JOURNAL OF COMPUTATIONAL PHYSICS **140,** 148–171 (1998)
ARTICLE NO. CP975880

Transport of Magnetic Flux in an Arbitrary Coordinate ALE Code

Robert E. Peterkin Jr., Michael H. Frese,[1] and Carl R. Sovinec[2]

Directed Energy Directorate, Air Force Research Laboratory: Phillips Lab, Kirtland AFB, New Mexico 87117
E-mail: bob@ppws07.plk.af.mil

Received October 28, 1997

We illustrate a new technique for computing the time-evolution of magnetic flux on a generally nonorthogonal computational grid of a time-dependent, arbitrary Lagrangian–Eulerian magnetohydrodynamics (MHD) simulation code and apply this technique to some classical MHD test problems. For a nontrivial application, we demonstrate the power of this technique for the interesting problem of compact toroid translation between a pair of converging conical electrodes.

I. INTRODUCTION

The ability to perform accurate simulation of the time evolution of a conducting magnetofluid is of great utility in a wide variety of space and laboratory situations. Of particular interest to the authors are collisional plasmas that are generated in the laboratory in finite regions defined by solid walls that may be either conductors or insulators. Often, the volume enclosed by the solid walls has a complex shape. A particular example of a laboratory geometry of interest to us, illustrated in Fig. 1, occurs in the Phillips Laboratory MARAUDER (magnetically accelerated ring to achieve ultra-high directed energy and radiation) compact toroid program. Its purpose is to study the formation, compression, and acceleration of magnetized plasma rings. An overview of the MARAUDER experiment is given by Degnan *et al.* [1]. The compact toroids are produced in a magnetized coaxial gun and are compressed between a pair of converging conical electrodes.

Simulation of plasma dynamics is often performed on a discrete three-dimensional mesh that is built from primitive polyhedra. A particularly simple polyhedron that is used by many code developers is the cube, but it is not possible to stack cubes to conform to the geometry of Fig. 1 without overlap. To simulate plasma dynamics within volumes of complex shape,

[1] Permanent address: NumerEx, Albuquerque, NM 87106.

[2] Present address: Los Alamos National Laboratory, MS K717, Los Alamos, NM 87545.

0021-9991/98 $25.00

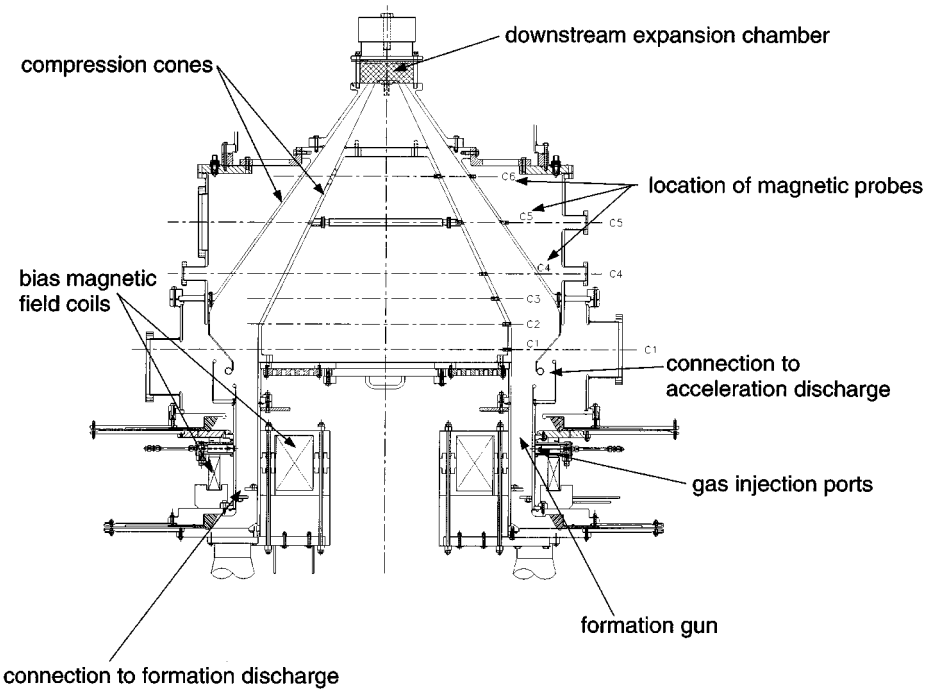

FIG. 1. MARAUDER high energy compact toroid accelerator design concept. A compact toroid is produced in the plasma gun and expansion volume that form the lower half of the apparatus. The toroid is then compressed between a pair of converging electrodes and accelerated in the converging coaxial gap downstream of the compression cones.

we have developed the MACH (multiblock arbitrary coordinate hydromagnetics) codes for simulating unsteady, collisional, plasma behavior. The two-dimensional version, MACH2, is discussed in [2], and the three-dimensional version, MACH3, is discussed in [3]. The two-dimensional code is obtained from the three-dimensional version by demanding that all derivatives in one of the coordinate directions vanish. The geometry in the codes is described in either a cylindrical, or a Cartesian orthonormal frame.

The MACH codes are of the Arbitrary Lagrangian/Eulerian (ALE) variety which allows for flexibility in the physics options at the expense of some numerical complexity. In an ALE code, Faraday's law is advanced in two steps: a Lagrangian advance followed by a remap of the magnetic field from the Lagrangian grid to the computational grid. The computational grid can be at rest in a laboratory frame (Eulerian), at rest in the fluid frame (Lagrangian), or in some other arbitrary state.

The Lagrangian advance of Faraday's law is discussed in Brackbill and Pracht [4] and is not reitereated here. The purpose of this paper is to describe a new magnetic flux-conserving algorithm for the remap portion of the numerical solution to Faraday's equation governing the dynamics of the magnetic field on an arbitrary coordinate mesh. The layout of this paper is as follows. The relevant equations solved by MACH and transport of conserved quantities through the mesh are described in Section II. The magnetic flux transport algorithm for an ALE code is described in Section III. Test problems that illustrate and validate the implementation of the magnetic flux transport algorithm are presented in Section IV.

Time-dependent simulations of compact toroid compression and translation are described in Section V. Our conclusions are given in Section VI.

II. THE MACH CODES

The MACH codes have been used to solve the time-dependent, single fluid, multitemperature, nonideal radiation magnetohydrodynamics (MHD) equations for real materials to guide and interpret a variety of collisional plasma experiments, as well as to perform numerical experiments on novel plasma physics ideas. In this paper we focus on the details of our new algorithm for the numerical convection of magnetic flux through an arbitrary mesh.

The dynamical equations are solved on a mesh that is composed of primitive cells. The cells are arranged in a patchwork of logically rectangular collections, called blocks, aligned corner to corner. The MACH grid lies in a family of planes equally spaced in the transverse coordinate—either θ or z. Furthermore, the subgrids within those planes are identical to each other. Thus, the three-dimensional grid is generated from a two-dimensional one by replication at fixed intervals in the transverse direction. This multiple block structure is suitable for performing numerical simulations of complex experimental configurations such as MARAUDER for which a block structured 2D grid is shown in Fig. 2.

