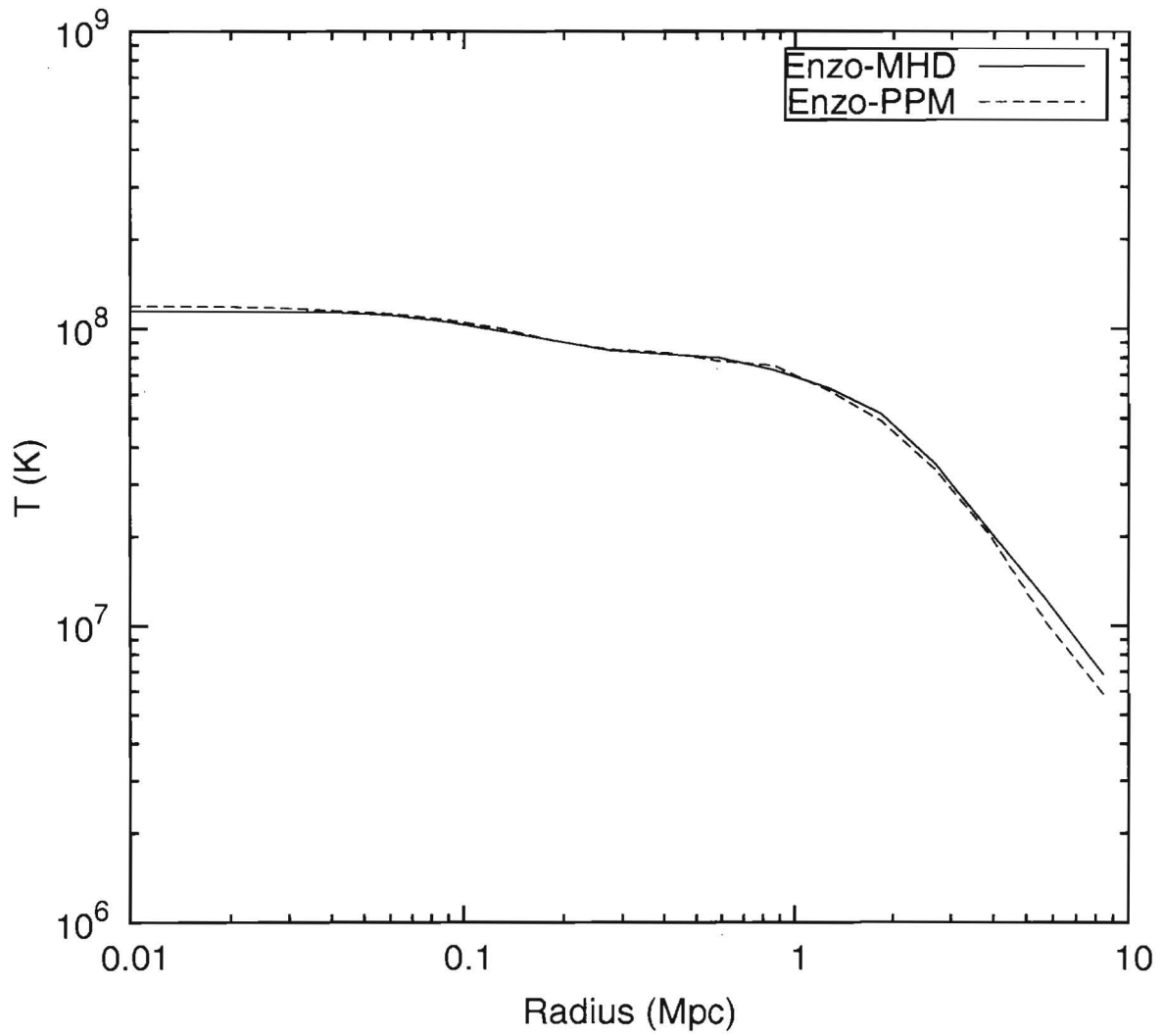


Fig. 24.— Spherically averaged temperature radial profiles at  $z = 0$  from MHD solver and PPM solver..

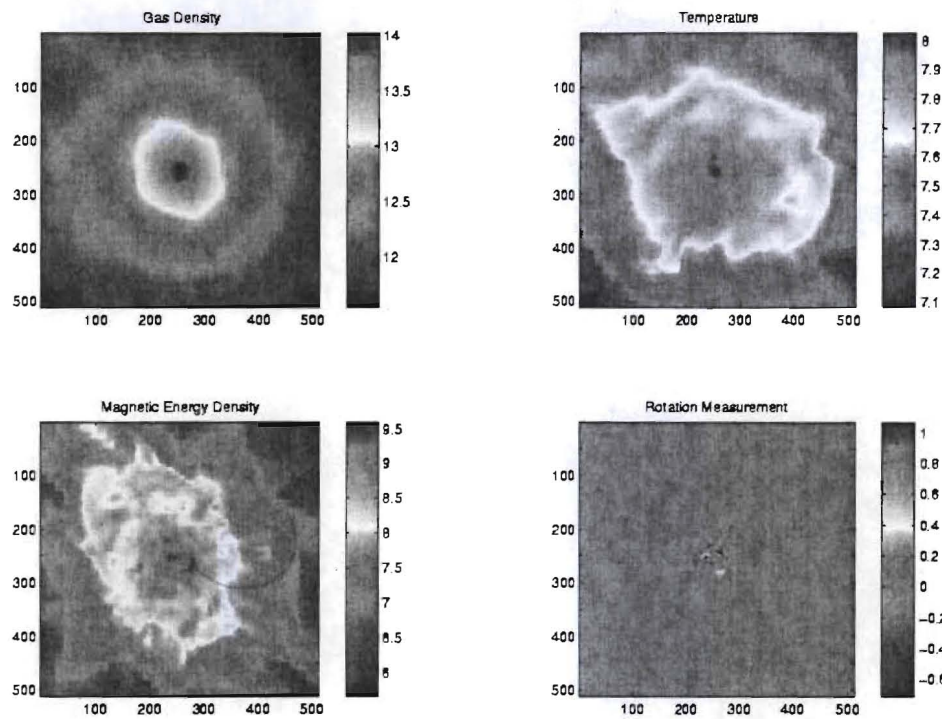


Fig. 25.— Images of gas density ( $M_{\odot} \text{ Mpc}^{-3}$ ), temperature (K), magnetic energy density ( $\text{erg cm}^{-2}$ ) and rotation measure ( $\text{rad m}^{-2}$ ) of the galaxy cluster simulation with an initial magnetic field  $B_y = 9.72753 \times 10^{-10} G$ . Projections are of the inner 4Mpc/h of cluster center at  $z = 0$ .

