SIMULATING RADIATING AND MAGNETIZED FLOWS IN MULTI-DIMENSIONS WITH ZEUS-MP

John C. Hayes[1], Michael L. Norman[1,2], Robert A. Fiedler[3], James O. Bordner[1,2], Pak Shing Li[4], Stephen E. Clark[2], Asif ud-Doula[5,6], and Mordecai-Mark MacLow[7]

Draft version April 11, 2006

ABSTRACT

This paper describes ZEUS-MP, a multi-physics, massively parallel, message-passing implementation of the ZEUS code. ZEUS-MP differs significantly from the thoroughly documented ZEUS-2D code, the completely undocumented (in peer-reviewed literature) ZEUS-3D code, and a marginally documented "version 1" of ZEUS-MP first distributed publicly in 1999. ZEUS-MP offers an MHD algorithm which is better suited for multidimensional flows than the ZEUS-2D module by virtue of modifications to the Method of Characteristics scheme first suggested by Hawley & Stone (1995). This MHD module is shown to compare quite favorably to the TVD scheme described by Ryu et al. (1998). ZEUS-MP is the first publicly-available ZEUS code to allow the advection of multiple chemical (or nuclear) species. Radiation hydrodynamic simulations are enabled via an implicit flux-limited radiation diffusion (FLD) module. The hydrodynamic, MHD, and FLD modules may be used, singly or in concert, in one, two, or three space dimensions. Additionally, so-called "1.5-D" and "2.5-D" grids, in which the "half-D" denotes a symmetry axis along which a constant but non-zero value of velocity or magnetic field is evolved, are supported. Self gravity may be included either through the assumption of a GM/r potential or a solution of Poisson's equation using one of three linear solver packages (conjugate-gradient, multigrid, and FFT) provided for that purpose. Point-mass potentials are also supported.

Because ZEUS-MP is designed for large simulations on parallel computing platforms, considerable attention is paid to the parallel performance characteristics of each module in the code. Strong-scaling tests involving pure hydrodynamics (with and without self-gravity), MHD, and RHD are performed in which large problems (256^3 zones) are distributed among as many as 1024 processors of an IBM SP3. Parallel efficiency is a strong function of the amount of communication required between processors in a given algorithm, but all modules are shown to scale well on up to 1024 processors for the chosen fixed problem size.

Subject headings: hydrodynamics – methods:numerical – methods:parallel – MHD – radiative transfer

1. INTRODUCTION

Since their formal introduction in the literature, the ZEUS simulation codes have enjoyed widespread use in the numerical astrophysics community, having been applied to such topics as planetary nebulae (García-Segura et al. 1999), molecular cloud turbulence (Mac Low 1999), solar magnetic arcades (Low & Manchester 2000), and galactic spiral arm formation (Martos et al. 2004a,b). The numerical methods used in the axisymmetric ZEUS-2D code are documented in an often-cited trio of papers (Stone & Norman 1992a,b; Stone et al. 1992b) well familiar to the computational astrophysics community. A reasonable first question to ask regarding this report might therefore be, "why write another ZEUS 'method' paper?" The first reason is that the code we describe in this paper, ZEUS-MP, is a significantly dif-

[1] Center for Astrophysics and Space Sciences, University of California, San Diego, CA 92093
[2] Physics Department, University of California, San Diego, CA 92093
[3] Center for Simulation of Advanced Rockets, University of Illinois, Urbana, IL 61801
[4] Astronomy Department, University of California at Berkeley, Berkeley, CA 94720
[5] Bartol Research Institute, University of Delaware, Newark, DE 19716
[6] Department of Physics and Astronomy, Swarthmore College, Swarthmore, PA 19081
[7] Department of Astrophysics, American Museum of Natural History, New York, NY 10024

ferent code when compared to the highly-documented ZEUS-2D code, the completely undocumented (in peer-reviewed literature) ZEUS-3D code, and a marginally documented "version 1" of ZEUS-MP made publicly available in 1999. The new version of ZEUS-MP we present is the first ZEUS code to unite 3D hydrodynamics (HD) and 3D magnetohydrodynamics (MHD) with implicit flux-limited radiation diffusion (FLD) and self-gravity in a software framework designed for execution on massively parallel architectures. This last feature anticipates a second major reason for offering a new method paper: the computing hardware landscape in which numerical astrophysicists operate has changed enormously since the ZEUS-2D trilogy was published. The enormous increase in computer processor speed and available memory has brought with it a new paradigm for computing in which large simulations are distributed across many (often hundreds to thousands) of parallel processors; this new environment has spawned additional figures of merit, such as parallel scalability, by which modern numerical algorithms must be judged. A major component of this paper is the demonstration of the suitability of ZEUS-MP's algorithms, both new and familiar, for parallel execution on thousands of processors.

In addition to describing the many new features and capabilities provided in ZEUS-MP, this paper fills significant gaps in the evolution history of the MHD and radiation modules offered in predecessors of ZEUS-MP. These gaps were partially a consequence of the evolu-

tion of the ZEUS series. The first code formally named ZEUS was developed by David Clarke (Clarke 1988; Clarke et al. 1986) for MHD simulations of radio jets. Thereafter, continued development of the ZEUS method proceeded along two parallel tracks. One track resulted in the release of the ZEUS-2D code, which solves the equations of self-gravitating radiation magnetohydrodynamics in two or 2.5 dimensions. ("2.5-D" denotes a problem computed in 2 spatial dimensions involving the 3-component of a vector quantity, such as velocity, that is invariant along the 3-axis but variable along the 1- and 2-axes.) The creation of ZEUS-2D occasioned the development and incorporation of several new algorithms, including (1) a covariant formulation, allowing simulations in various coordinate geometries; (2) a tensor artificial viscosity; (3) a new, more accurate MHD algorithm (MOCCT) combining the Constrained Transport algorithm (Evans & Hawley 1988) with a Method Of Characteristics treatment for Alfvén waves; and (4) a variable tensor Eddington factor (VTEF) solution for the equations of radiation hydrodynamics (RHD). The VTEF radiation module was described in Stone et al. (1992b) but not included in the version of ZEUS-2D offered for public distribution. An implicit FLD-based RHD module was publicly distributed with ZEUS-2D but never documented in a published report. (A draft of a technical report describing the 2-D FLD module is available on the World Wide Web[8].) The VTEF module described in Stone et al. (1992b) was later modified to incorporate a different matter-radiation coupling scheme, Eddington tensors computed from a time-dependent solution to the radiation transfer equation, and parallelization over two space dimensions and one angle cosine. This new VTEF code was coupled to an early version of ZEUS-MP and used to compare VTEF solutions to FLD results for test problems featuring strongly beamed radiation fields. This work was published by Hayes & Norman (2003), but as before the VTEF module remained reserved for private use. ZEUS-MP as described herein provides an updated FLD module designed for all dimensionalities and geometries in a parallel computing environment. This paper documents our new module at a level of detail which should aid users of both ZEUS-MP and the public ZEUS-2D code.

The second track resulted in the release of ZEUS-3D, the first ZEUS code capable of three-dimensional simulations. Written for the Cray-2 supercomputer by David Clarke, ZEUS-3D physics options included hydrodynamics, MHD, self-gravity, and optically thin radiative cooling. ZEUS-3D was the first ZEUS code with parallel capability, accomplished using Cray Autotasking compiler directives. The ZEUS-3D MHD module differed from that described in the ZEUS-2D papers with regard to both dimensionality and method: the MOC treatment of Alfvén waves was modified to incorporate the improvements introduced by John Hawley and James Stone (Hawley & Stone 1995). This modified "HSMOCCT" method is the basis for the MHD module adopted in ZEUS-MP.

Roughly speaking, ZEUS-MP encapsulates a qualitative union of the methods provided by the public ZEUS-2D and ZEUS-3D codes and enlarges upon them with new capabilities and solution techniques. The first public release of ZEUS-MP included HD, MHD, and self gravity, but was written exclusively for 3-D simulations, excluding at one stroke a long menu of worthy 2-D research applications and erecting an inconvenient barrier to expedient code testing. The new version we describe offers a substantially extended menu of physics, algorithm, and dimensionality options. The HD, MHD, and RHD modules accommodate simulations in 1, 1.5, 2, 2.5, and 3 dimensions. Arbitrary equations of state are supported; gamma-law and isothermal equations of state are provided. ZEUS-MP is the first ZEUS code to allow multispecies fluids to be treated; this is achieved with the addition of a concentration field array dimensioned for an arbitrary number of fluids. An implicit flux-limited photon diffusion module is included for RHD problems. As noted previously, the implicit FLD solution is based upon the method adopted for the public version of ZEUS-2D but has been extended to three dimensions. In addition, the FLD module sits atop a scalable linear system solver using the *conjugate gradient* (CG) method (Barret et al. 1994). Supplementing the FLD driver is an opacity module offering simple density and temperature-dependent power-law expressions for absorption and scattering coefficients; additional user-supplied modules (such as tabular data sets) are easily accommodated. Self-gravity is included in several ways: (1) spherical gravity (GM/r) is adopted for one-dimensional problems and may also be used in two dimensions; (2) two parallel Poisson solvers are included for problems with Neumann or Dirichlet boundary conditions in Cartesian or curvilinear coordinates, and (3) a fast Fourier Transform (FFTw) package is provided for problems with triply-periodic boundaries. In addition to self-gravity, a simple point-mass external potential may be imposed in spherical geometry.

While the ZEUS code line has evolved significantly, this process has not occurred in a vacuum. Indeed, the past decade has seen the emergence of several new MHD codes based upon Godunov methods (Ryu et al. 1998; Londrillo & Del Zanna 2000; Londrillo & del Zanna 2004; Balsara 2004; Gardiner & Stone 2005). Godunov-type schemes are accurate to second order in both space and time, are automatically conservative, and possess superior capability for resolving shock fronts when compared to ZEUS-type schemes at identical resolution. These advances, coupled with ZEUS's lower-order rate of convergence with numerical resolution (Stone et al. 1992a), and further with recent observations (Falle 2002) of vulnerabilities in the ZEUS-2D implementation of MOCCT might lead one to ask whether a new ZEUS code has a strong potential for future contributions of significance in areas of current astrophysical interest. We argue this point in the affirmative on two fronts. While we acknowledge that the ZEUS method possesses (as do all numerical methods) weaknesses which bound its scope of applicability, we note the encouraging comparisons (Stone & Gardiner 2005) of simulation results computed with Athena, an unsplit MHD code coupling PPM hydrodynamics to Constrained Transport, to ZEUS results in problems of supersonic MHD turbulence in 3D, and in shearing-box simulations of the magneto-rotational instability in two and three dimensions. The authors confirm the reliability of the ZEUS results and note that the numerical dissipation in Athena is equivalent to that of ZEUS at a

---

[8] http://cosmos.ucsd.edu/lca-publications/LCA013/ index.html

grid resolution of about 1.5 along each axis. In a similar vein, we examine a standard 2-D MHD test problem, due to Orszag & Tang (1979), in which the results from ZEUS-MP are found to compare quite favorably with those from the TVD code described in Ryu et al. (1998) and with results from different upwind Godunov codes presented by Dai & Woodward (1998) and Londrillo & Del Zanna (2000).

A second cause for optimism of ZEUS's future arises from the versatility inherent in ZEUS-MP's design. As this paper demonstrates, a wide variety of physics modules are easily implemented within ZEUS-MP's design framework. Additionally, different solution techniques for treating the same physics (self-gravity is an excellent example) are almost trivially accommodated. The theme of this paper is, therefore: physics, flexibility, and parallel performance. To demonstrate these traits we organize the paper as follows: the presentation begins by writing the fluid equations solved by ZEUS-MP in section 2; section 3 provides a largely descriptive overview of the various numerical methods used to solve these equations on a discrete mesh. Sections 4 and 5 present two groups of test problems. Section 4 provides a suite of tests to verify the correctness of the various physics modules. Section 5 examines a quartet of 3-D problems which measure performance on a massively parallel computer.

The main body of the paper is concluded with a summary in section 6. Full documentation of the finite-difference formulae describing hydrodynamic, magnetohydrodynamic, and radiation hydrodynamic evolution in three dimensions is provided in a series of appendices. These appendices are preceded by a tabular "code map" (Table A1) in appendix A associating the discrete equations written in appendices B-E with the subroutines in ZEUS-MP that compute them. We offer this as an aid to users wishing to familiarize themselves with the code and to code developers who desire to implement their own versions or improve upon what is currently offered in ZEUS-MP. The HD, MHD, and FLD modules are detailed in appendices B, C, and D, respectively. Appendix E documents the 3-D linearized form of Poisson's equation expressed in covariant grid coordinates, and details of our parallel implementation techniques and strategies are given in appendix F.

2. THE PHYSICAL EQUATIONS

Our description of the physical state of a fluid element is specified by the following set of fluid equations relating the mass density (ρ), velocity ($\mathbf{v}$), gas internal energy density (e), radiation energy density (E), radiation flux ($\mathbf{F}$), and magnetic field strength ($\mathbf{B}$):

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{v} = 0; \tag{1}$$

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p + \left(\frac{\chi}{c}\right)\mathbf{F} + \frac{1}{4\pi}\left(\nabla\times\mathbf{B}\right)\times\mathbf{B} - \rho\nabla\Phi; \tag{2}$$

$$\rho\frac{D}{Dt}\left(\frac{e}{\rho}\right) + p\nabla\cdot\mathbf{v} = c\kappa_E E - 4\pi\kappa_P B_p; \tag{3}$$

$$\rho\frac{D}{Dt}\left(\frac{E}{\rho}\right) + \nabla\cdot\mathbf{F} + \nabla\mathbf{v}:\mathbf{P} = 4\pi\kappa_P B_p - c\kappa_E E; \tag{4}$$

$$\frac{\partial\mathbf{B}}{\partial t} = \nabla\times\left(\mathbf{v}\times\mathbf{B}\right). \tag{5}$$

The Lagrangean (or comoving) derivative is given by the usual definition:

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla \tag{6}$$

The four terms on the RHS of the gas momentum equation (2) denote forces to due thermal pressure gradients, radiation stress, magnetic Lorentz acceleration, and the gravitational potential, respectively. The RHS of (3) gives source/sink terms due to absorption/emission of radiant energy. Each term serves an inverse role on the RHS of (4). In (3), B_p denotes the Planck function:

$$B_p = \frac{\sigma}{\pi}T^4, \tag{7}$$

where T is the local material temperature. Equations (3) and (4) are also functions of flux-mean, Planck-mean, and energy-mean opacities, which are formally defined as

$$\chi = \frac{1}{\mathbf{F}}\int_0^\infty \chi(\nu)\mathbf{F}(\nu)d\nu, \tag{8}$$

$$\kappa_P = \frac{1}{B_p}\int_0^\infty \chi(\nu)B_p(\nu)d\nu, \tag{9}$$

$$\kappa_E = \frac{1}{E}\int_0^\infty \chi(\nu)E(\nu)d\nu. \tag{10}$$

In the simple problems we discuss in this paper, the three opacities are computed from a single expression, differing only in that κ_E and κ_P are defined at zone centers and χ is computed at zone faces. In general, however, independent expressions or data sets for the three opacities are trivially accommodated in our discrete representation of the RHD equations.

We compute the radiation flux, $\mathbf{F}$, according to the diffusion approximation:

$$\mathbf{F} = -\left(\frac{c\Lambda_E}{\chi}\right)\nabla E, \tag{11}$$

where we have introduced a flux-limiter (Λ_E) to ensure a radiation propagation speed that remains bounded by the speed of light (c) in transparent media. Attenuation of the flux is regulated by the total extinction coefficient, χ, which in general may include contributions from both absorption and scattering processes.

Equation (4) also includes a term involving the radiation stress tensor, $\mathbf{P}$. In general, $\mathbf{P}$ is not known *a priori* and must be computed from a solution to the radiative transfer equation. In the VTEF methods described in Stone et al. (1992b) and Hayes & Norman (2003), $\mathbf{P}$ is written as the product of the (known) radiation energy, E, and an (unknown) tensor quantity, $\mathbf{f}$:

$$\mathbf{P} = \mathbf{f}E. \tag{12}$$

A solution for $\mathbf{f}$ may be derived from a formal solution to the radiative transfer equation (Mihalas 1978) or may be approximated analytically. For the RHD implementation in ZEUS-MP, we follow the latter approach, adopting the expression for $\mathbf{f}$ given by equation (13) in Turner & Stone (2001):

$$\mathbf{f} = \frac{1}{2}\left(1-f\right)\mathbf{I} + \frac{1}{2}\left(3f-1\right)\hat{\mathbf{n}}\hat{\mathbf{n}}, \tag{13}$$

where $\hat{\mathbf{n}} = \nabla\mathrm{E}/ \mid \nabla\mathrm{E} \mid$, $\mathbf{I}$ is the unit tensor, and f is a scalar "Eddington factor" expressed as a function of the flux limiter, Λ_E, and E as

$$f \;=\; \Lambda_\mathrm{E} \;+\; \left(\frac{\Lambda_\mathrm{E} \mid \nabla\mathrm{E} \mid}{\chi\mathrm{E}} \right)^2 \qquad (14)$$

The RHD equations are accurate only to order unity in $\mathbf{v}/c$ (Mihalas & Mihalas 1984), consistent with the radiation modules described in Stone et al. (1992b), Turner & Stone (2001), and Hayes & Norman (2003). The assumption of local thermodynamic equilibrium is reflected in our adoption of the Planck function in equations (3) and (4) evaluated at the local material temperature; our use of the FLD approximation gives rise to equation (11) and is discussed further in § 3.3.

Evolution of the magnetic field (5) is constrained by the assumption of ideal MHD. We are therefore, in common with previous public ZEUS codes, assuming wave modes which involve fluid motions but no charge separation. This equation for $\mathbf{B}$ also assumes zero resistivity, a valid approximation in astrophysical environments where the degree of ionization is sufficiently high to ensure a rapid collision rate between charged and neutral particles and thus strong coupling between the two. There exist astrophysical environments where this assumption is expected to break down (e.g. the cores of cold, dense molecular clouds), in which case an algorithm able to distinguish the dynamics of ionized and neutral particles is required. Stone (1999) published extensions to the ZEUS MHD algorithm to treat nonideal phenomena such as Ohmic dissipation and partially ionized plasmas where the ionic and neutral components of the fluid are weakly coupled via a collisional drag term. Incorporation of these algorithmic extensions into ZEUS-MP is currently left as an exercise to the interested user, but may be undertaken for a future public release.

The gravitational potential Φ appearing in (2) is computed from a solution to Poisson's equation:

$$\nabla^2\Phi \;=\; 4\pi\mathrm{G}\rho. \qquad (15)$$

Our various techniques for solving (15) are described in §3.4; the linear system which arises from discretizing (15) on a covariant coordinate mesh is derived in appendix E.

Our fluid equations are closed with an equation of state (EOS) expressing the thermal pressure as a function of the internal gas energy. The dynamic test problems considered in this paper adopt a simple ideal EOS with $\gamma = 5/3$ except where noted.

3. NUMERICAL METHODS: AN OVERVIEW

Figure 1 summarizes the dependencies of ZEUS-MP's major physics and I/O modules on underlying software libraries. The *Message Passing Interface* (MPI) software library is used to implement parallelism in ZEUS-MP and lies at the foundation of the code. This library is available on all NSF and DOE supercomputing facilities and may be freely downloaded[9] for installation on small private clusters. ZEUS-MP's linear system solvers and I/O drivers access MPI functions directly and act in service of the top layer of physics modules, which are described in the following subsections.
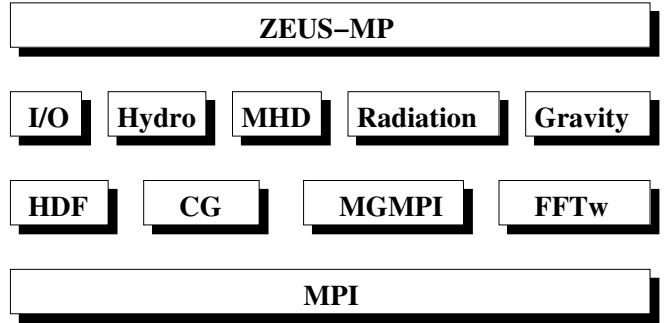
[9] http://www-unix.mcs.anl.gov/mpi/mpich/

FIG. 1.— Software implementation within ZEUS-MP.

3.1. *Hydrodynamics*

Equations (1) through (5) provide the most complete physical description which may be invoked by ZEUS-MP to characterize a problem of interest. There exist classes of problems, however, for which either radiation or magnetic fields (or both) are not relevant and thus may be ignored. In such circumstances, ZEUS-MP evolves an appropriately reduced subset of field variables and solves only those equations needed to close the system. ZEUS-MP may therefore define systems of equations for pure HD, MHD, RHD, or RMHD problems as necessary. We begin the description of our numerical methods by considering purely hydrodynamic problems in this section; we introduce additional equations as we expand the discussion of physical processes.

Tracing their ancestry back to a two-dimensional, Eulerian hydrodynamics (HD) code for simulations of rotating protostellar collapse (Norman et al. 1980), all ZEUS codes are rooted in an HD algorithm based upon the method of finite differences on a staggered mesh (Norman 1980; Norman & Winkler 1986), which incorporates a second order-accurate, monotonic advection scheme (van Leer 1977). The basic elements of the ZEUS scheme arise from consideration of how the evolution of a fluid element may be properly described on an *adaptive mesh* whose grid lines move with arbitrary velocity, $\mathbf{v_g}$. Following the analysis in Winkler et al. (1984), we identify three pertinent time derivatives in an adaptive coordinate system: (1) the Eulerian time derivative ($\partial/\partial t$), taken with respect to coordinates fixed in the laboratory frame, (2) the Lagrangean derivative (D/Dt; cf. equation 6), taken with respect to a definite fluid element, and (3) the adaptive-mesh derivative (d/dt), taken with respect to fixed values of the adaptive mesh coordinates. Identifying dV as a volume element bounded by fixed values of the adaptive mesh and $d\mathbf{S}$ as the surface bounding this element, one may employ the formalism of Winkler et al. (1984) to split the fluid equations into two distinct solution steps: the *source step*, in which we solve

$$\rho\frac{\partial\mathbf{v}}{\partial t} = -\nabla\mathrm{p} \;-\; \nabla\cdot\mathbf{Q} \;-\; \rho\nabla\Phi; \qquad (16)$$

$$\frac{\partial\mathrm{e}}{\partial t} = -\mathrm{p}\nabla\cdot\mathbf{v} \;-\; \nabla\mathbf{v} : \mathbf{Q}; \qquad (17)$$

and the *transport step*, whence

$$\frac{d}{dt}\int_V \rho\,dV = -\oint_{dV} \rho\,(\mathbf{v}-\mathbf{v_g})\cdot d\mathbf{S}; \qquad (18)$$

$$\frac{d}{dt}\int_V \rho\mathbf{v}\,dV = -\oint_{dV} \rho\mathbf{v}\,(\mathbf{v}-\mathbf{v_g})\cdot d\mathbf{S}; \qquad (19)$$

$$\frac{d}{dt}\int_V e\,dV = -\oint_{dV} e\,(\mathbf{v}-\mathbf{v_g})\cdot d\mathbf{S}; \qquad (20)$$

where $\mathbf{v_g}$ is the local grid velocity. Equations (16) and (17) have been further modified to include an artificial viscous pressure, $\mathbf{Q}$. ZEUS-MP employs the method due to von Neumann & Richtmyer (1950) to apply viscous pressure at shock fronts. This approach is known to provide spurious viscous heating in convergent coordinate geometries even when no material compression is present. Tscharnuter & Winkler (1979) describe a tensor formalism for artificial viscosity which avoids this problem; an implementation of this method will be provided in a future release of ZEUS-MP. For problems involving very strong shocks and stagnated flows, the artificial viscosity may be augmented with an additional term which is linear in velocity and depends upon the local sound speed. The precise forms of the quadratic and linear viscosity terms are documented in Appendix B.

3.2. MHD

The treatment of MHD waves in ZEUS-MP is by necessity more complex than that for HD waves because MHD waves fall into two distinct families: (1) longitudinal, compressive (fast and slow magnetosonic); and (2) transverse, non-compressive (Alfvén) waves. The former family may be treated in the source-step portion of the ZEUS solution in a similar fashion to their hydrodynamic analogs, but the latter wave family couples directly to the magnetic induction equation and therefore requires a more complex treatment. From the algorithmic perspective, the inclusion of MHD into the ZEUS scheme has two consequences: (1) fluid accelerations due to *compressive* MHD waves introduce additional terms in equation (16); (2) fluid accelerations due to *transverse* MHD introduce additional velocity acceleration terms which, owing to the coupling to the induction equation, are computed in a separate step which follows the source step update but precedes the "transport" (i.e. fluid advection) update. In this way, the updates due to fluid advection are deferred until all updates to the velocities are properly recorded. As will be extensively described in this section, the fluid accelerations due to transverse MHD waves are combined with the evolution equation for $\mathbf{B}$ because of the tight mathematical coupling between the two solutions.

We guide the following discussion by providing, in the continuum limit, the final result. With the inclusion of MHD, the source/transport solution sequence expands to begin with an MHD-augmented "source" step:

$$\rho\frac{\partial\mathbf{v}}{\partial t} = -\nabla p - \nabla(B^2/8\pi) - \nabla\cdot\mathbf{Q} - \rho\nabla\Phi; \quad (21)$$

$$\frac{\partial e}{\partial t} = -p\nabla\cdot\mathbf{v} - \nabla\mathbf{v}:\mathbf{Q}; \qquad (22)$$

This is followed by an MOCCT step, whence

$$\rho\frac{\partial\mathbf{v}}{\partial t}\bigg|_{\text{final}} = \rho\frac{\partial\mathbf{v}}{\partial t}\bigg|_{\text{source step}} + \frac{1}{4\pi}(\mathbf{B}\cdot\nabla)\mathbf{B}; \quad (23)$$

$$\frac{d}{dt}\int_S \mathbf{B}\cdot d\mathbf{S} = \vec{\epsilon}\cdot d\mathbf{l}, \qquad (24)$$

where $\vec{\epsilon}$ is the electromotive force (EMF) and is given by

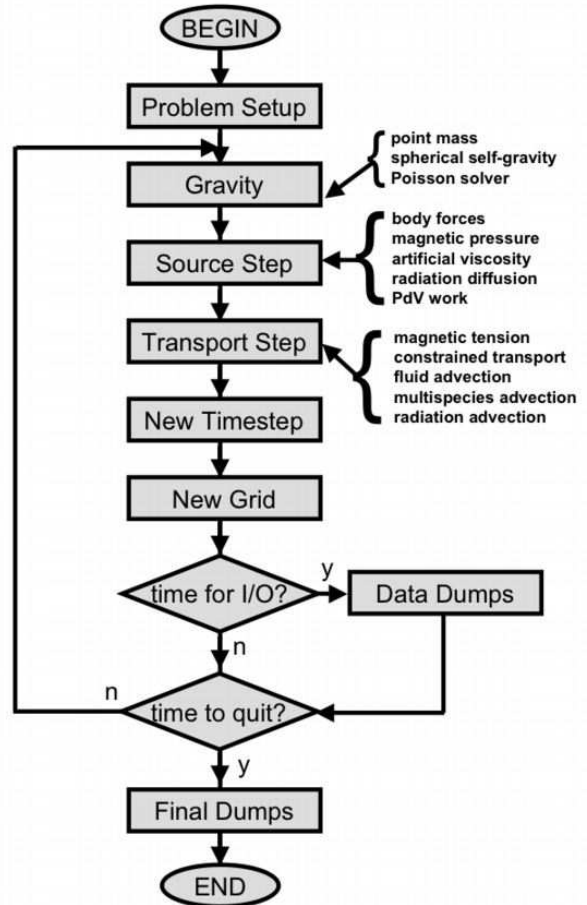$$\vec{\epsilon} = (\mathbf{v}-\mathbf{v_g})\times\mathbf{B}. \qquad (25)$$

FIG. 2.— Program control for ZEUS-MP.

With velocities and $\mathbf{B}$ fields fully updated, we then proceed to the "transport" step as written in equations (18) through (20).

Figure 2 shows the program control logic used to implement the solution outlined by equations (21)-(24) and the previously-written expressions (18)-(20). Once the problem has been initialized, Poisson's equation is solved to compute the gravitational potential. In the source step, updates due to longitudinal forces (thermal and magnetic), radiation stress, artificial viscosity, energy exchange between the gas and radiation field, and pdV work are performed at fixed values of the coordinates. Accelerations due to transverse MHD waves are then computed and magnetic field components are updated. Finally, velocities updated via (21) and (23) are used to advect field variables through the moving mesh in the transport step. Advection is performed in a series of directional sweeps which are cyclically permuted at each time step.

The remainder of this section serves to derive the new expressions appearing in (21), (23), and (24), and to document their solution. We begin by first considering the Lorentz acceleration term in the gas momentum equation (2). Using the vector identity

$$(\nabla\times\mathbf{B})\times\mathbf{B} = -\nabla\left(B^2/2\right) + (\mathbf{B}\cdot\nabla)\mathbf{B} \qquad (26)$$

we expand the Lorentz acceleration term such that (2) becomes (ignoring radiation)

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla\mathrm{p} \; - \; \nabla(B^2/8\pi) \; + \; \frac{1}{4\pi}\left(\mathbf{B}\cdot\nabla\right)\mathbf{B}$$
$$- \rho\nabla\Phi. \tag{27}$$

The second term on the RHS of (27) is the gradient of the magnetic pressure. This term, which provides the contribution from the compressive magnetosonic waves, is clearly a longitudinal force and is differenced in space and time identically to the thermal pressure term. This expression is thus evaluated simultaneously with the other contributions to the "source step" portion of the momentum equation (equation 21); contributions from the magnetic pressure to the discrete representation of (21) along each axis are shown in equations (B10) through (B12), with notational definitions provided by expressions (B19) through (B30), in appendix B.

The third term on the RHS of (27) represents magnetic tension in curved field lines and is transverse to the gradient of **B**. This term, which is added to the solution sequence as equation (23) couples to the magnetic induction equation to produce Alfvén waves; the magnetic tension force and the induction equation (5) are therefore solved by a single procedure: the Method of Characteristics + Constrained Transport (MOCCT).