The primitive cell in MACH is a hexahedron. In Cartesian coordinates (x, y, z), it is formed by translating an arbitrarily shaped xy-plane quadrilateral in the z-coordinate direction. In cylindrical coordinates (r, ϕ, z), the primitive cell is constructed by translating an rz-plane quadrilateral in the ϕ-coordinate direction. Henceforth, we refer to the z or ϕ coordinate as the "orthogonal" direction, and the xy or rz coordinates as "in-plane." These terms refer to the computational geometry of the two-dimensional code. For the geometry of a compact toroid, the "orthogonal" direction is usually called "toroidal" and the "in-plane" directions are referred to as "poloidal." A sample MACH hexahedral primitive cell is illustrated in Fig. 3. Each primitive cell has six faces, and each face is a quadrilateral. In the 3D code, each face is shared by two cells. In the 2D code, the four faces with in-plane normals are shared by two cells, and the two faces with orthogonal normals are not shared. To eliminate ambiguity, we assign only the left (L), bottom (B), and front (F) face to a particular cell.

A. *The Ideal MHD Equations of MACH*

For a charge-neutral magnetofluid of mass density, ρ, moving with velocity, $\boldsymbol{u}$, in the presence of a magnetic field, $\boldsymbol{B}$, the code solves the mass continuity equation,

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \boldsymbol{u}); \tag{1}$$

the fluid momentum equation,

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} = -\rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla \cdot (-\gamma (P_e + P_i + Q) + \boldsymbol{M}); \tag{2}$$

the ion and neutral particle energy equation,

$$\rho \frac{\partial \varepsilon_i}{\partial t} = -\rho \boldsymbol{u} \cdot \nabla \varepsilon_i - P_i \nabla \cdot \boldsymbol{u} + \Phi_{ei}; \tag{3}$$

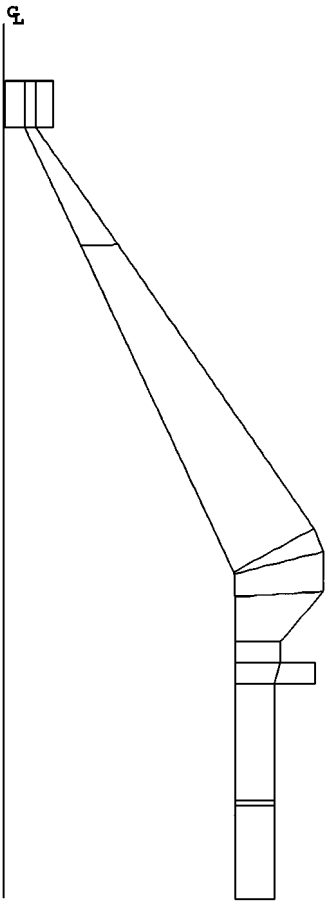

FIG. 2. Collections of hexahedral cells form a block. Blocks are patched together to form a computational domain. The computational region above is for the MARAUDER concept and is made from 14 blocks.

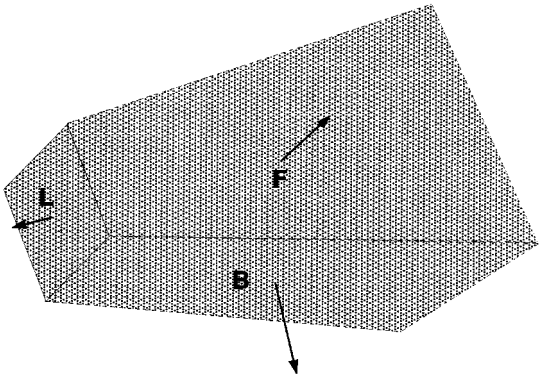

FIG. 3. Hexahedral cell for forming finite volumes in MACH. Each primitive cell has six faces, and each face is shared by two cells. To eliminate ambiguity, we assign only the left (L), bottom (B), and front (F) face to a particular cell.

the electron energy equation,

$$\rho \frac{\partial \varepsilon_e}{\partial t} = -\rho \boldsymbol{u} \cdot \nabla \varepsilon_e - P_e \nabla \cdot \boldsymbol{u} + \boldsymbol{J} \cdot \boldsymbol{E} - \Phi_{ei}; \tag{4}$$

and Faraday's law,

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E}. \tag{5}$$

In Eq. (2), γ is the spatial metric tensor with contravariant components, $\gamma^{\alpha\beta}$, and $\boldsymbol{M}$ is the contravariant Maxwell stress tensor, which in MKS units is

$$M^{\alpha\beta} = \frac{1}{\mu_0}\left(B^\alpha B^\beta - \frac{1}{2} B^2 \gamma^{\alpha\beta}\right), \tag{6}$$

where Greek indices run from 1 to 3 and represent the three orthogonal spatial dimensions. In this paper, we assume the summation convention over repeated indices. In the MACH codes, the equations are generally solved in an orthonormal frame in which case the spatial metric tensor is equivalent to the unit dyad. However, some of the analysis in the following sections will be done in a coordinate basis in which the spatial metric can differ from the unit dyad. Φ_{ei} is an electron–ion coupling term [5]. Q is an artificial numerical compressional viscosity pressure.

The electric field, $\boldsymbol{E}$, is obtained from a simple Ohm's law,

$$\boldsymbol{E} = -\boldsymbol{u} \times \boldsymbol{B}. \tag{7}$$

The displacement current is assumed to be relatively small so the current density is simply

$$\boldsymbol{J} = \frac{\nabla \times \boldsymbol{B}}{\mu_0}. \tag{8}$$

The numerical approximation to the dynamical equations should preserve the constraint on the magnetic field,

$$\nabla \cdot \boldsymbol{B} = 0. \tag{9}$$

This set of equations is closed by equations of state for the electron and ion plus neutral pressures: $P_{e(i)} = P(\rho, \varepsilon_{e(i)})$. In this paper, we use an ideal gas equation of state with constant ratio of specific heats Γ,

$$P_{e(i)} = (\Gamma - 1)\rho \varepsilon_{e(i)}. \tag{10}$$

B. *Spatial Centering of Variables in MACH*

As is common in ALE codes, the velocity vector field resides at the corners of the primitive cell. These corners are usually called the nodes of the mesh. The eight cell centers, C, of eight adjacent primitive cells that are placed in a $2 \times 2 \times 2$ array form a dual primitive cell. The nodes of the primitive cell move with fluid velocity during the Lagrangian advance of the dynamical equations (1)–(5). Because the primitive cells are in general not cubic,

finite volume differencing is used in the MACH codes. A finite volume approximation to the divergence of a vector field, $\boldsymbol{A}$, is then

$$\nabla \cdot \boldsymbol{A} \simeq \frac{1}{V} \int \nabla \cdot \boldsymbol{A}\, dV = \frac{1}{V} \oint \boldsymbol{A} \cdot d\boldsymbol{S}, \tag{11}$$

where we have applied Stokes' theorem for a control volume, V, that is bounded by a surface, $\boldsymbol{S}$, with outward pointing normal, $\hat{\boldsymbol{n}}$, so the components of $d\boldsymbol{S}$ are $dS_\alpha = n_\alpha dS$. On a discrete mesh, the control volume is that of the primitive cell or its dual, and the surface integral in Eq. (11) is approximated by a sum over the faces of the cell. Hence,

$$\nabla \cdot \boldsymbol{A} \simeq \frac{1}{V} \sum_{m=1}^{m_{\text{faces}}} \boldsymbol{A} \cdot \hat{\boldsymbol{n}}_m \Delta S_m, \tag{12}$$

where $\hat{\boldsymbol{n}}_m$ is the unit normal to the mth two-surface of area, ΔS_m. The number of faces, m_{faces}, depends on how exactly the control volume, V, is constructed. Specifically, m_{faces} need not be six.