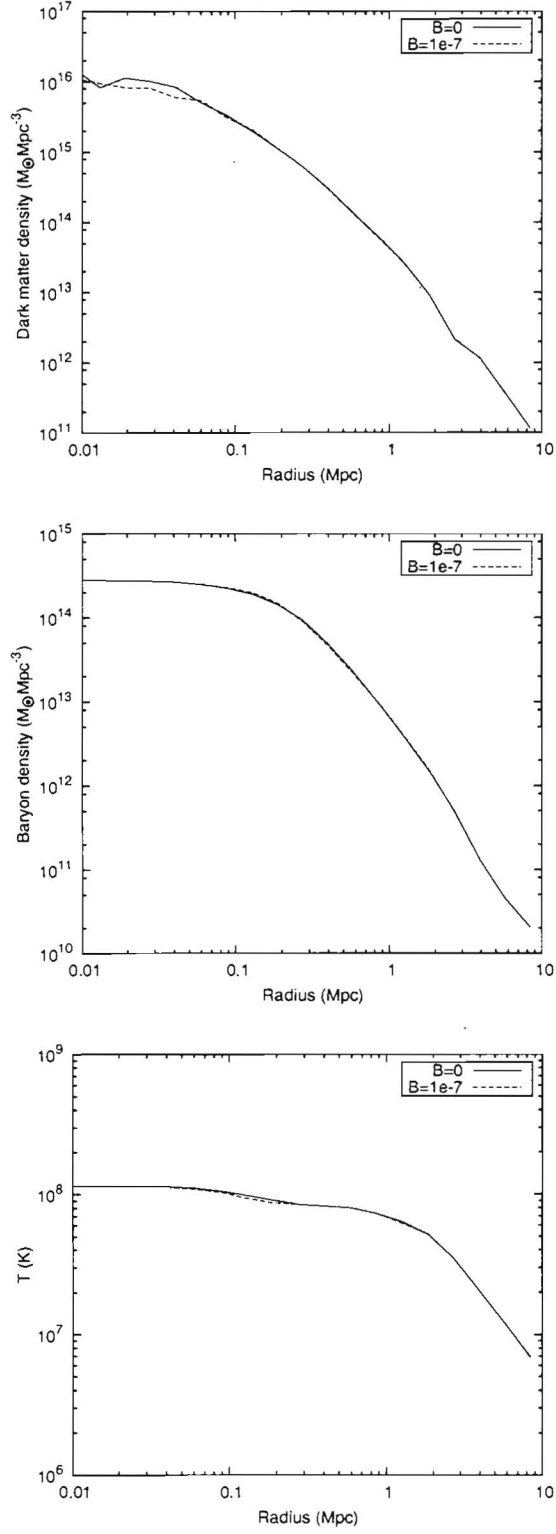


Fig. 26.— Specially averaged radial profiles of dark matter density, baryon density and temperature of MHD simulations with zero and  $B_y = 9.72753 \times 10^{-10} G$  initial magnetic fields.