ZEUS-MP employs the MOCCT method of Hawley & Stone (1995), which is a generalization of the algorithm described in Stone & Norman (1992b) to 3D, with some slight modifications that improve numerical stability. To describe MOCCT, we first derive the moving frame induction equation. Recall that equation (5) is derived from Faraday's law

$$\frac{\partial\mathbf{B}}{\partial t} \;=\; -c\nabla\times\mathbf{E}, \tag{28}$$

where **E**, **B** and the time derivative are measured in the Eulerian frame. The electric field **E** is specified from Ohm's law

$$c\mathbf{E} = -\mathbf{v}\times\mathbf{B} + \mathbf{J}/\sigma. \tag{29}$$

Equation (5) results when we substitute equation 29 into equation 28, and let the conductivity $\sigma \rightarrow \infty$. Integrating equation 28 over a moving surface element $S(t)$ bounded by a moving circuit $C(t)$, the general form of Faraday's law is

$$\frac{d}{dt}\int_S \mathbf{B}\cdot d\mathbf{S} = -c\oint_C \mathbf{E}'\cdot d\mathbf{l} \tag{30}$$

where $\mathbf{E}'$ is the electric field measured in the moving frame. To first order in v/c, $\mathbf{E}' = \mathbf{E} + (\mathbf{v_g}\times\mathbf{B})/c$. From equation 29, for a perfectly conducting fluid $\mathbf{E} = -\mathbf{v}\times\mathbf{B}/c$. Combining these two results and substituting into equation 30, we get

$$\frac{d}{dt}\int_S \mathbf{B}\cdot d\mathbf{S} = \oint_C (\mathbf{v} - \mathbf{v_g})\times\mathbf{B}\cdot d\mathbf{l}. \tag{31}$$

Equation (31) states that the time rate of change of the magnetic flux piercing S

$$\phi_S = \int_S \mathbf{B}\cdot d\mathbf{S} \tag{32}$$

is given by the line integral of the electromotive force (EMF) $\epsilon = (\mathbf{v} - \mathbf{v_g})\times\mathbf{B}$ along C:

$$\frac{d\phi_S}{dt} = \oint_C \epsilon\cdot d\mathbf{l}. \tag{33}$$
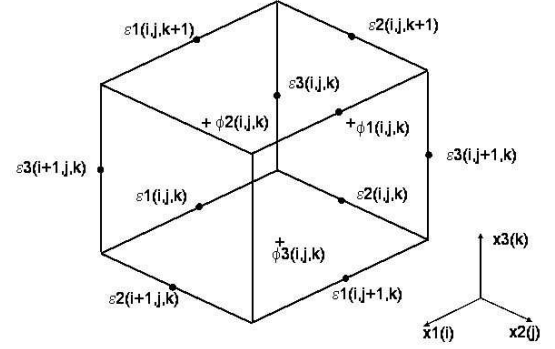

FIG. 3.— Centering of magnetic field variables in ZEUS-MP.

Equation (33), using (32), is equivalent to expression (24) appearing in our grand solution outline, and it forms, along with equation (23), the target for our MOCCT algorithm. Equation (33) is familiar from standard texts on electrodynamics, only now S and C are moving with respect to the Eulerian frame. If $\mathbf{v_g} = \mathbf{v}$, we recover the well known flux-freezing result, $d\phi_S/dt = D\phi_S/Dt = 0$.

As discussed in Evans & Hawley (1988); Stone & Norman (1992b); Hawley & Stone (1995), equation (33) is in the form which guarantees divergence-free magnetic field transport when finite differenced, provided the EMFs are evaluated once and only once per time step. Referring to the unit cell shown in Figure 3, we can write the discrete form of equation (33) as

$$\frac{\phi 1_{i,j,k}^{n+1} - \phi 1_{i,j,k}^{n}}{\Delta t} = \epsilon 2_{i,j,k}\Delta x2_{i,j} + \epsilon 3_{i,j+1,k}\Delta x3_{i,j+1,k}$$
$$- \epsilon 2_{i,j,k+1}\Delta x2_{i,j} - \epsilon 3_{i,j,k}\Delta x3_{i,j,k}; \tag{34}$$

$$\frac{\phi 2_{i,j,k}^{n+1} - \phi 2_{i,j,k}^{n}}{\Delta t} = \epsilon 1_{i,j,k+1}\Delta x1_i + \epsilon 3_{i,j+1,k}\Delta x3_{i,j,k}$$
$$- \epsilon 1_{i,j,k}\Delta x1_i - \epsilon 3_{i+1,j,k}\Delta x3_{i+1,j,k}; \tag{35}$$

$$\frac{\phi 3_{i,j,k}^{n+1} - \phi 3_{i,j,k}^{n}}{\Delta t} = \epsilon 1_{i,j,k}\Delta x1_i + \epsilon 2_{i+1,j,k}\Delta x2_{i+1,j}$$
$$- \epsilon 1_{i,j+1,k}\Delta x1_i - \epsilon 2_{i,j,k}\Delta x2_{i,j}, \tag{36}$$

where $\phi 1, \phi 2, \phi 3$ are the face-centered magnetic fluxes piercing the cell faces whose unit normals are in the $\mathbf{n_1}, \mathbf{n_2}, \mathbf{n_3}$ directions, respectively, $\epsilon 1, \epsilon 2, \epsilon 3$ are components of the edge-centered EMFs, and $\Delta x1, \Delta x2, \Delta x3$ are the coordinate distances of the cell edges. The peculiar subscripting of these line elements is made clear in Appendix C. It is easy to see that for any choice of EMFs, $\mathbf{B}^{n+1}$ will be divergence-free provided $\mathbf{B}^n$ is divergence-free. By Gauss's theorem, $\int_V \nabla\cdot\mathbf{B}dV = \oint_S \mathbf{B}\cdot d\mathbf{S} = 0$, where the second equality follows from the fact $\nabla\cdot\mathbf{B} = 0$. Analytically, the time derivative of $\nabla\cdot\mathbf{B}$ is also zero. Numerically,

$$\frac{d}{dt}\int_V \nabla\cdot\mathbf{B}dV = \frac{d}{dt}\oint_S \mathbf{B}\cdot d\mathbf{S} \approx \sum_{faces=1}^{6} d\phi/dt$$

$$= \sum_{faces=1}^{6}\sum_{edges=1}^{4} \epsilon\cdot d\mathbf{l} = 0. \tag{37}$$

The last equality results from the fact that when summing over all the faces and edges of a cell, each EMF appears twice with a change of sign, and thus cancel in pairs.

In principle, one could use *any* method to compute the EMF within the CT formalism and still maintain divergence-free fields. In practice, a method must be used which stably and accurately propagates both MHD wave types: longitudinal, compressive (fast and slow magnetosonic) waves, and the transverse, non-compressive (Alfvén) waves. As noted previously, the first wave type is straightforwardly incorporated into the treatment of the compressive hydrodynamic waves; the real difficulty arises in the treatment of the Alfvén waves.

In ideal MHD, Alfvén waves can exhibit discontinuities (rotational, transverse) at current sheets. Unlike hydrodynamical shocks, these structures are not dissipative, which rules out the use of dissipative numerical algorithms to model them. In addition, Alfvén waves tightly couple the evolution equations for the velocity and magnetic field components perpendicular to the direction of propagation. This rules out operator operator splitting these components. Finally, we need an algorithm that can be combined with CT to give both divergence-free transport of fields and correct local dynamics. This will be achieved if the EMFs used in the CT scheme contain information about all wave modes, which for stability, must be appropriately upwinded. These multiple requirements can be met using the Method of Characteristics (MOC) to compute the EMFs. The resulting hybrid scheme is MOCCT (Stone & Norman 1992b; Hawley & Stone 1995).

Schematically, the EMFs can be written as (ignoring $\mathbf{v_g}$ for simplicity)

$$\epsilon 1_{i,j,k} = v2^*_{i,j,k}\, b3^*_{i,j,k} - v3^*_{i,j,k}\, b2^*_{i,j,k} \qquad (38)$$

$$\epsilon 2_{i,j,k} = v3^*_{i,j,k}\, b1^*_{i,j,k} - v1^*_{i,j,k}\, b3^*_{i,j,k} \qquad (39)$$

$$\epsilon 3_{i,j,k} = v1^*_{i,j,k}\, b2^*_{i,j,k} - v2^*_{i,j,k}\, b1^*_{i,j,k} \qquad (40)$$

where the starred quantities represent time-centered values for these variables resulting from the solution of the characteristic equations at the centers of zone edges where the EMFs are located. To simplify, we apply MOC to the Alfvén waves only, as the longitudinal modes are adequately handled in a previous step by finite difference methods.

Because the MOC is applied only to transverse waves, we may derive the appropriate differential equations by considering the 1-D MHD wave equations for an incompressible fluid (Stone & Norman 1992b) which reduce to

$$\frac{\partial v}{\partial t} = \frac{B_x}{\rho}\frac{\partial B}{\partial x} - \frac{\partial}{\partial x}(v_x v), \qquad (41)$$

$$\frac{\partial B}{\partial t} = B_x \frac{\partial v}{\partial x} - \frac{\partial}{\partial x}(v_x B), \qquad (42)$$

where we have used the divergence-free constraint in one dimension (which implies $\partial B_x/\partial x \equiv 0$) and the non-compressive nature of Alfvén waves (which implies $\partial v_x/\partial x \equiv 0$).

We can rewrite the coupled equations (41) and (42) in characteristic form by multiplying equation (42) by $\rho^{-1/2}$ and then adding and subtracting them, yielding

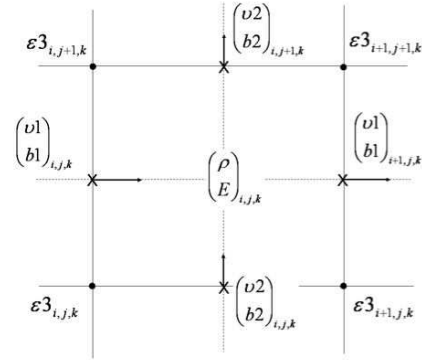$$\frac{\mathcal{D}v}{\mathcal{D}t} \pm \frac{1}{\rho^{1/2}}\frac{\mathcal{D}B}{\mathcal{D}t} = 0. \qquad (43)$$

FIG. 4.— A two dimensional $(x1-x2)$ slice through the unit cell in Figure 3 containing the four $\epsilon 3$'s. The computation of $\epsilon 3_{i,j,k}$ is illustrated in Figure 5.

The plus sign denotes the characteristic equation along the forward facing characteristic C^+, while the minus sign denotes the characteristic equation along the backward facing characteristic C^-. The comoving derivative used in equation (43) is defined as

$$\mathcal{D}/\mathcal{D}t \equiv \partial/\partial t + (v_x \mp B_x/\rho^{1/2})\partial/\partial x, \qquad (44)$$

where the minus (plus) sign is taken for the comoving derivative along the $C^+(C^-)$ characteristic. Note that the coefficient of the second term in equation (44) is just the Alfvén velocity in the moving fluid, $v_x \pm v_A$. Physically, equations (43) state that along characteristics, which are straight lines in spacetime with slopes $v_x \pm v_A$, the changes in the velocity and magnetic field components in each direction are not independent.

The finite-difference equations used to solve the characteristic equations (43) can be generated as follows. Consider the one dimensional space-time diagram centered at the position of one of twelve edge-centered EMFs where we require the values v^*_i, B^*_i (see Figure 5). Extrapolating back in time along the characteristics C^+ and C^- to time level n defines the "footpoints". By using upwind van Leer (1977) interpolation, we can compute the time-averaged values for these variables in each domain of dependence. For both the velocities and the magnetic fields the characteristic speed $v_x \pm v_A$ are used to compute the footpoint values $v^{+,n}_i, B^{+,n}_i, v^{-,n}_i, B^{-,n}_i$. The finite difference equations along C^+ and C^- become

$$(v^*_i - v^{+,n}_i) + (B^*_i - B^{+,n}_i)/(\rho^+_i)^{1/2} = 0, \qquad (45)$$

$$(v^*_i - v^{-,n}_i) - (B^*_i - B^{-,n}_i)/(\rho^-_i)^{1/2} = 0, \qquad (46)$$

where the subscript i refers to cell i, not the i-th component of the vectors $\mathbf{v}, \mathbf{B}$. For simplicity, we set $\rho^+_i = \rho^n_{i-1}$ and $\rho^-_i = \rho^n_i$. The two linear equations for the two unknowns v^*_i and B^*_i are then solved algebraically.

For our multidimensional calculations, the characteristic equations are solved in a directionally split fashion on planes passing through the center of the cell and the cell edges where the EMFs are to be evaluated. To illustrate, consider the calculation of $\epsilon 3_{i,j,k}$ (Eq. 40). The plane parallel to the $x1 - x2$ plane passing through the four $\epsilon 3$'s in Figure 3 is shown in Figure 4. Evaluating
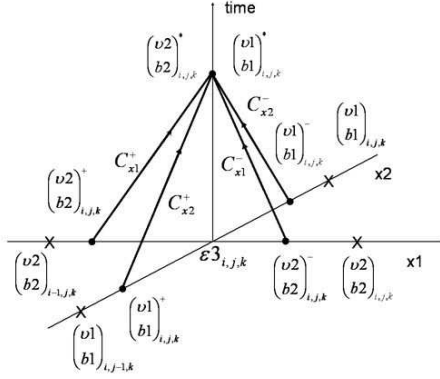
FIG. 5.— The computation of $\epsilon3_{i,j,k}$ involves the solution of two 1-D characteristic equations for Alfvén waves confined to the $(x1-x2)$ plane. The space-time diagrams for the solution of $b2^*, v2^*$ due to Alfvén waves propagating parallel to the $x1$ axis, and for the solution of $b1^*, v1^*$ due to Alfvén waves propagating parallel to the $x2$ axis are displayed.

$\epsilon3_{i,j,k}$ requires values for $b1^*, v1^*, b2^*$, and $v2^*$ at the cell corner $(x1_i, x2_j, x3_k)$. First, as outlined above, one computes $b2^*, v2^*$ by solving the characteristic equations for Alfvén waves propagating in the $x1$ direction (Figure 5). When calculating the location of the footpoints, $v1$ and the Alfvén speed are averaged to the zone corner. Then, the procedure is repeated for $b1^*, v1^*$ by solving the characteristic equations for Alfvén waves propagating in the $x2$ direction, using $v2$ and the Alfvén speed averaged to the zone corner. Once all the $\epsilon3$'s are evaluated in this way, the analogous procedure is followed for slices containing $\epsilon1$ and $\epsilon2$. Only after all the EMFs have been evaluated over the entire grid can the induction equation equation be updated in a divergence-free fashion.

Finally, we consider how the fluid momentum is updated due to the Lorentz Force. The key point is that we do not want to throw away the fluid accelerations arising from Alfvén waves that are implicit in the solution to the characteristic equations (45) and (46). For example, the acceleration of $v3$ by the transverse magnetic forces is given by

$$\rho \frac{\partial v3}{\partial t} = -\nabla_{(3)}(B^2/8\pi) + \frac{1}{4\pi}(\mathbf{B} \cdot \nabla)B_3, \quad (47)$$

or in terms of the EMFs:

$$\rho_{i,j,k}\left(\frac{v3^{n+1}_{i,j,k} - v3^n_{i,j,k}}{\Delta t}\right) = \frac{1}{4\pi}[\overline{b1}(\frac{b3^*_{\epsilon2(i+1,j,k)} - b3^*_{\epsilon2(i,j,k)}}{\Delta x1})$$
$$+ \overline{b2}(\frac{b3^*_{\epsilon1(i,j+1,k)} - b3^*_{\epsilon1(i,j,k)}}{\Delta x2})](48)$$

where $\overline{b1}$ and $\overline{b2}$ are four-point averages of the magnetic field to the spatial location of $v3$, and the $b3^*$'s are those values that enter into the EMF's referred to in the subscripts (see Figure 3.) Similarly, the magnetic pressure calculation is

$$\rho_{i,j,k}\left(\frac{v3^{n+1}_{i,j,k} - v3^n_{i,j,k}}{\Delta t}\right) = -\frac{1}{4\pi}[\overline{b1}(\frac{\Delta b1}{\Delta x3}) + \overline{b2}(\frac{\Delta b2}{\Delta x3})]. \quad (49)$$

The evaluation of the EMF's outlined by equations (38) through (40) has been modified according to a prescription due to Hawley & Stone (1995), in which each of the two vB product terms is computed from a mix of quantities computed directly from the characteristic equations with quantities estimated from simple advection. Full details are provided in Appendix C, but the idea may illustrated by an appropriate rewrite of (40) for the evaluation of $\epsilon3_{i,j,k}$:

$$\epsilon3_{i,j,k} = 0.5 * \left(v1^*_{i,j,k}\ \overline{b2}_{i,j,k} + \overline{v1}_{i,j,k}\ b2^*_{i,j,k}\right)$$
$$- 0.5 * \left(v2^*_{i,j,k}\ \overline{b1}_{i,j,k} + \overline{v2}_{i,j,k}\ b1^*_{i,j,k}\right), \quad (50)$$

where the starred quantities are derived from characteristic solutions and the barred quantities arise from upwinded averages along the appropriate fluid velocity component. This modification (which engenders the "HS" in "HSMOCCT") introduces a measure of diffusivity into the propagation of Alfvén waves which is not present in the MOC scheme described in Stone & Norman (1992b). Hawley & Stone (1995) note that this change resulted in a more robust algorithm when applied to fully multidimensional problems characterized by strong magnetic discontinuities.

3.3. Radiation Diffusion

The inclusion of radiation in the system of fluid equations to be solved introduces four changes in the numerical solution. First, an additional contribution to the source-step momentum equation arises from the radiation flux:

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p - \nabla \cdot \mathbf{Q} + \left(\frac{\chi}{c}\right)\mathbf{F} - \rho \nabla \Phi; . \quad (51)$$

Second, new source/sink terms appear in the source-step gas energy equation:

$$\frac{\partial e}{\partial t} = c\kappa_E E - 4\pi\kappa_P B_p - p\nabla \cdot \mathbf{v} - \nabla \mathbf{v} : \mathbf{Q}. \quad (52)$$

Third, a new source-step equation is added for the radiation energy density:

$$\frac{\partial E}{\partial t} = 4\pi\kappa_P B_p - c\kappa_E E - \nabla \cdot \mathbf{F} - \nabla \mathbf{v} : \mathbf{P}; \quad (53)$$

and fourth, an additional advection equation for E is added to the transport step:

$$\frac{d}{dt}\int_V E\,dV = -\oint_{dV} E\,(\mathbf{v} - \mathbf{v_g}) \cdot d\mathbf{S}. \quad (54)$$

In comparing equations (51) - (53) with either the pure HD equations (16) - (17) or the MHD analogs (21) - (24), it is clear that the inclusion of radiation physics may be made to the HD or MHD systems of equations with equal ease. Similary, the transport step is augmented with a solution of (54) in either scenario.

ZEUS-MP computes the evolution of radiating fluid flows through an implicit solution to the coupled gas and radiation energy equations (52) and (53). Rather than solve the time-dependent radiation momentum equation and treat the flux, $\mathbf{F}$, as an additional dependent variable, we adopt the flux-limited diffusion (FLD) approximation as shown in (11). This allows an algebraic substitution for $\mathbf{F}$ in the flux-divergence term of the source-step equation (53) for E and avoids the need for an additional advection equation for the flux. The FLD approximation is an attractive choice for multidimensional RHD applications for which local heating/cooling approximations are

inadequate. With regard to computational expense, FLD offers enormous economy relative to exact Boltzmann solutions for the photon distribution function because the dimensionality of the problem is reduced by 2 when the angular variation of the radiation field is integrated away. Additionally, the mathematical structure of the FLD equations makes the solution amenable to parallel implementation. Fundamentally, however, the flux-limiter is a mathematical construction which interpolates between the limiting cases of transparency and extreme opacity in a manner that (hopefully) retains sufficient accuracy in the more difficult semi-transparent regime. Precisely what constitutes "sufficient accuracy" is dictated by the needs of the particular application, and the techniques for meeting that requirement may likewise depend upon the research problem. Levermore & Pomraning (1981) (LP) constructed an FLD theory which derived a form of Λ_E widely adopted in astrophysical applications (and in this paper). In their work, LP use simple test problems to check the accuracy of their FLD solution against exact transport solutions. In simulations of core-collapse supernovae, Liebendörfer et al. (2004) compared calculations employing energy-dependent multi-group FLD (MGFLD) calculations against those run with an exact Boltzmann solver and found alternate forms of the limiter which better treated the transport through the semi-transparent shocked material in the post-bounce environment. These calculations and others have shown that FLD or MGFLD techniques can yield solutions that compare favorably with exact transport, but that a "one size fits all" prescription for the flux limiter is not to be expected.

In the context of applications, two other vulnerabilities of FLD bear consideration. Hayes & Norman (2003) have compared FLD to VTEF solutions in problems characterized by highly anisotropic radiation fields. Because the FLD equation for flux is a function of the local gradient in E, radiation tends to flow in the direction of radiation energy gradients even when such behavior is not physically expected, as in cases where radiation from a distant source is shadowed by an opaque object. In applications where the directional dependence of the radiation is relevant (Turner et al. (2005) discuss a possible example), an FLD prescription may be of limited reliability. A second vulnerability concerns numerical resolution. As discussed in detail by Mihalas & Mihalas (1984), the diffusion equation must be flux-limited because numerically, the discrete diffusion equation advances a radiation wave one mean-free path (λ) per time step. Flux limiters are designed to act when $\lambda/\Delta t$ exceeds c, which ordinarily one expects in a transparent medium. A problem can arise in extremely opaque media however, when the physical mean-free path is much smaller than the width of a grid cell. In this case, λ is unresolved, and the effective propagation length scale is now determined by the much larger zone size. Because the signal speed is much less than c, the flux-limiter provides no constraint on the propagation of radiation. This can lead to unphysically rapid heating of irradiated boundary layers which are spatially unresolved. This problem has long bedeviled terrestrial transport applications; whether this represents a liability for a given astrophysical application should be carefully assessed by the user.

We consider now some basic details of our FLD module. In our RHD prescription, matter and radiation may exchange energy through absorption and emission processes represented by the right-hand sides of equations (3) and (4), and the radiation stress term on the LHS of (4). The radiation energy budget is further influenced by spatial transport, which we treat with the diffusion operator. The high radiation signal speed in transparent media mandates an implicit solution to the radiation energy equation. Coupling between the radiation and matter is treated via an iterative scheme based upon Newton's method. Recently, Turner & Stone (2001) published a new FLD module for the ZEUS-2D code. The physical assumptions underlying their method are consistent with those identified here, but the mathematical treatment for solving the coupled system differs from what we describe below.

Our construction of a linear system for the 3-D RHD equations begins with expressions for the spatially and temporally discretized gas and radiation energy equations. Consider the gas and radiation energy densities to be defined at discrete points along 3 orthogonal axes denoted by i, j, and k; i.e. $e \rightarrow e_{i,j,k}$ and $E \rightarrow E_{i,j,k}$. We approximate the partial time derivative in terms of a time-centered difference between two adjacent time levels, t^n and t^{n+1}: $\Delta t \equiv t^{n+1} - t^n$. We then define two functions in $e_{i,j,k}$ and $E_{i,j,k}$:

$$
\begin{aligned}
f_{i,j,k}^{(1)} = & E_{i,j,k}^{n+1} - E_{i,j,k}^n \\
& - \Delta t \left[4\pi\kappa_P B_p - c\kappa_E E_{i,j,k}^{n+1} \right] \\
& - \Delta t \left[\nabla \cdot \mathbf{F}_{i,j,k}^{n+1} + \nabla\mathbf{v} : \mathbf{P}_{i,j,k}^{n+1} \right];
\end{aligned} \tag{55}
$$

$$
\begin{aligned}
f_{i,j,k}^{(2)} = & e_{i,j,k}^{n+1} - e_{i,j,k}^n \\
& - \Delta t \left[-4\pi\kappa_P B_p + c\kappa_E E_{i,j,k}^{n+1} \right] \\
& + \Delta t p \nabla \cdot \mathbf{v}.
\end{aligned} \tag{56}
$$

For notational economy, we have confined explicit reference to coordinate indices and time level to the gas and radiation energy variables. As written above, the functions $f_{i,j,k}^{(1)}$ and $f_{i,j,k}^{(2)}$ are identically zero for a consistent solution for $e_{i,j,k}$ and $E_{i,j,k}$. We employ a Newton-Raphson iteration scheme to find the roots of (56) and (55). To construct a discrete system of equations which is linear in both energy variables, we evaluate the E-dependent flux limiter from values of E at the previous time level n. The thermal pressure, Planck functions, and opacities are updated at each iteration. The velocities are roughly (but not formally) time-centered, having been updated with contributions from body forces and artificial viscosity prior to the radiation solution (cf. figure 2). We may write the linear system to be solved as

$$
J(x)\,\delta x = -f(x). \tag{57}
$$

In (57), x is the vector of gas and radiation energy variables: $x \equiv (E_{i,j,k}, e_{i,j,k})$. Likewise, the solution vector δx is the set of corrections to these variables, $(\delta E_{i,j,k}, \delta e_{i,j,k})$. f represents the vector of discrete functions $\left(f_{i,j,k}^{(1)}, f_{i,j,k}^{(2)} \right)$, and $J(x)$ is the Jacobian, $\partial f^i / \partial x^j$.

As written above, expression (57) represents a matrix of size (2N)x(2N), where N is the product of the numbers of mesh points along each coordinate axis. As will

be shown in appendix D the corrections, $\delta e_{i,j,k}$, to the gas energies may be analytically expressed as functions of the radiation energy corrections, $\delta E_{i,j,k}$. This allows the solution of a reduced system of size NxN for the vector of $\delta E_{i,j,k}$, from which the set of $\delta e_{i,j,k}$ are immediately obtained. These corrections are used to iteratively update the trial values of $e_{i,j,k}$ and $E_{i,j,k}$. We have found in a variety of test problems that typically 2 to 10 N-R iterations are required for a converged solution at each time step.

Each iteration in the N-R loop involves a linear system which must be solved with matrix algebra. Our solution for Poisson's equation also requires a linear system solver. We discuss the types of linear solvers implemented in ZEUS-MP separately in §3.8, with additional details provided in appendices D, E, and F.

3.4. *Self Gravity*

ZEUS-MP treats Newtonian gravity at three different levels of approximation: (1) point-mass potentials, (2) spherically-symmetric gravity ($\nabla \Phi = GM/r^2$), and (3) exact solutions to Poisson's equation (15). The first two options are trivial to implement; in this discussion we therefore focus on the final option. In three dimensions, the discrete Laplacian operator connects a mesh point at coordinate (i,j,k) with both adjacent neighbors along each axis; thus a finite-differenced form of (15) takes the following form:

$$a_1 \Phi_{i,j,k-1} \ + \ a_2 \Phi_{i,j-1,k} \ + \ a_3 \Phi_{i-1,j,k} +$$
$$a_4 \Phi_{i+1,j,k} \ + \ a_5 \Phi_{i,j+1,k} \ + \ a_6 \Phi_{i,j,k+1} +$$
$$a_7 \Phi_{i,j,k} = 4\pi G \rho_{i,j,k} \quad (58)$$

If (58) is defined on a Cartesian mesh in which the zone spacing along each axis is uniform, then the "a" coefficients in equation 58 are constant over the problem domain. If, in addition to uniform gridding, the problem data is characterized by spatial periodicity, then *Fast Fourier Transform* (FFT) algorithms offer a highly efficient method for solving (58). For this class of problems (see Li et al. (2004) for a recent example), ZEUS-MP provides an FFT module based upon the publicly available "FFTw" software (Frigo & Johnson 2005). While FFT-based methods are not in general restricted only to periodic problems, we emphasize that the module implemented in ZEUS-MP is valid only for 3-D problems on uniform Cartesian meshes with triply-periodic boundaries.

For multidimensional problems which do not meet all of the validity criteria for the FFTw solver, ZEUS-MP provides two additional solution modules. The most general of these accommodates 2-D and 3-D grids in Cartesian, cylindrical, and spherical geometries and is based upon the same CG solver provided for the FLD radiation equations. A second module currently written for 3-D Cartesian meshes with Dirichlet or Neumann boundary conditions is based upon the *multigrid* (MG) method (cf. §3.8.2).

When equation (58) is formulated as a function of ZEUS covariant grid variables, the matrix elements represented by the "a" coefficients take on a much more complicated functional form than the constant values which obtain for a uniform Cartesian mesh. The form of (58) written for the general class of 3-D covariant grids is

documented in appendix E. Details of all three solution techniques for Poisson's equation are written in §3.8.

3.5. *Multi-species Advection*

Prior public versions of ZEUS codes have treated the gas as a single species fluid. In order to be forward-compatible with physical processes such as chemistry, nuclear reactions, or lepton transport, ZEUS-MP offers a straightforward mechanism for identifying and tracking separate chemical or nuclear species in a multi-species fluid mixture. Because ZEUS-MP solves one set of fluid equations for the total mass density, ρ, we have no facility for modeling phenomena in which different species possess different momentum distributions and thus move with respect to one another. Nonetheless, a wide variety of astrophysical applications are enabled with a mechanism for quantifying the abundances of separate components in a mixed fluid. Our multi-species treatment considers only the physical advection of different species across the coordinate mesh; physics modules which compute the local evolution of species concentrations (such as a nuclear burning network) must be provided by the user.

Our implementation of multispecies advection proceeds by defining a concentration array, X_n, such that ρX_n is the fractional mass density of species n. The advection equations in the ZEUS transport step therefore include:

$$\frac{d}{dt} \int_V (\rho X_n) \ dV \ = \ - \oint_{dV} (\rho X_n) \ (\mathbf{v} - \mathbf{v_g}) \cdot d\mathbf{S}. \quad (59)$$

This construction is evaluated such that the mass fluxes used to advect individual species across the mesh lines are consistent with those used to advect the other field variables defined in the application. Discrete formulae for the conservative advection of X_n and the other hydrodynamic field variables are provided in appendix B.

3.6. *Time Step Control*

Maintainence of stability and accuracy in a numerical calculation requires proper management of time step evolution. The general expression regulating the time step in ZEUS-MP is written

$$\Delta t_{\text{new}} = C_{\text{cfl}}$$
$$\times \left\{ \frac{1}{\Delta t_{\text{cs}}^2} + \frac{1}{\Delta t_{v1}^2} + \frac{1}{\Delta t_{v2}^2} + \frac{1}{\Delta t_{v3}^2} + \right.$$
$$\left. \frac{1}{\Delta t_{\text{al}}^2} + \frac{1}{\Delta t_{\text{av}}^2} + \frac{1}{\Delta t_{\text{rd}}^2} \right\}^{1/2} ,$$
$$(60)$$

in which C_{cfl} is the Courant factor and the Δt^2 terms are squares of the minimum values of the following quantities:

$$\Delta t_{\text{cs}}^2 = \gamma (\gamma - 1) \cdot (e/\rho) \ / \ (\Delta X_{\text{min}})^2 ; \quad (61)$$
$$\Delta t_{v1}^2 = [(v1 - vg1) \ / \ dx1a]^2 ; \quad (62)$$
$$\Delta t_{v2}^2 = [(v2 - vg2) \ / \ (g2b \ dx2a)]^2 ; \quad (63)$$
$$\Delta t_{v3}^2 = [(v3 - vg3) \ / \ (g31b \ g32b \ dx3a)]^2 ; \quad (64)$$
$$\Delta t_{\text{al}}^2 = \left(\overline{b1}^2 + \overline{b2}^2 + \overline{b3}^2 \right) \ / \ \left[4\rho (\Delta X)^2 \right] ; \quad (65)$$

$$\Delta t_{\mathrm{av}}^2 = \left[4 \text{ qcon } \left| \frac{dv}{dx} \right|_{\mathrm{max}} \right]^{-2} ; \qquad (66)$$

$$\Delta t_{\mathrm{rd}}^2 = \left[\mathbf{ertol} \left| \frac{\mathrm{E}}{\Delta \mathrm{E}} \right| \right]^2 \qquad (67)$$

These values represent, respectively, the local sound-crossing time, the local fluid crossing time along each coordinate, the local Alfvén wave crossing time, the local viscous timescale, and a radiation timescale determined by dividing the value of E returned from the FLD solver by the time rate of change in E determined by comparing the new E to that from the previous time step. **ertol** is a specified tolerance which retrodictively limits the maximum fractional change in E allowed in a timestep. ΔX_{min} represents the minimum length of a 3-D zone edge, i.e. MIN[$dx1a$, $g2b\ dx2a$, $g31b\ g32b\ dx3a$], where each zone length is expressed in terms of the local covariant grid coefficients (cf. appendix B). As expressed in (60), dt_{new} represents a trial value of the new time step, which is allowed to exceed the previous value of the time step by no more than a preset factor; i.e. $\Delta t_{\mathrm{final}} = \min[\Delta t_{\mathrm{new}}, \text{fac} \times \Delta t_{\mathrm{old}}]$, with "fac" typically equaling 1.26. This value allows the time step to increase by up to a factor of 10 every 10 cycles.