If $\boldsymbol{A}$ resides at cell centers, then $\nabla \cdot \boldsymbol{A}$ resides at cell nodes, and vice versa. This is a general property; each derivative moves the attention between cell center and cell node. Thus, since forces are applied to the nodes in the Lagrangian step and are computed by taking the divergence of a stress tensor as in Eq. (2), the stresses should be located at cell centers. Hence, P_e, P_i, $\boldsymbol{B}$, and $\boldsymbol{\sigma}$ reside fundamentally at cell centers in MACH.

C. *Faraday's Law for Ideal MHD*

In ideal MHD, the electric field is simply $-\boldsymbol{u} \times \boldsymbol{B}$, so Eq. (5) can be written in component form as

$$\frac{\partial B^\alpha}{\partial t} = \epsilon^{\alpha\beta\gamma} \nabla_\beta (\epsilon_{\gamma\kappa\lambda} u^\kappa B^\lambda), \tag{13}$$

where $\epsilon^{\alpha\beta\gamma}$ is the Levi–Civita tensor density

$$\epsilon^{\alpha\beta\gamma} = \gamma^{-1/2} [\alpha\beta\gamma], \tag{14}$$

where γ is the determinant of the spatial metric tensor in the particular coordinate system, and $[\alpha\beta\gamma]$ is the well-known totally antisymmetric symbol. If, as suggested in Ref. [6], we define the contravariant magnetic vector density of weight 1 constructed from the magnetic field and the 3-space metric as

$$\beta^\alpha \equiv \sqrt{\gamma} B^\alpha, \tag{15}$$

then Faraday's law for ideal MHD takes on the particularly simple form in a coordinate basis (as contrasted to an orthonormal frame)

$$\frac{\partial \beta^\alpha}{\partial t} = \frac{\partial}{\partial x^\gamma} (u^\alpha \beta^\gamma - u^\gamma \beta^\alpha). \tag{16}$$

It is worth noting that the spatial metric in a cylindrical coordinate system (r, ϕ, z) is diagonal with components $\gamma_{rr} = 1$, $\gamma_{\phi\phi} = r^2$, $\gamma_{zz} = 1$, so $\sqrt{\gamma} = r$. Therefore, the physical poloidal

components of the magnetic field in a cylindrical orthonormal frame are identical to the poloidal components in a cylindrical coordinate basis, and the physical toroidal component in an orthonormal frame is equal to rB^{ϕ} in a cylindrical coordinate basis. Hence, the poloidal components of $\boldsymbol{\beta}$ in a coordinate basis are equal to the poloidal components of $r\boldsymbol{B}$ in an orthonormal frame, and the toroidal component of $\boldsymbol{\beta}$ is equal to the physical B^{ϕ} in an orthonormal frame.

D. *Conserved Quantities and Transport through the Mesh*

The differential equations (1)–(5) are consistent with the integral conservation laws of mass, momentum, internal energy, and magnetic flux. These quantities are conserved globally, but not, generally, within each primitive cell where mass, momentum, energy, and magnetic flux can be transported between adjacent cells.

As an illustration, let us consider the total time derivative of the mass, m, in the generally time-dependent volume, V, of a primitive cell with bounding surface, $\boldsymbol{S}$. Let the grid velocity, $\boldsymbol{u}_g$, be the velocity at which the boundary of the primitive cell moves. In the case of pure Lagrangian motion, the grid velocity is identically the fluid velocity, $\boldsymbol{u}$, but generally, the relative velocity, defined as

$$\boldsymbol{u}_{\text{rel}} \equiv \boldsymbol{u}_g - \boldsymbol{u} \tag{17}$$

is nonzero. Then the total time derivative (or convective derivative) is given by

$$D_t = \frac{\partial}{\partial t} + \boldsymbol{u}_g \cdot \nabla. \tag{18}$$

Therefore, the total time derivative of the cell mass is

$$D_t m = D_t \int_{V(t)} \rho \, dV = \int_{V(t)} [(D_t \rho)\, dV + \rho (D_t \, dV)], \tag{19}$$

where $D_t \rho$ is obtained by combining Eq. (18) with Eq. (1), and $D_t \, dV$ is given by the well-known Euler expansion formula [7]:

$$D_t \, dV = \nabla \cdot \boldsymbol{u}_g \, dV. \tag{20}$$

Hence, Eq. (19) becomes

$$D_t m = \int \nabla \cdot (\rho \boldsymbol{u}_{\text{rel}})\, dV \tag{21}$$

which can be written as a surface integral by applying Stokes' theorem to obtain

$$D_t m = \oint \rho \boldsymbol{u}_{\text{rel}} \cdot d\boldsymbol{S}. \tag{22}$$

The meaning of this is clear: the mass in a cell changes by an amount equal to an appropriately chosen mass density, ρ, times the volume flux out of a cell, $\boldsymbol{u}_{\text{rel}} \cdot d\boldsymbol{S}$. For pure Lagrangian motion in which $\boldsymbol{u}_{\text{rel}}$ vanishes, the mass in a cell does not change.

A similar analysis can be performed for the magnetic flux, Φ, through a surface, $\boldsymbol{S}$,

$$\Phi = \int_{S(t)} B^{\alpha} dS_{\alpha} = \int_{S(t)} \beta^{\alpha} dS^{c}_{\alpha}, \tag{23}$$

where dS^{c}_{α} is the coordinate area element which is related to the physical area element, dS_{α}, via

$$dS_{\alpha} \equiv \sqrt{\gamma} dS^{c}_{\alpha}. \tag{24}$$

The total time derivative of the magnetic flux is then

$$D_t \Phi = D_t \int \beta^{\alpha} dS^{c}_{\alpha} = \int \left[(D_t \beta^{\alpha})\, dS^{c}_{\alpha} + \beta^{\alpha} \left(D_t dS^{c}_{\alpha} \right) \right], \tag{25}$$

where $D_t \beta^{\alpha}$ is obtained by combining Eq. (18) with Eq. (16) and Eq. (9), and $D_t dS^{c}_{\alpha}$ is (see, for example, Ref. [6])

$$D_t dS_{\alpha} = \left(\nabla_{\beta} u^{\beta}_{g} \right) dS^{c}_{\alpha} - \left(\nabla_{\alpha} u^{\beta}_{g} \right) dS^{c}_{\beta}. \tag{26}$$

Hence, Eq. (25) becomes

$$D_t \Phi = -\int \nabla \times (\boldsymbol{u}_{\rm rel} \times \boldsymbol{\beta}) \cdot d\boldsymbol{S}^{c} \tag{27}$$

which can be written as a line integral by applying Stokes' theorem to obtain

$$D_t \Phi = -\oint (\boldsymbol{u}_{\rm rel} \times \boldsymbol{\beta}) \cdot d\boldsymbol{x}^{c}, \tag{28}$$

where the line integral is over the one-dimensional closed coordinate path that encloses the surface, $\boldsymbol{S}^{c}$. The meaning of this is clear: the flux through a surface changes by an amount equal in magnitude, and opposite in sign, to the integral of the electric field around a closed loop that bounds the surface. An alternate interpretation, similar to that for the change in cell mass, can be seen by rewriting Eq. (28) as

$$D_t \Phi = -\oint \boldsymbol{\beta} \cdot (\boldsymbol{u}_{\rm rel} \times d\boldsymbol{x}^{c}) \tag{29}$$

which means that the flux change is equal to an appropriately chosen magnetic vector density, $\boldsymbol{\beta}$, times the area flux out of a cell $(\boldsymbol{u}_{\rm rel} \times d\boldsymbol{x}^{c})$. If the relative velocity vanishes, as it does for pure Lagrangian motion, the magnetic flux does not change.