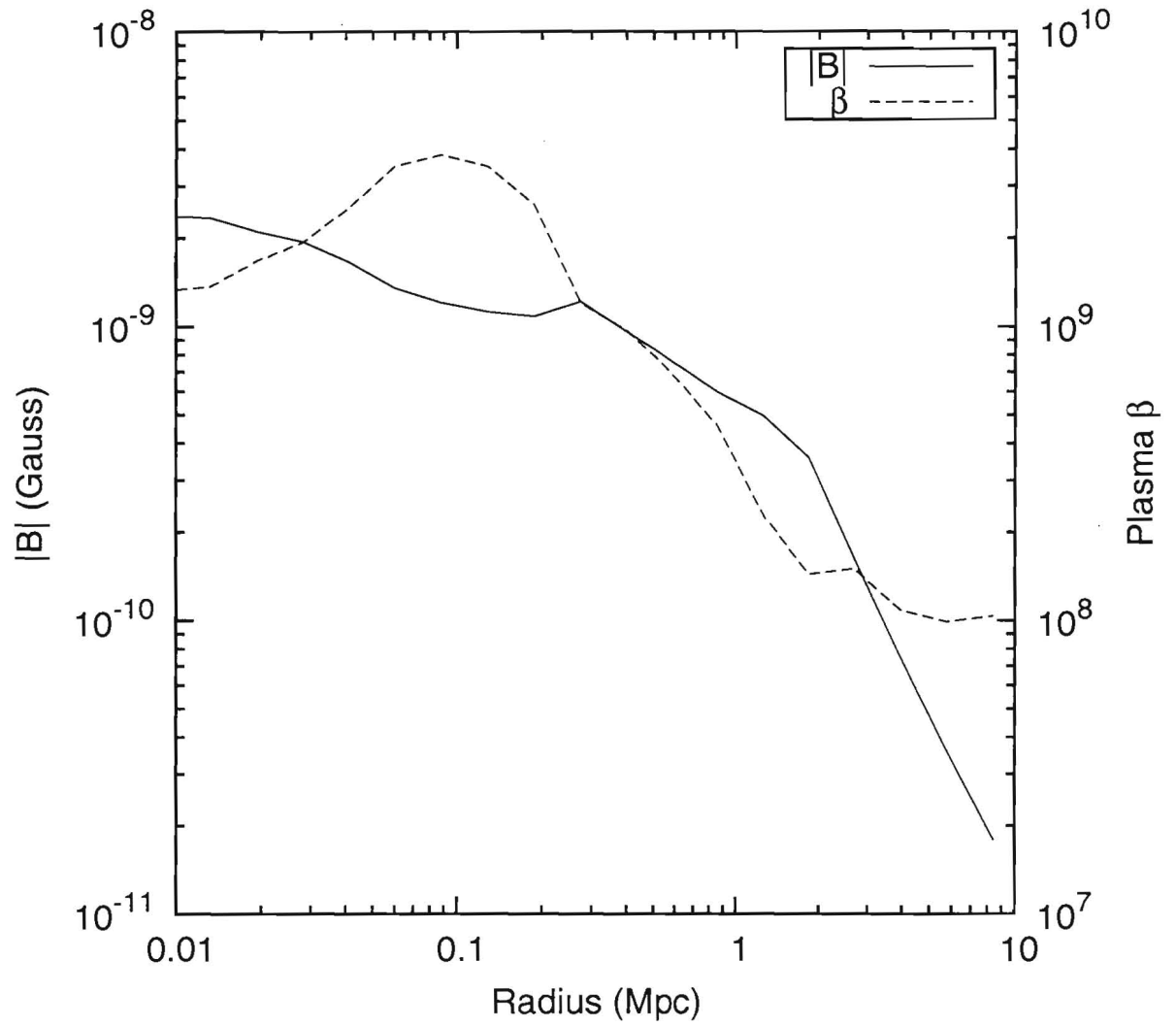


Fig. 27.— Spherically averaged radial profiles of magnetic field strength and plasma beta of MHD simulation with  $B_y = 9.72753 \times 10^{10}$  G initial magnetic fields.

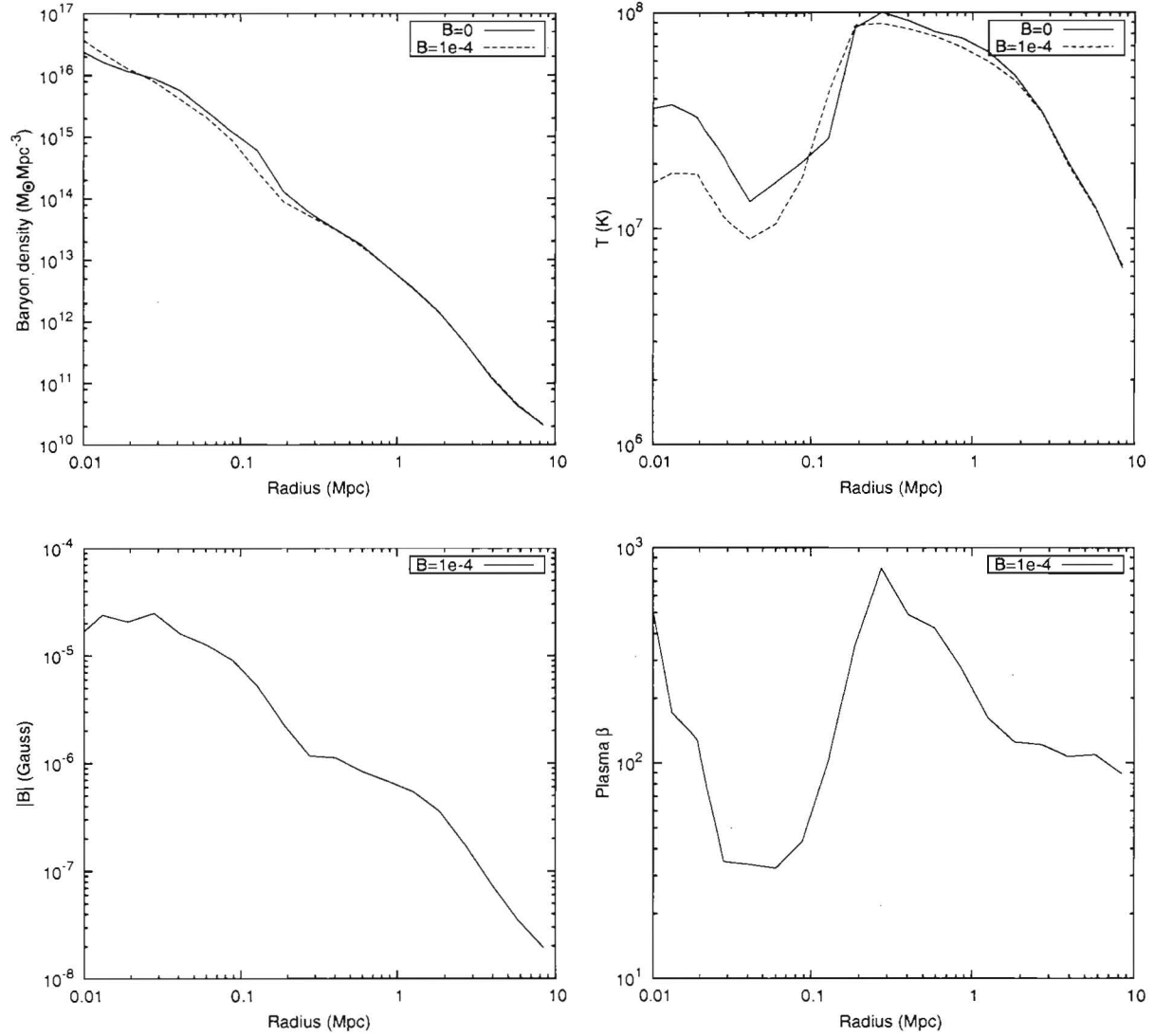


Fig. 28.— Radial profiles of MHD simulations with zero and  $B_y = 9.72753 \times 10^{-7} G$  initial magnetic fields with radiative cooling, star formation and star formation feedback.

#### 4. Conclusion

In this work, we have presented the implementation of MHD in the AMR cosmology code Enzo in order to serve as a single complete document for future simulations done with Enzo MHD. For the fixed resolution patches, we used the PDE solver of Li et al. (2008) to solve the ideal MHD equations (section 2.6); we have operator split the gravitational (2.8) and cosmological expansion (2.9) terms; and we have used the CT methods of Balsara & Spicer (1999) and Gardiner & Stone (2005) to maintain  $\nabla \cdot \mathbf{B} = 0$ . We use a slightly modified version of the AMR algorithm procedure of Balsara (2001) to create fine grids and communicate the more accurate fine grid data to the cheaper coarse grid data. A broad array of tests were done in section 3 to demonstrate that the implementation correctly matches the expected results.