3.7. *Parallelism*

The most powerful computers available today are parallel systems with hundreds to thousands of processors connected into a cluster. While some systems offer a shared-memory view to the applications programmer, others, such as Beowulf clusters, do not. Thus, to maximize portability we have assumed "shared nothing" and implemented ZEUS-MP as an SPMD (Single Program, Multiple Data) parallel code using the MPI message-passing library to accomplish interprocessor communication. In this model, parallelism is affected via *domain decomposition* (Foster 1995), in which each CPU stores data for and performs operations upon a unique sub-block of the problem domain. Because finite-difference forms of gradient, divergence, and Laplacian operators couple data at multiple mesh points, data must be exchanged between neighboring processors when such operations are performed along processor data boundaries. ZEUS-MP employs "asynchronous" or "non-blocking" communication functions which allow interprocessor data exchange to proceed simultaneously with computational operations. This approach provides the attractive ability to hide a large portion of the communication costs and thus improve parallel scalability. Details of our method for overlapping communication and computation operations in ZEUS-MP are provided in appendix F.

3.8. *Linear Solvers*

Our implicit formulation of the RHD equations and our solution to Poisson's equation for self gravity require the use of an efficient linear system solver. Linear systems for a single unknown may involve of order 10^6 solution variables for a 3-D mesh at low to moderate resolution; the number of unknowns in a high-resolution 3-D simulation can exceed 10^9. In this regime, the CPU cost of the linear system solution can easily dominate the cost of the full hydrodynamic evolution. The choice of solution technique, with its associated implementation requirements

and performance attributes, is therefore critically important. Direct inversion methods such as Gauss-Seidel are ruled out owing both to extremely high operation counts and a spatially recursive solution which precludes parallel implementation. As with radiation flux limiters, there is no "best" method, practical choices being constrained by mathematical factors such as the matrix *condition number* (cf. §3.8.1), coordinate geometry, and boundary conditions, along with performance factors such as sensitivity to problem size and ease of parallel implementation. This variation of suitability with problem configuration motivated us to instrument ZEUS-MP with three separate linear solver packages: a *preconditioned conjugate gradient* (CG) solver, a *multigrid* (MG) solver, and a *fast Fourier transform* (FFT) solver. We describe each of these below.

3.8.1. *The Conjugate Gradient Solver*

The conjugate gradient (CG) method is one example of a broad class of *non-stationary iterative* methods for sparse linear systems. A concise description of the theory of the CG method and a pseudo-code template for a numerical CG module is available in Barret et al. (1994). While a full discussion of the CG method is beyond the scope of this paper, several key elements will aid our discussion. The linear systems we wish to solve may be written in the form $Ax = b$, where A is the linear system matrix, x is the unknown solution vector, and b is a known RHS vector. An associated quadratic form, $f(x)$, may be constructed such that

$$f(x) = \frac{1}{2} x^T A x - b^T x + c, \qquad (68)$$

where c is an arbitrary constant (the "T" superscript denotes a transpose). One may show algebraically that if A is symmetric ($A^T = A$) and positive-definite ($x^T A x > 0$ for all non-zero x), then the vector x which satisfies $Ax = b$ also satisfies the condition that $f(x)$ is minimized, i.e.

$$f'(x) \equiv \left\{ \begin{array}{c} \frac{\partial}{x_1} f(x) \\ \frac{\partial}{x_2} f(x) \\ . \\ . \\ . \\ \frac{\partial}{x_n} f(x) \end{array} \right\} = 0. \qquad (69)$$

The CG method is an iterative technique for finding elements of x such that (69) is satisfied. A key point to consider is that the convergence rate of this approach is strongly sensitive to the *spectral radius* or *condition number* of the matrix, given by the ratio of the largest to smallest eigenvalues of A. For matrices that are poorly conditioned, the CG method is applied to "preconditioned" systems such that

$$M^{-1} A x = M^{-1} b, \qquad (70)$$

where the *preconditioner*, M^{-1}, is chosen such that the eigenvalues of $(M^{-1}A)$ span a smaller range in value. (Typically, one equates the preconditioner with M^{-1} rather than M.)

From (70) it follows at once that the "ideal" preconditioner for A is simply A^{-1}, which of course is unknown.

TABLE 1
MULTIGRID V-CYCLE ITERATION.

Keyword	Operation	Description
smooth	$\mathcal{L}_h \bar{x}_h^n = b_h$	*Smooth error on fine grid via stationary method*
compute	$r_h^n \leftarrow b_h - \mathcal{L}_h \bar{x}_h^n$	*Compute residual on fine grid*
restrict	$r_h^n \rightarrow r_{2h}^n$	*Transfer residual down to coarse grid*
solve	$\mathcal{L}_{2h} e_{2h}^n = r_{2h}^n$	*Obtain coarse grid correction from residual equation*
prolong	$e_h^n \leftarrow e_{2h}^n$	*Transfer coarse grid correction up to fine grid*
update	$x_h^{n+1} \leftarrow \bar{x}_h^n + e_h^n$	*Update solution with coarse grid correction*

However, for matrices in which the main diagonal elements are much larger in magnitude than the off-diagonal elements, a close approximation to A^{-1} may be constructed by defining a diagonal matrix whose elements are given by the reciprocals of the corresponding diagonal elements of A. This technique is known as *diagonal preconditioning*, and we have adopted it in the implementation of our CG solver. The property in which the diagonal elements of A strongly exceed (in absolute value) the values of the off-diagonal elements is known as *diagonal dominance*. Diagonal dominance is a prerequisite for the profitable application of diagonal preconditioning. Diagonal preconditioning is an attractive technique due to its trivial calculation, the fact that it poses no barrier to parallel implementation, and its fairly common occurrence in linear systems. Nonetheless, sample calculations in §5 will demonstrate cases in which diagonal dominance breaks down, along with the associated increase in cost of the linear solution.

3.8.2. *The Multigrid Solver*

Unlike the conjugate gradient method, multigrid methods (Brandt 1977) are based on stationary iterative methods. A key feature of multigrid is the use of a hierarchy of nested coarse grids to dramatically increase the rate of convergence. Ideally, multigrid methods are fast, capable of numerically solving elliptic PDE's with computational cost proportional to the number of unknowns, which is optimal. For example, for a k^3 problem in three dimensions, multigrid (specifically the *full multigrid* method, discussed below) requires only $O(k^3)$ operations. Compare this to $O(k^3 \log k)$ for FFT methods, $O(k^4)$ for non-preconditioned CG, and approximately $O(k^{3.5})$ for preconditioned CG (Heath 1997). Multigrid has disadvantages as well, however; they are relatively difficult to implement correctly, and are very sensitive to the underlying PDE and discretization. For example, anisotropies in the PDE coefficients or grid spacing, discontinuities in the coefficients, or the presence of advection terms, can all play havoc with standard multigrid's convergence rate.

Stationary methods, on which multigrid is based, are very simple, but also very slow to converge. For a linear system $Ax = b$, the two main stationary methods are Jacobi's method ($a_{ii} x_i^{n+1} = b_i - \sum_{j \neq i} a_{ij} x_j^n$) and the Gauss-Seidel method ($a_{ii} x_i^{n+1} = b_i - \sum_{j < i} a_{ij} x_j^{n+1} - \sum_{j > i} a_{ij} x_j^n$), where subscripts denote matrix and vector components, and superscripts denote iterations. While stationary methods are very slow to converge to the solution (the computational cost for both Jacobi and Gauss-

Seidel methods is $O(k^5 \log k)$ for a k^3 elliptic problem in $3D$), they do reduce the high-frequency components of the error very quickly; that is, they efficiently "smooth" the error. This is the first part of understanding how multigrid works. The second part is that a problem with a smooth solution on a given grid can be accurately represented on a coarser grid. This can be a very useful thing to do, because problems on coarser grids can be solved faster.

Multigrid combines these two ideas as follows. First, a handful of iterations of a stationary method (frequently called a "smoother" in multigrid terminology) is applied to the linear system to smooth the error. Next, the residual for this smoothed problem is transfered to a coarse grid, solved there, and the resulting coarse grid correction is used to update the solution on the original ("fine") grid. Table 1 shows the main algorithm for the *multigrid V-cycle iteration*, applied to the linear system $\mathcal{L}_h x_h^n = b_h$ associated with a grid with zone spacing h.

Note that the coarse grid problem (keyword "solve" in table 1) is solved recursively. The recursion bottoms out when the coarsest grid has a single unknown; or, more typically, when the coarse grid is small enough to be quickly solved using some other method, such as CG, or with a small number of applications of the smoother. Also, the multigrid V-cycle can optionally have additional applications of the smoother at the end of the iteration. This is helpful to smooth errors introduced in the coarse grid correction, or to symmetrize the iteration when used as a preconditioner.

The *full multigrid* method uses V-cycles in a bootstrapping approach, first solving the problem on the the coarsest grid, then interpolating the solution up to the next-finer grid to use as a starting guess for a V-cycle. Ideally, just a single V-cycle at each successively finer grid level is required to obtain a solution whose error is commensurate with the discretization error.

Multigrid methods in ZEUS-MP are provided using an external MPI-parallel C++/C/Fortran package called MGMPI (Bordner 2002). It includes a suite of Krylov subspace methods as well as multigrid solvers. The user has flexible control over the multigrid cycling strategy, boundary conditions, depth of the multigrid mesh hierarchy, choice of multigrid components, and even whether to use Fortran or C computational kernels. Parallelization is via MPI using the same domain decomposition as ZEUS-MP. Currently there are limitations to grid sizes in MGMPI: there must be $M2^L - 1$ zones along axes bounded by Dirichlet or Neumann boundary conditions, and $M2^L$ zones along periodic axes, where L is the number of coarse grids used, and M is an arbitrary integer.
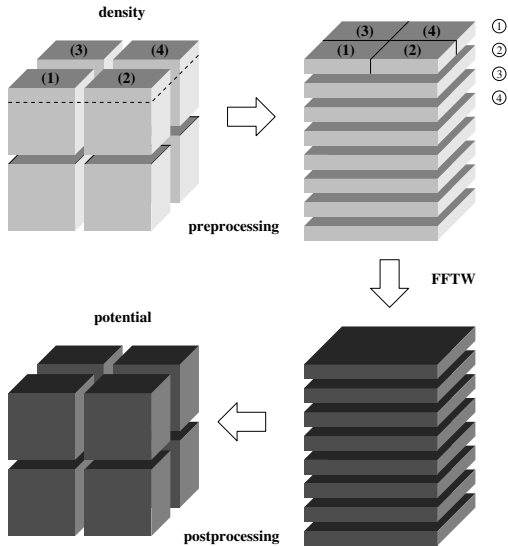
FIG. 6.— Block-to-slab decomposition of density data before calling FFTW library and slab-to-block decomposition of gravitational potential afterward.

This restriction is expected to change in future versions of MGMPI.

3.8.3. *The Fast Fourier Transform Solver*

As mentioned in §3.4, FFT algorithms offer a highly efficient method in solving the Poisson equation. The publicly available "Fastest Fourier Transform in the West" (FFTw) algorithm (Frigo & Johnson 2005) is used as one of the gravity solvers available in ZEUS-MP. Note that the parallelized version of the FFTw library using MPI is only available in version 2.1.5 or before. Based on this version of MPI FFTw, the gravity solver implemented in ZEUS-MP is valid only for Cartesian meshes with triply-periodic boundaries.

The transform data used by the FFTw routines is distributed, which means that a distinct portion of data resides in each processor during the transformation. In particular, the data array is divided along the first dimension of the data cube, which is sometimes called a slab decomposition. Users can design their data layout using slab decomposition as the FFTw requires but it is inconvenient for solving general problems. Therefore, there is a preprocessing of the density data distribution from a general block domain decomposition to a slab decomposition before calling the FFTw routines. After the potential is calculated, the slab decomposition of the potential is transformed back to block decomposition in the postprocessing stage. Figure 6 shows the idea of these two additional processes.

In figure 6, the initial density data block is first rearranged from block decomposition into slab decomposition. In this example using a $2 \times 2 \times 2$ topology, the first layer of blocks will be divided into four slabs. The four processors exchange the data slabs to ensure the density data remains organized at the same spatial location. Therefore, the data exchange can be viewed as a rearrangement of the spatial location of processors. Since data in the second layer of blocks does not overlap

with the first layer of data blocks, the non-blocking data communication among blocks in different layers can proceed simultaneously. After the gravitational potential is calculated, the reverse rearrangement of potential data from slab decomposition back to block decomposition is performed in an analogous manner.

Because of the required slab decomposition in FFTw, the number of processors that can be used for a given problem size is limited. For example, for a problem with 512^3 zones on 512 processors, each slab least one cell thick. Using more than 512 processors in this example will not lessen the time to solution for the potential as extra processors would simply stand idle.

4. VERIFICATION TESTS

In this section we present results from a suite of test problems designed to stress each of ZEUS-MP's physics modules and verify the correct function of each. We begin with a pure HD problem which follows shock propagation in spherical geometry. We then examine a trio of MHD problems, two of which were considered in Stone & Norman (1992b), and the final of which has become a standard multidimensional test among developers of Godunov-based MHD codes. The section concludes with two radiation problems, the first treating radiation diffusion waves through static media; the second following the evolution of radiating shock waves. All problems with shocks use a quadratic (von Neumann-Richtmyer) artificial viscosity coefficient qcon of 2.0. The Orszag-Tang vortex test uses an additional linear viscosity with a value of qlin = 0.25.

Three of the test problems discussed below which include hydrodynamic effects are also adiabatic: these are the Sedov-Taylor HD blast wave, the MHD Riemann problem, and the Orszag-Tang MHD vortex problem. In these cases, the total fluid energy integrated over the grid should remain constant. Because ZEUS-MP evolves an internal gas energy, rather than total fluid energy, equation, the ZEUS-MP solution scheme is non-conservative by construction. It therefore behooves us to monitor and disclose errors in total energy conservation as appropriate. For the three adiabatic dynamics problems noted above, the total energy conservation errors (measured relative to the initial integrated energy) were 1.4%, 0.8%, and 1.6%, respectively.

For problems involving magnetic field evolution, an additional metric of solution fidelity is the numerical adherence to the divergence-free constraint. As shown analytically in §3.2 and previously in Hawley & Stone (1995); Stone & Norman (1992b); Evans & Hawley (1988), the Constrained Transport algorithm is divergence-free by construction, regardless of the method chosen to compute the EMF's. Nonetheless, all numerical codes which evolve discrete representations of the fluid equations are vulnerable to errors bounded from below by machine round-off; we therefore compute $\nabla \cdot \mathbf{B}/|\mathbf{B}|$ at each mesh point and record the largest positive and negative values thus obtained. For the Alfvén rotor and MHD Riemann problems, which are computed on 1D grids, the maximum normalized divergence values at the end of the calculations remain formally zero to machine tolerance in double precision (15 significant figures). For the 2D Orszag-Tang vortex, the divergence-free constraint remains satisfied to within roughly 1 part in 10^{12}, con-
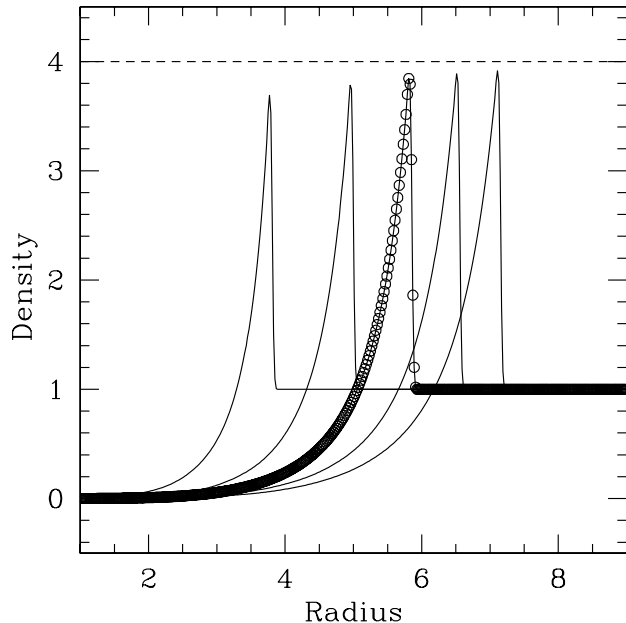
FIG. 7.— Density vs. radius for the Sedov-Taylor blast wave. Density is plotted in units of 10^{-8} g cm^{-3}; radius is plotted in units of 10^{13} cm. Profiles represent numerical results plotted in intervals of 6×10^4 seconds. The dashed line shows the analytic value for the peak density.

sistent with machine round-off error over an evolution of roughly 1000 timesteps.

4.1. *Hydrodynamics: Sedov-Taylor Blast Wave*

Our first test problem is a classic hydrodynamic test due to Sedov (1959), in which a point explosion is induced at the center of a cold, homogeneous sphere in the absence of gravity. Problem parameters are chosen such that the explosion energy is orders of magnitude larger than the total internal energy of the cloud. In this case, the resulting shock wave evolves in a self-similar fashion in which the shock radius and velocity evolve with time according to

$$r_{sh} = \xi_{sh} \left(\frac{E_o}{\rho_o} \right)^{1/5} t^{2/5}, \qquad (71)$$

and

$$v_{sh} = \frac{2}{5} \xi_{sh} \left(\frac{E_o}{\rho_o} \right)^{1/5} t^{-3/5}, \qquad (72)$$

where E_o and ρ_o are the explosion energy and the initial density, respectively. ξ_{sh} is a dimensionless constant which is equal to 1.15 for an ideal gas with $\gamma = 5/3$. The density, pressure, temperature, and fluid velocity at the shock front are given by

$$\rho_s = 4\rho_o, \qquad (73)$$

$$P_s = \frac{3}{4}\rho_o v_{sh}^2, \qquad (74)$$

$$T_s = \frac{3}{16} \frac{\mu m_h}{k_B} v_{sh}^2, \qquad (75)$$

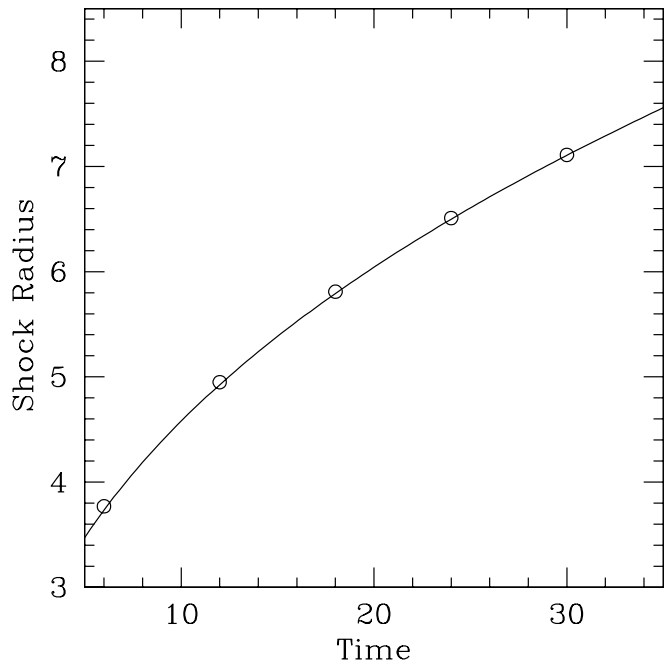and

$$v_s = \frac{3}{4} v_{sh}. \qquad (76)$$

FIG. 8.— Shock radius vs. time for the Sedov-Taylor blast wave. Time is plotted in units of 10^4 seconds in the right-hand figure. Open circles represent numerical results plotted at the times corresponding to profiles in Figure 7. The solid line indicates the analytic solution.

Our problem was run in one dimension on a mesh of 500 zones equally spaced in radius. We initialize a spherical cloud of radius 10^{14} cm with a uniform density of 10^{-8} g/cm^3. The initial temperature is 50 K. At t = 0, 10^{50} ergs of internal energy are deposited within a radius of 10^{12} cm, which spreads the blast energy over 5 zones. Depositing the energy over a few zones within a small region centered on the origin maintains the point-like nature of the explosion and markedly improves the accuracy of the solution relative to that obtained if the all of the energy is deposited in the first zone.

Figures 7 and 8 provide results of our Sedov-Taylor blast wave test. Figure 7 shows radial plots of density separated in time by 6×10^4 seconds. The density and radius are expressed in units of 10^{-8} g/cm^{-3} and cm, respectively. The dashed line indicates the analytic value for the density at the shock front. Figure 8 shows numerical values (open circles) of the shock front at times identical to those used in the Figure 7. These data are superimposed upon the analytic solution (solid line) for the shock radius given by equation 71.

4.2. *MHD*

4.2.1. *Magnetic Braking of an Aligned Rotor*

Our first MHD test examines the propagation of torsional Alfvén waves generated by an aligned rotor. A disk of uniform density ρ_d, thickness z_d, and angular velocity Ω_0 lies in an ambient, initially static, medium with density ρ_m. The disk and medium are threaded by an initially uniform magnetic field oriented parallel to the rotation axis of the disk. Considered in cylindrical geometry, rotation of the disk produces transverse Alfvén waves which propagate along the Z axis and generate non-zero ϕ components of velocity and magnetic field. Analytic solutions for v_ϕ and B_ϕ were calculated by Mouschovias & Paleologou (1980) under the assumption that only the
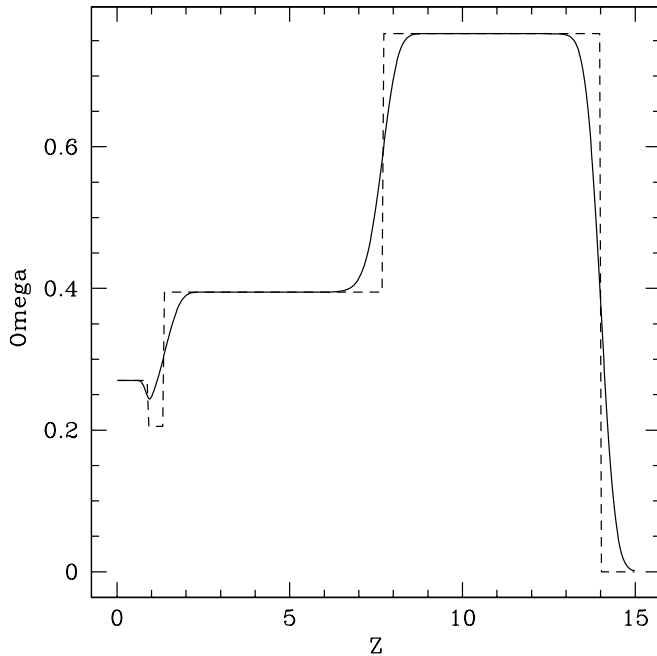
FIG. 9.— MHD aligned rotor: angular velocity vs. Z. The dashed line indicates the analytic solution of Mouschovias & Paleologou (1980).
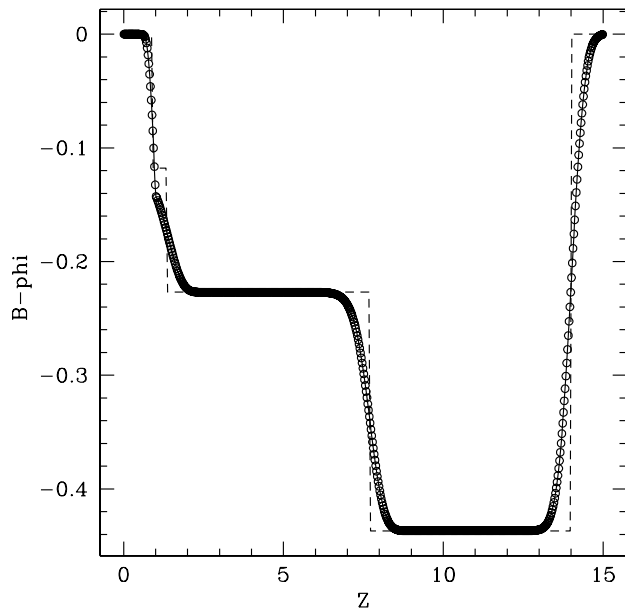

FIG. 11.— MHD Riemann problem: density vs. X at t = 80 sec.

FIG. 10.— MHD aligned rotor: B_ϕ vs. Z. The dashed line indicates the analytic solution of Mouschovias & Paleologou (1980).
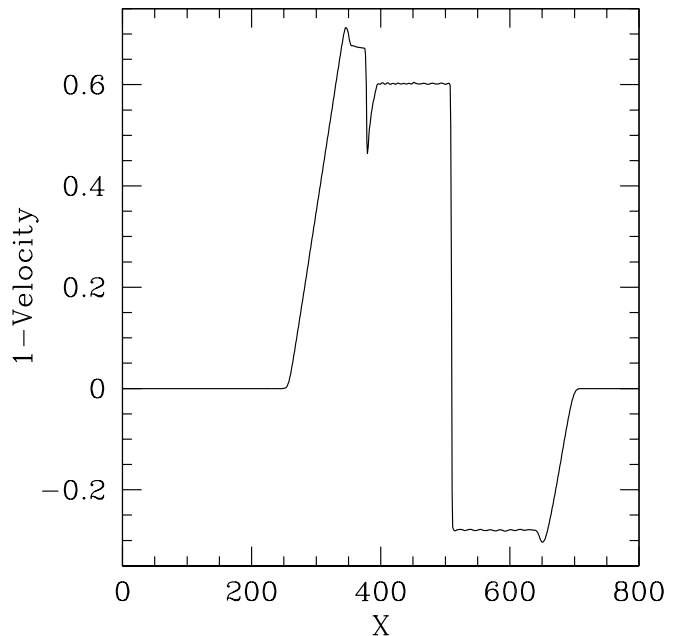
FIG. 12.— MHD Riemann problem: 1-velocity vs. X at t = 80 sec.

transverse Alfvén wave modes are present; to reproduce these conditions in ZEUS-MP, compressional wave modes due to gradients in gas and magnetic pressures are artificially suppressed. The utility of this restriction lies in the fact that in more general calculations, errors in the propagation of Alfvén waves may easily be masked by the effects of other wave modes in the problem.

The problem parameters as described above correspond to the case of discontinuous initial conditions considered in Mouschovias & Paleologou (1980). We consider a half-plane spanning the range $0 \leq Z \leq 15$,
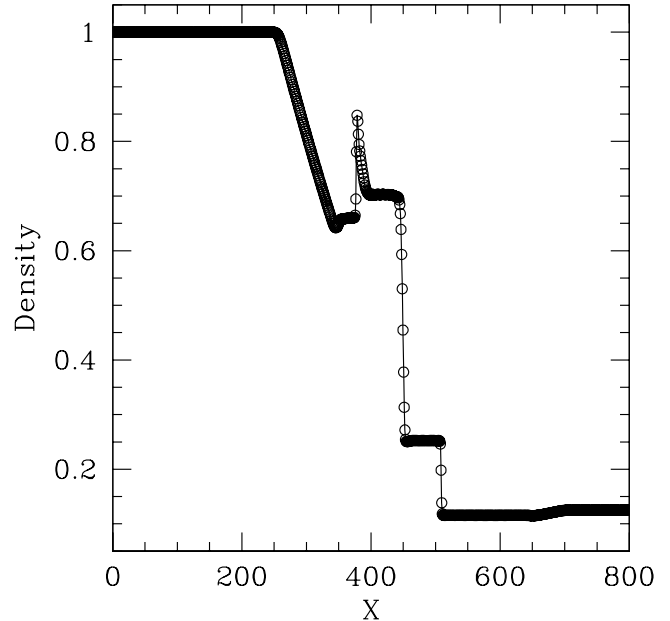
with $\rho_d = 10$ and $\rho_m = 1$. Because there are no dynamical phenomena acting along the radial coordinate, we may compute the problem on a 1-D grid of Z on which R- and ϕ-invariant values of v_ϕ and B_ϕ are computed. Figures 9 and 10 show results of the calculation at a time t = 13. Solid curves indicate the solutions from ZEUS-MP; dashed lines show the analytic solution of Mouschovias & Paleologou (1980). These results are consistent with those obtained with ZEUS-2D and shown in Stone & Norman (1992b); the only salient difference between the two calculations is that we used twice as many zones (600) as reported for the ZEUS-2D calculation. The increased resolution is mandated by the
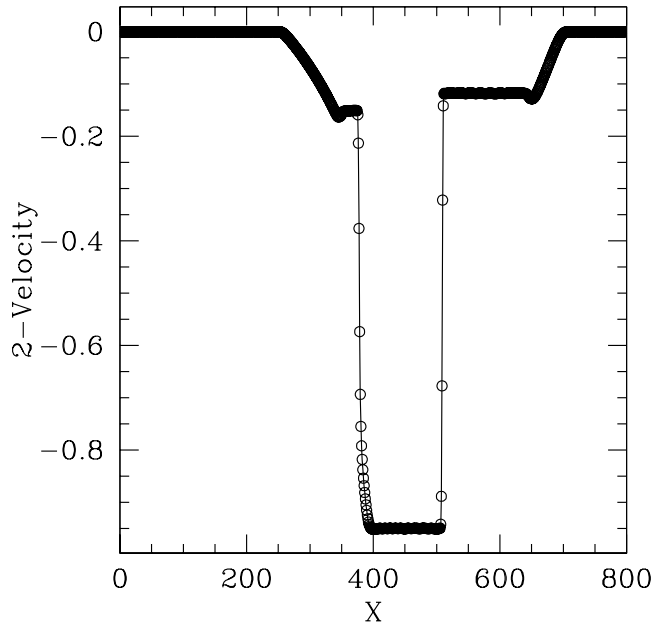
FIG. 13.— MHD Riemann problem: 2-velocity vs. X at t = 80 sec.

FIG. 14.— MHD Riemann problem: 3-velocity vs. X at t = 80 sec.

fact that the HSMOCCT algorithm is by construction more diffusive than the original MOCCT algorithm documented in Stone & Norman (1992b). As noted previously in section 3.2 and discussed in detail in Hawley & Stone (1995), this added diffusivity makes the MOCCT algorithm more robust in fully multidimensional calculations. The requirement within HSMOCCT of higher resolution with respect to ZEUS-2D's older MOCCT algorithm is maximized in this test problem due to the artificial suppression of compressive hydrodynamic waves and longitudinal MHD waves; the true resolution requirements of HSMOCCT as implemented in ZEUS-MP will depend in part upon the relative importance of various wave modes to the calculation and will in general be problem dependent.