III. ALGORITHM FOR TRANSPORTING MAGNETIC FLUX

In 1988, Evans and Hawley published an "optimal strategy" for numerically evolving the magnetic field equation of MHD in a manner that maintains the divergencelessness of the magnetic field to within machine round-off error [6]. A key element of their approach is the choice of spatial location of the three components of the magnetic field, $\boldsymbol{B}$. They relied on a staggered mesh on which each component on $\boldsymbol{B}$ is placed on one of the three unique

generating orthogonal faces of each primitive cell. Their algorithm is thus unsuitable for the MACH codes in which all three components of $\boldsymbol{B}$ are co-located at the center of each primitive cell. We describe here a similar, but new, algorithm that accounts for the magnetic fluxes differently than is done in Ref. [6].

The strategy of Evans and Hawley is to compute a magnetic flux for each component of $\boldsymbol{B}$ through the two-surface on which each component of $\boldsymbol{B}$ resides. The time-derivative of each flux is evaluated by taking a line-integral of $-\boldsymbol{u}_{\rm rel} \times \boldsymbol{\beta}$ along the one-dimensional closed coordinate boundary of each two-surface (see Eq. (28)). In our new algorithm for MACH, we use a similar strategy, but the flux surfaces for the in-plane components of the magnetic field are not the faces of the primitive cell.

In an ALE code, the grid and the fields supported by it are advanced during each computational cycle of incremental time, Δt, in two steps. First, the grid and its fields are moved in a Lagrangian fashion; that is, each node of the grid is moved with the local magnetofluid velocity, $\boldsymbol{u}$, by a displacement, $\Delta \boldsymbol{x}_L = \boldsymbol{u}\Delta t$. We label the fields of the state of the system with a subscript L to indicate the state after the Lagrangian step, but before the transport step. Second, the grid and its fields generally undergo a pull-back (or remap) to a final state, F. During the pull-back, the displacement of each node of the grid is $\Delta \boldsymbol{x}_{\text{pull-back}} = \boldsymbol{u}_{\rm rel}\Delta t$. The extreme in which there is no pull-back is the limit of fully Lagrangian motion. The other extreme in which the pull-back brings the grid completely back to the original state is the pure Eulerian case.

The Lagrangian advance of the physical components of $\boldsymbol{B}$ is described by Brackbill and Pracht [4]. The pull-back that conserves magnetic flux is done separately for each of the three orthogonal components of $\boldsymbol{\beta}$. In our treatment, all three components of $\boldsymbol{\beta}$ are co-located at each cell center, C. We construct three flux surfaces—one for each component of $\boldsymbol{\beta}$—each of which contains the point, C. The construction is illustrated in Fig. 4.

A. *Finite-Volume Magnetic Flux Advance*

The prescription has three steps: (1) compute three fluxes through three Lagrangian surfaces from three Lagrangian field components for each cell; (2) redistribute the three magnetic fluxes in response to the relative velocity of the grid and fluid; and (3) compute three new magnetic field components in each pulled-back cell from the three new fluxes.

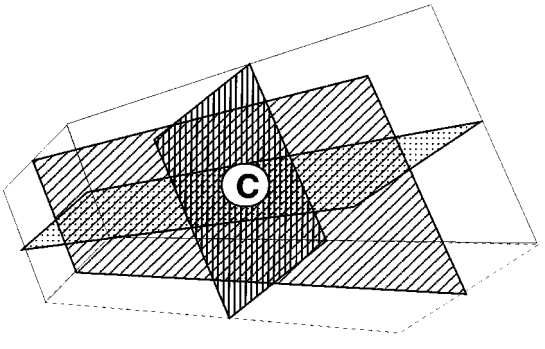

FIG. 4. Three magnetic flux surfaces for computing three different fluxes, Φ, in a cell centered at point, C.

Let the three flux surfaces be labeled with the index n that runs from 1 to 3. The normals to the $n = 1$ and $n = 2$ flux surfaces lie in the grid plane. Because the normal to the front face of each primitive cell is in the orthogonal direction, so too is the normal to the $n = 3$ flux surface.

The flux through the nth surface is surface is then

$$\Phi_n = A^c_{n\alpha}\beta^\alpha, \tag{30}$$

where $A^c_{n\alpha}$ is a 3×3 matrix for the coordinate areas of the n flux surfaces. Explicitly,

$$\begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{pmatrix} = \begin{pmatrix} A^c_{1x} & A^c_{1y} & 0 \\ A^c_{2x} & A^c_{1y} & 0 \\ 0 & 0 & A^c_{3z} \end{pmatrix} \cdot \begin{pmatrix} \beta^x \\ \beta^y \\ \beta^z \end{pmatrix}, \tag{31}$$

where we use x, y, z to denote the three coordinate directions—with x, y the in-plane and z the orthogonal coordinates—that are not necessarily Cartesian. In cylindrical coordinates, x, y, z represent respectively r, z, ϕ. The decoupling of the orthogonal component of $\boldsymbol{\beta}$ from the in-plane components is a consequence of the orientation of the primitive cell in the MACH codes where, by definition, two of the cell faces are aligned with the orthogonal coordinate direction. In the more general case, none of the elements of Eq. (31) will vanish.

Step one in our three-step process entails computing the three fluxes with data from just after the Lagrangian step via Eq. (31).

The second step is to update the fluxes with the relative velocity of the grid via Eq. (28). The integral over the path enclosing the flux surface must be replaced by a sum over the edges that form the boundary of the flux surface. In the MACH codes, Eq. (28) becomes a sum over four edges. The change in the magnetic flux through flux surface n during an increment of time, Δt, is

$$\Delta\Phi_n = -\Delta t \sum_{l=1}^{4} \left(\boldsymbol{u}^{e_l}_{\text{rel}} \times \boldsymbol{\beta}^{e_l}_{\text{up}}\right) \cdot \Delta\boldsymbol{x}^{ce_l}, \tag{32}$$

where $\boldsymbol{u}^{e_l}_{\text{rel}}$ is the relative velocity of edge l, $\boldsymbol{\beta}^{e_l}_{\text{up}}$ is an appropriately chosen, essentially upwind, magnetic vector density at this edge, and $\Delta\boldsymbol{x}^{ce_l}$ is the coordinate length of the edge of flux surface n. The relative velocity in an ALE code is defined at the primitive cell nodes which are the endpoints of an edge. The edge-value of the relative velocity is simply the arithmetic mean of the two edge-point values.

The magnetic flux through the pulled-back flux surface n is

$$\Phi'_n = \Phi_n + \Delta\Phi_n. \tag{33}$$

A pair of adjacent generic flux surfaces of $\boldsymbol{\beta}$, each with outward-pointing normal, is illustrated in Fig. 5. The right-handed direction of integration for the boundary of each surface is illustrated in the figure. Because of the direction of integration, the contribution to the flux change for the two adjacent flux surfaces on each side of an edge are equal in magnitude, but opposite in sign. Consequently, we compute the contribution to the flux change from the motion of a particular edge only once and add it to the flux through surface n on both sides of the edge, but with opposite signs so that the total flux change from a single edge contribution is identically zero.