We present this work as a reference for future work to be done with EnzoMHD. Currently underway are simulations involving protostellar core formation, MHD Turbulence, and galaxy cluster formation and evolution with magnetic fields, and work has begun to include cosmic ray acceleration into the code.

The authors gratefully acknowledge support from NSF grants Grant for David and Los Alamos National Lab How to cite Hao's support?, and We probably need to cite some LRAC awards

## 5. Discussion

### A. Schematic for the Cosmological MHD Code

In this section, we present a schematic of the MHD code, for clarity and easy reference.

Step 0.– We start with conserved quantities density, total energy, and momentum  $(\rho_{BM}^n, E_{total}^n, \mathbf{p}_{DM}^n)$ , and primitive quantities velocity and gas pressure  $(v_{BM}^n, P_{gas}^n)$  for the baryonic matter; face and cell centered magnetic fields  $(B_c^n, B_f^n)$ ; and Lagrangian dark matter mass, position, and velocity  $(\rho_{DM}^n, \mathbf{x}^n, \mathbf{v}_{DM}^n)$ . These are all at time  $t^n$ . Where needed, primitive quantities will be described by  $U = (\rho_{DM}, P_{gas}, \mathbf{v}_{DM}, \mathbf{B})$ , and conserved quantities by  $V = (\rho_{DM}, E_{total}, \mathbf{p}_{DM}, \mathbf{B})$ . Conversion between the two is done as needed.

Step 1. Solve Poisson's equation for the acceleration field at  $t^{n+\frac{1}{2}}$

$$\phi^n \leftarrow \rho_{BM}^n + \rho_{DM}^n \quad (\text{A1})$$

$$\phi^{n+1/2} = \phi^n \left(1 + \frac{\Delta t^n}{2\Delta t^{n-1}}\right) - \phi^{n-1} \frac{\Delta t^n}{2\Delta t^{n-1}} \quad (\text{A2})$$

$$g_i^{n+1/2} = \frac{1}{2a^{n+1/2}\delta x_i} (\phi_{i+1}^{n+1/2} - \phi_{i-1}^{n+1/2}) \quad (\text{A3})$$

Step 2.– Update particle positions and velocities. (Strictly speaking, this happens after the Expansion step, but the narrative works better if it's here.)

$$\mathbf{v}_{DM}^{n+1/2} = \mathbf{v}_{DM}^n - \frac{\Delta t^n}{2} \frac{\dot{a}^{n+1/2}}{a^{n+1/2}} \mathbf{v}_{DM}^n - \frac{\Delta t^n}{2} \mathbf{g}^{n+1/2} \quad (\text{A4})$$

$$\mathbf{x}_{DM}^{n+1} = \mathbf{x}_{DM}^n + \Delta t^n (\mathbf{v}_{i,DM}^{n+1/2} / a^{n+1/2}) \quad (\text{A5})$$

$$v_{i,DM}^{n+1} = v_{i,DM}^{n+1/2} - \frac{\Delta t^n}{2} \frac{\dot{a}^{n+1/2}}{a^{n+1/2}} v_{i,DM}^{n+1/2} - \frac{\Delta t^n}{2} g_i^{n+1/2} \quad (\text{A6})$$

Step 3.– Apply half of the gravitational and expansion update to the fields that require it, to obtain the temporary state  $\tilde{U} = (\rho, \tilde{P}_{total}^n, \tilde{\mathbf{v}}_{BM}^n, \tilde{B}_c^n)$

$$\tilde{\mathbf{v}}_{BM}^n = \mathbf{v}_{BM}^n - \frac{\Delta t^n}{2} \frac{\dot{a}^n}{a^n} \mathbf{v}_{BM}^n - \frac{\Delta t^n}{2} \frac{1}{a^n} \mathbf{g}^{n+1/2} \quad (\text{A7})$$

$$\tilde{p}^n = p^n - \frac{\Delta t^n}{2} \frac{2\dot{a}^n}{a^n} p^n \quad (\text{A8})$$

$$\tilde{B}_c^n = B_c^n - \frac{\delta t^n}{4} \frac{\dot{a}^n}{a^n} B_c^n \quad (\text{A9})$$

$$\tilde{U} = (\rho, \tilde{P}_{total}^n, \tilde{\mathbf{v}}_{BM}^n, \tilde{B}_c^n) \quad (\text{A10})$$

Step 4. Compute interface states at  $i \pm \frac{1}{2}, n + \frac{1}{2}$  using linear spatial reconstruction and second order time integration:

$$U_{i+\frac{1}{2},L}^{n+\frac{1}{2}}, U_{i+\frac{1}{2},R}^{n+\frac{1}{2}} \leftarrow \tilde{U}_{i-1}, \tilde{U}_i, \tilde{U}_{i+1}, \tilde{U}_{i+2} \quad (\text{A11})$$

Step 4. Compute approximation of the flux in equation 16 at the interface  $i + \frac{1}{2}$ . This is done by solving the Riemann problem using one of the solvers mentioned in section 2.6

$$\hat{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \text{Riemann}(U_{i+\frac{1}{2},L}^{n+\frac{1}{2}}, U_{i+\frac{1}{2},R}^{n+\frac{1}{2}}) \quad (\text{A12})$$

Step 5. Update the conserved quantities with the new fluxes:

$$(V_i^{n+1})_{MHD} = V_i^n - \frac{\Delta t}{\Delta x} [\hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}}] \quad (\text{A13})$$

Step 5. Compute Electric field from Fluxes

$$E_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \Leftarrow \hat{F}_{i+\frac{1}{2}} \quad (\text{A14})$$

Step 6.-Gravitational step for the baryonic matter, with time centered density

$$(p_{i,BM}^{n+1})_{MHD,Grav} = (p_{i,BM}^{n+1})_{MHD} - \Delta t^n \frac{(\rho^n + \rho_{MHD}^{n+1})}{2} g_i^{n+1/2} \quad (\text{A15})$$