4.2.2. *MHD Riemann Problem*

Our second MHD problem is a magnetic shock tube problem due to Brio & Wu (1988). Test results with ZEUS-2D using the van Leer advection algorithm were also published in Stone & Norman (1992b). This test problem is described by "left" and "right" states in which the discontinuous medium is threaded by a magnetic field which is uniform on both sides but exhibits a kink at the material interface. Our formulation of the problem differs from that of Brio & Wu (1988) and Stone & Norman (1992b) only in that we have oriented the transverse component of the magnetic field to have non-zero components along both the Y and Z axes in Cartesian geometry. At t = 0, our left state is given by $(\rho_l, \ p_l, \ B_l^x, \ B_l^y, \ B_l^z) =$ (1.0, 1.0, 0.75, 0.6, 0.8), and the right state is given by $(\rho_r, \ p_r, \ B_r^x, \ B_r^y, \ B_r^z) =$ (0.125, 0.1, 0.75, -0.6, -0.8). All velocities are initially zero. The ratio of specific heats for this problem is 2.0. As with the calculation in Stone & Norman (1992b), the problem is computed in 1-D on an 800-zone grid and run to a time of t = 80. The spatial domain length is 800.
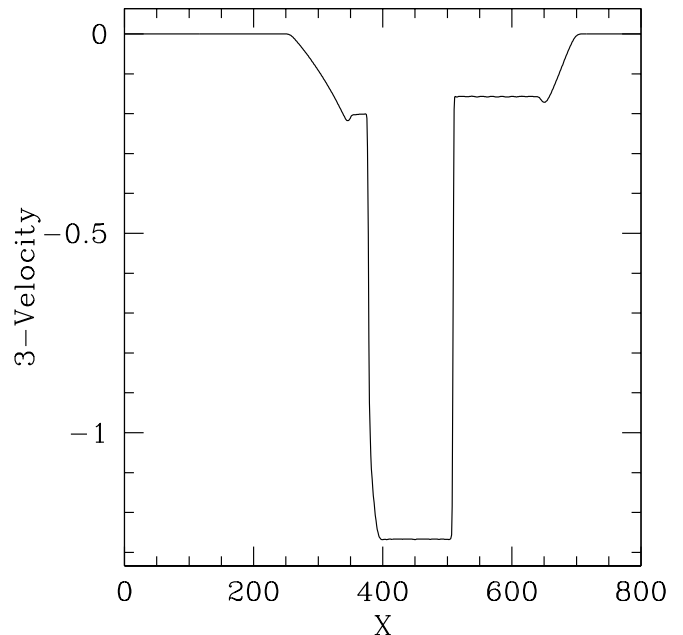
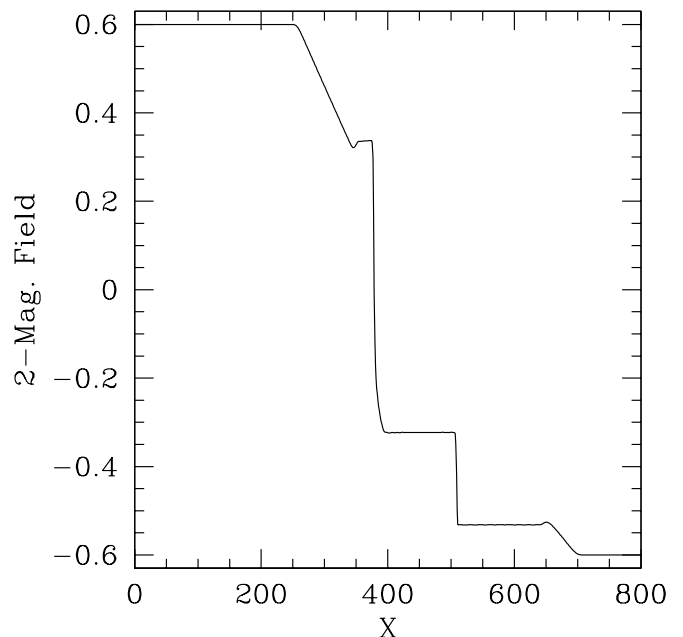Figures 11 through 16 show the results obtained

FIG. 15.— MHD Riemann problem: 2-component of magnetic field vs. X at t = 80 sec.

with ZEUS-MP. The 1-component of B (not included in the figures) remained flat over the domain at its initial value, as expected. The grid resolution is identical to that used in the ZEUS-2D calculation, and the results are evidently consistent (see Fig. 6 of Stone & Norman (1992b)). While this problem is not truly multidimensional, it does exhibit both transverse and compressional wave modes, in contrast with the previous test problem. In this case, we may qualitatively match the results from ZEUS-2D without an increase in grid resolution.

4.2.3. *Orszag-Tang Vortex*

Our final MHD test problem is a multidimensional problem due to Orszag & Tang (1979) which has been
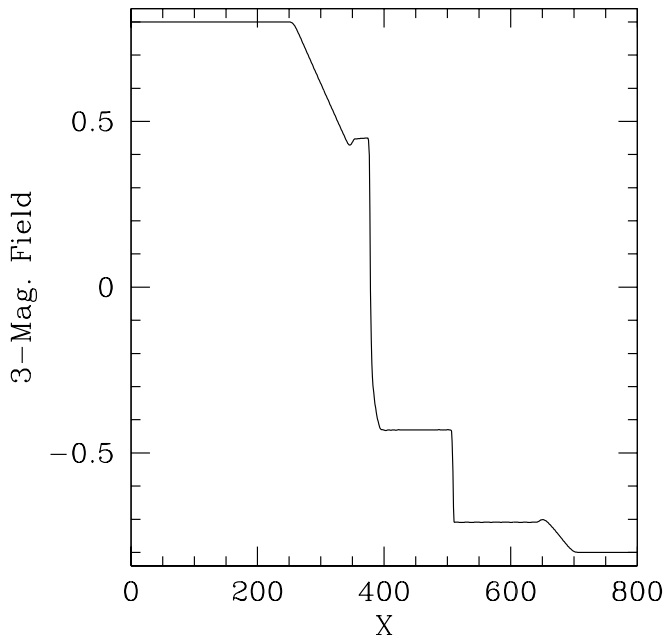
Fig. 16.— MHD Riemann problem: 3-component of magnetic field vs. X at t = 80 sec.

Fig. 17.— Orszag-Tang vortex: pressure at t = 0.48 seconds. 256^2 zones were used.

Fig. 18.— Orszag-Tang vortex: pressure at t = 0.48 seconds. 512^2 zones were used.

featured in a number of recent MHD method papers, such as Dai & Woodward (1998); Ryu et al. (1998) and Londrillo & Del Zanna (2000); Londrillo & del Zanna (2004). This problem follows the evolution of a 2-D periodic box of gas with $\gamma = 5/3$, in which fluid velocities and magnetic field components are initialized according to $\mathbf{v} = v_0 \left[-\sin\left(2\pi y\right) \hat{\mathbf{x}} + \sin\left(2\pi x\right) \hat{\mathbf{y}} \right]$ and $\mathbf{B} = B_0 \left[-\sin\left(2\pi y\right) \hat{\mathbf{x}} + \sin\left(4\pi x\right) \hat{\mathbf{y}} \right]$, with $v_0 = 1$ and $B_0 = 1/(4\pi)^{1/2}$. The box has length 1.0 along each side. The density and pressure are initially uniform with values of $25/(36\pi)$ and $5/(12\pi)$, respectively (these choices lead to an initial adiabatic sound speed of 1.0). Subsequent evolution leads to a complex network of waves, shocks, rarefactions, and stagnant flows. Ryu et al. (1998) provide greyscale snapshots of the flow field at t = 0.48; in addition, they provide 1-D cuts through the data along the line given by $y = 0.4277$, over which the gas and magnetic pressures are plotted as functions of x. The Ryu et al. (1998) results were computed on a 256^2-zone Cartesian mesh. For consistency, we also computed the problem on a 256^2-zone mesh, from which comparison values of pressure at the identical cut in y may be extracted. To explore the effect of resolution, we also provide 2-D greyscale images from a 512^2-zone calculation.

Our multidimensional flow structures at t = 0.48 are given in Figures 17 through 20, which are to be compared to the grey-scale panels on the left-hand side of Figure 3 in Ryu et al. (1998). Figure 21 presents line plots of gas and magnetic pressure along a line of x located at $y = 0.4277$. Save for a very small notch in the gas pressure near x = 0.5, our pressure profiles from the 256^2 calculation appear to be virtually identical to those from Ryu et al. (1998) at identical resolution. With respect to the 2-D images, the effect of resolution is most apparent in maps of the velocity divergence (Figures 19 and 20). Again, the ZEUS-MP results at a grid resolution of 256^2 compare quite favorably to those from Ryu et al. (1998), with subtle flow features marginally less well resolved. Our results are likewise consistent with
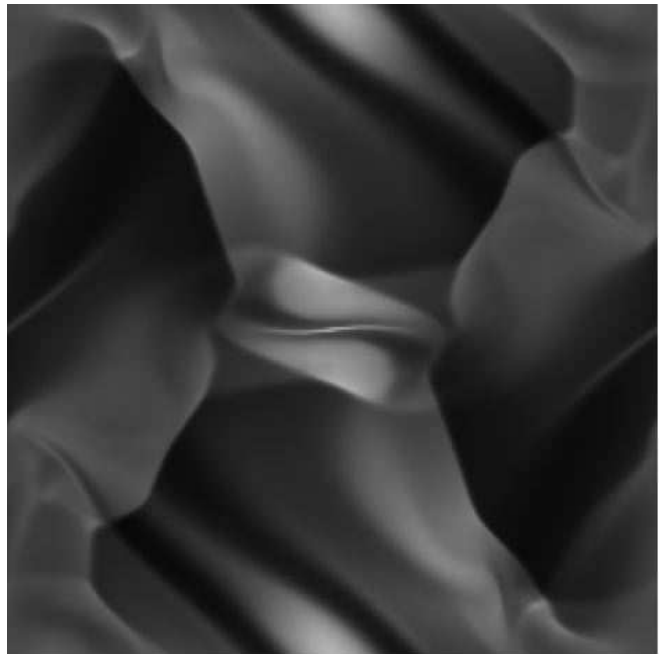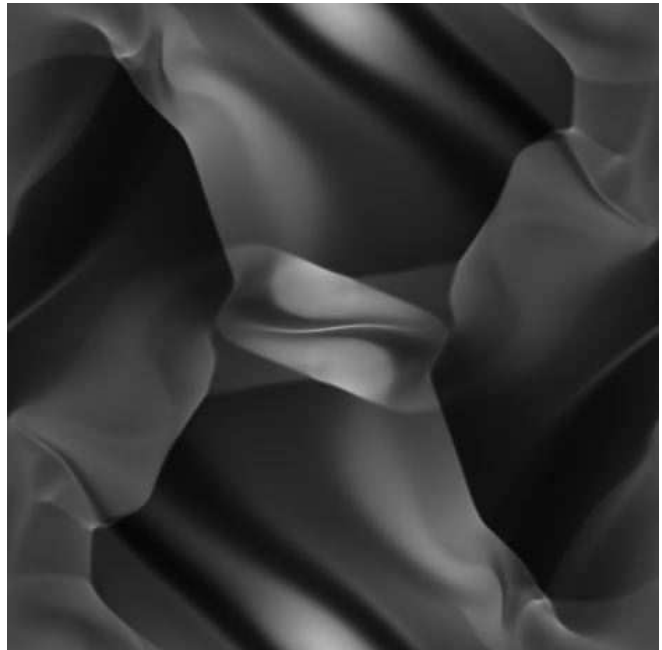
those computed at similar resolution by Dai & Woodward (1998) and Londrillo & Del Zanna (2000) (Londrillo & del Zanna (2004) also computed the problem but did not include figures matching the other cited works). The 256^2 and 512^2 results clearly bracket those of Ryu et al. (1998); thus we see that in this problem axial resolution requirements of the two codes differ by *at most* a factor of 2, which we consider an agreeable result for a finite-difference staggered-mesh code.

4.3. *Radiation*

4.3.1. *Marshak Waves*

FIG. 19.— Orszag-Tang vortex: $\nabla \cdot \mathbf{v}$ at t = 0.48 seconds for the 256^2 zone calculation.

FIG. 20.— Orszag-Tang vortex: $\nabla \cdot \mathbf{v}$ at t = 0.48 seconds for the 512^2 zone calculation.

We begin our examination of radiation physics with a test problem emphasizing the coupling between matter and radiation. The Marshak wave problem we compute was formulated by Su & Olson (1996) after a descriptio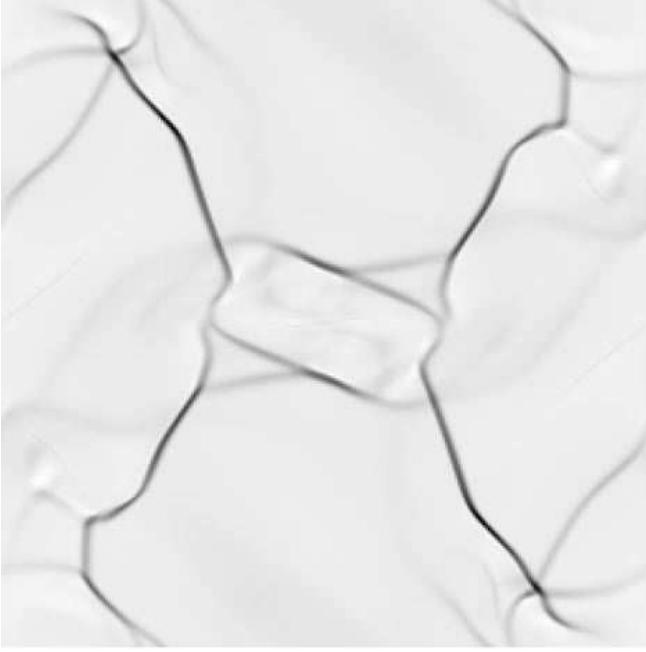n in Pomraning (1979). The problem considers the heating of a uniform, semi-infinite slab initially at $T = 0$ everywhere. The material is characterized by a T-independent (and therefore constant) opacity (κ) and a specific heat (α) proportional to T^3, in which case the gas and radiation energy equations become linear in the quantities E and T^4. Pomraning defined dimensionless space and
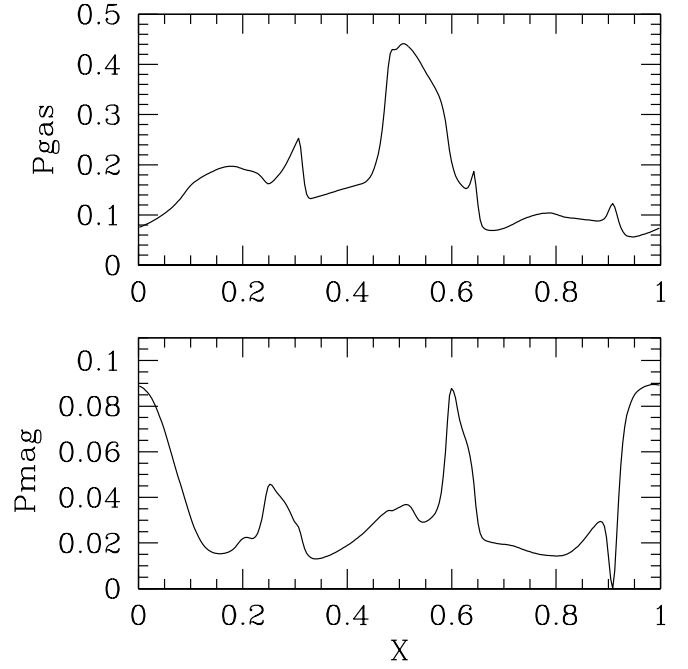

FIG. 21.— Gas and magnetic pressures vs. x along $y = 0.4277$ at time t = 0.48 sec.

time coordinates as

$$x \equiv \sqrt{3}\kappa z, \qquad (77)$$

and

$$\tau \equiv \left(\frac{4ac\kappa}{\alpha}\right)t, \qquad (78)$$

and introduced dimensionless dependent variables, defined as

$$u(x,\tau) \equiv \left(\frac{c}{4}\right)\left[\frac{E(z,t)}{F_{\mathrm{inc}}}\right], \qquad (79)$$

and

$$v(x,\tau) \equiv \left(\frac{c}{4}\right)\left[\frac{aT^4(z,t)}{F_{\mathrm{inc}}}\right]. \qquad (80)$$

In (79) and (80), F_{inc} is the incident boundary flux. With the definitions given by (77) through (80), Pomraning showed that the radiation and gas energy equations could be rewritten, respectively, as

$$\epsilon \frac{\partial u(x,\tau)}{\partial \tau} - \frac{\partial^2 u(x,\tau)}{\partial x^2} = v(x,\tau) - u(x,\tau), \qquad (81)$$

and

$$\frac{\partial v(x,\tau)}{\partial \tau} = u(x,\tau) - v(x,\tau), \qquad (82)$$

subject to the following boundary conditions:

$$u(0,\tau) - \frac{2}{\sqrt{3}}\frac{\partial u(0,\tau)}{\partial x} = 1, \qquad (83)$$

and

$$u(\infty,\tau) = u(x,0) = v(x,0) = 0. \qquad (84)$$

The user-specified parameter ϵ is related to the radiation constant and specific heat through

$$\epsilon = \frac{4a}{\alpha}. \qquad (85)$$

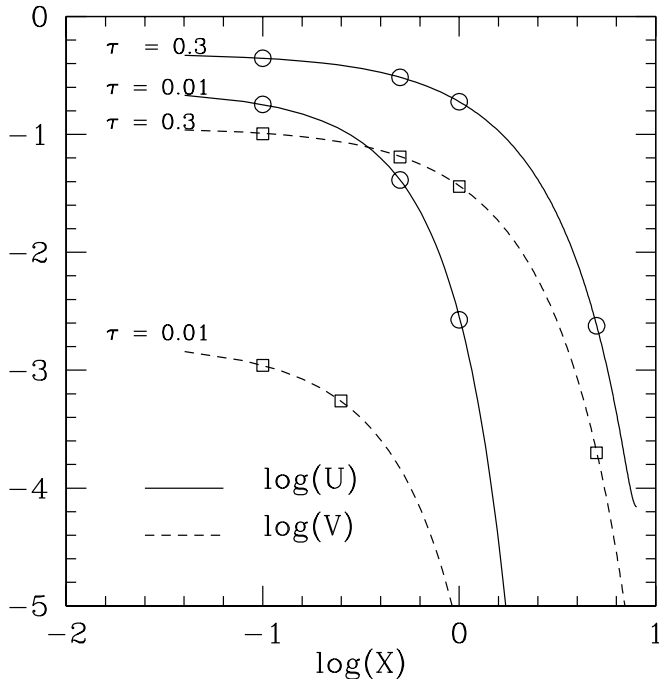With a choice of ϵ, the problem is completely specified and may be solved both numerically and analytically. For

FIG. 22.— Curves of log(U) and log(V) vs. log(X) for 2 values of τ. Curves show analytic solutions; circles indicate numerical data.

the ZEUS-MP test, we chose a 1-D Cartesian grid with 200 zones and a uniform density of 1 g cm^{-3}. The domain length is set to 8 cm, and the photon mean-free path (κ^{-1}) is chosen to be 1.73025 cm. Because this problem was designed for a pure diffusion equation, no flux limiters were used in the FLD module. ϵ was chosen to be 0.1, allowing direct comparison between our results and those given by Su & Olson (1996). Our results are shown in Figure 22, in which the dimensionless energy variables u and v are plotted against the dimensionless space coordinate x at two different values of the dimensionless time, τ. The open circles indicate benchmark data taken from the tabulated solutions of Su & Olson (1996); solid curves indicate ZEUS-MP results. The agreement is excellent.

4.3.2. *Radiating Shock Waves*

The classic text on the theory of shock waves and associated radiative phenomena is due to Zel'Dovich & Raizer (1967) (see also Zel'Dovich & Raizer (1969) for a short review article on shock waves and radiation). A more recent summary of basic concepts is available in Mihalas & Mihalas (1984). Radiating shock waves differ qualitatively from their purely hydrodynamic counterparts due the presence of a radiative precursor created by radiative preheating of material upstream from the shock front. The existence of this precursor gives rise to the identification of so-called *subcritical* and *supercritical* radiating shocks, which are distinguished by a comparison of the gas temperature behind the shock front to that in the material immediately upstream from the shock. In the case of subcritical shocks, the post-shock gas temperature exceeds the upstream value, and the radiative precursor is relatively weak. As the shock velocity is increased beyond a critical value, however, the upstream gas temperature becomes equal to (but never exceeds) the post-shock temperature; such shocks show very strong radiative preheating of the unshocked gas and are identified
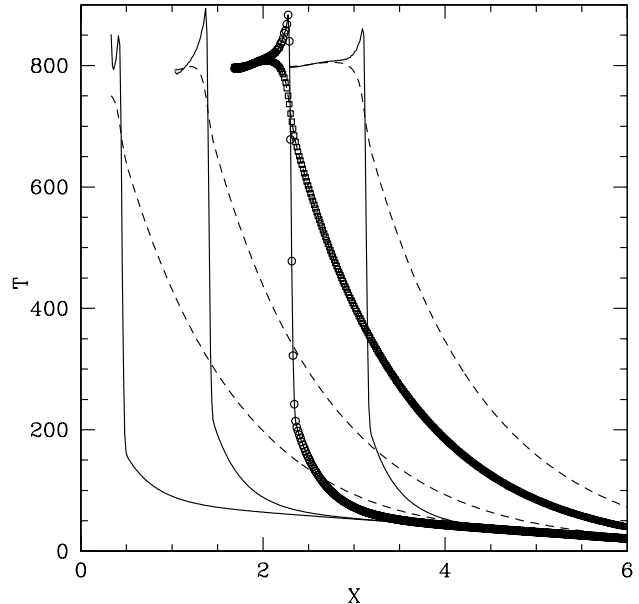
FIG. 23.— Subcritical radiating shock; matter and radiation temperatures vs. comoving X coordinate. Plot times are 5400, 1.7×10^4, 2.8×10^4, and 3.8×10^4 seconds.
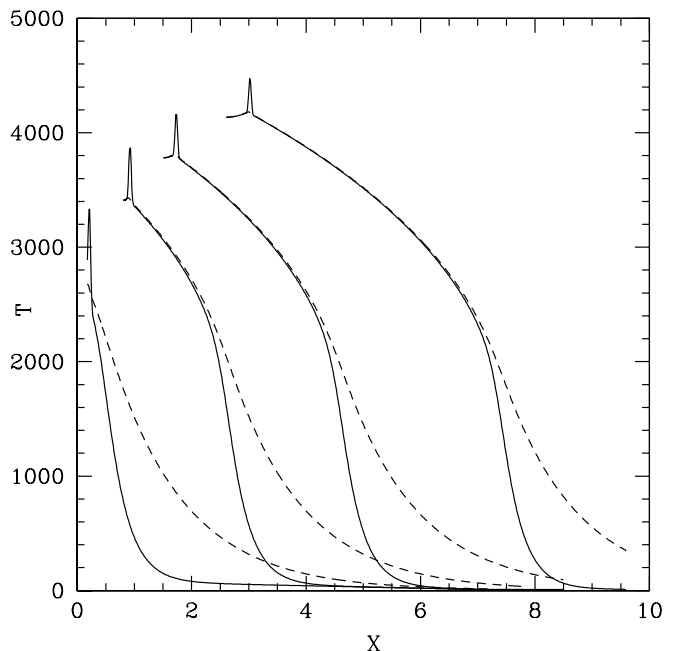
FIG. 24.— Supercritical radiating shock; matter and radiation temperatures vs. comoving X coordinate. Plot times are 860, 4000, 7500, and 1.3×10^4 seconds.

as supercritical shocks.

A numerical prescription for radiating shock test problems appropriate for astrophysical simulation codes was published by Ensman (1994); this configuration was revisited by Gehmeyr & Mihalas (1994) and again by Sincell et al. (1999a,b) and Hayes & Norman (2003). In this model, a domain of length or radius 7×10^{10} cm and an initially uniform density of 7.78×10^{-8} g cm^{-3} is given an initial temperature profile such that T falls smoothly from a value of 85 K at the inner boundary to 10 K at the outer boundary. The non-zero gradient

was necessary to avoid numerical difficulties in Ensman's VISPHOT code. A constant opacity of 3.1×10^{-10} is chosen, which yields a photon mean-free path roughly 5% of the domain length. Because the VISPHOT code uses a Lagrangean mesh, the shock is created by a "piston" affected by choosing an inner boundary condition on the fluid velocity. ZEUS-MP recreates this condition on an Eulerian grid by initializing the fluid velocity throughout the domain and outer boundary to the (negative of) the required piston velocity. The subcritical shock and supercritical shock tests share all problem parameters save for the piston velocity, chosen to be 6 km/s in the former case and 20 km/s for the latter. 512 zones were used to execute the problem on a 1-D mesh.

Figures 23 and 24 present temperature profiles for the subcritical and supercritical cases, respectively. To aid comparison of our Eulerian results to the Lagrangean results of Ensman (1994), we transform the coordinate axis into the rest frame of the unshocked matter. Solid lines indicate gas temperature; dashed lines indicate a radiation "temperature" defined by $T_r \equiv (E/a_r)^{1/4}$, where a_r is the radiation constant. Note the strongly preheated material ahead of the shock front in the supercritical case. Our results were computed in Cartesian geometry; those from Ensman (1994) were computed in a thin spherical shell with a large radius of curvature. This problem was also treated by Hayes & Norman (2003) using ZEUS-MP coupled to a parallel VTEF algorithm. Because that code was designed specifically for 2-D cylindrically-symmetric problems, the problem geometry was different in that the radiating surface was planar yet of finite transverse extent, whereas these results consider a formally infinite plane. This difference results in somewhat different peak values for temperature, but otherwise the results are qualitatively consistent.

5. PERFORMANCE TESTS

Section 4 considered test problems which gauge the accuracy of the code. This section considers issues of numerical performance, with a particular emphasis on problems distributed among large numbers of parallel processors.

5.1. *Aspects of Scalability*

The topic of parallel performance is most often encapsulated in the notion of *scalability*, which in this context is typically assessed by measuring the reduction in CPU time for a given quantity of numerical work as this work is distributed among an increasing number of processors. Relative to the cost on one CPU, perfect scalability would be represented by a cost reduction factor of $1/N$ when the same job is distributed across N processors. For tasks in which each processor can operate upon its portion of data independently of all other processors (a so-called *embarrassingly parallel* operation), perfect scalability is trivially achieved. Algorithms which compute solutions to spatially-discretized PDE's are by construction not embarrassingly parallel because the discrete spatial derivative operators employ stencils that overlap processor tile boundaries along the tile edges. On distributed-memory computers, data communication will therefore be required. Efficient management of this

TABLE 2
RADIATION DIFFUSION TEST PARAMETERS.

Medium	Outer Radius (cm)	ρ (g/cm^3)	Temperature (eV)
D-T gas	0.087	0.025	0.025
D-T ice	0.095	0.25	0.025
C-H foam	0.111	1.20	0.025
He gas	0.490	0.01	300

communication is thus a key ingredient to an efficient parallelization strategy.

More generally, scalability describes the sensitivity of an algorithm's CPU cost to a number of factors, of which parallelism is a leading but by no means unique member. Section 3.8.2 compared the cost of MG linear solvers to traditional stationary methods for a given problem size; the costs of the two methods exhibit very different dependencies upon the number of unknowns in the linear system. For iterative methods such as CG and MG, the required number of iterations for a converged solution is the primary factor in algorithm cost. Solvers whose iteration counts vary more weakly with problem size are to be favored for very large problems. In ideal cases, the required number of iterations for convergence of an MG solver can be virtually independent of the problem size, thus MG is often said to scale well to large problems. Because this behavior is orthogonal to the issue of parallel decomposition, we identify this as an independent definition of scalability.

An additional factor bearing on the cost of an iterative linear system solution is diagonal dominance, a condition in which matrix elements along the main diagonal are much larger in magnitude than off-diagonal elements along a given row. Matrices resulting from the discretization of the time-dependent diffusion equation exhibit diagonal dominance that varies directly with the size of the time step. To see this, consider the static diffusion equation discretized on a 1-D mesh with uniform spacing, Δx. For a static medium, the radiation energy equation becomes

$$\frac{\partial E}{\partial t} - \nabla \cdot D \nabla E = S, \qquad (86)$$

where S contains local source terms. We assume a spatially uniform medium (yielding a spatially constant D) and write this equation in discrete form as

$$\frac{E_i^{n+1} - E_i^n}{\Delta t} - \left[\frac{D}{(\Delta x)^2}\right]\left(E_{i+1}^{n+1} - 2E_i^{n+1} + E_{i-1}^{n+1}\right) = S_i, \qquad (87)$$

which may be rearranged to make the linear system structure obvious:

$$-\left[\frac{D\Delta t}{(\Delta x)^2}\right]E_{i-1}^{n+1} + \left[1 + \frac{2D\Delta t}{(\Delta x)^2}\right]E_i^{n+1}$$
$$-\left[\frac{D\Delta t}{(\Delta x)^2}\right]E_{i+1}^{n+1} = E_i^n + S_i\Delta t. \qquad (88)$$

The relationship between time step and diagonal dominance is manifest: in the limit that $\Delta t \to 0$, the linear system represented by (88) reduces to the identity matrix! In the opposite limit, the off-diagonal elements are
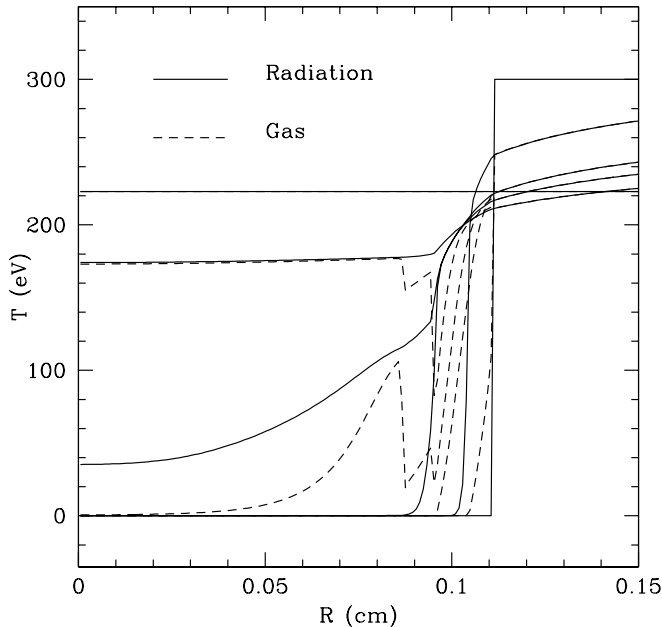
FIG. 25.— Radiation diffusion (1D): profiles of radiation temperature (solid lines) and gas temperature (dashed lines) at times of 0.0, 0.1, 0.5, 0.7, 1.0, and 10 ns. The initial profiles are step functions with the discontinuities located at R = 0.11 cm. The radiation wave propagates leftward, with the gas temperature lagging the radiation temperature, particularly in the optically thick regions.
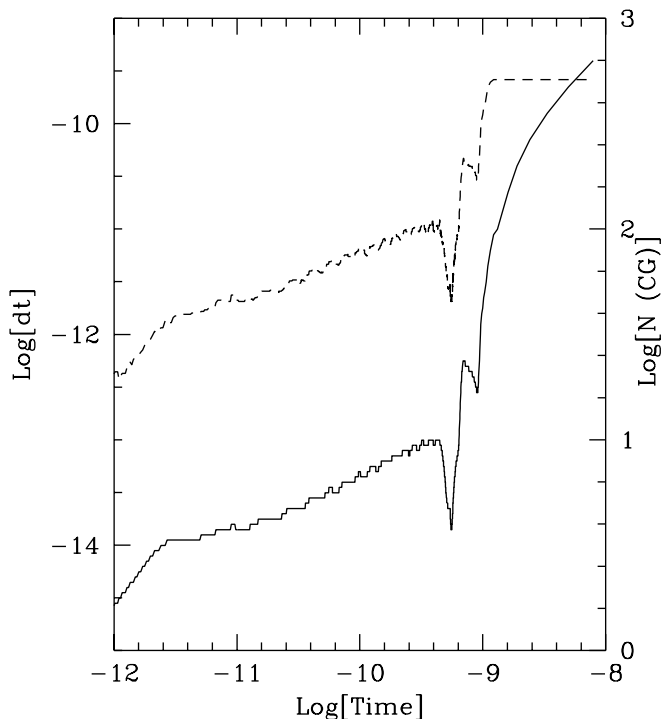

FIG. 26.— Radiation diffusion (1D): logarithms of the time step (solid line) and mean number of CG iterations per N-R iteration (dashed line) as functions of the evolution time. Times are measured in seconds.

comparable in magnitude to the main diagonal, a situation which results in greatly increased numbers of iterations required for convergence in a CG linear solver such as that implemented in our FLD module.