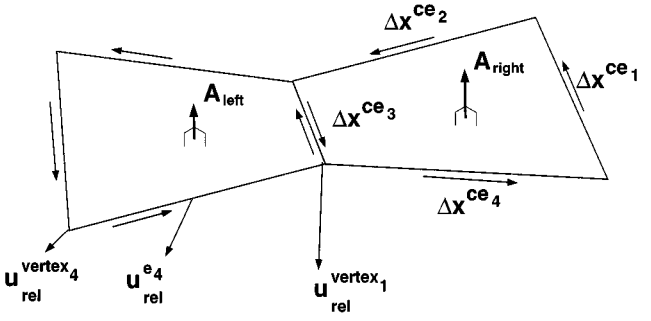

FIG. 5. A pair of adjacent magnetic flux surfaces for magnetic vector density with the direction of path integration shown for both surfaces.

Finally, the new magnetic field densities are computed from the pulled-back fluxes and cell areas by inverting Eq. (30).

$$\beta'^{\alpha} = (A'^{c\ \alpha n})^{-1}\Phi'_n, \tag{34}$$

where prime denotes pulled-back data. Explicitly,

$$\begin{pmatrix} \beta'^{x} \\ \beta'^{y} \\ \beta'^{z} \end{pmatrix} = \begin{pmatrix} A'^{c}_{2y}/\wp & -A'^{c}_{2x}/\wp & 0 \\ -A'^{c}_{1y}/\wp & A'^{c}_{1x}/\wp & 0 \\ 0 & 0 & 1/A'^{c}_{3z} \end{pmatrix} \cdot \begin{pmatrix} \Phi'_1 \\ \Phi'_2 \\ \Phi'_3 \end{pmatrix}, \tag{35}$$

where $\wp$ is the minor determinant of A'^{c}, and is explicitly $A'^{c}_{1x}A'^{c}_{2y} - A'^{c}_{1y}A'^{c}_{2x}$.

B. *Divergence Constraint on the Magnetic Field*

The finite volume approximation to the constraint on the magnetic field, Eq. (9), is obtained by replacing the generic vector field, $\boldsymbol{A}$ with $\boldsymbol{B}$ in Eqs. (11) and (12). Because the components of $\boldsymbol{B}$ reside at cell centers, the divergence of $\boldsymbol{B}$ resides at grid vertices. A finite volume approximation to the divergence of $\boldsymbol{B}$ is obtained by selecting an appropriate control volume, V that surrounds a given vertex. Hence,

$$\nabla \cdot \boldsymbol{B} \simeq \frac{1}{V}\sum_{m=1}^{m_{\text{faces}}} \boldsymbol{B}\cdot\hat{\boldsymbol{n}}_m \Delta S_m = \frac{1}{V}\sum_{m=1}^{m_{\text{faces}}} \boldsymbol{\beta}\cdot\hat{\boldsymbol{n}}_m \Delta S^c_m, \tag{36}$$

where the sum is over the number of faces that define the control volume containing the vertex at which $\nabla\cdot\boldsymbol{B}$ is centered, and ΔS^c_m is the mth coordinate two-surface area. $\boldsymbol{\beta}\cdot\hat{\boldsymbol{n}}_m\Delta S^c_m$ is the magnetic flux through the mth flux surface.

If the coordinate two-surface areas, $\hat{\boldsymbol{n}}_m\Delta S^c_m$ are chosen to be constructed from the $A^c_{n\alpha}$ used in the computation of the three magnetic fluxes via Eq. (31), then it is straightforward to show that the magnetic flux transport algorithm does not introduce spurious divergence $\boldsymbol{B}$.

Let us assume that the initial value problem has been correctly specified to be divergence free. Then we must show that the new magnetic field density is divergence free after it has been time advanced via the new flux transport algorithm.

Each grid vertex is surrounded by a total of eight primitive cells. A closed surface that encloses the control volume of a particular vertex can be constructed from the three flux

surfaces from each of the eight cells that surround the vertex. The three flux surfaces that are associated with each primitive cell do not share any edges, and in fact share only a single point, the cell center, C. However, a surface that encloses the vertex-centered control volume can be made from all three of the flux surfaces in all of the eight surrounding primitive cells if only that quarter of each flux surface that is closest to the vertex is used. Then there are a total of $m_{\text{faces}} = 24$ flux surface elements required to evaluate the divergence of $\boldsymbol{B}$. Therefore, we can write Eq. (36) as

$$\nabla \cdot \boldsymbol{B} \simeq \frac{1}{V} \sum_{n=1}^{3} \sum_{l=1}^{8} \Phi_{n,l}/4, \tag{37}$$

where $\Phi_{n,l}$ is the magnetic flux through the nth flux surface of the lth cell that surrounds a vertex.

Let the spatial indices i, j, and k label cell centers, with i and j spanning the in-plane directions, with i increasing from left to right and j increasing from bottom to top, and k increasing in the orthogonal direction. Then a vertex, located at $(i+1/2,\ j+1/2, k+1/2)$ is surrounded by four cells in the k-plane, starting from the bottom left and continuing in a clockwise manner, at $(i,\ j, k)$, $(i,\ j+1, k)$, $(i+1,\ j+1, k)$, and $(i+1,\ j, k)$, with an additional four cells in the $k+1$ plane.

With this labeling convention, Eq. (37) for the divergence of the time-advanced field, $\boldsymbol{B}'$ can be written as

$$\nabla \cdot \boldsymbol{B}' \simeq \frac{1}{4V} \begin{pmatrix} -\Phi'_{1,i,j,k} & +\Phi'_{2,i,j,k} & +\Phi'_{3,i,j,k} \\ -\Phi'_{1,i,j+1,k} & -\Phi'_{2,i,j+1,k} & +\Phi'_{3,i,j+1,k} \\ +\Phi'_{1,i+1,j+1,k} & -\Phi'_{2,i+1,j+1,k} & +\Phi'_{3,i+1,j+1,k} \\ +\Phi'_{1,i+1,j,k} & +\Phi'_{2,i+1,j,k} & +\Phi'_{3,i+1,j,k} \\ -\Phi'_{1,i,j,k+1} & +\Phi'_{2,i,j,k+1} & -\Phi'_{3,i,j,k+1} \\ -\Phi'_{1,i,j+1,k+1} & -\Phi'_{2,i,j+1,k+1} & -\Phi'_{3,i,j+1,k+1} \\ +\Phi'_{1,i+1,j+1,k+1} & -\Phi'_{2,i+1,j+1,k+1} & -\Phi'_{3,i+1,j+1,k+1} \\ +\Phi'_{1,i+1,j,k+1} & +\Phi'_{2,i+1,j,k+1} & -\Phi'_{3,i+1,j,k+1} \end{pmatrix}, \tag{38}$$

where $\Phi'_{n,l}$ is the time-advanced magnetic flux through the nth flux surface of the lth cell, and where the correct sign has been inserted in front of each $\Phi'_{n,l}$ to account for the relative orientation of the flux surface and the surface that surrounds the vertex $(i+1/2,\ j+1/2, k+1/2)$. Equation (38) is a correct finite volume version of the divergence $\boldsymbol{B}$ operator applied to $\boldsymbol{B}'$.