Step 7.-Expansion step for the baryonic matter,

$$(v_{BM}^{n+1})_{MHD,Grav,exp} = \frac{1 - (\Delta t^n/2)(\dot{a}^{n+1/2}/a^{n+1/2})}{1 + (\Delta t^n/2)(\dot{a}^{n+1/2}/a^{n+1/2})} (v_{BM}^{n+1})_{MHDGrav} \quad (\text{A16})$$

$$p^{n+1} = \frac{1 - (\Delta t^n)(\dot{a}^{n+1/2}/a^{n+1/2})}{1 + (\Delta t^n)(\dot{a}^{n+1/2}/a^{n+1/2})} (p^{n+1})_{MHD} \quad (\text{A17})$$

Step 8. Recurse to finer grids. Integrate fine grids from  $t^n$  to  $t^{n+1}$

$$V_{FineGrids}^{n+1} \Leftarrow V_{FineGrids}^n \quad (\text{A18})$$

Step 9.-Flux correction step for conserved baryon field quantities

$$V_{MHD,Grav,exp,fc}^{n+1} \Leftarrow (\hat{F}^{n+1/2}), (\hat{F}^{n+1/2})_{FineGrids}, V_{MHDGrav,exp}^{n+1} \quad (\text{A19})$$

Step 10.-Project conserved baryon field quantities and electric field from fine grids to coarse grids. This is done *after* the flux correction step to avoid any bookkeeping errors. The average is taken over  $\Delta t^n$  and the surface of each *FineGrid*.

$$V_{ParentGrid}^{n+1} = \langle V_{FineGrid}^{n+1} \rangle_{t,surface} \quad (\text{A20})$$

$$E_{ParentGrid}^{n+\frac{1}{2}} = \langle E_{FineGrid}^{n+\frac{1}{2}} \rangle_{t,surface} \quad (\text{A21})$$

Step 8.-Update magnetic fields from electric fields,

$$B_f^{n+1} = B_f^n - \frac{\Delta t}{a} \nabla \times E_{ParentGrid}^{n+\frac{1}{2}} \quad (\text{A22})$$



Step 9. Apply expansion to the Face Centered Fields

$$B_{f,exp}^{n+1} = \frac{1 - (\Delta t^n/4)(\dot{a}^{n+1/2}/a^{n+1/2})}{1 + (\Delta t^n/4)(\dot{a}^{n+1/2}/a^{n+1/2})} (B_f^{n+1}) \quad (\text{A23})$$

Step 10. Compute cell centered magnetic field from face centered (with the expansion subscript from step 9 dropped for clarity)

$$\begin{aligned} B_{c,x,i,j,k}^{n+1} &= 0.5 * (B_{f,x,i+\frac{1}{2},j,k} + B_{f,x,i-\frac{1}{2},j,k}) \\ B_{c,y,i,j,k}^{n+1} &= 0.5 * (B_{f,y,i,j+\frac{1}{2},k} + B_{f,y,i,j-\frac{1}{2},k}) \\ B_{c,z,i,j,k}^{n+1} &= 0.5 * (B_{f,z,i,j,k+\frac{1}{2}} + B_{f,z,i,j,k-\frac{1}{2}}) \end{aligned} \quad (\text{A24})$$

Step 11. We have now finished an update of this level. Rebuild the hierarchy from this level down.

$$V_{New \text{ FineGrids}}^{n+1} \Leftarrow V^{n+1} \quad (\text{A25})$$

$$B_{f,New \text{ FineGrids}}^{n+1} \Leftarrow B_f^{n+1} \quad (\text{A26})$$

## B. AMR MHD Reconstruction

### B.1. Hydrodynamic Reconstruction

I don't have any idea how the hydro reconstruction works. I've never looked into it. I probably should, before the paper comes out.

### B.2. MHD Reconstruction

For completeness, we will briefly outline the AMR reconstruction used in EnzoMHD. The reader is encouraged to see the details in the original paper by Balsara (2001).

In this appendix, we have dropped the subscript  $f$  from the face centered fields, as the face centered field is the only one in question.

Balsara's reconstruction method for the magnetic field is a 3 dimensional, quadratic reconstruction of all 3 vector fields simultaneously. If we let  $b$  be the polynomial fit to the discrete face centered field  $B$ , the general reconstruction is

$$b_x(x, y, z) = a_0 + a_x x + a_y y + a_z z + a_{xx} x^2 + a_{xy} xy + a_{xz} xz \quad (\text{B1})$$

$$b_y(x, y, z) = b_0 + b_x x + b_y y + b_z z + b_{xy} xy + b_{yy} y^2 + b_{yz} yz \quad (\text{B2})$$

$$b_z(x, y, z) = c_0 + c_x x + c_y y + c_z z + c_{xz} xz + c_{yz} yz + c_{zz} z^2 \quad (\text{B3})$$

The coefficients are found by the following constraints:

1. The analytic reconstruction should be divergence free.
2. At the faces of the parent cell, the reconstruction should reduce to a bilinear reconstruction, where the slopes are monotonized with the minmod slope limiter. For instance,

$$b_x(x = \frac{\Delta x}{2}, y) = B_{x, i+\frac{1}{2}, j, k} + \frac{\Delta_y B_{x, i+\frac{1}{2}}}{\Delta y} y + \frac{\Delta_z B_{x, i+\frac{1}{2}}}{\Delta z} z \quad (\text{B4})$$

where

$$\Delta_y B_{x, i+\frac{1}{2}} = \text{minmod}(B_{x, i+\frac{1}{2}, j+1} - B_{x, i+\frac{1}{2}, j}, B_{x, i+\frac{1}{2}, j} - B_{x, i+\frac{1}{2}, j-1}) \quad (\text{B5})$$

$$\text{minmod}(x, y) = \begin{cases} x, & |x| < |y| \text{ and } xy > 0 \\ y, & |y| < |x| \text{ and } xy > 0 \\ 0, & xy < 0 \end{cases} \quad (\text{B6})$$

The *minmod* slope is used in order to minimize oscillations. Area weighted averages over these polynomials are then used to assign the fine grid values.