We demonstrate this behavior with a radiation diffusion problem in which a cold sphere is immersed in a

high-temperature radiation field. Problem parameters are given in Table 2. The initial values of density and temperature are taken from a test problem given in Hayes & Norman (1999) designed to qualitatively mimic features of an Inertial Confinement Fusion (ICF) simulation such as that used by Baldwin et al. (1999) to compare numerical performance of CG-based and MG-based linear system solvers. The irradiated sphere is constructed of layers with strongly disparate densities and photon mean-free paths. The actual physical system this problem imitates (albeit crudely) is a sphere of D-T gas surrounded by a solid D-T "ice" of higher density, itself surrounded by a carbon-hydrogen foam of yet higher density. This assembly is immersed in a low density He gas subjected at t = 0 to an intense radiation field with a characteristic temperature of 300 eV. Opacities for real ICF materials have complex dependencies on energy and composition; our toy problem captures the gross features of the mean-free path variation via the following expression:

$$\Lambda(\rho, T) = \Lambda_0 \left(\frac{\rho}{\rho_0}\right)^\nu \left(\frac{T}{T_0}\right)^\mu, \qquad (89)$$

with Λ_0, ρ_0, T_0, ν, and μ given by 10^{-6}, 1.2 g/cm^3, 0.025 eV, 2.0, and 1.2, respectively. A further restriction is placed on the resulting opacities such that the minimum and maximum allowed values of χ are 10 and 10^6 cm^{-1}. This restriction filters out unphysically high and low values of the absorption coefficients. The important feature of this problem is that low-density gas is surrounded by two solids of much higher (and differing) densities. This construction results in an inward-propagating radiation diffusion wave with a highly variable rate of progress. Snapshots of the gas and radiation temperatures at times of 0, 0.1, 0.5, 0.7, 1.0, and 10 nanoseconds are given in Figure 25. Histories of the time step and average number of CG iterations required for NR iteration in the FLD module are plotted against evolution time in Figure 26. The choppy appearance of the plots is an artifact of sampling (every tenth cycle was archived for plotting). For static diffusion problems or RHD problems characterized by rapidly time-varying radiation fields, evolution of the time step will be strongly constrained by the maximum allowed fractional change in the radiation energy (= 0.01 in this problem). The initial time step progression is upward as the exterior radiation field slowly diffuses through the opaque foam layer. By 0.5 ns, the radiation has diffused through the foam layer and begun to penetrate the less opaque D-T ice layer, at which time the time step drops sharply owing to the more rapid evolution of the radiation energy. The time step trends upward again until the radiation wave breaks through the D-T gas/ice boundary. The time step then drops again as the radiation streams into the central region. The final evolution of the time step is upward as the problem domain reaches its final equilibrium. The equilibrium temperature is lower than the initial exterior value because a reflecting outer boundary was chosen rather than an imposed boundary flux.

The dashed line in Figure 26 shows that the number of CG linear system iterations required to solve the FLD matrix during an outer N-R iteration closely parrots the time step behavior. This is understood as a natural consequence of time-dependent diagonal domi-
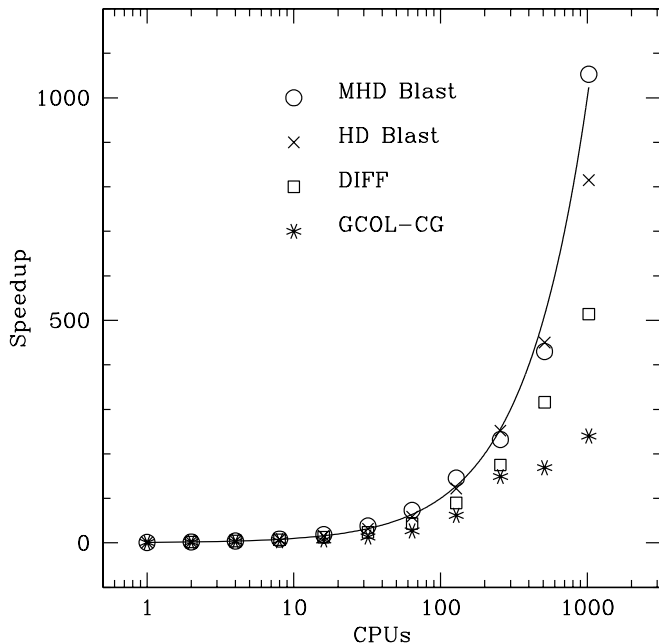
Fig. 27.— Strong-scaling comparison: relative speedup vs. number of CPUs for problems of fixed size. The solid curves indicates perfect N-fold scaling with CPU count. 256^3 zones were used for each problem; the maximum CPU number was 1024 for each case.

nance of the matrix as illustrated by equation 88. This exercise demonstrates the existence of a third dimension of scalability of particular relevance to time-dependent simulations: the dependence of CPU cost upon time step size. When a linear system is very nearly diagonal, both CG and MG will converge rapidly, but because a single iteration of full MG is more expensive than a single CG iteration, one may expect situations in which CG presents a more economical solution strategy for a given problem size. That increasing time step size could provide a "cross over" point with regard to optimal method is a logical consequence. This issue was investigated extensively by Baldwin et al. (1999) in the context of 2-D RHD simulations in spherical geometry. They considered RHD problems in which the time step varied naturally by orders of magnitude during the course of the simulation, and noted indeed that no one method provided the best economy over the entire calculation. While adaptive selection of linear solvers in a particular physics module has not been implemented in ZEUS-MP, we note that experimention along such lines in the context of astrophysical problems is an enticing candidate for future research.

5.2. *Parallel Performance Results*

We explore additional aspects of algorithm performance with a quartet of test problems computed on 3-D grids with 256^3 zones. The first two problems used are non-magnetic and magnetic variants of a simple blast wave test in which a sphere with initial overdensity and overpressure ratios of 100 and 10^6 is defined with respect to a uniform background medium. The problem is defined on a Cartesian grid. The magnetic version augments the problem setup with a uniform magnetic field aligned with the Z axis. The third problem is a 3-D calculation of radiation diffusion into an ICF capsule, with problem parameters as given previously. The fourth problem is the gravitational collapse of a pressureless cloud, using problem parameters given in Stone

& Norman (1992a).

For each problem, parallel performance is measured by a so-called *strong-scaling* test in which the total number of zones (and therefore the total amount of computational work) is held constant as the problem is repeated with increasing numbers of CPU's. Each problem is run for a small number of cycles (typically 30 to 50) which is held fixed for each trial. Figure 27 and Tables 3, 4, 5, and 6 summarize the results for the MHD blast wave, HD blast wave, radiation diffusion, and gravitational collapse tests. The number of timesteps for which each test was run is indicated in the title of each table, from which single time step costs may be derived. In this example, the gravitational collapse problem solved Poisson's equation with the CG linear solver, which is also used in the diffusion test. It is important to note that in this type of scaling study where the total problem size is held fixed, parallel scalability will inevitably break down for a sufficiently large number of processors, due in large part to surface-to-volume effects: when the local processor data block is too small, the communication cost of shipping data along the block's surfaces will compete with the computational cost of processing the full block volume. The number processors necessary to induce a turnover in a code's parallel scalability behavior will depend strongly on the level of communication required by the algorithm, a point we demonstrate in the experiments that follow. A competing technique for measuring scalability, known as a *weak-scaling* test, holds the processor block size constant and thus scales the total problem size with the

TABLE 3
FIXED-WORK SCALING: 256^3 MHD (30 TIME STEPS)

Processors	CPU Time (sec)	Speedup	Parallel Efficiency (%)
1	9391	—	—
2	4624	2.031	102
4	2236	4.200	105
8	1097	8.561	107
16	504.9	18.60	116
32	248.1	37.85	118
64	128.4	73.14	114
128	64.60	145.4	114
256	40.45	232.2	90
512	21.84	430.0	84
1024	8.91	1053	103

TABLE 4
FIXED-WORK SCALING: 256^3 HD (50 TIME STEPS)

Processors	CPU Time (sec)	Speedup	Parallel Efficiency (%)
1	5256	—	—
2	2746	1.91	96
4	1323	3.97	99
8	669.8	7.85	98
16	335.4	15.7	98
32	165.4	31.8	99
64	89.25	58.9	92
128	43.14	122	95
256	20.85	252	98
512	11.69	450	88
1024	6.450	815	80

TABLE 5
FIXED-WORK SCALING: 256^3 FLD (20 TIME STEPS)

Processors	CPU Time (sec)	Speedup	Parallel Efficiency (%)
2	3184	—	—
4	1569	2.03	102
8	887.1	3.59	90
16	509.8	6.25	78
32	269.9	11.8	74
64	146.9	21.7	68
128	71.12	44.8	70
256	36.38	87.5	68
512	20.15	158	62
1024	12.38	257	50

TABLE 6
FIXED-WORK SCALING: 256^3 GRAV-CG (5 TIME STEPS)

Processors	CPU Time (sec)	Speedup	Parallel Efficiency (%)
1	11430	—	—
2	7013	1.63	82
4	3696	3.09	77
8	2300	4.97	62
16	1666	6.86	83
32	866.5	13.2	41
64	422.8	27.0	42
128	184.8	61.9	48
256	76.97	149	58
512	67.47	169	33
1024	47.64	240	23

number of processors. This alternative has some utility: if, for example, one determines that twice the grid resolution is required to satisfy a given accuracy metric, one may investigate if doubling the number of processors along the axis preserves the cost of computing a time step without degrading parallel performance. While this is a relevant consideration, we eschew weak-scaling studies in this paper because (1) with a sufficiently large block of data on each processor, even poor message-passing implementations of parallelism can perform reasonably well, and (2) the characteristics of the problem under study change as the zone number increases. For Courant-limited calculations, doubling the resolution will double the number of time steps needed to complete a calculation, which rather offsets the virtues of maintaining a constant cost per time step. For problems using implicit linear solvers, increasing the total number of zones will, to a degree depending on the solver method, increase the number of iterations required for convergence at each cycle. Strong-scaling studies, while providing a harsher test of a parallel implementation, speak directly to the question: how rapidly may a research problem of a given size be solved?

The behaviors in Figure 27 reflect the relative impact of MPI communication operations on each module. The superlative scaling of the MHD tests derives from the highly computation-intensive nature of the algorithm. The HD test is actually a subset of the MHD problem, as both the MHD-specific routines and the HD advection algorithms must be used in any MHD problem. The
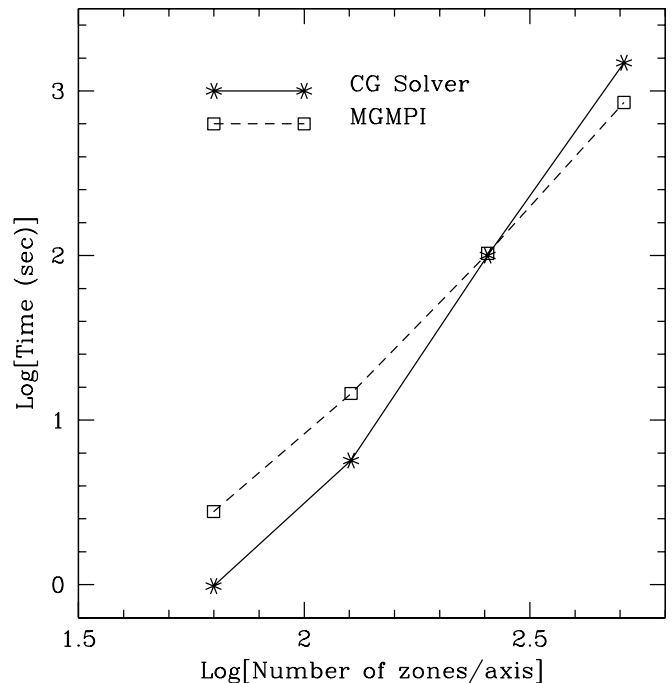

FIG. 28.— A comparison of timings for the gravitational collapse problem in 3-D using the general CG solver and MGMPI's multigrid solver for the Poisson equation.

radiation and gravity problems are both dominated by the cost of CG linear solver. The diffusion problem was run for a sufficiently limited number of time steps such that an average of eight CG iterations were required at each time step. In contrast, when used for the Poisson equation, of order 10^2 iterations are required for a mesh size of 256^3. Because each CG iteration requires both MPI data exchanges at tile boundaries and global searches for error minima, high iteration counts result in very communication-intensive operations. Parallel efficiency, which is computed by dividing the speedup relative to 1 processor by the processor number, is displayed in the fourth column of Tables 3-6. Superlinear speedup is observed most dramatically for the MHD test; this behavior is a by-product of strong-scaling studies and arises because single-CPU performance is degraded when the local data chunk is too large to fit in a processor's cache memory. This effect decreases as the per-CPU data size shrinks; the deleterious effects of communication then begin to appear as the processor counts run into the hundreds. Some of the peculiar variations in parallel efficiency in the MHD example are likely consequences of system and network effects associated with the particular machine used. Memory, bandwidth, and latency characteristic vary tremendously among different architectures; but the major trends shown in Figure 27 and the associated tables are representative and instructive, and are internally consistent with the relative reliance of each module upon data exchange among processors.

The fact that the CG solver requires – irrespective of parallelism – more iterations for larger problems brings with it two liabilities: increased iteration counts boost both the total operation cost *and* the number of MPI messages sent and received. The fact that multi-grid methods exhibit convergence behavior with little sensitivity to problem size motivated us to implement the independently-developed MGMPI package for use as an

alternative Poisson solver in ZEUS-MP. In its current form, MGMPI is restricted to 3-D Cartesian grids (which are nevertheless a common choice in large astrophysical calculations) with Dirichlet or Neumann boundary conditions (recall that the FFTw solver is offered for triply-periodic Cartesian meshes). Figure 28 shows the behavior of solution time against problem size for the gravitational collapse problem computed on a 3-D Cartesian mesh. Grid sizes of 63^3, 127^3, 255^3, and 511^3 zones were run. (Odd numbers of zones are required by the multigrid V-cycle in MGMPI.) Each trial was distributed across 64 processors to ensure that the larger problems would not exceed single-CPU memory limits. At small grid sizes, the CG solver is less expensive than the MGMPI solution, but at a mesh size of 511^3 the CG solution has clearly diverged with respect to the MGMPI solver. The fundamental difference lies in the average number of solver iterations required per time step. For the CG solver, this number was 32, 56, 99, and 190, respectively, for the four problem sizes tested. For MGMPI, this number is 2.2 for the smallest problem and grows only to 3 for the 511^3 run. Despite MGMPI's fairly high operation cost per iteration, the insensitivity of its convergence behavior to problem size guarantees, for a given parallel distribution, a performance advantage over the CG solver for a sufficiently large problem. In its current form, MGMPI does not employ asynchronous MPI calls for its message passing, as does the CG solver. The problem size for which MGMPI enjoys a clear advantage over the CG solver may therefore depend in part on the number of processors chosen. Nonetheless, for very large problems involving self-gravity on a Cartesian mesh, MGMPI is likely to be the preferred option in a ZEUS-MP calculation.

6. SUMMARY

In the introduction, we advertised the theme of this paper as "physics, flexibility, and parallelism." That these features are defining traits of ZEUS-MP is manifest: hydrodynamics, MHD, and radiation diffusion may be deployed, singly or in concert, on Cartesian, cylindrical, or spherical meshes in one to three dimensions. ZEUS-MP demonstrates parallel scalability suitable for computing platforms ranging from small clusters to the largest platforms currently available for unclassified research. Features of a code designed for community use must also include accuracy and computational expediency. The accuracy of ZEUS-MP has been verified both by traditional test problems and a multidimensional MHD problem frequently touted by developers of Godunov-based MHD codes. Even when additional resolution is required to ensure accuracy of a calculation, ZEUS-MP's parallel performance provides a powerful mechanism for keeping the required solution times manageable.

Virtues notwithstanding, we note that there are several ways in which ZEUS-MP may be modified and improved within its solution paradigm. Non-ideal MHD effects such as Ohmic dissipation and ambipolar diffusion are requisite in a variety of topics in interstellar physics such as star formation and interstellar shocks; methodologies for including these effects in the ZEUS framework have been documented by Stone (1999) in the case of Ohmic dissipation and likewise by Stone (1997) for ambipolar diffusion. A 3-D version of the VTEF algorithm

described by Hayes & Norman (2003) would be a major undertaking, but we note an approximation suggested by Ruszkowski & Begelman (2003) as an improvement to FLD suitable for the ZEUS codes. Because ZEUS-MP is intended for public distribution, one mission of this paper is to provide reference documentation at a sufficiently high level of detail so that ambitious code developers may modify it for their particular needs.

An additional area of improvement in which we are currently engaged concerns the iterative solvers offered with the code. The much higher computational cost of simulations with FLD and the CG-based self-gravity module derives from the very high numbers of iterations required for the CG linear solver to converge when the matrix loses its diagonal dominance. Because the matrix generated by the discrete Poisson equation is *never* strongly diagonally dominant, CG methods lose favor as the tool of choice for the Poisson problem on large grids. The suitability of CG to radiation problems is very dependent on the physical and temporal character of the problem at hand. As shown by Baldwin et al. (1999), suitably optimized MG methods may be preferable to CG for some classes of radiation problems. Our current MGMPI solver is not yet flexible enough for use in radiation applications, but enlarging its scope of applicability is a high priority item for future research.

We also note that the convergence requirements of our current CG solver may be dramatically improved with the use of a more effective preconditioner (recall the discussion in §3.8.1). Our solver uses diagonal preconditioning, which simultaneously boasts maximal ease of implementation and minimal range of effectiveness with respect to the condition numbers for which convergence is notably improved. Despite the importance of preconditioning to the performance of linear solvers upon which many astrophysical simulations must depend, this topic has received relatively little attention in the numerical astrophysics literature. One study which has focused on astrophysical applications was performed by Swesty et al. (2004), who considered a class of preconditioners known as "sparse approximate inverse" (SPAI) preconditioners. As the name implies, SPAI preconditioners attempt to construct an approximation to the inverse of a matrix which is more sophisticated than the purely diagonal approximation, but far less expensive to compute than the full inverse. Swesty et al. (2004) have constructed SPAI preconditioners designed for the linearized, discrete, energy-dependent version of the FLD equation, the so-called *multigroup* flux-limited diffusion (MGFLD) equation. The scientific focus in the Swesty et al. (2004) paper is on 2-D and 3-D MGFLD linear systems written to compute multidimensional neutrino diffusion coupled to hydrodynamic flows in core-collapse supernova simulations. While their analysis is designed for the MGFLD equations, they consider special cases of isoenergetic diffusion directly analogous to the reduced system of energy-averaged (or "grey") FLD equations adopted in ZEUS-MP (and ZEUS-2D). The results reported in Swesty et al. (2004) suggest that SPAI preconditioners may offer a very profitable line of research for future FLD implementations in ZEUS-MP or other application codes.

The release of a vastly redesigned and augmented "Version 2" of ZEUS-MP was occassioned by ZEUS-MP's adoption by the Terascale Supernova Initiative as a computational platform for simulating core-collapse supernova explosions in multidimensions. The TSI project, led by Dr. Anthony Mezzacappa, provided both the demand and much of the financial support for the effort which created this code. We are indebted to Tony Mezzacappa for his continued support, enthusiasm, and remarkable patience during the development phase of ZEUS-MP 2.0. Additionally, we gratefully acknowledge the continued support and wise counsel from Dr. Frank Graziani at the Lawrence Livermore National Laboratory. This work was supported by SciDAC grants from the DOE Office of Science High-Energy, Nuclear, and Advanced Scientific Computing Research Programs, and by DOE contract W-7405-ENG-48.

APPENDIX

A. A MAP OF ZEUS-MP

TABLE A1
The Mapping Between ZEUS-MP Subroutines and Equations Solved.

Action	Equation	Routine	Hydro	MHD	RHD
Compute Φ	E21	**GRAVITY**	√	√	√
SRCSTEP Updates					
$v1_{i,j,k}$ force update	B10	**FORCES**	√	√	√
$v2_{i,j,k}$ force update	B11	**FORCES**	√	√	√
$v3_{i,j,k}$ force update	B12	**FORCES**	√	√	√
$v1_{i,j,k}$ viscosity update	B35	**AVISC**	√	√	√
$v2_{i,j,k}$ viscosity update	B36	**AVISC**	√	√	√
$v3_{i,j,k}$ viscosity update	B37	**AVISC**	√	√	√
$e_{i,j,k}$ viscosity update	B38	**AVISC**	√	√	
FLD solution	D15	**GREY_FLD**			√
$e_{i,j,k}$ pdV update	B41	**PDV**	√	√	in **GREY_FLD**
TRANSPRT Updates					
$v1_{i,j,k}$ Lorentz acc.	C58	**LORENTZ**		√	
$v2_{i,j,k}$ Lorentz acc.	C59	**LORENTZ**		√	
$v3_{i,j,k}$ Lorentz acc.	C60	**LORENTZ**		√	
$\mathcal{E}1_{i,j,k}$ update	C34	**HSMOC**		√	
$\mathcal{E}2_{i,j,k}$ update	C35	**HSMOC**		√	
$\mathcal{E}3_{i,j,k}$ update	C33	**HSMOC**		√	
$b1_{i,j,k}$ update	C70	**CT**		√	
$b2_{i,j,k}$ update	C71	**CT**		√	
$b3_{i,j,k}$ update	C72	**CT**		√	
1-Advect $\rho_{i,j,k}$	B44	**TRANX1**	√	√	√
1-Advect $e_{i,j,k}$	B45	**TRANX1**	√	√	√
1-Advect $E_{i,j,k}$	B46	**TRANX1**			√
1-Advect $X_{i,j,k}$	B47	**TRANX1**	√	√	√
1-Advect $v1_{i,j,k}$	B63	**MOMX1**	√	√	√
1-Advect $v2_{i,j,k}$	B64	**MOMX1**	√	√	√
1-Advect $v3_{i,j,k}$	B65	**MOMX1**	√	√	√
2-Advect $\rho_{i,j,k}$	B50	**TRANX2**	√	√	√
2-Advect $e_{i,j,k}$	B51	**TRANX2**	√	√	√
2-Advect $E_{i,j,k}$	B52	**TRANX2**			√
2-Advect $X_{i,j,k}$	B53	**TRANX2**	√	√	√
2-Advect $v1_{i,j,k}$	B69	**MOMX2**	√	√	√
2-Advect $v2_{i,j,k}$	B70	**MOMX2**	√	√	√
2-Advect $v3_{i,j,k}$	B71	**MOMX2**	√	√	√
3-Advect $\rho_{i,j,k}$	B56	**TRANX3**	√	√	√
3-Advect $e_{i,j,k}$	B57	**TRANX3**	√	√	√
3-Advect $E_{i,j,k}$	B58	**TRANX3**			√
3-Advect $X_{i,j,k}$	B59	**TRANX3**	√	√	√
3-Advect $v1_{i,j,k}$	B76	**MOMX3**	√	√	√
3-Advect $v2_{i,j,k}$	B77	**MOMX3**	√	√	√
3-Advect $v3_{i,j,k}$	B78	**MOMX3**	√	√	√
Time Step Control					
New Δt	60	**NUDT**	√	√	√

Table A1 provides a reference listing of the major equations derived in the following appendices and the ZEUS-MP subroutines which compute them. The first three columns of the table indicate the solution substep, pertinent equation, and associated subroutine, respectively. The latter three columns are headed by labels defining three classes of simulation: purely hydrodynamic, MHD, and RHD. In each column a "√" mark indicates that the equation on that line is include in the solution update. Minor headings reading "SRCSTEP" and "TRANSPRT" (which reference subroutines with those names), respectively, indicate the two major groups of solution substeps introduced in section 3

Entries in the table are ordered corresponding to the sequence in which these operations occur during exection, save that advection operations along each coordinate axis in the "TRANSPRT" section are cyclically permuted from one time step to the next.

B. THE 3-D DISCRETE GAS HYDRODYNAMIC EQUATIONS

B.1. *Metric Factors*

ZEUS-MP expresses the discrete fluid equations in the coordinate-independent fashion documented in Stone & Norman (1992a). For convenience, we reproduce the basic metric definitions here. The *metric tensor*, $g_{i,j}$, relates the length, ds, of a line element in one coordinate space, y^k, to the equivalent expression in a second coordinate space, x^i, where we assume that the y^k can be expressed as functions of the x^i. Thus:

$$ds^2 = \left(\frac{\partial y^k}{\partial x^i}\right)\left(\frac{\partial y^k}{\partial x^j}\right) dx^i dx^j \equiv g_{i,j} dx^i dx^j, \tag{B1}$$

where k is summed from 1 to n, where n is the number of dimensions of x. For orthogonal coordinate bases, $g_{i,j}$ is diagonal; following the convention in Stone & Norman (1992a) we write:

$$g_{i,j} = \begin{pmatrix} h_1^2 & 0 & 0 \\ 0 & h_2^2 & 0 \\ 0 & 0 & h_3^2 \end{pmatrix}. \tag{B2}$$

In Cartesian coordinates, we then have

$$(x_1, x_2, x_3) = (x, y, z); \quad (h_1, h_2, h_3) = (1, 1, 1), \tag{B3}$$

while in cylindrical coordinates, we have

$$(x_1, x_2, x_3) = (z, r, \phi); \quad (h_1, h_2, h_3) = (1, 1, r), \tag{B4}$$

and in spherical coordinates, we have

$$(x_1, x_2, x_3) = (r, \theta, \phi); \quad (h_1, h_2, h_3) = (1, r, r\sin\theta). \tag{B5}$$

Following the convention introduced in the ZEUS-2D papers, the h factors are re-expressed as separable functions of g factors which are not to be confused with $g_{i,j}$ defined above:

$$h_1 = 1 \equiv g_1, \tag{B6}$$

$$h_2 = f(x_1) \equiv g_2, \tag{B7}$$

$$h_3 = f(x_1) f(x_3) \equiv g_{31} g_{32}. \tag{B8}$$

The explicit expressions for g_2, g_{31}, and g_{32} are apparent by comparing expressions (B6) - (B8) with (B3) - (B5).

B.2. *Coordinate Meshes*

The staggered-mesh formalism relies upon an "A" mesh, whose points are centered on zone faces, and a "B" mesh, whose points are located at zone centers. The coordinates of the A mesh along each axis are given by $x1a_i$, $x2a_j$, and $x3a_k$, with corresponding arrays for the B mesh. Associated values for the metric coefficients $g2$, $g31$, $g32$, and the derivatives of these coefficients with respect to $x1$ and $x2$ are likewise evaluated on both meshes and stored in 1-D arrays.

In many (but not all) instances, spatial derivatives are written as functions of volume differences rather than coordinate differences. Along the three axes, transformation from coordinate to volume derivatives are written as

$$\begin{aligned} \partial/\partial x_1 &\rightarrow g_2 g_{31} \ \partial/\partial V_1, \\ \partial/\partial x_2 &\rightarrow g_{32} \ \partial/\partial V_2, \\ \partial/\partial x_3 &\rightarrow \partial/\partial V_3. \end{aligned} \tag{B9}$$

Scalar field variables (ρ, e, E, and X(l)) are centered on the B mesh. Velocity and magnetic field vector arrays $(v1_{i,j,k}, v2_{i,j,k}, v3_{i,j,k}); (b1_{i,j,k}, b2_{i,j,k}, b3_{i,j,k})$ are centered on the appropriate zone faces. Magnetic EMF's $(\mathcal{E}1_{i,j,k}, \mathcal{E}2_{i,j,k}, \mathcal{E}3_{i,j,k})$ are defined at midpoints of zone edges.