Each $\Phi'_{n,l}$ in Eq. (38) is related to $\Phi_{n,l}$ by $\Phi'_{n,l} = \Phi_{n,l} + \Delta\Phi_{n,l}$ as in Eq. (33). The change in the divergence of $\boldsymbol{B}$, $\Delta(\nabla \cdot \boldsymbol{B})$, can be replaced by the sum of the 24 $\Delta\Phi_{n,l}$ with signs as in Eq. (38). Here each $\Delta\Phi_{n,l}$ is simply the magnetic flux change that is a consequence of the relative motion between the magnetofluid and the grid given in Eq. (32). The sum is zero since the 24 flux changes can be grouped into 12 pairs so that each pair vanishes identically. For example, $\Delta\Phi_{1,i,j,k} = -\Delta\Phi_{1,i+1,j,k}$ because these flux surfaces share an edge and the flux change at this edge is computed just once, but applied twice with opposite signs to the adjacent surfaces. Therefore, the magnetic flux transport algorithm preserves the divergence of the magnetic field to machine precision.

IV. VALIDATION TEST PROBLEMS WITH MACH2

In what follows, the technique described in the previous section is applied to six simple numerical magnetic convection problems with the MACH2 code. This test suite is complimentary to that proposed by Stone, Evans, Hawley, and Norman [8].

A. *Cartesian Numerical Convection*

A standard test problem is the numerical convection of an initially rectangular pulse through a fixed grid. A good numerical scheme will do this with a tolerable degree of numerical diffusion and dispersion. We performed two calculations with different prescriptions for choosing the upwind value of the magnetic vector density in Eq. (28).

Initially, a test field in the y direction with magnitude 10^{-5} T is placed in the 16 left-most cells of a problem domain with fixed mesh that is 80 cells wide. In this case, the plasma β (ratio of the fluid pressure to the magnetic pressure) is 10^7. The fluid that supports the pulse of magnetic field is given an initial speed on 10^4 m/s (8×10^5 times the Alfvén speed) to the right. The small magnitude of the magnetic field is chosen so as not to produce an appreciable magnetic pressure that would otherwise influence the dynamics. A plot of the magnetic field distribution vs cell number at cycle 0 as well as after 500 computational cycles, by which time the pulse has traveled through approximately 40 cells, is shown in Fig. 6 for two different choices for the upwind value of $\boldsymbol{B}$. The computation that produced the graph on the top of the figure used a first-order donor cell scheme, whereas the computation that produced the bottom graph used a second-order van Leer scheme [9].

While neither scheme generates substantial dispersion, not surprisingly the van Leer scheme is superior for reducing numerical diffusion. In both cases, however, the implementation of Eq. (32) conserves magnetic flux to machine accuracy. Because the van Leer prescription is obviously superior to donor cell, the second-order scheme is used in all subsequent simulations.

B. *Radial Numerical Convection through a Nonorthogonal Grid*

A more challenging test problem is the numerical convection of an initially rectangular pulse of magnetic field through a fixed, but nonorthogonal, grid. In this case, the magnetic vector field is not aligned with the normals to the magnetic flux surfaces. Again, we performed two calculations with identical initial data and numerical algorithm, but with two different computational grids: one composed of only rectangular cells, and the other composed of nonrectangular cells. Both problems were performed in cylindrical rather than Cartesian coordinates.

A test field is applied in the z cylindrical direction but with magnitude 0.6366 T which yields a total magnetic flux of 1 mWb. This pulse is placed in the eight right-most cells of a problem domain that is 24 cells wide and eight cells high. The fluid moves with a speed on 10^4 m/s to the left (in the $-r$ direction). The ratio of the initial kinetic energy to the magnetic energy is $2/3$. Isocontours of magnetic flux that are superimposed on the computational mesh after 200 computational cycles, by which time the inner edge of the pulse has traveled approximately 1.5 times its initial thickness, are illustrated in Fig. 7. The ratio of the total flux after 200 cycles to the total initial flux is 1.003 on the rectangular grid and is 1.021 on the nonorthogonal grid. The figure shows the flux lines to be perfectly straight on the rectangular grid and to bend slightly inward at the top boundary of the nonorthogonal grid.

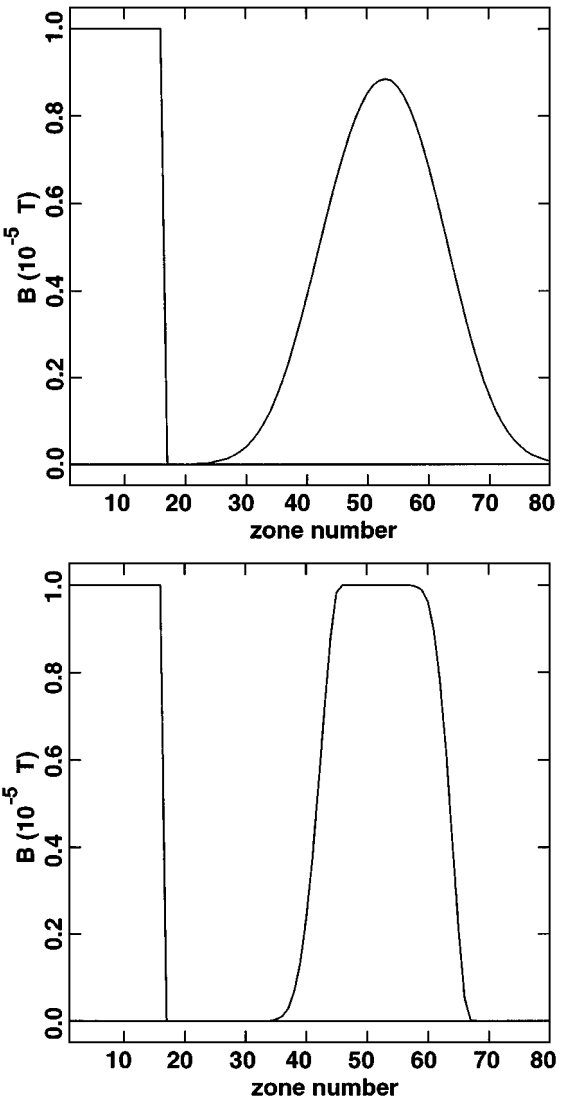

FIG. 6. A first-order donor cell scheme (top) contrasted to a second-order van Leer scheme (bottom) for numerical convection of a rectangular pulse through a fixed Cartesian grid. The original pulse and the pulse after 500 cycles are shown respectively on the left edge and centered at about cell 52.

Part of this bend is caused by the nature of the discrete data set since the magnetic flux exists only at the nodes of the grid. The plotting package performs a bilinear interpolation between nodes to approximate the placement of the contour lines between the nodes.