Often, a fine grid patch will encroach on unrefined territory. This results in the refinement of coarse zones that a.) share a face with fine grids but b.) don't have corresponding fine grids of their own. Balsara refers to this as "Prolongation" of the fine grid. To avoid generating any divergence at the boundary of the face, the interpolation polynomials need to match the old fine data. The interpolation equations above (eqns B1 - B3) do not have enough degrees of freedom to accommodate that many data points. In this case, Balsara describes a new polynomial that DOES have enough degrees of freedom, by adding 3<sup>rd</sup> order cross terms to equations B1 - B3:

$$\begin{aligned} b_x(x, y, z) = & a_0 + a_x x + a_y y + a_z z + a_{xx} x^2 + a_{xy} xy + a_{xz} xz \\ & + a_{yz} yz + a_{xyz} xyz + a_{xxz} x^2 z + a_{xxy} x^2 y \end{aligned} \quad (B7)$$

$$\begin{aligned} b_y(x, y, z) = & b_0 + b_x x + b_y y + b_z z + b_{xy} xy + b_{yy} y^2 + b_{yz} yz \\ & + b_{xz} xz + b_{yyz} y^2 z + b_{xyz} xyz + b_{yyx} xy^2 \end{aligned} \quad (B8)$$

$$\begin{aligned} b_z(x, y, z) = & c_0 + c_x x + c_y y + c_z z + c_{xz} xz + c_{yz} yz + c_{zz} z^2 \\ & + c_{xy} xy + c_{yzz} yz^2 + c_{xzz} xz^2 + c_{xyz} xyz \end{aligned} \quad (B9)$$

The yet undetermined coefficients are found by matching the polynomial to a bilinear fit on the face:

$$b(x = \frac{\Delta x}{2}, y, z) = B_{x, i+\frac{1}{2}, j, k} + \frac{\Delta_y B_{x, i+\frac{1}{2}}}{\Delta y} y + \frac{\Delta_z B_{x, i+\frac{1}{2}}}{\Delta z} z + \frac{\Delta_{yz} B_{x, i+\frac{1}{2}}}{\Delta y \Delta z} yz \quad (B10)$$

and now the finite differences are taken from the finest grid:

$$\begin{aligned} \Delta_{yz} B_{x, i+\frac{1}{2}} = & 4((B_{x, i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2}} - B_{x, i+\frac{1}{2}, j-\frac{1}{2}, k+\frac{1}{2}}) - \\ & (B_{x, i+\frac{1}{2}, j+\frac{1}{2}, k-\frac{1}{2}} - B_{x, i+\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}})) \end{aligned} \quad (B11)$$

$$\begin{aligned} \Delta_y B_{x, i+\frac{1}{2}} = & ((B_{x, i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2}} - B_{x, i+\frac{1}{2}, j-\frac{1}{2}, k+\frac{1}{2}}) + \\ & (B_{x, i+\frac{1}{2}, j+\frac{1}{2}, k-\frac{1}{2}} - B_{x, i+\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}})) \end{aligned} \quad (B12)$$

where  $B$  is the field on the fine grid. Note that since this is now a centered difference, the minmod slope limiter is not used.

### B.3. Implementation in Enzo

In order to avoid complicated book keeping routines to determine which cells are being Prolonged into, and from which direction, we formulate only one interpolation polynomial, given by equations B7-B9. The necessary finite differences for a given refinement region are taken from the finest data available, as in equations B11 and B12. The last four terms in each reconstruction polynomial are there exclusively to ensure consistency of Old Fine Grid Data, so for faces that have no Fine Data before the reconstruction, these are set to zero. Since the reconstruction polynomial exactly matches the old fine grid data, this also eliminates the need to copy the old fine grid data to the newly refined patch.

### C. Flux Correction

At any given time in an AMR simulation, there are points in space that are described by more than one data structure. In a finite volume hydro calculation, with cell centered data fields, this occurs at the boundary between coarse and fine grids in the Flux fields,  $\vec{F}$ . In an AMR MHD calculation, with face centered magnetic fields, this occurs at the same boundary, in the face centered magnetic field, and the edge centered electric field. Ensuring consistency between data is vital for the conservation of quantities like mass, energy, momentum, and  $\nabla \cdot \mathbf{B}$ . Flux Correction is essential for this consistency.

#### C.1. Conservation Form

I thought it would be useful to briefly describe the basic formulation of the methods used in Enzo and EnzoMHD before moving on to the flux correction mechanism.

Take any conservative system, such as ideal MHD. It can be written in a differential form as

$$\frac{\partial V}{\partial t} + \nabla \cdot F = 0 \quad (\text{C1})$$

where  $V$  and  $F$  are suitably defined, in our case by 15 and 16. Here we ignore any source terms.

In finite volume methods, we store average quantities of  $V$  and  $F$ , and re-write the conservation law in Conservation Form, using the Fundamental Theorem and Stokes Theorem. Starting with eqn C1, and integrating, we get:

$$\int_t^{t+\Delta t} \int_V \frac{\partial V}{\partial t} dV dt = - \int_t^{t+\Delta t} \int_A F \cdot dA dt \quad (\text{C2})$$

where the volume  $V$  is taken from the point  $(x, y, z)$  to  $(x + \Delta x, y + \Delta y, z + \Delta z)$ . Now let

$$\hat{V}^n = \frac{1}{\Delta V} \int_V V(x, y, z, t^n) dV \quad (\text{C3})$$

$$\tilde{F}_{x, I+\frac{1}{2}, J, K} = \frac{1}{\Delta y \Delta z} \int_{\Delta y, \Delta z} F(x = I + \frac{1}{2}, y, z) \cdot \hat{x} dy dz \quad (\text{C4})$$

where  $\hat{x}$  is the unit vector in the  $x$  direction. Similar definitions apply  $\tilde{F}_y$  and  $\tilde{F}_z$ , and

$$\hat{F}_x = \frac{1}{\Delta t} \int_{\Delta t} \tilde{F}_x dt \quad (\text{C5})$$