B.3. *The "Source Step" Equations*

B.3.1. *Body Forces*

In this subsection we document the updates to velocity due to body forces and artificial viscosity, and the updates to internal energy due to artificial viscosity and compressional heating. The three components of fluid velocity are

updated from body forces due to pressure gradients, self-gravity, rotational pseudo-forces, magnetic pressure, and radiation stress according to the following expressions:

$$
\frac{v1_{i,j,k}^{n+a} - v1_{i,j,k}^{n}}{\Delta t} = -\left(\frac{2}{\rho_{i,j,k} + \rho_{i-1,j,k}}\right)\frac{p_{i,j,k} - p_{i-1,j,k}}{\Delta x1 b_i} \ + \ \frac{\Phi_{i,j,k} - \Phi_{i-1,j,k}}{\Delta x1 b_i}
$$

$$
+ \frac{\partial g2a_i}{\partial x1}\left[\mathcal{S}2_i^{oo} + \mathcal{S}2_i^{po} + \mathcal{S}2_{i-1}^{oo} + \mathcal{S}2_{i-1}^{po}\right] \ \times
$$

$$
\left(\frac{1}{8}\right)\frac{\left[v2_{i,j,k}^{n} + v2_{i,j+1,k}^{n} + v2_{i-1,j,k}^{n} + v2_{i-1,j+1,k}^{n}\right]}{g2a_i^2\left(\rho_{i,j,k} + \rho_{i-1,j,k}\right)}
$$

$$
+ \frac{\partial g31a_i}{\partial x1}\left[\mathcal{S}3_i^{oo} + \mathcal{S}3_i^{op} + \mathcal{S}3_{i-1}^{oo} + \mathcal{S}3_{i-1}^{op}\right] \ \times
$$

$$
\left(\frac{1}{8}\right)\frac{\left[v3_{i,j,k}^{n} + v3_{i,j,k}^{n} + v3_{i-1,j,k}^{n} + v3_{i-1,j,k+1}^{n}\right]}{g31a_i^2 g32b_j\left(\rho_{i,j,k} + \rho_{i-1,j,k}\right)}
$$

$$
- \frac{(d1b2_i^{oo} + d1b2_i^{po} + d1b3_i^{oo} + d1b3_i^{op})}{2\left(\rho_{i,j,k} + \rho_{i-1,j,k}\right)\Delta x1 b_i}
$$

$$
- 2\Lambda_{\mathrm{E}}^{(1)}\ \frac{\mathrm{E}_{i,j,k} - \mathrm{E}_{i-1,j,k}}{\left(\rho_{i,j,k} + \rho_{i-1,j,k}\right)\Delta x1 b_i}; \tag{B10}
$$

$$
\frac{v2_{i,j,k}^{n+a} - v2_{i,j,k}^{n}}{\Delta t} = -\left(\frac{2}{\rho_{i,j,k} + \rho_{i,j-1,k}}\right)\frac{p_{i,j,k} - p_{i,j-1,k}}{g2b_i \Delta x2 b_j} \ + \ \frac{\Phi_{i,j,k} - \Phi_{i,j-1,k}}{g2b_i \Delta x2 b_j}
$$

$$
+ \frac{\partial g32a_j}{\partial x2}\left[\mathcal{S}3_i^{oo} + \mathcal{S}3_i^{op} + \mathcal{S}3_i^{mo} + \mathcal{S}3_i^{mp}\right] \ \times
$$

$$
\left(\frac{1}{8}\right)\frac{\left[v3_{i,j,k}^{n} + v3_{i,j,k+1}^{n} + v3_{i,j-1,k}^{n} + v3_{i,j-1,k+1}^{n}\right]}{g31b_i g32a_j^2\left(\rho_{i,j,k} + \rho_{i,j-1,k}\right)}
$$

$$
- \frac{(d2b3_i^{oo} + d2b3_i^{op} + d2b1_i^{oo} + d2b1_i^{po})}{2\left(\rho_{i,j,k} + \rho_{i,j-1,k}\right)g2b_i \Delta x2 b_j}
$$

$$
- 2\Lambda_{\mathrm{E}}^{(2)}\ \frac{\mathrm{E}_{i,j,k} - \mathrm{E}_{i,j-1,k}}{\left(\rho_{i,j,k} + \rho_{i,j-1,k}\right)g2b_i \Delta x2 b_j}; \tag{B11}
$$

$$
\frac{v3_{i,j,k}^{n+a} - v3_{i,j,k}^{n}}{\Delta t} = -\left(\frac{2}{\rho_{i,j,k} + \rho_{i,j,k-1}}\right)\frac{p_{i,j,k} - p_{i,j,k-1}}{g31b_i g32b_j \Delta x3 b_k} \ + \ \frac{\Phi_{i,j,k} - \Phi_{i,j,k-1}}{g31b_i g32b_j \Delta x3 b_k}
$$

$$
- \frac{(d3b1_i^{oo} + d3b1_i^{po} + d3b2_i^{oo} + d3b2_i^{po})}{2\left(\rho_{i,j,k} + \rho_{i,j,k-1}\right)g31b_i g32b_j \Delta x3 b_k}
$$

$$
- 2\Lambda_{\mathrm{E}}^{(3)}\ \frac{\mathrm{E}_{i,j,k} - \mathrm{E}_{i,j,k-1}}{\left(\rho_{i,j,k} + \rho_{i,j,k-1}\right)g31b_i g32b_j \Delta x3 b_k}. \tag{B12}
$$

Equations (B10) through (B12) make use of the following functions in the rotational pseudo-force terms:

$$
\mathcal{S}2_i^{oo} = \left(\frac{1}{2}\right)v2_{i,j,k}^{n}g2b_i\left(\rho_{i,j,k} + \rho_{i,j-1,k}\right), \tag{B13}
$$

$$
\mathcal{S}2_i^{po} = \left(\frac{1}{2}\right)v2_{i,j+1,k}^{n}g2b_i\left(\rho_{i,j+1,k} + \rho_{i,j,k}\right), \tag{B14}
$$

$$
\mathcal{S}3_i^{oo} = \left(\frac{1}{2}\right)v3_{i,j,k}^{n}g31b_i g32b_j\left(\rho_{i,j,k} + \rho_{i,j,k-1}\right), \tag{B15}
$$

$$
\mathcal{S}3_i^{op} = \left(\frac{1}{2}\right)v3_{i,j,k+1}^{n}g31b_i g32b_j\left(\rho_{i,j,k+1} + \rho_{i,j,k}\right), \tag{B16}
$$

$$
\mathcal{S}3_i^{mo} = \left(\frac{1}{2}\right)v3_{i,j-1,k}^{n}g31b_i g32b_{j-1}\left(\rho_{i,j,k} + \rho_{i,j,k-1}\right), \tag{B17}
$$

$$
\mathcal{S}3_i^{mp} = \left(\frac{1}{2}\right)v3_{i,j-1,k+1}^{n}g31b_i g32b_{j-1}\left(\rho_{i,j-1,k+1} + \rho_{i,j-1,k}\right). \tag{B18}
$$

Similarly, the magnetic pressure terms employ the following:

$$
d1b2^{oo} = \frac{\left[(g2b_i b2_{i,j,k})^2 - (g2b_{i-1} b2_{i-1,j,k})^2\right]}{g2a_i^2}, \tag{B19}
$$

$$d1b2^{po} = \frac{\left[(g2b_i b2_{i,j+1,k})^2 - (g2b_{i-1} b2_{i-1,j+1,k})^2\right]}{g2a_i^2}, \tag{B20}$$

$$d1b3^{oo} = \frac{\left[(g31b_i b3_{i,j,k})^2 - (g31b_{i-1} b3_{i-1,j,k})^2\right]}{g31a_i^2}, \tag{B21}$$

$$d1b3^{op} = \frac{\left[(g31b_i b3_{i,j,k+1})^2 - (g31b_{i-1} b3_{i-1,j,k+1})^2\right]}{g31a_i^2}, \tag{B22}$$

$$d2b3^{oo} = \frac{\left[(g32b_j b3_{i,j,k})^2 - (g32b_{j-1} b3_{i,j-1,k})^2\right]}{g32a_j^2}, \tag{B23}$$

$$d2b3^{op} = \frac{\left[(g32b_j b3_{i,j,k+1})^2 - (g32b_{j-1} b3_{i,j-1,k+1})^2\right]}{g32a_j^2}, \tag{B24}$$

$$d2b1^{oo} = (b1_{i,j,k})^2 - (b1_{i,j-1,k})^2, \tag{B25}$$

$$d2b1^{po} = (b1_{i+1,j,k})^2 - (b1_{i+1,j-1,k})^2, \tag{B26}$$

$$d3b1^{oo} = (b1_{i,j,k})^2 - (b1_{i,j,k-1})^2, \tag{B27}$$

$$d3b1^{po} = (b1_{i+1,j,k})^2 - (b1_{i+1,j,k-1})^2, \tag{B28}$$

$$d3b2^{oo} = (b2_{i,j,k})^2 - (b2_{i,j,k-1})^2, \tag{B29}$$

$$d3b2^{op} = (b2_{i,j+1,k})^2 - (b2_{i,j+1,k-1})^2. \tag{B30}$$

B.3.2. Artificial Viscosity

Once the velocity update from forces is complete, velocities which were known at time level "n" are now known at an intermediate time level which we designate as level "$n + a$". These intermediate velocity components are then updated due to the Von Neumann and Richtmyer prescription as follows: define

$$\Delta v1_{i,j,k} = \begin{cases} v1^{n+a}_{i,j,k} - v1^{n+a}_{i-1,j,k}, & v1^{n+a}_{i,j,k} < v1^{n+a}_{i-1,j,k}; \\ 0, & v1^{n+a}_{i,j,k} \geq v1^{n+a}_{i-1,j,k}; \end{cases} \tag{B31}$$

$$\Delta v2_{i,j,k} = \begin{cases} v2^{n+a}_{i,j,k} - v2^{n+a}_{i-1,j,k}, & v2^{n+a}_{i,j,k} < v2^{n+a}_{i,j-1,k}; \\ 0, & v2^{n+a}_{i,j,k} \geq v2^{n+a}_{i,j-1,k}; \end{cases} \tag{B32}$$

$$\Delta v3_{i,j,k} = \begin{cases} v3^{n+a}_{i,j,k} - v3^{n+a}_{i,j,k-1}, & v3^{n+a}_{i,j,k} < v3^{n+a}_{i,j,k-1}; \\ 0, & v3^{n+a}_{i,j,k} \geq v3^{n+a}_{i,j,k-1}; \end{cases} \tag{B33}$$

and

$$q1_{i,j,k} = C_{av}\rho\left(\Delta v1_{i,j,k}\right)^2; \quad q2_{i,j,k} = C_{av}\rho\left(\Delta v2_{i,j,k}\right)^2; \quad q3_{i,j,k} = C_{av}\rho\left(\Delta v3_{i,j,k}\right)^2. \tag{B34}$$

The velocity updates are then computed as

$$v1^{n+b}_{i,j,k} = v1^{n+a}_{i,j,k} - \frac{q1_{i,j,k} - q1_{i-1,j,k}}{\Delta x1b_i\left(\rho_{i,j,k} + \rho_{i-1,j,k}\right)/2}, \tag{B35}$$

$$v2^{n+b}_{i,j,k} = v2^{n+a}_{i,j,k} - \frac{q2_{i,j,k} - q2_{i,j-1,k}}{g2b_i\Delta x2b_j\left(\rho_{i,j,k} + \rho_{i,j-1,k}\right)/2}, \tag{B36}$$

$$v3^{n+b}_{i,j,k} = v3^{n+a}_{i,j,k} - \frac{q3_{i,j,k} - q3_{i,j,k-1}}{g31b_i g32b_j\Delta x3b_k\left(\rho_{i,j,k} + \rho_{i,j,k-1}\right)/2}. \tag{B37}$$

The gas internal energy is simultaneously updated via

$$e^{n+b}_{i,j,k} = e^n_{i,j,k} - \frac{q1_{i,j,k}\Delta v1_{i,j,k}}{dx1a_i} - \frac{q2_{i,j,k}\Delta v2_{i,j,k}}{g2b_i dx2a_j} - \frac{q3_{i,j,k}\Delta v3_{i,j,k}}{g31b_i g32b_j dx3a_k}. \tag{B38}$$

For problems with strong shocks, an additional linear viscosity may used. ZEUS-MP includes a linear viscosity of the form described in Stone & Norman (1992a), in which the linear viscosity depends upon the local sound speed:

$$\text{qlin}_{i,j,k} = C_l\left(\frac{\gamma P}{\rho}\right)^{1/2}\rho\,\Delta v, \tag{B39}$$

where C_l is a constant (typically of order 0.1) and Δv is the difference in neighboring velocities along the coordinate under consideration. As with the quadratic viscosity, $\text{qlin}_{i,j,k}$ is evaluated independently along each axis. The updates to velocity and gas energy are identical to those for the quadratic viscosity save for the replacement of "q" with "qlin" in equations (B35) through (B38).

B.3.3. *Compressional Heating*

For an ideal EOS the *pdV* compressional heating term is evaluated exactly as outline in Stone & Norman (1992a): to improve energy conservation, the updated gas energy can be written as an implicit function of a time-centered pressure, whence

$$\left(e^{n+1} - e^n\right)/\Delta t \; = \; -p^{n+1/2}\nabla\cdot\mathbf{v}, \tag{B40}$$

where $p^{n+1/2} \equiv 0.5\left(p^n + p^{n+1}\right)$. Using the equation of state, $p = (\gamma - 1)e$, (B40) may be rewritten to yield

$$e_{i,j,k}^{n+c} \; = \; \left[\frac{1 - (\Delta t/2)(\gamma - 1)\nabla\cdot\mathbf{v}_{i,j,k}}{1 + (\Delta t/2)(\gamma - 1)\nabla\cdot\mathbf{v}_{i,j,k}}\right]e_{i,j,k}^{n+b}, \tag{B41}$$

where e^{n+b} and e^{n+c} are the gas energies immediately prior to and after the *pdV* update. For non-ideal equations of state, predictor-corrector techniques or Newton-Raphson iterations over temperature may be employed.

B.4. *The "Transport Step" Equations*

In the transport step, ZEUS field variables are advected through the computational mesh using the technique of *consistent transport*, introduced by Norman et al. (1980). Consistent transport attempts to minimize local conservation errors due to numerical diffusion by defining face-centered fluxes of each field variable consistent with the mass flux used to advect the matter density. In this procedure, the quantities advected are the mass density (ρ), the specific internal energy (e/ρ), the specific radiation energy (E/ρ), and the specific momenta $S1 = \rho\,v1$, $S2 = \rho\,g2\,v2$, and $S3 = \rho\,g31\,g32\,v3$. The metric factors introduced into the definitions of $S2$ and $S3$ transform these quantities into angular momenta in curvilinear coordinates.

B.4.1. *Scalar Variables*

We first consider the advection of mass density along the i coordinate. The amount of mass crossing a cell face perpendicular to the i axis in a time step, Δt, is given by

$$\dot{M}_{i,j,k}^1\Delta t \; = \; \tilde{\rho}_{i,j,k}\tilde{A}_i\left(v1_{i,j,k} - vg1_i\right)\Delta t, \tag{B42}$$

where $\tilde{A}_i$ is the time-centered area factor for cell i-face i, and $\tilde{\rho}$ is the matter density average to cell face i. ZEUS-MP uses second-order Van Leer (van Leer 1977) averaging to construct monotonic, upwinded averages of all advected quantities. For advection across the i faces, the time-centered area factor (which accounts for grid motion) is

$$\tilde{A}_i \; = \; g2a_i^{n+1/2}g31a_i^{n+1/2}. \tag{B43}$$

The computed mass flux, $\dot{M}_{i,j,k}^1$, is then used to advect $\rho_{i,j,k}$ according to

$$\rho_{i,j,k}^{n+1} \; = \; \left[\rho_{i,j,k}^n dvl1a_i^n + \left(\dot{M}_{i,j,k}^1 - \dot{M}_{i+1,j,k}^1\right)\Delta t\right]/dvl1a_i^{n+1}. \tag{B44}$$

Consistent transport of the gas and radiation energy densities proceeds by defining specific energies (erg/gm) for each of these quantities, averaging the specific energy to cell faces via Van Leer interpolation, and computing fluxes across each face with the mass fluxes computed in (B42). We thus have:

$$e_{i,j,k}^{n+1} = \left\{e_{i,j,k}^n dvl1a_i^n \; + \; \left[\widetilde{\left(\frac{e}{\rho}\right)}_i\dot{M}_{i,j,k}^1 - \widetilde{\left(\frac{e}{\rho}\right)}_{i+1}\dot{M}_{i+1,j,k}^1\right]\Delta t\right\}/dvl1a_i^{n+1}; \tag{B45}$$

$$E_{i,j,k}^{n+1} = \left\{E_{i,j,k}^n dvl1a_i^n \; + \; \left[\widetilde{\left(\frac{E}{\rho}\right)}_i\dot{M}_{i,j,k}^1 - \widetilde{\left(\frac{E}{\rho}\right)}_{i+1}\dot{M}_{i+1,j,k}^1\right]\Delta t\right\}/dvl1a_i^{n+1}. \tag{B46}$$

The multi-species composition advection uses the X(l) variables to define partial densities, which are then advected and converted back to dimensionless mass fractions. Thus:

$$X(l)_{i,j,k}^{n+1} \; = \; \left[X(l)_{i,j,k}^n\rho_{i,j,k}^n dvl1a_i^n + \left(\widetilde{X}(l)_i\dot{M}_{i,j,k}^1 - \widetilde{X}(l)_{i+1}\dot{M}_{i+1,j,k}^1\right)\Delta t\right]/\left(\rho_{i,j,k}^{n+1}dvl1a_i^{n+1}\right). \tag{B47}$$

For the advection of scalar variables across cell faces perpendicular to the j axis, we write mass fluxes and time-centered face areas as

$$\dot{M}_{i,j,k}^2 \; = \; \tilde{\rho}_{i,j,k}\tilde{A}_j\left(v2_{i,j,k} - vg2_j\right), \tag{B48}$$

and

$$\tilde{A}_j \; = \; g31b_i^n\,g32a_j^{n+1/2}\,dx1a_i^n/dvl1a_i^n. \tag{B49}$$

Advection of ρ, e, E, and X(l) along the j coordinate then proceeds as

$$\rho_{i,j,k}^{n+1} \; = \; \left[\rho_{i,j,k}^n dvl2a_j^n + \left(\dot{M}_{i,j,k}^2 - \dot{M}_{i,j+1,k}^2\right)\Delta t\right]/dvl2a_j^{n+1}, \tag{B50}$$

$$e_{i,j,k}^{n+1} = \left\{ e_{i,j,k}^n dvl2a_j^n \; + \; \left[\left(\widetilde{\frac{e}{\rho}}\right)_j \dot{M}_{i,j,k}^2 - \left(\widetilde{\frac{e}{\rho}}\right)_{j+1} \dot{M}_{i,j+1,k}^2 \right] \Delta t \right\} / dvl2a_j^{n+1}, \tag{B51}$$

$$E_{i,j,k}^{n+1} = \left\{ E_{i,j,k}^n dvl2a_j^n \; + \; \left[\left(\widetilde{\frac{E}{\rho}}\right)_j \dot{M}_{i,j,k}^2 - \left(\widetilde{\frac{E}{\rho}}\right)_{j+1} \dot{M}_{i,j+1,k}^2 \right] \Delta t \right\} / dvl2a_j^{n+1}, \tag{B52}$$

and

$$X(l)_{i,j,k}^{n+1} \; = \; \left[X(l)_{i,j,k}^n \rho_{i,j,k}^n dvl2a_j^n + \left(\widetilde{X}(l)_j \dot{M}_{i,j,k}^2 - \widetilde{X}(l)_{j+1} \dot{M}_{i,j+1,k}^2 \right) \Delta t \right] / \left(\rho_{i,j,k}^{n+1} dvl2a_j^{n+1} \right). \tag{B53}$$

Similarly, the advection of scalar quantities along the k axis is done as follows: define

$$\dot{M}_{i,j,k}^3 \; = \; \tilde{\rho}_{i,j,k} \tilde{A} \left(v3_{i,j,k} - vg3_k \right), \tag{B54}$$

and

$$\tilde{A} \; = \; g2b_i^n \; dx1a_i^n \; dx2a_j^n / \left(dvl1a_i^n \; dvl2a_j^n \right). \tag{B55}$$

Thus

$$\rho_{i,j,k}^{n+1} \; = \; \left[\rho_{i,j,k}^n dvl3a_k^n + \left(\dot{M}_{i,j,k}^3 - \dot{M}_{i,j,k+1}^3 \right) \Delta t \right] / dvl3a_k^{n+1}, \tag{B56}$$

$$e_{i,j,k}^{n+1} = \left\{ e_{i,j,k}^n dvl3a_k^n \; + \; \left[\left(\widetilde{\frac{e}{\rho}}\right)_k \dot{M}_{i,j,k}^3 - \left(\widetilde{\frac{e}{\rho}}\right)_{k+1} \dot{M}_{i,j,k+1}^3 \right] \Delta t \right\} / dvl3a_k^{n+1}, \tag{B57}$$

$$E_{i,j,k}^{n+1} = \left\{ E_{i,j,k}^n dvl3a_k^n \; + \; \left[\left(\widetilde{\frac{E}{\rho}}\right)_k \dot{M}_{i,j,k}^3 - \left(\widetilde{\frac{E}{\rho}}\right)_{k+1} \dot{M}_{i,j,k+1}^3 \right] \Delta t \right\} / dvl3a_k^{n+1}, \tag{B58}$$

and

$$X(l)_{i,j,k}^{n+1} \; = \; \left[X(l)_{i,j,k}^n \rho_{i,j,k}^n dvl3a_k^n + \left(\widetilde{X}(l)_k \dot{M}_{i,j,k}^3 - \widetilde{X}(l)_{k+1} \dot{M}_{i,j,k+1}^3 \right) \Delta t \right] / \left(\rho_{i,j,k}^{n+1} dvl3a_k^{n+1} \right). \tag{B59}$$

B.4.2. Momentum Variables

Each component of the specific momentum is computed (modulo metric factors) by dividing the appropriate velocity component by an arithmetic average of the density at the corresponding cell face. Thus

$$S1_{i,j,k} = 0.5 \; v1_{i,j,k} \left(\rho_{i,j,k}^n + \rho_{i-1,j,k}^n \right), \tag{B60}$$

$$S2_{i,j,k} = 0.5 \; v2_{i,j,k} \left(\rho_{i,j,k}^n + \rho_{i,j-1,k}^n \right) \; g2b_i, \tag{B61}$$

$$S3_{i,j,k} = 0.5 \; v3_{i,j,k} \left(\rho_{i,j,k}^n + \rho_{i,j,k-1}^n \right) \; g31b_i \; g32b_j. \tag{B62}$$

Along the i coordinate, the specific momenta are transported according to the following:

$$S1_{i,j,k}^{n+1} = \left[S1_{i,j,k}^n \; dvl1b_i^n + \dot{S}1_i - \dot{S}1_{i-1} \right] / dvl1b_i^{n+1}; \tag{B63}$$

$$S2_{i,j,k}^{n+1} = \left[S2_{i,j,k}^n \; dvl1a_i^n + \dot{S}2_i - \dot{S}2_{i-1} \right] / dvl1a_i^{n+1}; \tag{B64}$$

$$S3_{i,j,k}^{n+1} = \left[S3_{i,j,k}^n \; dvl1a_i^n + \dot{S}3_i - \dot{S}3_{i-1} \right] / dvl1a_i^{n+1}. \tag{B65}$$

Note that the volume factors used to transport $S2_{i,j,k}$ and $S3_{i,j,k}$ differ from those used to transport $S1_{i,j,k}$ owing to the different centering of $S2_{i,j,k}$ and $S3_{i,j,k}$ with respect to the staggered i mesh. The momentum fluxes are constructed from the previously computed i components of the mass flux as:

$$\dot{S}1_i = \left(\dot{M}_{i,j,k}^1 + \dot{M}_{i+1,j,k}^1 \right) \tilde{v}_1 \left(0.5 \; g2b_i^{n+1/2} \; g31b_i^{n+1/2} \right), \tag{B66}$$

$$\dot{S}2_i = \left(\dot{M}_{i,j-1,k}^1 + \dot{M}_{i,j,k}^2 \right) \tilde{v}_2 \left(0.5 \; g2a_i^n \; g2a_i^{n+1/2} \; g31a_i^{n+1/2} \right), \tag{B67}$$

$$\dot{S}3_i = \left(\dot{M}_{i,j,k-1}^1 + \dot{M}_{i,j,k}^3 \right) \tilde{v}_3 \left(0.5 \; g31a_i^n \; g2a_i^{n+1/2} \; g31a_i^{n+1/2} \right). \tag{B68}$$

In the definition of the momentum fluxes, $\tilde{v}_1$, $\tilde{v}_2$, and $\tilde{v}_3$ denote cell-centered Van Leer averages of the three relative velocity components: $v1_{i,j,k} - vg1_i$, $v2_{i,j,k} - vg2_j$, and $v3_{i,j,k} - vg3_k$.

Along the j coordinate, momentum advection is computed via

$$S1_{i,j,k}^{n+1} = \left[S1_{i,j,k}^n \; dvl2a_j^n + \dot{S}1_j - \dot{S}1_{j+1} \right] / dvl2a_j^{n+1}; \tag{B69}$$

$$S2_{i,j,k}^{n+1} = \left[S2_{i,j,k}^n \; dvl2b_j^n + \dot{S}2_{j-1} - \dot{S}2_j \right] / dvl2b_j^{n+1}; \tag{B70}$$

$$S3_{i,j,k}^{n+1} = \left[S3_{i,j,k}^n \; dvl2a_j^n + \dot{S}3_j - \dot{S}3_{j+1} \right] / dvl2a_j^{n+1}, \tag{B71}$$

with

$$\dot{S}1_j = \left(\dot{M}^2_{i-1,j,k} + \dot{M}^2_{i,j,k} \right) \tilde{v}_1 \left(0.5 \; g31a^n_i \; g32a^{n+1/2}_j \; dx1b^n_i \; dvl1b^n_i \right), \tag{B72}$$

$$\dot{S}2_j = \left(\dot{M}^2_{i,j,k} + \dot{M}^2_{i,j+1,k} \right) \tilde{v}_2 \left(0.5 \; g2b^n_i \; g31b^n_i \; g32b^{n+1/2}_i \; dx1a^n_i \; dvl1a^n_i \right), \tag{B73}$$

$$\dot{S}3_j = \left(\dot{M}^2_{i,j,k-1} + \dot{M}^2_{i,j,k} \right) \tilde{v}_3 \left(0.5 \; (g31b^n_i)^2 \; g32a^n_i \; g32a^{n+1/2}_i \; dx1a^n_i \; dvl1a^n_i \right). \tag{B74}$$

$$\tag{B75}$$

As with the definitions of momentum fluxes along the i axis, the $\tilde{v}$ terms in the j-flux expressions represent Van Leer averages of the relative velocity components, but the numerical values differ owing to the change of axis.

Finally, the k-axis equations for momentum advection are written as

$$S1^{n+1}_{i,j,k} = \left[S1^n_{i,j,k} \; dvl3a^n_k + \dot{S}1_k - \dot{S}1_{k+1} \right] / dvl3a^{n+1}_k; \tag{B76}$$

$$S2^{n+1}_{i,j,k} = \left[S2^n_{i,j,k} \; dvl3a^n_k + \dot{S}2_k - \dot{S}2_{k+1} \right] / dvl3a^{n+1}_k; \tag{B77}$$

$$S3^{n+1}_{i,j,k} = \left[S3^n_{i,j,k} \; dvl3b^n_k + \dot{S}3_{k-1} - \dot{S}3_k \right] / dvl3b^{n+1}_k, \tag{B78}$$

with

$$\dot{S}1_k = \left(\dot{M}^3_{i-1} + \dot{M}^3_i \right) \tilde{v}_1 \left(0.5 \; g2a^n_i \; dx1b^n_i \; dx2a^n_j / dvl1b^n_i \; dvl2a_j \right); \tag{B79}$$

$$\dot{S}2_k = \left(\dot{M}^3_{j-1} + \dot{M}^3_j \right) \tilde{v}_2 \left(0.5 \; (g2b^n_i)^2 \; dx1a^n_i \; dx2b^n_j / dvl1a^n_i \; dvl2b^n_j \right); \tag{B80}$$

$$\dot{S}3_k = \left(\dot{M}^3_k + \dot{M}^3_{k+1} \right) \tilde{v}_3 \left(0.5 \; g31b^n_i \; g2b^n_i \; dx1a^n_i \; g32b^n_j \; dx2a^n_j / dvl1a^n_i \; dvl2a^n_j \right). \tag{B81}$$

C. THE 3-D DISCRETE MHD EQUATIONS

C.1. *Construction of the EMF's*

In a 3-D geometry expressed upon covariant mesh variables, the characteristic equations for Alfvén wave propagation (43) along the 1-axis become

$$\frac{1}{\sqrt{\rho}} \frac{\mathcal{D}}{\mathcal{D}t} \left(b1_{i,j,k} \right) \; \pm \; \frac{\mathcal{D}^\pm}{\mathcal{D}t} \left(v1_{i,j,k} \right) \; = \; \pm \mathcal{S}, \tag{C1}$$

in which the Lagrangean derivative is expanded as

$$\frac{\mathcal{D}^\pm}{\mathcal{D}t} \; = \; \frac{\partial}{\partial t} \; + \; \left(v2 \mp v^{(2)}_A \right) \frac{\partial}{\partial x2} \; + \; \left(v3 \mp v^{(3)}_A \right) \frac{\partial}{\partial x3}, \tag{C2}$$

the Alfvén velocities are given by

$$v^{(2)}_A = b2/\sqrt{\rho}, \tag{C3}$$

$$v^{(3)}_A = b3/\sqrt{\rho}, \tag{C4}$$

and $\mathcal{S}$ is a source term arising from derivatives of the coordinate metric factors ($\equiv 0$ in Cartesian geometry). We difference the temporal derivatives along each characteristic as

$$\mathcal{D}^+[b1] = b1^* - b1^+, \tag{C5}$$

$$\mathcal{D}^-[b1] = b1^* - b1^-, \tag{C6}$$

$$\mathcal{D}^+[v1] = v1^* - v1^+, \tag{C7}$$

$$\mathcal{D}^-[v1] = v1^* - v1^-, \tag{C8}$$

$$\tag{C9}$$

and solve the characteristic equations for $v1^*$ and $b1^*$:

$$b1^* \; = \; \frac{\sqrt{\rho^+ \rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}} \left[\frac{b1^+}{\sqrt{\rho^+}} + \frac{b1^-}{\sqrt{\rho^-}} + v1^+ - v1^- \right], \tag{C10}$$

and

$$v1^* \; = \; \frac{1}{\sqrt{\rho^+} + \sqrt{\rho^-}} \left[v1^+ \sqrt{\rho^+} + v1^- \sqrt{\rho^-} + b1^+ - b1^- \right] + \mathcal{S}\Delta t. \tag{C11}$$

As discussed in the main text, ρ^+ and ρ^- are the densities at the footpoints of the respective characteristics. Equation C1, in view of equations (C2)-(C4), suggests that when evaluating $\mathcal{E}3_{i,j,k}$ as outlined in section 3.2, $v1$ and $b1$

should be upwinded along both the 2- and 3-components of the characteristic velocity. The numerical impracticality of this approach leads us to adopt the approach of Hawley & Stone (1995), in which only partial characteristics are used to upwind velocity and magnetic field components. Quantities upwinded along Alfvén characteristics are then combined with quantities upwinded along hydrodynamic fluid-flow characteristics in a self-consistent fashion. We illustrate the procedure by outlining the calculation of $\epsilon3$, written schematically as

$$\epsilon3 \ = \ v1^*b2^* \ - \ v2^*b1^*. \tag{C12}$$

To aid the documentation, we introduce an "ADV" function in which $\text{ADV}[b2, v1]$ denotes a mean value for b2 computed from a Van Leer average upwinded according to the v1 velocity. This functional notation will be used to describe quantities upwinded along coordinate axes (fluid flow characteristics) or Alfvén characteristics. Our method subdivides into two stages. In stage I, partial characteristics along the 2-axis are used in the construction of values for $v1^*$ and $b1^*$ as follows:
Step Ia: upwind b2 and v2 along 1-axis:

$$\overline{b2}^{(1)} \equiv \text{ADV}[b2, v1]; \tag{C13}$$

$$\overline{v2}^{(1)} \equiv \text{ADV}[v2, v1]. \tag{C14}$$

Step Ib: compute 2-characteristic Alfvén speeds:

$$v^{(2+)} = \overline{v2}^{(1)} \ - \ |\overline{b2}^{(1)}|/\sqrt{\rho^+}; \tag{C15}$$

$$v^{(2-)} = \overline{v2}^{(1)} \ + \ |\overline{b2}^{(1)}|/\sqrt{\rho^-}. \tag{C16}$$

Step Ic: upwind v1 and b1 along the +/- characteristics:

$$\overline{v1}^{(2+)} = \text{ADV}\left[v1, v^{(2+)}\right]; \tag{C17}$$

$$\overline{v1}^{(2-)} = \text{ADV}\left[v1, v^{(2-)}\right]; \tag{C18}$$

$$\overline{b1}^{(2+)} = \text{ADV}\left[b1, v^{(2+)}\right]; \tag{C19}$$

$$\overline{b1}^{(2-)} = \text{ADV}\left[b1, v^{(2-)}\right]. \tag{C20}$$

Step Id: solve the characteristic equations (C10 and C11) for $b1^*$ and $v1^*$:

$$b1^* = \left(\frac{\sqrt{\rho^+\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}\right)\left[\frac{\overline{b1}^{(2+)}}{\sqrt{\rho^+}} + \frac{\overline{b1}^{(2-)}}{\sqrt{\rho^-}} + \overline{v1}^{(2+)} - \overline{v1}^{(2-)}\right]; \tag{C21}$$

$$v1^* = \left(\frac{1}{\sqrt{\rho^+} + \sqrt{\rho^-}}\right)\left[\overline{v1}^{(2+)} \sqrt{\rho^+} \ + \ \overline{v1}^{(2-)} \sqrt{\rho^-} \ + \ \overline{b1}^{(2+)} \ - \ \overline{b1}^{(2-)}\right]$$
$$+ \mathcal{S}\Delta t \tag{C22}$$

Step Ie: compute and store the products $v1^* \ \overline{b2}^{(1)}$ and $\overline{v2}^{(1)} \ b1^*$.
   Stage II is analogous to stage I, except that now we solve for $v2^*$ and $b2^*$ by examining partial characteristics in the 1 direction:
Step IIa:

$$\overline{b1}^{(2)} \equiv \text{ADV}[b1, v2]; \tag{C23}$$

$$\overline{v1}^{(2)} \equiv \text{ADV}[v1, v2]. \tag{C24}$$

Step IIb:

$$v^{(1+)} = \overline{v1}^{(2)} \ - \ |\overline{b1}^{(2)}|/\sqrt{\rho^+}; \tag{C25}$$

$$v^{(1-)} = \overline{v1}^{(2)} \ + \ |\overline{b1}^{(2)}|/\sqrt{\rho^-}. \tag{C26}$$

Step IIc:

$$\overline{v2}^{(1+)} = \text{ADV}\left[v2, v^{(1+)}\right]; \tag{C27}$$

$$\overline{v2}^{(1-)} = \text{ADV}\left[v2, v^{(1-)}\right]; \tag{C28}$$

$$\overline{b2}^{(1+)} = \text{ADV}\left[b2, v^{(1+)}\right]; \tag{C29}$$

$$\overline{b2}^{(1-)} = \text{ADV}\left[b2, v^{(1-)}\right]. \tag{C30}$$

Step IId:

$$b2^* = \left(\frac{\sqrt{\rho^+ \rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}} \right) \left[\frac{\overline{b2}^{(1+)}}{\sqrt{\rho^+}} + \frac{\overline{b2}^{(1-)}}{\sqrt{\rho^-}} + \overline{v2}^{(1+)} - \overline{v2}^{(1-)} \right] ; \tag{C31}$$

$$v2^* = \left(\frac{1}{\sqrt{\rho^+} + \sqrt{\rho^-}} \right) \left[\overline{v2}^{(1+)} \sqrt{\rho^+} + \overline{v2}^{(1-)} \sqrt{\rho^-} + \overline{b2}^{(1+)} - \overline{b2}^{(1-)} \right]$$
$$+ \mathcal{S} \Delta t \tag{C32}$$

Step IIe: compute and store the products $v2^* \, \overline{b1}^{(2)}$ and $\overline{v1}^{(2)} \, b2^*$.