C. *Equilibrium Calculation and Unphysical Forces Parallel to the Magnetic Field*

A magnetofluid initially at rest with a uniform axial magnetic field develops spurious forces parallel to $\boldsymbol{B}$ when the conservative form of the momentum equation is solved numerically. As explained by Brackbill and Barnes [10], these spurious forces are a consequence of nonzero $\nabla \cdot \boldsymbol{B}$. Motivated by the numerical test discussed in their paper, we monitor the

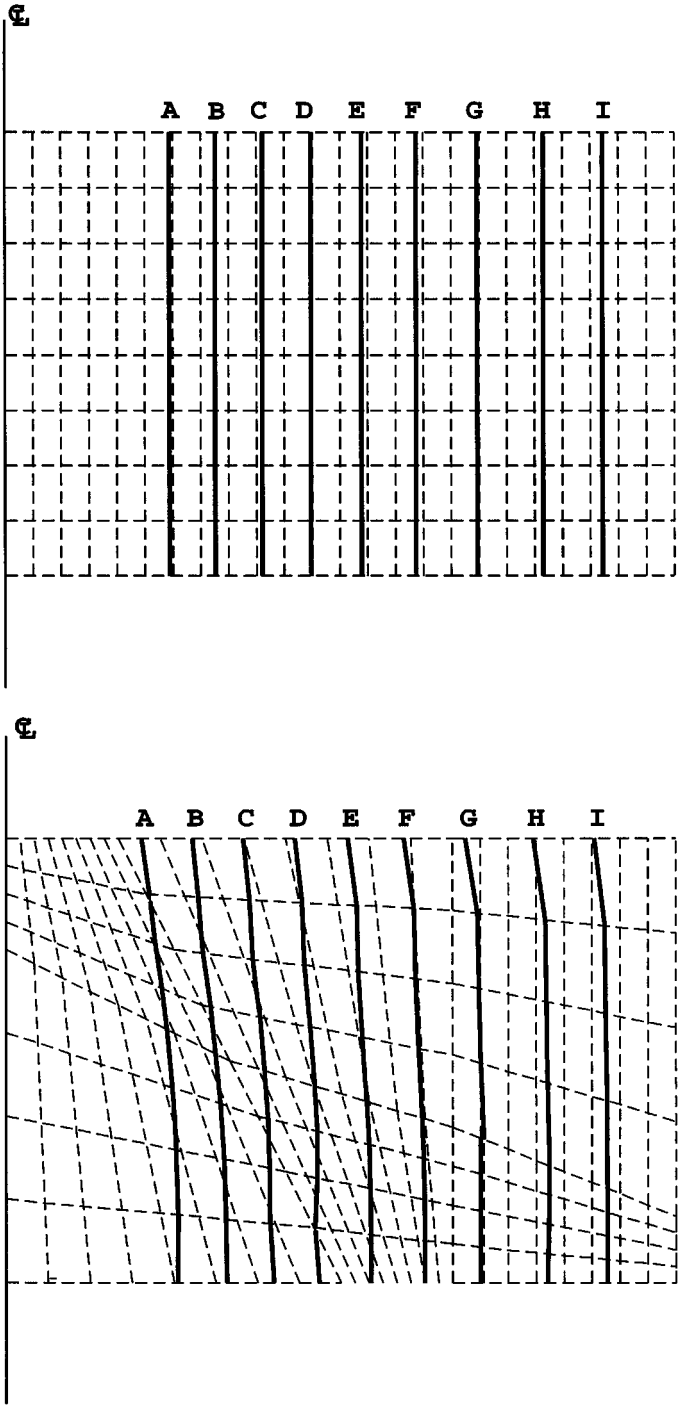

FIG. 7. Magnetic flux of an initially pure axial magnetic field on a fixed rectangular (top) and non-orthogonal (bottom) grid after 200 cycles of inward radial motion. The contours begin with A at 10^{-4} Wb are separated by 10^{-4} Wb. Initially, all of the flux was placed in the right 1/3 of the simulated area.

possible development of an unphysical acceleration parallel to a magnetic field when we use the conservative form of the momentum equation Eq. (2). The projection of the magnetic force density onto the magnetic field is proportional to $|\boldsymbol{B}|\,\nabla\cdot\boldsymbol{B}$ [11].

A numerical algorithm that generates even small $\nabla\cdot\boldsymbol{B}$ will generate an unphysical force parallel to $\boldsymbol{B}$. Therefore, the size of such unphysical forces is a direct measure of the degree to which a numerical algorithm preserves the solenoidal character of the magnetic field.

We simulated the evolution of a uniform plasma in a uniform 1 T axial magnetic field on the same pair of Eulerian meshes used in the previous discussion—one composed of only rectangular cells and the other composed of nonrectangular cells.

For this equilibrium calculation, the initial plasma velocity is identically zero. The plasma β is 1.3×10^{-3}. Boundary conditions on the three sides away from the symmetry axis allow the magnetic field in the interior to continue to the outside and set the velocity field to zero. In Fig. 8 we show the velocity field of this perfectly conducting fluid after 10,000 time steps (390 Alfvén transit times across the entire mesh). As expected, the velocity vector field is everywhere parallel to the magnetic field (and not, for example, to the grid lines of the nonorthogonal mesh). However, the ratio of the maximum fluid speed to the Alfvén speed is only $\sim3\times10^{-10}$ and the ratio of the final kinetic energy to the magnetic energy is $\sim3\times10^{-22}$. This is direct evidence that our magnetic flux conserving transport algorithm does not generate $\nabla\cdot\boldsymbol{B}$ beyond that generated by finite precision machine arithmetic.

Numerical algorithms for other physical processes not discussed in this paper (such as magnetic field diffusion) may not be as good at maintaining the solenoidal constraint on the magnetic field. Therefore, generally, it continues to be a good idea to use the nonconservative form of the momentum equation by subtracting from the left-hand side of Eq. (2) the term $\boldsymbol{B}\nabla\cdot\boldsymbol{B}/\mu_0$.

D. *Steady Flow Parallel to a Uniform Magnetic Field*

In contrast to the previous section in which the magnetofluid was initially at rest with respect to the computational mesh, we now consider the same uniform magnetofluid with a uniform 1 T axial magnetic field throughout the computational domain to be given an initial uniform velocity in the axial direction. All other components of $\boldsymbol{B}$ and $\boldsymbol{u}$ vanish identically at cycle 0. Boundary conditions are applied to the three exterior edges (but not the axis) that allow the magnetic field in the interior to continue to the outside by setting the ghost cell values of the magnetic field to 1 T in the axial direction. The velocity field on these three edges is set to the same value as the interior at $t=0$.

We have performed numerical simulations on computational grids composed of both rectangular and nonrectangular cells, and with both sub- and super-Alfvénic fluid velocities. In all cases, because $\boldsymbol{u}$ is parallel to $\boldsymbol{B}$, we expect the initial data to represent a stationary solution. In other words, the velocity and magnetic vector fields should be independent of time and should not influence each other.