The averaging here was taken explicitly in two steps to emphasize that  $\Delta x, \Delta y$  and  $\Delta z$  are possibly functions of  $t$ , as the are in cosmological hydrodynamics. Putting this all together, we get the



equations in their final analytical form before discretization (also the last form we'll be using here)

$$\begin{aligned}\hat{V}_{I,J,K}^{n+1} = \hat{V}_{I,J,K}^n - \Delta t & \left( \frac{1}{\Delta x} (\hat{F}_{x,I+\frac{1}{2},J,K} - \hat{F}_{x,I-\frac{1}{2},J,K}) + \right. \\ & \frac{1}{\Delta y} (\hat{F}_{y,I,J+\frac{1}{2},K} - \hat{F}_{y,I,J-\frac{1}{2},K}) + \\ & \left. \frac{1}{\Delta z} (\hat{F}_{z,I,J,K+\frac{1}{2}} - \hat{F}_{z,I,J,K-\frac{1}{2}}) \right)\end{aligned}\quad (C6)$$

Note that equation C6 is an exact equation, since only averages and the fundamental theorem of calculus have been used up to this point. The trick in finite volume methods such as our MHD is finding finding appropriate approximations to  $\hat{F}$  that are both accurate and stable.

### C.2. Conservation Form and AMR: Enter Flux Correction.

As mentioned at the beginning of the section, an AMR simulation has multiple data structures representing a single point in space. In entirely cell centered codes such as PPM, the only such instance is at the surface of a fine grid boundary, where both the fine grid and coarse grid represent the flux at that point. Moreover, after the fine grid field is projected into the coarse, there's a mismatch on the coarse grid itself as to the value of the flux at the surface. The value of that discrepancy can be easily found. After the projection, a coarse grid at a point  $(I, J)$  has the value (restricting to 2d, for clarity)

$$\hat{V}_{I,J}^{n+1} = \sum_{\substack{i=I\pm\frac{1}{4} \\ j=J\pm\frac{1}{4}}} \hat{q}_{i,j}^{n+1} \quad (C7)$$

where lower case quantities denote the value of the fine grid data. Expanding the time update for  $\hat{q}^{n+1}$  in space and time, we find that

$$\hat{V}_{I,J}^{n+1} = \sum_{\substack{i=I\pm\frac{1}{4} \\ j=J\pm\frac{1}{4}}} \hat{q}_{i,j}^n - \left( \sum_{m=n}^{n+1} \sum_{x,j=J\pm\frac{1}{4}} \frac{\Delta t^m}{\Delta V^m} \hat{f}_{I+\frac{1}{2},j}^m + \sum_{m=n}^{n+1} \sum_{x,j=J\pm\frac{1}{4}} \frac{\Delta t^m}{\Delta V^m} \hat{f}_{I-\frac{1}{2},j}^m \right) \quad (C8)$$

$$-(y \text{ and } z \text{ terms}) \quad (C9)$$

By construction of the interpolation polynomial (and projection at the last timesteps) the first term is just equal to  $\hat{V}_{I,J}^n$ , which means that, by equation C6  $\hat{V}_{I,J}$  effectively sees, at the point  $I + \frac{1}{2}$ ,

$$\frac{\Delta t}{\Delta V} \hat{F}_x = \sum_{m=n}^{n+1} \sum_{x,j=J\pm\frac{1}{4}} \frac{\Delta t^m}{\Delta V^m} \hat{f}_{I+\frac{1}{2},j}^m := \langle f_x \rangle \quad (C10)$$

However, for the cell  $(I-1, J)$  depicted in figure (Working on this figure), which has no corresponding fine grid flux,  $\hat{F}_{I+\frac{1}{2}}$  come from the discretization method on the coarse grid. There is absolutely no reason for the two to match, so we have a discrepancy in the descriptions of the data. This can

be solved by simply replacing the less refined data that  $\hat{V}_{I+1,J}$  used with the more refined average, given by equation C10:

$$\hat{V}_{I+1,J,f_c} = \hat{V}_{I+1,J} + \frac{\Delta t}{\Delta V} \hat{F}_{x,I+\frac{1}{2},J} - \sum_m \sum_j \frac{\Delta t^m}{\Delta V^m} \hat{f}_{x,I+\frac{1}{2},j}^m \quad (\text{C11})$$

Now every place  $\hat{F}_{x,I,J}$  show up in our method, the exact same approximation is used.

### C.3. Flux Correction and MHD

A similar formalism to that described in C.1 is used for to advance the magnetic fields in EnzoMHD, but instead of using volume averages, we use area averages. The magnetic evolution is given by the induction equation:

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E} \quad (\text{C12})$$

When discretized, equation C12 yields the equation

$$\begin{aligned} \hat{B}_{x,I+\frac{1}{2},J}^{n+1} = \hat{B}_{x,I+\frac{1}{2},J}^n - \frac{\Delta t}{\Delta y \Delta z} & (\Delta z (\hat{E}_{z,I+\frac{1}{2},J+\frac{1}{2},K} - \hat{E}_{z,I+\frac{1}{2},J-\frac{1}{2},K}) + \\ & \Delta y (\hat{E}_{y,I+\frac{1}{2},J,K+\frac{1}{2}} - \hat{E}_{y,I+\frac{1}{2},J,K-\frac{1}{2}})) \end{aligned} \quad (\text{C13})$$

where

$$\hat{B}_{x,I+\frac{1}{2},J,K}^n = \frac{1}{\Delta y \Delta z} \int_A \vec{B}(x = I + \frac{1}{2}, y, z, t^n) \cdot \hat{x} dy dz \quad (\text{C14})$$

$$\hat{E}^n = \frac{1}{\Delta t} \int_t^{t+\Delta t} \frac{1}{\Delta x} \int_x \vec{E} \cdot d\vec{l} dt \quad (\text{C15})$$

which is also exact, and the main problem is finding a suitable approximation for  $\hat{E}$ .