With these two stages complete, we now write the 3-EMF as

$$\epsilon 3 \;=\; \frac{v1^* \overline{b2}^{(1)} + \overline{v1}^{(2)} b2^*}{2} - \frac{v2^* \overline{b1}^{(2)} + \overline{v2}^{(1)} b1^*}{2} . \tag{C33}$$

The 1-emf and 2-emf expressions are derived and expressed analogously as

$$\epsilon 1 \;=\; \frac{v2^* \overline{b3}^{(2)} + \overline{v2}^{(3)} b3^*}{2} - \frac{v3^* \overline{b2}^{(3)} + \overline{v3}^{(2)} b2^*}{2} , \tag{C34}$$

and

$$\epsilon 2 \;=\; \frac{v3^* \overline{b1}^{(3)} + \overline{v3}^{(1)} b1^*}{2} - \frac{v1^* \overline{b3}^{(1)} + \overline{v1}^{(3)} b3^*}{2} . \tag{C35}$$

Because each component of the magnetic field (e.g. $b1$) depends upon EMF's computed around both transverse axes (e.g. $\epsilon 2$ and $\epsilon 3$), the evolution of each B-field component will depend upon the full set of characteristics. This method is effectively a simple directional splitting of the full MOC algorithm. As discussed in Hawley & Stone (1995), each term in a given EMF expression is composed of 1-D advection solutions in which hydrodynamic characteristics are mixed with Alfvén characteristics in a consistent fashion; i.e. in the leading term of equation C33, $b2$ has been passively advected along the same coordinate axis for which the characteristic velocity equation is solved. This consistency is maintained in all terms of the EMF equations. Additional discussion in Hawley & Stone (1995) notes that the practice of consistently mixing partial Alfvén characteristic solutions with hydrodynamic advection retains the relative simplicity of 1-D upwinding yet is less prone to error in the presence of strong magnetic discontinuities.

C.2. *Lorentz Acceleration of Velocities*

The Lorentz accelerations are computed by a procedure analogous to the calculation of the EMF's outlined above. In what follows we make extensive use of the notation introduced in the previous section. We demonstrate the method in detail by writing expressions for the 1-component of the Lorentz acceleration, which depends upon information propagating along Alfvén characteristics in the 2- and 3-directions. Stage I of the solution considers the Alfvén 2-characteristics:

Step Ia: define footpoint densities as

$$\sqrt{\rho^+} \;=\; (\rho_{i,j,k} \cdot \rho_{i-1,j,k})^{1/2} ; \quad \sqrt{\rho^-} \;=\; (\rho_{i,j-1,k} \cdot \rho_{i-1,j-1,k})^{1/2} . \tag{C36}$$

Step Ib: define average of $b2_{i,j,k}$ and compute Alfvén speeds:

$$\overline{b2}_j = 0.5 \, (b2_{i,j,k} + b2_{i-1,j,k}) ; \tag{C37}$$

$$v_A^{(2+)} = -|\overline{b2}| \, / \, \sqrt{\rho^+} ; \tag{C38}$$

$$v_A^{(2-)} = |\overline{b2}| \, / \, \sqrt{\rho^-} . \tag{C39}$$

Step Ic: upwind $b1_{i,j,k}$ and $v1_{i,j,k}$ along the (+) and (-) Alfvén characteristics:

$$b1^{(2+)} = \text{ADV} \left[b1, v_A^{(2+)} \right] ; \tag{C40}$$

$$b1^{(2-)} = \text{ADV} \left[b1, v_A^{(2-)} \right] ; \tag{C41}$$

$$v1^{(2+)} = \text{ADV} \left[v1, v_A^{(2+)} \right] ; \tag{C42}$$

$$v1^{(2-)} = \text{ADV} \left[v1, v_A^{(2-)} \right] . \tag{C43}$$

Step Id: solve characteristic equation for b:

$$b1^{2*} \;=\; \left\{ \frac{\overline{b1}^{(2+)}}{\sqrt{\rho^+}} + \frac{\overline{b1}^{(2-)}}{\sqrt{\rho^-}} + \text{SGN} \left[1, \overline{b2} \right] \cdot \left(\overline{v1}^{(2+)} - \overline{v1}^{(2-)} \right) \right\} \left(\frac{1}{\sqrt{\rho^+}} + \frac{1}{\sqrt{\rho^-}} \right) , \tag{C44}$$

where "SGN $\left[1,\overline{b2}\right]$" is plus or minus 1 depending on the sign of $\overline{b2}$. Finally,
Step Ie: evaluate the first contribution to the Lorentz 1-acceleration:

$$S1_{i,j,k}^{\mathcal{L}} \;=\; \left(\overline{b2}_{j+1} + \overline{b2}_j\right)\left(b1_{j+1}^{2*} - b1_j^{2*}\right)/\left(g2a_i \; dx2a_j\right). \tag{C45}$$

Stage II examines evolution along the Alfvén 3-characteristics as follows:
Step IIa: define footpoint densities as

$$\sqrt{\rho^+} \;=\; \left(\rho_{i,j,k}\cdot\rho_{i,j,k-1}\right)^{1/2}; \quad \sqrt{\rho^-} \;=\; \left(\rho_{i,j,k-1}\cdot\rho_{i-1,j,k-1}\right)^{1/2}. \tag{C46}$$

Step IIb: define average of $b3_{i,j,k}$ and compute Alfvén speeds:

$$\overline{b3}_k = 0.5\left(b3_{i,j,k} + b2_{i-1,j,k}\right); \tag{C47}$$

$$v_A^{(3+)} = -|\overline{b3}| \; / \; \sqrt{\rho^+}; \tag{C48}$$

$$v_A^{(3-)} = |\overline{b3}| \; / \; \sqrt{\rho^-}. \tag{C49}$$

Step IIc: upwind $b1_{i,j,k}$ and $v1_{i,j,k}$ along the (3+) and (3-) Alfvén characteristics:

$$b1^{(3+)} = \text{ADV}\left[b1, v_A^{(3+)}\right]; \tag{C50}$$

$$b1^{(3-)} = \text{ADV}\left[b1, v_A^{(3-)}\right]; \tag{C51}$$

$$v1^{(3+)} = \text{ADV}\left[v1, v_A^{(3+)}\right]; \tag{C52}$$

$$v1^{(3-)} = \text{ADV}\left[v1, v_A^{(3-)}\right]. \tag{C53}$$

Step IId: solve characteristic equation for b:

$$b1^{3*} \;=\; \left\{\frac{\overline{b1}^{(3+)}}{\sqrt{\rho^+}} + \frac{\overline{b1}^{(3-)}}{\sqrt{\rho^-}} + \text{SGN}\left[1,\overline{b3}\right]\cdot\left(\overline{v1}^{(3+)} - \overline{v1}^{(3-)}\right)\right\}\left(\frac{1}{\sqrt{\rho^+}} + \frac{1}{\sqrt{\rho^-}}\right). \tag{C54}$$

Step IIe: add the second contribution to the Lorentz 1-acceleration to the first:

$$\begin{aligned}
S1_{i,j,k}^{\mathcal{L}} = &\left(\overline{b2}_{j+1} + \overline{b2}_j\right)\left(b1_{j+1}^{2*} - b1_j^{2*}\right)/\left(g2a_i \; dx2a_j\right) \\
&+ \left(\overline{b3}_{k+1} + \overline{b3}_k\right)\left(b1_{k+1}^{3*} - b1_k^{3*}\right)/\left(g31a_i \; g32b_j \; dx3a_k\right).
\end{aligned} \tag{C55}$$

The 2- and 3-components of $S^{\mathcal{L}}$ are similarly written as

$$\begin{aligned}
S2_{i,j,k}^{\mathcal{L}} = &\left(\overline{b3}_{k+1} + \overline{b3}_k\right)\left(b2_{j+1}^{3*} - b2_j^{3*}\right)/\left(g31a_i \; g32b_j \; dx3a_k\right) \\
&+ \left(\overline{b1}_{i+1} + \overline{b1}_i\right)\left(b2_{i+1}^{1*} - b2_i^{1*}\right)/\left(g2b_i \; dx1a_i\right);
\end{aligned} \tag{C56}$$

$$\begin{aligned}
S3_{i,j,k}^{\mathcal{L}} = &\left(\overline{b1}_{i+1} + \overline{b1}_i\right)\left(b3_{i+1}^{1*} - b3_i^{1*}\right)/\left(g31b_i \; dx1a_i\right) \\
&+ \left(\overline{b2}_{j+1} + \overline{b2}_j\right)\left(b3_{j+1}^{2*} - b3_j^{2*}\right)/\left(g2a_i \; g32b_j \; dx3a_k\right).
\end{aligned} \tag{C57}$$

With the accelerations thus defined, the fluid velocities are accelerated according to

$$v1_{i,j,k}^{n+c} = v1_{i,j,k}^{n+b} \;+\; \frac{1}{2}\Delta t \cdot S1_{i,j,k}^{\mathcal{L}} \; / \; \sqrt{\rho_{i,j,k}\cdot\rho_{i-1,j,k}}; \tag{C58}$$

$$v2_{i,j,k}^{n+c} = v2_{i,j,k}^{n+b} \;+\; \frac{1}{2}\Delta t \cdot S2_{i,j,k}^{\mathcal{L}} \; / \; \sqrt{\rho_{i,j,k}\cdot\rho_{i,j-1,k}}; \tag{C59}$$

$$v3_{i,j,k}^{n+c} = v3_{i,j,k}^{n+b} \;+\; \frac{1}{2}\Delta t \cdot S3_{i,j,k}^{\mathcal{L}} \; / \; \sqrt{\rho_{i,j,k}\cdot\rho_{i,j,k-1}}, \tag{C60}$$

where the $n+b$ superscript denotes velocities which have been updated in the source step via local body forces (step "a") and artificial viscosity (step "b").

C.3. Evolution of the Field Components

With EMF's suitably computed on each edge of the 3-D grid cell, the three magnetic field components are evolved from time level n to level $n+1$ via the 3-D CT formalism. As with the gas hydrodynamic advection equations, the HSMOCCT algorithm is formulated to account for grid motion along all three coordinate axes, thus special care must be taken to time-center spatial coordinate terms correctly. The line integral equations describing the temporal evolution of the magnetic fluxes through the i, j, and k cell faces were introduced in §3.2; for ease of reference we repeat them here:

$$\frac{\phi1_{i,j,k}^{n+1} - \phi1_{i,j,k}^n}{\Delta t} \;=\; \epsilon2_{i,j,k}\Delta x2_{i,j} + \epsilon3_{i,j+1,k}\Delta x3_{i,j+1,k} - \epsilon2_{i,j,k+1}\Delta x2_{i,j} - \epsilon3_{i,j,k}\Delta x3_{i,j,k}; \tag{C61}$$

$$\frac{\phi2^{n+1}_{i,j,k} - \phi2^n_{i,j,k}}{\Delta t} = \epsilon1_{i,j,k+1}\Delta x1_i + \epsilon3_{i,j+1,k}\Delta x3_{i,j,k} - \epsilon1_{i,j,k}\Delta x1_i - \epsilon3_{i+1,j,k}\Delta x3_{i+1,j,k}; \tag{C62}$$

$$\frac{\phi3^{n+1}_{i,j,k} - \phi3^n_{i,j,k}}{\Delta t} = \epsilon1_{i,j,k}\Delta x1_i + \epsilon2_{i+1,j,k}\Delta x2_{i+1,j} - \epsilon1_{i,j+1,k}\Delta x1_i - \epsilon2_{i,j,k}\Delta x2_{i,j}. \tag{C63}$$

The magnetic fluxes are related to the field components and covariant metric tensor coefficients through the following relations:

$$\phi1 = h_2 \ h_3 \ dx2 \ dx3 \ b1 \ \rightarrow \ g2a_i \ g31a_i \ g32b_j \ dx2a_j \ dx3a_k \ b1_{i,j,k}; \tag{C64}$$

$$\phi2 = h_1 \ h_3 \ dx1 \ dx3 \ b2 \ \rightarrow \ g31b_i \ g32a_j \ dx1a_i \ dx3a_k \ b2_{i,j,k}; \tag{C65}$$

$$\phi3 = h_1 \ h_2 \ dx1 \ dx2 \ b3 \ \rightarrow \ g2b_i \ dx1a_i \ dx2a_j \ b3_{i,j,k}. \tag{C66}$$

Cell edge line elements are transformed to covariant coordinates via

$$\Delta x1 \rightarrow dx1a_i; \tag{C67}$$

$$\Delta x2 \rightarrow g2a_i \ dx2a_j; \tag{C68}$$

$$\Delta x3 \rightarrow g31a_i \ g32a_j \ dx3a_k. \tag{C69}$$

The evolution equations for $b1_{i,j,k}$, $b2_{i,j,k}$, and $b3_{i,j,k}$ are then written as:

$$(\mathcal{A}1_{i,j,k})^{n+1} b1^{n+1}_{i,j,k} = (\mathcal{A}1_{i,j,k})^n b1^n_{i,j,k} + \Delta t \left[\mathcal{E}2_{i,j,k} + \mathcal{E}3_{i,j+1,k} - \mathcal{E}2_{i,j,k+1} - \mathcal{E}3_{i,j,k} \right]; \tag{C70}$$

$$(\mathcal{A}2_{i,j,k})^{n+1} b2^{n+1}_{i,j,k} = (\mathcal{A}2_{i,j,k})^n b2^n_{i,j,k} + \Delta t \left[\mathcal{E}3_{i,j,k} + \mathcal{E}1_{i,j,k+1} - \mathcal{E}3_{i+1,j,k} - \mathcal{E}1_{i,j,k} \right]; \tag{C71}$$

$$(\mathcal{A}3_{i,j,k})^{n+1} b3^{n+1}_{i,j,k} = (\mathcal{A}3_{i,j,k})^n b3^n_{i,j,k} + \Delta t \left[\mathcal{E}1_{i,j,k} + \mathcal{E}2_{i+1,j,k} - \mathcal{E}1_{i,j+1,k} - \mathcal{E}2_{i,j,k} \right]. \tag{C72}$$

Equations (C70) - (C72) make use of area factors, $\mathcal{A}1_{i,j,k}$, $\mathcal{A}2_{i,j,k}$, and $\mathcal{A}3_{i,j,k}$, which are simply the metric coefficients multiplying the corresponding b component in equations (C64) - (C66), evaluated at time level n or $n+1$ according to the associated superscript. In ZEUS-MP, the "emf[1,2,3](i,j,k)" arrays store the $\mathcal{E}$ values indicated in (C70) - (C72), and are given by the true EMF components multiplied by the appropriate time-centered line element:

$$\mathcal{E}1_{i,j,k} = \epsilon1_{i,j,k} \left(dx1a_i \right)^{n+1/2}; \tag{C73}$$

$$\mathcal{E}2_{i,j,k} = \epsilon2_{i,j,k} \left(g2a_i \ dx2a_j \right)^{n+1/2}; \tag{C74}$$

$$\mathcal{E}3_{i,j,k} = \epsilon3_{i,j,k} \left(g31a_i \ g32a_j \ dx3a_k \right)^{n+1/2}. \tag{C75}$$

D. THE 3-D DISCRETE RADIATION DIFFUSION MATRIX

D.1. *Flux Limiters*

We write the three components of radiation flux as

$$\mathbf{F}1 = -D1 \left(\nabla E \right)^1, \tag{D1}$$

$$\mathbf{F}2 = -D2 \left(\nabla E \right)^2, \tag{D2}$$

$$\mathbf{F}3 = -D3 \left(\nabla E \right)^3, \tag{D3}$$

where the quantities $(D1, D2, D3)$ represent flux-limited diffusion coefficients computed independently along each axis, and the superscripts on ∇E indicate the appropriate component of the gradient operator. Recall from equation 11 that each diffusion coefficient takes the following form:

$$D = \left(\frac{c\Lambda_E}{\chi} \right). \tag{D4}$$

ZEUS-MP currently implements two forms of the flux-limiter, Λ_E. The first is due to Levermore & Pomraning (1981), (c.f. equation 28 of their paper):

$$\Lambda_E(LP) = \frac{2+R}{6+3R+R^2}, \tag{D5}$$

where R is given by

$$R \equiv \|\nabla E\|/E. \tag{D6}$$

The second option is a construction derived by Minerbo (1978):

$$\Lambda_E(Mi) = \begin{cases} \frac{2}{3+\sqrt{9+12R^2}}, & R \le 1.5; \\ \frac{1}{1+R+\sqrt{1+2R}}, & R > 1.5; \end{cases} \tag{D7}$$

where R is as defined previously. An important feature of the implementation is that the numerical value of R is *lagged* in time because it is evaluated with converged values of E from the previous time step:

$$R_{i,j,k} = \frac{\|\nabla \mathrm{E}_{i,j,k}^n\|}{\mathrm{E}_{i,j,k}^n}. \tag{D8}$$

This choice preserves the linearity of our discrete solution for $\mathrm{E}_{i,j,k}^{n+1}$.

Because $\nabla\cdot\mathbf{F}$ must be defined at cell centers for consistency with E in (4), the flux components are considered to be centered on cell faces. This introduces an additional subtlety in the computation of diffusion coefficients, as the opacities (χ) and R values (and hence $\nabla \mathrm{E}^n$) must be colocated with $\mathbf{F}$. Thus, while R is manifestly a scalar quantity, the face-centered opacity must be computed from an average of neighboring cell-centered values whose spatial relationship depends upon the cell face in question. Face-centered gradients in E are subject to a similar constraint. At a given cell, each component of flux acquires a (generally) unique value of the E-dependent flux-limiter, which further underscores the simplification gained by time-lagging the evaluation of R as a function of E.

D.2. *The Matrix*

Recall from §3.3 that the discrete radiation and gas energy equations solved in the ZEUS source step are written as

$$f_{i,j,k}^{(1)} = \mathrm{E}_{i,j,k}^{n+1} - \mathrm{E}_{i,j,k}^n - \Delta t \left[4\pi\kappa_\mathrm{P}\mathrm{B}_\mathrm{p} - c\kappa_\mathrm{E}\mathrm{E}_{i,j,k}^{n+1} - \nabla\cdot\mathbf{F}_{i,j,k}^{n+1} - \nabla\mathbf{v} : \mathbf{P}_{i,j,k}^{n+1} \right]; \tag{D9}$$

$$f_{i,j,k}^{(2)} = \mathrm{e}_{i,j,k}^{n+1} - \mathrm{e}_{i,j,k}^n - \Delta t \left[-4\pi\kappa_\mathrm{P}\mathrm{B}_\mathrm{p} + c\kappa_\mathrm{E}\mathrm{E}_{i,j,k}^{n+1} - \mathrm{p}\nabla\cdot\mathbf{v} \right]. \tag{D10}$$

Our derivation of the FLD matrix proceeds by first differentiating equations (D9) and (D10) with respect to $\mathrm{e}_{i,j,k}$ and $\mathrm{E}_{i,j,k}$. Considering first the radiation energy equation, we note that $f_{i,j,k}^{(1)}$ depends on the value of $\mathrm{e}_{i,j,k}$ through the evaluation of B_p, which requires an (in general) energy-dependent material temperature. The dependence of $f_{i,j,k}^{(1)}$ on E is more complex, owing to the flux-divergence term. As will be documented below, $\nabla\cdot\mathbf{F}_{i,j,k}$ is written as a 7-point function in E coupling $\mathrm{E}_{i,j,k}$ to nearest-neighbor values along all 3 coordinate axes. Evaluating the Jacobian for the radiation energy equation will yield a system of the following form:

$$\begin{aligned}
\mathcal{A}_{i,j,k}\delta\mathrm{E}_{i,j,k-1} &+ \mathcal{B}_{i,j,k}\delta\mathrm{E}_{i,j-1,k} + \mathcal{C}_{i,j,k}\delta\mathrm{E}_{i-1,j,k} + \mathcal{D}_{i,j,k}\delta\mathrm{E}_{i,j,k} + \\
\mathcal{E}_{i,j,k}\delta\mathrm{E}_{i+1,j,k} &+ \mathcal{F}_{i,j,k}\delta\mathrm{E}_{i,j+1,k} + \mathcal{G}_{i,j,k}\delta\mathrm{E}_{i,j,k+1} + \mathcal{H}_{i,j,k}\delta\mathrm{e}_{i,j,k} = -f_{i,j,k}^{(1)},
\end{aligned} \tag{D11}$$

where $\mathcal{A}_{i,j,k}$, through $\mathcal{H}_{i,j,k}$ are given by:

$$\begin{aligned}
\mathcal{A}_{i,j,k} &= \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i,j,k-1}^{n+1}}; \quad \mathcal{B}_{i,j,k} = \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i,j-1,k}^{n+1}}; \\
\mathcal{C}_{i,j,k} &= \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i-1,j,k}^{n+1}}; \quad \mathcal{D}_{i,j,k} = \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i,j,k}^{n+1}}; \\
\mathcal{E}_{i,j,k} &= \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i+1,j,k}^{n+1}}; \quad \mathcal{F}_{i,j,k} = \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i,j+1,k}^{n+1}}; \\
\mathcal{G}_{i,j,k} &= \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{E}_{i,j,k+1}^{n+1}}; \quad \mathcal{H}_{i,j,k} = \frac{\partial f_{i,j,k}^{(1)}}{\partial \mathrm{e}_{i,j,k}^{n+1}}.
\end{aligned} \tag{D12}$$

Because the gas energy equation involves no space derivatives in the solution variables, the Jacobian expression is considerably simpler:

$$\frac{\partial f_{i,j,k}^{(2)}}{\partial \mathrm{e}_{i,j,k}}\delta\mathrm{e}_{i,j,k} + \frac{\partial f_{i,j,k}^{(2)}}{\partial \mathrm{E}_{i,j,k}}\delta\mathrm{E}_{i,j,k} = -f_{i,j,k}^{(2)}. \tag{D13}$$

The fact that $f_{i,j,k}^{(2)}$ depends on the gas energy only through $\mathrm{e}_{i,j,k}^{n+1}$ allows $\delta\mathrm{e}_{i,j,k}$ to be written algebraically as

$$\delta\mathrm{e}_{i,j,k} = -\frac{f_{i,j,k}^{(2)} + \left(\partial f_{i,j,k}^{(2)} / \partial \mathrm{E}_{i,j,k}^{n+1} \right) \delta\mathrm{E}_{i,j,k}}{\partial f_{i,j,k}^{(2)} / \partial \mathrm{e}_{i,j,k}^{n+1}}. \tag{D14}$$

Substitution of (D14) into (D11) eliminates the explicit dependence of the radiation energy Jacobian on $\delta\mathrm{e}_{i,j,k}$, resulting in a reduced linear system for the radiation energy corrections:

$$\begin{aligned}
& \mathcal{A}_{i,j,k}\,\delta\mathrm{E}_{i,j,k-1} + \\
& \mathcal{B}_{i,j,k}\,\delta\mathrm{E}_{i,j-1,k} + \\
\mathcal{C}_{i,j,k}\,\delta\mathrm{E}_{i-1,j,k} + & \mathcal{D}_{i,j,k}'\,\delta\mathrm{E}_{i,j,k} + \mathcal{E}_{i,j,k}\,\delta\mathrm{E}_{i+1,j,k} + \\
& \mathcal{F}_{i,j,k}\,\delta\mathrm{E}_{i,j+1,k} + \\
& \mathcal{G}_{i,j,k}\,\delta\mathrm{E}_{i,j,k+1} \\
& \qquad = \mathcal{I}_{i,j,k}f_{i,j,k}^{(2)} - f_{i,j,k}^{(1)},
\end{aligned} \tag{D15}$$

where

$$\mathcal{D}' = \left\{ \frac{\partial f^{(1)}_{i,j,k}}{\partial \mathrm{E}^{n+1}_{i,j,k}} - \frac{\partial f^{(1)}_{i,j,k}}{\partial \mathrm{e}^{n+1}_{i,j,k}} \left(\frac{\partial f^{(2)}_{i,j,k}/\partial \mathrm{E}^{n+1}_{i,j,k}}{\partial f^{(2)}_{i,j,k}/\partial \mathrm{e}^{n+1}_{i,j,k}} \right) \right\}; \tag{D16}$$

$$\mathcal{I} = \left(\frac{\partial f^{(1)}_{i,j,k}/\partial \mathrm{e}^{n+1}_{i,j,k}}{\partial f^{(2)}_{i,j,k}/\partial \mathrm{e}^{n+1}_{i,j,k}} \right). \tag{D17}$$

The $\mathcal{H}$ coefficient of $\delta\mathrm{e}_{i,j,k}$ has been absorbed into $\mathcal{D}'$; coefficients $\mathcal{B}$ and $\mathcal{D}$ through $\mathcal{H}$ remain unchanged. The terms on the LHS of (D15) have been been arranged along multiple lines in a manner illustrating the band structure of the resulting matrix, which is described by a tridiagonal structure coupling points (i-1,j,k), (i,j,k), and (i+1,j,k), accompanied by subdiagonals coupling points (i,j-1,k) and (i,j,k-1) and superdiagonals coupling points (i,j+1,k) and (i,j,k+1).

Equation (D15) is equivalent to a matrix equation of the form $\mathcal{M}\delta\vec{\mathrm{E}} = \vec{\mathcal{R}}$, where $\mathcal{M}$ is a 7-banded matrix whose diagonals are specified by the values of $\mathcal{A}$ through $\mathcal{G}$. As with the covariant form of the Poisson equation matrix (appendix E), $\mathcal{M}$ may be symmetrized by multiplying each row by a total volume element for zone (i,j,k): $\Delta V1a_i\Delta V2a_j\Delta V3a_k$. Written in this way, it is necessary to evaluate (and document) only the five bands $\mathcal{D}'$ through $\mathcal{G}$, and the RHS vector, $\vec{\mathcal{R}}$.

The main diagonal of the symmetrized matrix is given by

$$\mathcal{D}^{sym}_{i,j,k} = \left\{ \frac{\partial f^{(1)}_{i,j,k}}{\partial \mathrm{E}^{n+1}_{i,j,k}} - \frac{\partial f^{(1)}_{i,j,k}}{\partial \mathrm{e}^{n+1}_{i,j,k}} \left(\frac{\partial f^{(2)}_{i,j,k}/\partial \mathrm{E}^{n+1}_{i,j,k}}{\partial f^{(2)}_{i,j,k}/\partial \mathrm{e}^{n+1}_{i,j,k}} \right) \right\} \times \Delta V1a_i\Delta V2a_j\Delta V3a_k. \tag{D18}$$

We evaluate the four required function derivatives as a function of a time-centering parameter, θ, such that $\theta = 1$ gives fully implicit time differencing. (The time step, Δt, is by definition time centered.) We present the derivatives in order of increasing complexity, thus:

$$\frac{\partial f^{(2)}}{\partial \mathrm{E}^{n+1}_{i,j,k}} = -\theta\Delta t c\kappa_{\mathrm{E}}{}^{n+\theta}_{i,j,k}; \tag{D19}$$

$$\frac{\partial f^{(1)}}{\partial \mathrm{e}^{n+1}_{i,j,k}} = -\theta\Delta t \left[4\pi\mathrm{B}_{\mathrm{P}}{}^{n+\theta}_{i,j,k} \left(\frac{\partial \kappa_{\mathrm{P}}}{\partial \mathrm{e}} \right)^{n+\theta}_{i,j,k} + \kappa_{\mathrm{P}}{}^{n+\theta}_{i,j,k} \left(\frac{\partial \mathrm{B}}{\partial \mathrm{e}} \right)^{n+\theta}_{i,j,k} \right]$$
$$+ \theta\Delta t\, c\mathrm{E}^{n+\theta}_{i,j,k} \left(\frac{\partial \kappa_{\mathrm{E}}}{\partial \mathrm{e}} \right)^{n+\theta}_{i,j,k}; \tag{D20}$$

$$\frac{\partial f^{(2)}}{\partial \mathrm{e}^{n+1}_{i,j,k}} = 1 + \theta\Delta t \left[4\pi\mathrm{B}_{\mathrm{P}}{}^{n+\theta}_{i,j,k} \left(\frac{\partial \kappa_{\mathrm{P}}}{\partial \mathrm{e}} \right)^{n+\theta}_{i,j,k} + \kappa_{\mathrm{P}}{}^{n+\theta}_{i,j,k} \left(\frac{\partial \mathrm{B}}{\partial \mathrm{e}} \right)^{n+\theta}_{i,j,k} \right]$$
$$+ \theta\Delta t \left[-c\mathrm{E}^{n+\theta}_{i,j,k} \left(\frac{\partial \kappa_{\mathrm{E}}}{\partial \mathrm{e}} \right)^{n+\theta}_{i,j,k} + p^{n+\theta}_{i,j,k} (\nabla\cdot\mathbf{v})_{i,j,k} \right]. \tag{D21}$$

The final derivative expression is written schematically as

$$\frac{\partial f^{(1)}}{\partial \mathrm{E}^{n+1}_{i,j,k}} = 1 + \theta\Delta t \left[c\kappa_{\mathrm{E}}{}^{n+\theta}_{i,j,k} + \frac{\partial}{\partial \mathrm{E}^{n+1}_{i,j,k}} (\nabla\cdot\mathbf{F}_{i,j,k}) + \frac{\partial}{\partial \mathrm{E}^{n+1}_{i,j,k}} (\nabla\mathbf{v}:\mathbf{P})_{i,j,k} \right]. \tag{D22}$$

Because we assume that $\mathbf{P} = \mathbf{f}\mathrm{E}$, the final term in (D22) is simply $\nabla\mathbf{v}:\mathbf{f}$, where $\mathbf{f}$ is assumed known and held fixed during the N-R iteration. To evaluate $\nabla\cdot\mathbf{F}$, we assume that the three components of $\mathbf{F}$ are given by (D1) - (D3), and we express the divergence operator in covariant coordinates using equation (116) of Stone & Norman (1992a):

$$\nabla\cdot\mathbf{F} = \frac{1}{h_1 h_2 h_3} \left[\frac{\partial}{\partial x_1} (h_2 h_3 \mathbf{F}1) + \frac{\partial}{\partial x_2} (h_1 h_3 \mathbf{F}2) + \frac{\partial}{\partial x_3} (h_1 h_2 \mathbf{F}3) \right]$$
$$\equiv \frac{\partial (g_2 g_{31} \mathbf{F}1)}{\partial V_1} + \frac{1}{g_2} \frac{\partial (g_{32} \mathbf{F}2)}{\partial V_2} + \frac{1}{g_{31} g_{32}} \frac{\partial (\mathbf{F}3)}{\partial V_3}, \tag{D23}$$

where in the latter expression we have transformed spatial derivatives into volume derivatives. A similar operation is performed on each component of the gradient operator:

$$\nabla\mathrm{E} = \left(\frac{1}{h_1} \frac{\partial \mathrm{E}}{\partial x_1}, \frac{1}{h_2} \frac{\partial \mathrm{E}}{\partial x_2}, \frac{1}{h_3} \frac{\partial \mathrm{E}}{\partial x_3} \right)$$
$$\equiv \left(g_2 g_{31} \frac{\partial \mathrm{E}}{\partial V_1}, \frac{g_{32}}{g_2} \frac{\partial \mathrm{E}}{\partial V_2}, \frac{1}{g_{31} g_{32}} \frac{\partial \mathrm{E}}{\partial V_3} \right). \tag{D24}$$

With (D23) and (D24) in hand, we may construct a discrete form of $\nabla\cdot\mathbf{F}$ explicitly in terms of the 7-point stencil in $\mathrm{E}_{i,j,k}$, from which derivatives of $\nabla\cdot\mathbf{F}_{i,j,k}$ with respect to the appropriate E variables may be read by inspection. We provide the result here:

$$\nabla\cdot\mathbf{F}_{i,j,k} = -\frac{(g2a_{i+1}g31a_{i+1})^2 D1_{i+1,j,k}}{\Delta V1a_i}\left(\frac{\mathrm{E}_{i+1,j,k} - \mathrm{E}_{i,j,k}}{\Delta V1b_{i+1}}\right)$$

$$+\frac{(g2a_ig31a_i)^2 D1_{i,j,k}}{\Delta V1a_i}\left(\frac{\mathrm{E}_{i,j,k} - \mathrm{E}_{i-1,j,k}}{\Delta V1b_i}\right)$$

$$-\frac{(g32a_{j+1})^2 D2_{i,j+1,k}}{g2b_i^2\Delta V2a_j}\left(\frac{\mathrm{E}_{i,j+1,k} - \mathrm{E}_{i,j,k}}{\Delta V2b_{j+1}}\right)$$

$$+\frac{(g32a_j)^2 D2_{i,j,k}}{g2b_i^2\Delta V2a_j}\left(\frac{\mathrm{E}_{i,j,k} - \mathrm{E}_{i,j-1,k}}{\Delta V2b_j}\right)$$

$$-\frac{D3_{i,j,k+1}}{(g31b_ig32b_j)^2\Delta V3a_k}\left(\frac{\mathrm{E}_{i,j,k+1} - \mathrm{E}_{i,j,k}}{\Delta V3b_{k+1}}\right)$$

$$+\frac{D3_{i,j,k}}{(g31b_ig32b_j)^2\Delta V3a_k}\left(\frac{\mathrm{E}_{i,j,k} - \mathrm{E}_{i,j,k-1}}{\Delta V3b_k}\right). \tag{D25}$$

The middle term in (D22) then follows at once:

$$\frac{\partial\nabla\cdot\mathbf{F}_{i,j,k}}{\partial\mathrm{E}_{i,j,k}} = \frac{(g2a_{i+1}g31a_{i+1})^2 D1_{i+1,j,k}}{\Delta V1a_i\Delta V1b_{i+1}} + \frac{(g2a_ig31a_i)^2 D1_{i,j,k}}{\Delta V1a_i\Delta V1b_i}$$

$$+\frac{(g32a_{j+1})^2 D2_{i,j+1,k}}{g2b_i^2\Delta V2a_j\Delta V2b_{j+1}} + \frac{(g32a_j)^2 D2_{i,j,k}}{g2b_i^2\Delta V2a_j\Delta V2b_j}$$

$$+\frac{D3_{i,j,k+1}}{(g31b_ig32b_j)^2\Delta V3a_k\Delta V3b_{k+1}} + \frac{D3_{i,j,k}}{(g31b_ig32b_j)^2\Delta V3a_k\Delta V3b_k}. \tag{D26}$$

The three super-diagonal bands of the symmetric matrix, $\mathcal{E}_{i,j,k}$, $\mathcal{F}_{i,j,k}$, and $\mathcal{G}_{i,j,k}$, originate in the derivatives of $\nabla\cdot\mathbf{F}_{i,j,k}$ with respect to $\mathrm{E}_{i+1,j,k}$, $\mathrm{E}_{i,j+1,k}$, and $\mathrm{E}_{i,j,k+1}$, respectively. We therefore have:

$$\mathcal{E}_{i,j,k}^{sym} \equiv \frac{\partial f_{i,j,k}^{(1)}}{\partial\mathrm{E}_{i+1,j,k}}\times\Delta V1a_i\Delta V2a_j\Delta V3a_k$$

$$= -\theta\Delta t\,\frac{(g2a_{i+1}g31a_{i+1})^2 D1_{i+1,j,k}}{\Delta V1b_{i+1}}\times\Delta V2a_j\Delta V3a_k; \tag{D27}$$

$$\mathcal{F}_{i,j,k}^{sym} \equiv \frac{\partial f_{i,j,k}^{(1)}}{\partial\mathrm{E}_{i,j+1,k}}\times\Delta V1a_i\Delta V2a_j\Delta V3a_k$$

$$= -\theta\Delta t\,\frac{(g32a_{j+1})^2 D2_{i,j+1,k}}{g2b_i^2\Delta V2b_{j+1}}\times\Delta V1a_i\Delta V3a_k; \tag{D28}$$

$$\mathcal{G}_{i,j,k}^{sym} \equiv \frac{\partial f_{i,j,k}^{(1)}}{\partial\mathrm{E}_{i,j,k+1}}\times\Delta V1a_i\Delta V2a_j\Delta V3a_k$$

$$= -\theta\Delta t\,\frac{D3_{i,j,k+1}}{(g31b_ig32b_j)^2\Delta V3b_{k+1}}\times\Delta V1a_i\Delta V2a_j. \tag{D29}$$

Finally, the RHS of the symmetrized linear system is evaluated as

$$\mathcal{R}_{i,j,k} = \left\{\left(\frac{\partial f_{i,j,k}^{(1)}/\partial\mathrm{e}_{i,j,k}^{n+1}}{\partial f_{i,j,k}^{(2)}/\partial\mathrm{e}_{i,j,k}^{n+1}}\right) f_{i,j,k}^{(2)} - f_{i,j,k}^{(1)}\right\}\times\Delta V1a_i\Delta V2a_j\Delta V3a_k, \tag{D30}$$

with $\partial f_{i,j,k}^{(1)}/\partial\mathrm{e}_{i,j,k}^{n+1}$ and $\partial f_{i,j,k}^{(2)}/\partial\mathrm{e}_{i,j,k}^{n+1}$ given by equations (D20) and (D21), respectively.

E. THE 3-D DISCRETE POISSON EQUATION MATRIX

The 2-D form of Poisson's equation was written (although not formally derived) in Stone & Norman (1992a); here we extend the discrete expression to 3-D and explicitly derive and document the matrix elements. Following Stone & Norman (1992a), we write the general tensor form of the Laplacian operating on a scalar function, Φ, as

$$\nabla^2\Phi = \frac{1}{h_1h_2h_3}\left[\frac{\partial}{\partial x_1}\left(\frac{h_2h_3}{h_1}\frac{\partial\Phi}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left(\frac{h_1h_3}{h_2}\frac{\partial\Phi}{\partial x_2}\right) + \frac{\partial}{\partial x_3}\left(\frac{h_1h_2}{h_3}\frac{\partial\Phi}{\partial x_3}\right)\right] \tag{E1}$$

The inner partial derivatives of Φ are rewritten as functions of ZEUS metric coefficients:

$$q_1 \equiv g_2 g_{31} g_{32} \frac{\partial \Phi}{\partial x_1}; \quad q_2 \equiv \frac{g_{31} g_{32}}{g_2} \frac{\partial \Phi}{\partial x_2}; \quad q_3 \equiv \frac{g_2}{g_{31} g_{32}} \frac{\partial \Phi}{\partial x_3}; \tag{E2}$$

and the outer derivatives over the "q" functions so defined are transformed into volume derivatives and written in discrete form as

$$\frac{1}{g_{32}} \frac{\partial q_1}{\partial V_1} = \frac{1}{g32b_j} \left(\frac{q1_{i+1} - q1_i}{\Delta V1a_i} \right), \tag{E3}$$

$$\frac{1}{g_2 g_{31}} \frac{\partial q_2}{\partial V_2} = \frac{1}{g2b_i g31b_i} \left(\frac{q2_{j+1} - q2_j}{\Delta V2a_j} \right), \tag{E4}$$

$$\frac{1}{g_2 g_{31} g_{32}} \frac{\partial q_3}{\partial V_3} = \frac{1}{g2b_i g31b_i g32b_j} \left(\frac{q3_{k+1} - q3_k}{\Delta V3a_k} \right). \tag{E5}$$

The derivatives inside of the q functions are left as discrete coordinate differences:

$$q1_i = g2a_i g31a_i g32b_j \left(\Phi_{i,j,k} - \Phi_{i-1,j,k} \right) / \Delta x1b_i, \tag{E6}$$

$$q2_j = (g31b_i g32a_j / g2b_i) \left(\Phi_{i,j,k} - \Phi_{i,j-1,k} \right) / \Delta x2b_j, \tag{E7}$$

$$q3_k = (g2b_i / g31b_i g32b_j) \left(\Phi_{i,j,k} - \Phi_{i,j,k-1} \right) / \Delta x3b_k. \tag{E8}$$

Leaving the inner derivatives as functions of coordinate differences was done for consistency with the formulation in the public ZEUS-2D code. We have also formulated the linear system for the case in which the inner derivatives are also transformed into volume differences. We have not discovered an application in which this distinction has a measurable effect. We therefore adopt the former approach for the purposes of this document. Evaluating (E3) - (E5) with (E6) - (E8) yields

$$\frac{1}{g_{32}} \frac{\partial q_1}{\partial V_1} \to \frac{1}{\Delta V1a_i} \times$$
$$\{ \mathcal{P}1 \Phi_{i+1,j,k} + \mathcal{D}1 \Phi_{i,j,k} + \mathcal{M}1 \Phi_{i-1,j,k} \}, \tag{E9}$$

$$\frac{1}{g_2 g_{31}} \frac{\partial q_2}{\partial V_2} \to \frac{1}{(g2b_i)^2 \Delta V2a_j} \times$$
$$\{ \mathcal{P}2 \Phi_{i,j+1,k} + \mathcal{D}2 \Phi_{i,j,k} + \mathcal{M}2 \Phi_{i,j-1,k} \}, \tag{E10}$$

$$\frac{1}{g_2 g_{31} g_{32}} \frac{\partial q_3}{\partial V_3} \to \left(\frac{1}{g31b_i g32b_j} \right)^2 \left(\frac{1}{\Delta V3a_k} \right) \times$$
$$\{ \mathcal{P}3 \Phi_{i,j,k+1} + \mathcal{D}3 \Phi_{i,j,k} + \mathcal{M}3 \Phi_{i,j,k+1} \}. \tag{E11}$$

In (E9) - (E11), the $\mathcal{P}$, $\mathcal{D}$, and $\mathcal{M}$ functions are written as

$$\mathcal{P}1 = \left(\frac{g2a \, g31a}{\Delta x1b} \right)_{i+1}, \tag{E12}$$

$$\mathcal{D}1 = - \left[\left(\frac{g2a \, g31a}{\Delta x1b} \right)_{i+1} + \left(\frac{g2a \, g31a}{\Delta x1b} \right)_i \right], \tag{E13}$$

$$\mathcal{M}1 = \left(\frac{g2a \, g31a}{\Delta x1b} \right)_i; \tag{E14}$$

$$\mathcal{P}2 = \left(\frac{g32a}{\Delta x2b} \right)_{j+1}, \tag{E15}$$

$$\mathcal{D}2 = - \left[\left(\frac{g32a}{\Delta x2b} \right)_{j+1} + \left(\frac{g32a}{\Delta x2b} \right)_j \right], \tag{E16}$$

$$\mathcal{M}2 = \left(\frac{g32a}{\Delta x2b} \right)_j; \tag{E17}$$

$$\mathcal{P}3 = \left(\frac{1}{\Delta x3b} \right)_{k+1}, \tag{E18}$$

$$\mathcal{D}3 = - \left[\left(\frac{1}{\Delta x3b} \right)_{k+1} + \left(\frac{1}{\Delta x3b} \right)_k \right], \tag{E19}$$

$$\mathcal{M}3 = \left(\frac{1}{\Delta x3b} \right)_k. \tag{E20}$$

The LHS of the discrete Poisson equation may be constructed by a direct summation of expressions (E9) - (E11). Such action results in a 7-banded sparse matrix (cf. equation 58) in which elements along the main diagonal are given by the sum of the 3 $\mathcal{D}$ expressions listed above, multiplied by the inverse volume factor $(1/\Delta V1a)_i$. Similarly, the first superdiagonal band (coupling $\Phi_{i,j,k}$ to $\Phi_{i+1,j,k}$) is given by the $\mathcal{P}1$ expression multiplied by the associated volume factor in (E10). The remaining two superdiagonals and the three subdiagonal bands are derived in analogous fashion. The matrix may be symmetrized, however, if expressions (E9) - (E11) are first multiplied by a total volume element $\Delta V \equiv \Delta V1a_i \Delta V2a_j \Delta V3a_k$. The resulting transpose symmetry allows explicit calculation, storage, and operation upon the three subdiagonals to be avoided. The symmetric linear system may be written symbolically (compare with equation 58) as

$$s_4 \Phi_{i+1,j,k} \; + \; s_5 \Phi_{i,j+1,k} \; + \; s_6 \Phi_{i,j,k+1} +$$
$$s_7 \Phi_{i,j,k} = 4\pi G \left(\Delta V1a_i \Delta V2a_j \Delta V3a_k\right) \rho_{i,j,k}, \tag{E21}$$

with

$$s_4 = \Delta V2a_j \Delta V3a_k \; \times \; \mathcal{D}1 \; + \; \frac{\Delta V1a_i \Delta V3a_k}{(g2b_i)^2} \; \times \; \mathcal{D}2 \; +$$

$$\frac{\Delta V1a_i \Delta V2a_j}{(g31b_i g32b_j)^2} \times \mathcal{D}3, \tag{E22}$$

$$s_5 = \Delta V2a_j \Delta V3a_k \; \times \; \mathcal{P}1, \tag{E23}$$

$$s_6 = \frac{\Delta V1a_i \Delta V3a_k}{(g2b_i)^2} \; \times \; \mathcal{P}2, \tag{E24}$$

$$s_7 = \frac{\Delta V1a_i \Delta V2a_j}{(g31b_i g32b_j)^2} \times \mathcal{P}3. \tag{E25}$$

F. IMPLEMENTATION TECHNIQUES AND STRATEGIES

The ZEUS algorithm solves the partial differential equations describing astrophysical fluid flows by means of an "operator-split" finite difference scheme. The field variables are advanced in time through a series of substeps corresponding to each operator (physical process) contributing to the full evolution equations. Whether the field variables are updated in an explicit or implicit manner, they use values of quantities computed during the previous substep. Therefore, a parallel algorithm in which multiple substeps are executed concurrently is not feasible. Instead, our parallelization strategy is based on domain decomposition, in which the spatial mesh is divided into "tiles" and the field variables are updated in each tile concurrently. Each substep in the time-stepping scheme is completed in all tiles before moving on to the next substep, so that the time levels of all variables remain synchronized between tiles.

Gradients and other spatial derivatives appearing in the evolution equations are approximated by linear combinations of field variable values evaluated at discrete points in a set of several neighboring mesh zones comprising the "stencils" of the difference operators. Evaluating spatial derivatives in mesh zones near tile boundaries requires values of some quantities at locations in zones belonging to neighboring tiles. Therefore, before we can update the field variables in zones near the boundaries of a tile, we must receive some data from neighboring tiles as required by the stencils. We perform the required exchange of data between tiles by means of "message passing", using the MPI library. MPI enables the code to execute efficiently on many types of parallel architectures, from systems with globally shared memory to clusters of workstations.

Optimal paralled efficiency is achieved by minimizing the ratio of communication overhead to computational work (updating field variables). The amount of data that needs to be exchanged between tiles is proportional to the number of zones near tile boundaries (not physical boundaries, unless periodic boundary conditions are prescribed there). We therefore minimize the ratio of the number of zones near tile surfaces to zones in tile interiors by decomposing the domain along each active spatial dimension. We balance the load by assigning nearly the same number of zones to each tile.

Communication overhead involves more than merely the transit time for the messages (which is proportional to message size, i.e., the number of array elements). It also includes network latency (same for any message size), time for the CPUs to copy/pack/unpack the data to be passed, and context switching delays as the CPUs alternate between updating variables and passing messages. Fortunately much of the communication overhead is comprised of idle cycles, some of which can be spent doing other useful work, provided one makes use of the "non-blocking" communications operations in MPI.

One way to reduce communication overhead is to minimize the number of mesages that are sent. Of particular concern is the exchange of data between tiles that share only one corner point. Only a few zones near a corner require any data from tiles sharing only that corner, but each tile has 8 corners, each of which are shared by 7 neighboring tiles. Each tile also has 8 edges which are shared by 3 neighboring tiles. In contrast, each of the 6 tile faces has at most just 1 neighboring face. We can avoid passing a large number of small messages by exchanging messages across tile faces in 3 stages, sending and receiving messages along just one dimension per stage (see Figure F1). We begin the next communication stage only after the previous stage is completed. Data from neighboring tiles is stored in the 2 layers of "ghost" zones on the surfaces of each tile. This ghost cell data is included in all messages and automatically carries edge and corner cell values to the tiles that share only those edges and corners.
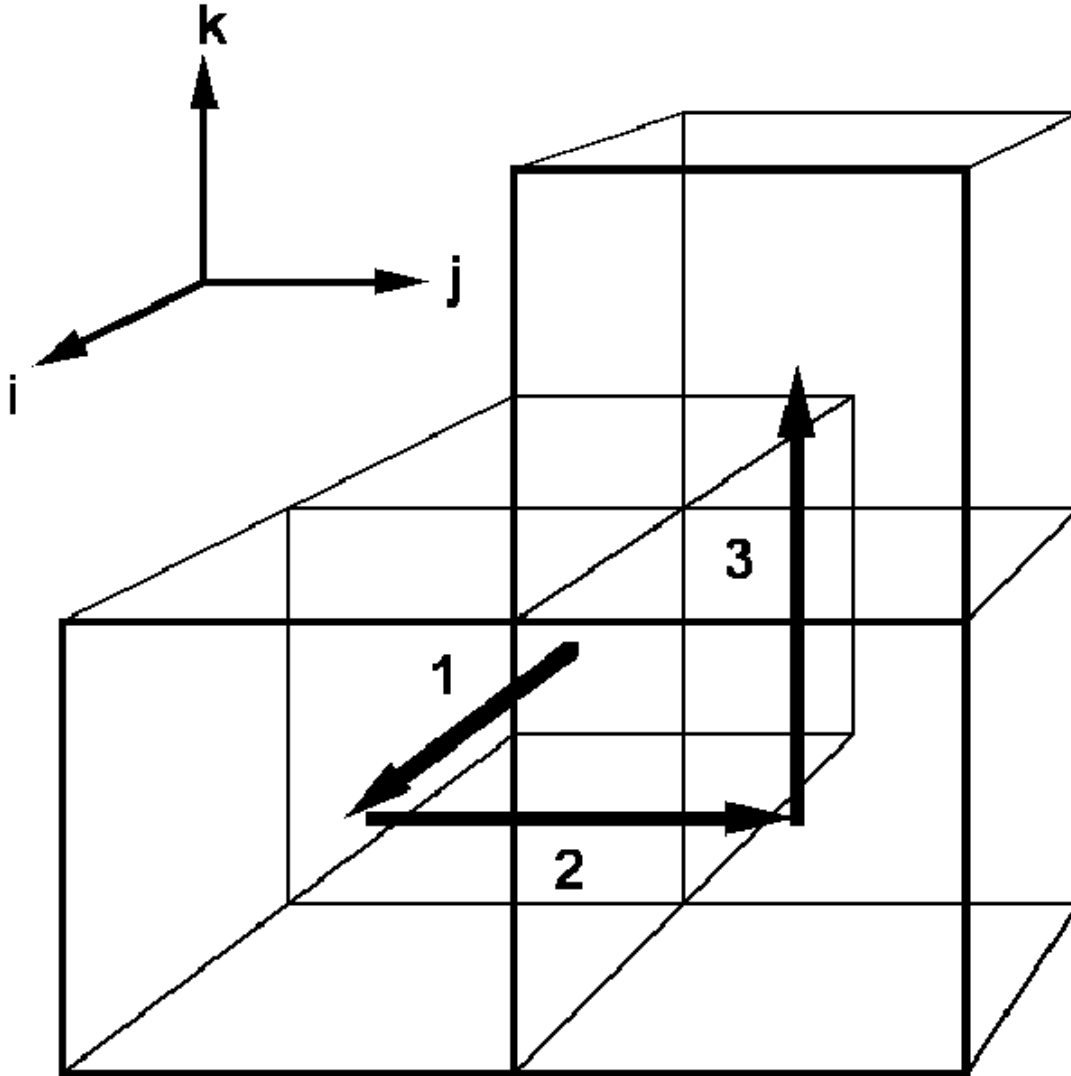
Fig. F1.— MPI communication flow in ZEUS-MP.

In some of the substeps in the ZEUS algorithm, such as advection along the "i" direction for pure hydrodynamics, updating the field variables involves relatively little computational work. In such cases, we employ a more agressive strategy to overlap more communication operations with computations (at the expense of a more complicated code). We subdivide the zones in each tile into 3 roughly equal groups, so that one third of the interior zones can be updated while the messages for each communication stage are in transit. After the messages for a given stage are received, we update field variables in zones near tile boundaries for which all data is available. The precise procedure is as follows:

1. Boundary data is exchanged with neighbors along the i-axis while updates are performed for the 1st third of interior zones.

2. Boundary data is exchanged along the j-axis while updates are performed (a) for the values of i skipped in step (1) in the 1st third of interior zones, and (b) for the 2nd third of interior zones (all i).

3. Boundary data is exchanged along the k-axis while updates are performed (a) for the values of j skipped in step (2) in the first two thirds of interior zones, and (b) for the 3rd third of interior zones (all i and j).

4. Updates are performed for the values of k skipped in the previous steps (all i and j).

The procedure outlined above is adopted for several stages in the source step portion of the hydrodynamics update, including the artificial viscosity, body forces, compressional heating (pdV), and EOS updates. Because the radiation module employs an implicit solution which updates $E_{i,j,k}$ and $e_{i,j,k}$ at all mesh points simultaneously, partial mesh updates are not possible. A related procedure *is* employed, however, in both the subroutines which compute matrix elements and within the CG linear solver routine which returns corrections for $E_{i,j,k}$ during a Newton-Raphson iteration. By construction, the FLD stencil never accesses data which lies outside of a tile boundary along more than one

axis; i.e. "ghost corner" zones are never accessed. Because of this, the asynchronous MPI exchanges of tile faces may be initiated for all three axes simultaneously, since only ghost corner cells depend upon the ordering of face updates. Within the FLD module, therefore, we exploit a simplified procedure in which updates are performed along tile faces, MPI "ISEND" and "IRECV" calls are posted for face data along all axes, interior updates are performed, and an MPI "WAIT" operation is performed to ensure that message-passing has completed before proceeding. This allows considerable opportunity for overlapping communication with computation both in the evaluation of matrix elements and the processing of matrix data during the CG linear solution step.

At the very end of a time step, the size of the next explicit time step is computed from the minimum value of the Courant limit over all zones. MPI and other message-passing libraries provide routines to perform such global reduction operations efficiently in parallel. The time spent waiting for this operation to complete comprises only a fraction of the total communication time.

REFERENCES

Baldwin, C., Brown, P. N., Falgout, R., Graziani, F., & Jones, J. 1999, Journal of Computational Physics, 154, 1

Balsara, D. S. 2004, ApJS, 151, 149

Barret, R., Berry, M., Chan, T., J., D., Donato, J., Dongarra, J., Eijhkout, V., Pozo, R., Romine, C., & Van der Vorst, H. 1994, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (Philadelphia: SIAM)

Bordner, J. 2002, Using MGMPI-1.0, http://cosmos.ucsd.edu/~jbordner/ mgmpi-user-versions/mgmpi-user.ps

Brandt, A. 1977, Math. Comp., 31, 333

Brio, M. & Wu, C. C. 1988, Journal of Computational Physics, 75, 400

Clarke, D. A. 1988, Ph.D. Thesis

Clarke, D. A., Norman, M. L., & Burns, J. O. 1986, ApJ, 311, L63

Dai, W. & Woodward, P. R. 1998, ApJ, 494, 317

Ensman, L. 1994, ApJ, 424, 275

Evans, C. R. & Hawley, J. F. 1988, ApJ, 332, 659

Falle, S. A. E. G. 2002, ApJ, 577, L123

Foster, I. 1995, Designing and Building Parallel Programs (Addison-Wesley)

Frigo, M. & Johnson, S. G. 2005, Proceedings of the IEEE, 93, 216, special issue on "Program Generation, Optimization, and Platform Adaptation"

García-Segura, G., Langer, N., Różyczka, M., & Franco, J. 1999, ApJ, 517, 767

Gardiner, T. A. & Stone, J. M. 2005, Journal of Computational Physics, 205, 509

Gehmeyr, M. & Mihalas, D. 1994, Physica D Nonlinear Phenomena, 77, 320

Hawley, J. F. & Stone, J. M. 1995, Computer Physics Communications, 89, 127

Hayes, J. C. & Norman, M. L. 1999, Scaling and Performance of a 3-D Radiation Hydrodynamics Code on Message-Passing Parallel Computers: Final Report, Tech. Rep. UCRL-CR-136348, Lawrence Livermore National Laboratory

—. 2003, ApJS, 147, 197

Heath, M. T. 1997, SCIENTIFIC COMPUTING: An Introductory Survey (McGraw-Hill)

Levermore, C. D. & Pomraning, G. C. 1981, ApJ, 248, 321

Li, P. S., Norman, M. L., Mac Low, M., & Heitsch, F. 2004, ApJ, 605, 800

Liebendörfer, M., Messer, O. E. B., Mezzacappa, A., Bruenn, S. W., Cardall, C. Y., & Thielemann, F.-K. 2004, ApJS, 150, 263

Londrillo, P. & Del Zanna, L. 2000, ApJ, 530, 508

Londrillo, P. & del Zanna, L. 2004, Journal of Computational Physics, 195, 17

Low, B. C. & Manchester, W. I. 2000, ApJ, 528, 1026

Mac Low, M. 1999, ApJ, 524, 169

Martos, M., Hernandez, X., Yáñez, M., Moreno, E., & Pichardo, B. 2004a, MNRAS, 350, L47

Martos, M., Yañez, M., Hernandez, X., & Moreno, E. P. B. 2004b, Journal of Korean Astronomical Society, 37, 199

Mihalas, D. 1978, Stellar atmospheres /2nd edition/ (San Francisco, W. H. Freeman and Co., 1978. 650 p.)

Mihalas, D. & Mihalas, B. 1984, Foundations of radiation hydrodynamics (New York: Oxford University Press, 1984)

Minerbo, G. N. 1978, Journal of Quantitative Spectroscopy and Radiative Transfer, 20, 541

Mouschovias, T. C. & Paleologou, E. V. 1980, ApJ, 237, 877

Norman, M. L. 1980, Ph.D. Thesis

Norman, M. L., Wilson, J. R., & Barton, R. T. 1980, ApJ, 239, 968

Norman, M. L. & Winkler, K.-H. A. 1986, in NATO Advanced Research Workshop on Astrophysical Radiation Hydrodynamics, ed. K.-H. A. Winkler & M. L. Norman (Norwell, MA: Kluwer Academic Publishers), 187–221

Orszag, S. A. & Tang, C.-M. 1979, Journal of Fluid Mechanics, 90, 129

Pomraning, G. C. 1979, Journal of Quantitative Spectroscopy and Radiative Transfer, 21, 249

Ruszkowski, M. & Begelman, M. C. 2003, ApJ, 586, 384

Ryu, D., Miniati, F., Jones, T. W., & Frank, A. 1998, ApJ, 509, 244

Sedov, L. I. 1959, Similarity and Dimensional Methods in Mechanics (Similarity and Dimensional Methods in Mechanics, New York: Academic Press, 1959)

Sincell, M. W., Gehmeyr, M., & Mihalas, D. 1999a, Shock Waves, 9, 391

—. 1999b, Shock Waves, 9, 403

Stone, J. M. 1997, ApJ, 487, 271

—. 1999, Journal of Computational and Applied Mathematics, 109, 261

Stone, J. M. & Gardiner, T. A. 2005, in AIP Conf. Proc. 784: Magnetic Fields in the Universe: From Laboratory and Stars to Primordial Structures., 16–26

Stone, J. M., Hawley, J. F., Evans, C. R., & Norman, M. L. 1992a, ApJ, 388, 415

Stone, J. M., Mihalas, D., & Norman, M. L. 1992b, ApJS, 80, 819

Stone, J. M. & Norman, M. L. 1992a, ApJS, 80, 753

—. 1992b, ApJS, 80, 791

Su, B. & Olson, G. L. 1996, Journal of Quantitative Spectroscopy and Radiative Transfer, 56, 337

Swesty, F. D., Smolarski, S. J., & Saylor, P. E. 2004, ApJS, 153, 369

Tscharnuter, W. M. & Winkler, K.-H. 1979, Computer Physics Communications, 18, 171

Turner, N. J., Blaes, O. M., Socrates, A., Begelman, M. C., & Davis, S. W. 2005, ApJ, 624, 267

Turner, N. J. & Stone, J. M. 2001, ApJS, 135, 95

van Leer, B. 1977, Journal of Computational Physics, 23, 276

von Neumann, J. & Richtmyer, R. D. 1950, J. Appl. Phys., 21, 232

Winkler, K.-H. A., Norman, M. L., & Mihalas, D. 1984, Journal of Quantitative Spectroscopy and Radiative Transfer, 31, 473

Zel'Dovich, Y. B. & Raizer, Y. P. 1967, Physics of shock waves and high-temperature hydrodynamic phenomena (New York: Academic Press, 1966/1967, edited by Hayes, W.D.; Probstein, Ronald F.)

—. 1969, Annual Review of Fluid Mechanics, 1, 385