Again, after the area-weighted projection of the fine grid field  $\hat{b}_x$  into the coarse grid  $\hat{B}_x$ , there's a discrepancy between the electric field at a refined point on the surface of a refined grid, as it's seen by both grids that have subgrids and grids that don't. In Balsara (2001), he suggests a similar flux correction mechanism to that of the standard hydro, described in C.2. However, due to an issue with the initial implementation of Flux Correction in Enzo (which has since been fixed), we chose a different route. In EnzoMHD, instead of projecting fine zones into coarse zones and then correcting zones in the coarse grid, we project as before, but then take the curl of the entire coarse grid. Thus, all coarse grid magnetic fields see the most accurate data at the same time, and no a-posteriori correction needs to be done. Where there are no subgrids, the coarse grid sees an electric field that comes from the CT module in section 2.7, and where there are subgrids it sees

$$\begin{aligned} \hat{E}_{z,i-\frac{1}{2},j-\frac{1}{2},k}^n = \frac{\Delta t^n}{\Delta t} & (e_{z,i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{4}}^{n+\frac{1}{2}} + e_{z,i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{4}}^{n+\frac{1}{2}}) + \\ & \frac{\Delta t^{n+\frac{1}{2}}}{\Delta t} (e_{z,i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{4}}^{n+\frac{3}{4}} + e_{z,i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{4}}^{n+\frac{3}{4}}) \end{aligned} \quad (\text{C16})$$

While a complete flux correction treatment would potentially save on memory and flops, in practice the extra memory is negligible compared to the total memory and time used by the rest of Enzo, and the extra floating point operations done here are offset by increase cache utilization of the data, as the entire grid is done in a single stride one sweep instead of an essentially random access pattern.

## REFERENCES

- Abel, T., Bryan, G. L., & Norman, M. L. 2000, *ApJ*, 540, 39
- . 2002, *Science*, 295, 93
- Anninos, P., Zhang, Y., Abel, T., & Norman, M. L. 1997, *New Astronomy*, 2, 209
- Balsara, D. S. 2001, *Journal of Computational Physics*, 174, 614
- Balsara, D. S. & Kim, J. 2004, *ApJ*, 602, 1079
- Balsara, D. S. & Spicer, D. S. 1999, *Journal of Computational Physics*, 149, 270
- Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, *ApJ*, 304, 15
- Berger, M. J. & Colella, P. 1989, *Journal of Computational Physics*, 82, 64
- Brackbill, J. U. & Barnes, D. C. 1980, *Journal of Computational Physics*, 35, 426
- Brio, M. & Wu, C. C. 1988, *Journal of Computational Physics*, 75, 400
- Bryan, G. L., Machacek, M., Anninos, P., & Norman, M. L. 1999, *ApJ*, 517, 13
- Bryan, G. L. & Norman, M. L. 1998, *ApJ*, 495, 80
- Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R., & Ostriker, J. P. 1995, *Computer Physics Communications*, 89, 149
- Carilli, C. L. & Taylor, G. B. 2002, *ARA&A*, 40, 319
- Cen, R. & Ostriker, J. P. 1993, *ApJ*, 417, 404
- Colella, P. & Woodward, P. R. 1984, *Journal of Computational Physics*, 54, 174
- Dedner, A., Kemm, F., Kröner, D., Munz, C.-D., Schnitzer, T., & Wesenberg, M. 2002, *Journal of Computational Physics*, 175, 645
- Dolag, K., Bartelmann, M., & Lesch, H. 1999, *A&A*, 348, 351
- Evans, C. R. & Hawley, J. F. 1988, *ApJ*, 332, 659
- Gardiner, T. A. & Stone, J. M. 2005, *Journal of Computational Physics*, 205, 509
- Hallman, E. J., Motl, P. M., Burns, J. O., & Norman, M. L. 2006, *ApJ*, 648, 852
- Hockney, R. & Eastwood, j. 1985, *Computer Simulation Using Particles* (McGraw Hill, New York)
- Iapichino, L. & Niemeyer, J. C. 2008, *MNRAS*, 388, 1089

- Jena, T., Norman, M. L., Tytler, D., Kirkman, D., Suzuki, N., Chapman, A., Melis, C., Paschos, P., O'Shea, B., So, G., Lubin, D., Lin, W.-C., Reimers, D., Janknecht, E., & Fechner, C. 2005, *MNRAS*, 361, 70
- Kim, J., Ryu, D., Jones, T. W., & Hong, S. S. 1999, *ApJ*, 514, 506
- Kritsuk, A. G. & Norman, M. L. 2002, *ApJ*, 569, L127
- . 2004, *ApJ*, 601, L55
- Kritsuk, A. G., Norman, M. L., Padoan, P., & Wagner, R. 2007, *ApJ*, 665, 416
- Li, S. 2005, *Journal of Computational Physics*, 203, 344
- Li, S., Li, H., & Cen, R. 2008, *ApJS*, 174, 1
- Loken, C., Norman, M. L., Nelson, E., Burns, J., Bryan, G. L., & Motl, P. 2002, *ApJ*, 579, 571
- Mignone, A. 2007, *Journal of Computational Physics*, 225, 1427
- Miyoshi, T. & Kusano, K. 2005, *AGU Fall Meeting Abstracts*, B1295–
- Motl, P. M., Burns, J. O., Loken, C., Norman, M. L., & Bryan, G. 2004, *ApJ*, 606, 635
- Norman, M. L. 2005, in *Background Microwave Radiation and Intracluster Cosmology*, ed. F. Melchiorri & Y. Rephaeli, 1–58603
- Orszag, S. A. & Tang, C.-M. 1979, *Journal of Fluid Mechanics*, 90, 129
- O'Shea, B. W., Abel, T., Whalen, D., & Norman, M. L. 2005, *ApJ*, 628, L5
- O'Shea, B. W. & Norman, M. L. 2007, *ApJ*, 654, 66
- Powell, K. G., Roe, P. L., Linde, T. J., Gombosi, T. I., & de Zeeuw, D. L. 1999, *Journal of Computational Physics*, 154, 284
- Ryu, D., Ostriker, J. P., Kang, H., & Cen, R. 1993, *ApJ*, 414, 1
- Xu, H., O'Shea, B. W., Collins, D. C., Norman, M. L., Li, H., & Li, S. 2008, *ApJ*, 688, L57