To illustrate the state of the simulation after 10,000 computational cycles, a plot of magnetic flux isocontours is superimposed on the nonorthogonal computational mesh in Fig. 9. For this particular problem, the initial flow speed was chosen to be 10^5 m/s which is $0.256\times v_{\text{Alfvén}}$. At this speed and after 10^4 cycles, a fluid element will travel a distance equal to 15 times the height of the problem domain. Figure 9 is indistinguishable from a plot of the corresponding data at cycle 0. The magnetic flux lines should be straight lines oriented in the axial direction. The slight bending of the flux lines occurs even at

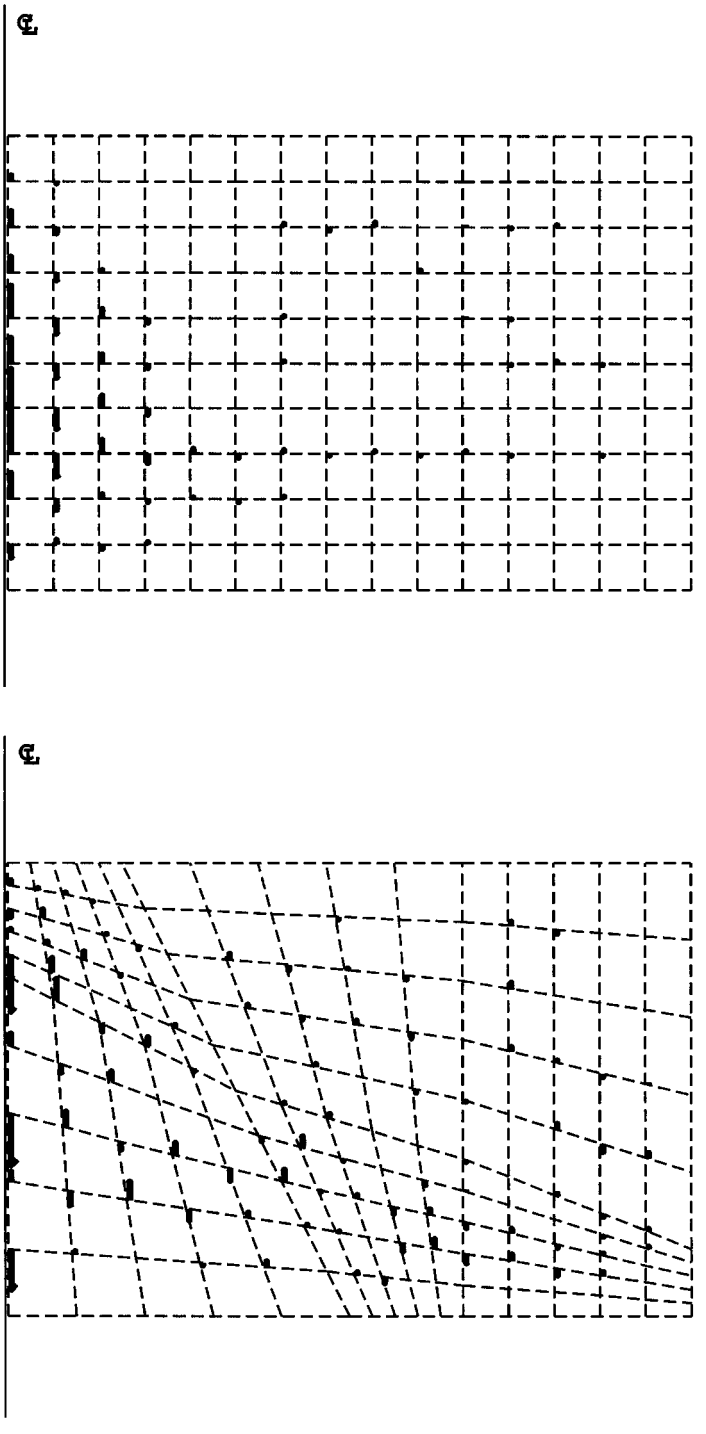

FIG. 8. Spurious plasma velocity vector field superimposed on Eulerian rectangular (top) and nonorthogonal (bottom) grid after 10,000 cycles of on computation from initially static data. The maximum plasma speed ever achieved is only 3×10^{-10} times the Alfvén speed.

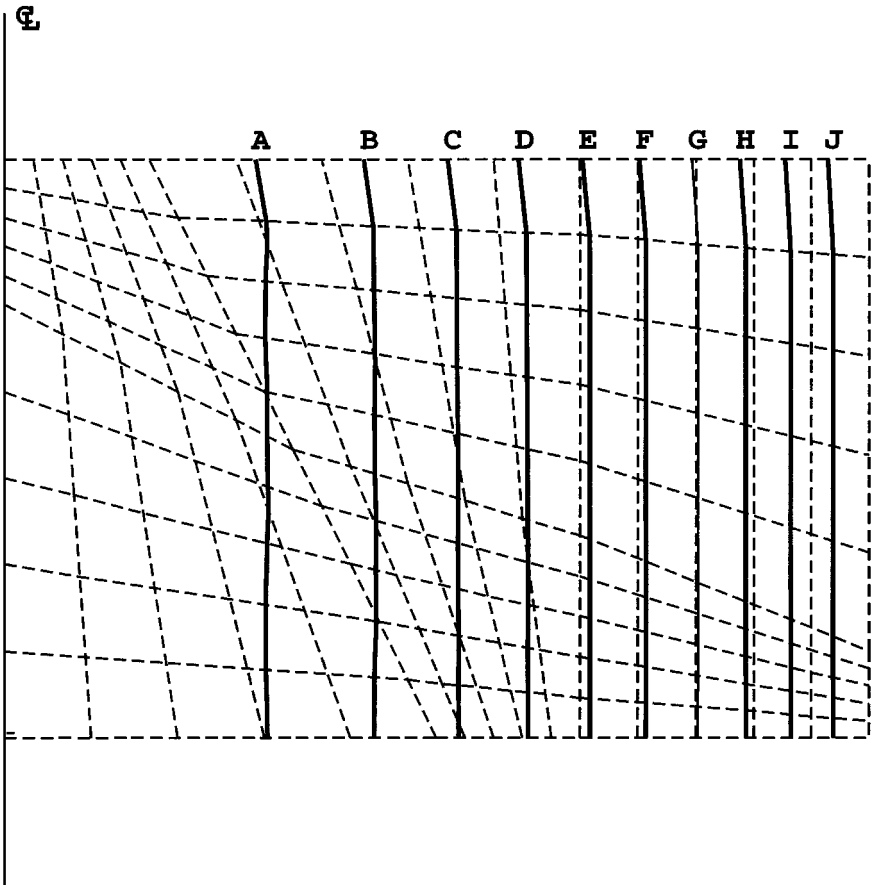

FIG. 9. Magnetic flux lines superimposed on a nonorthogonal Eulerian grid in a cylindrical coordinate system after 10,000 computational cycles during which the fluid supporting the field is moving with a speed comparable to the Alfvén speed in the direction parallel to the magnetic field. The contours are equally spaced between A and J in increments of 0.026 Wb.

cycle 0 and is caused by the interpolation between discrete data points on the relatively coarse nonorthogonal computational mesh. Even after 10,000 cycles, the velocity and magnetic fields are qualitatively the same as the initial state. For example, the largest ratio of radial to axial speed anywhere in the domain is $\sim 10^{-6}$. Similarly, the largest ratio B_r / B_z is also $\sim 10^{-6}$. We achieve similar results for other meshes and other flow speeds. For rectangular meshes, these ratios are consistently below 1 part in 10^{15}.

E. *Transverse Flow through Adjacent Regions with Antiparallel Magnetic Field Lines*

Initially, a finite region of space is filled with magnetic field: the field in the left half is in the $+\boldsymbol{y}$ direction, and the field in the right half is in the $-\boldsymbol{y}$ direction in Cartesian geometry as illustrated in the top picture of Fig. 10. The total initial magnetic flux is identically zero. The fluid that supports the downward pointing field is given an initial velocity to the left. The right and left boundaries are perfect, immovable, flux-excluding walls. The magnetic field boundary conditions on the top and bottom boundaries allow the field lines to continue unimpeded.

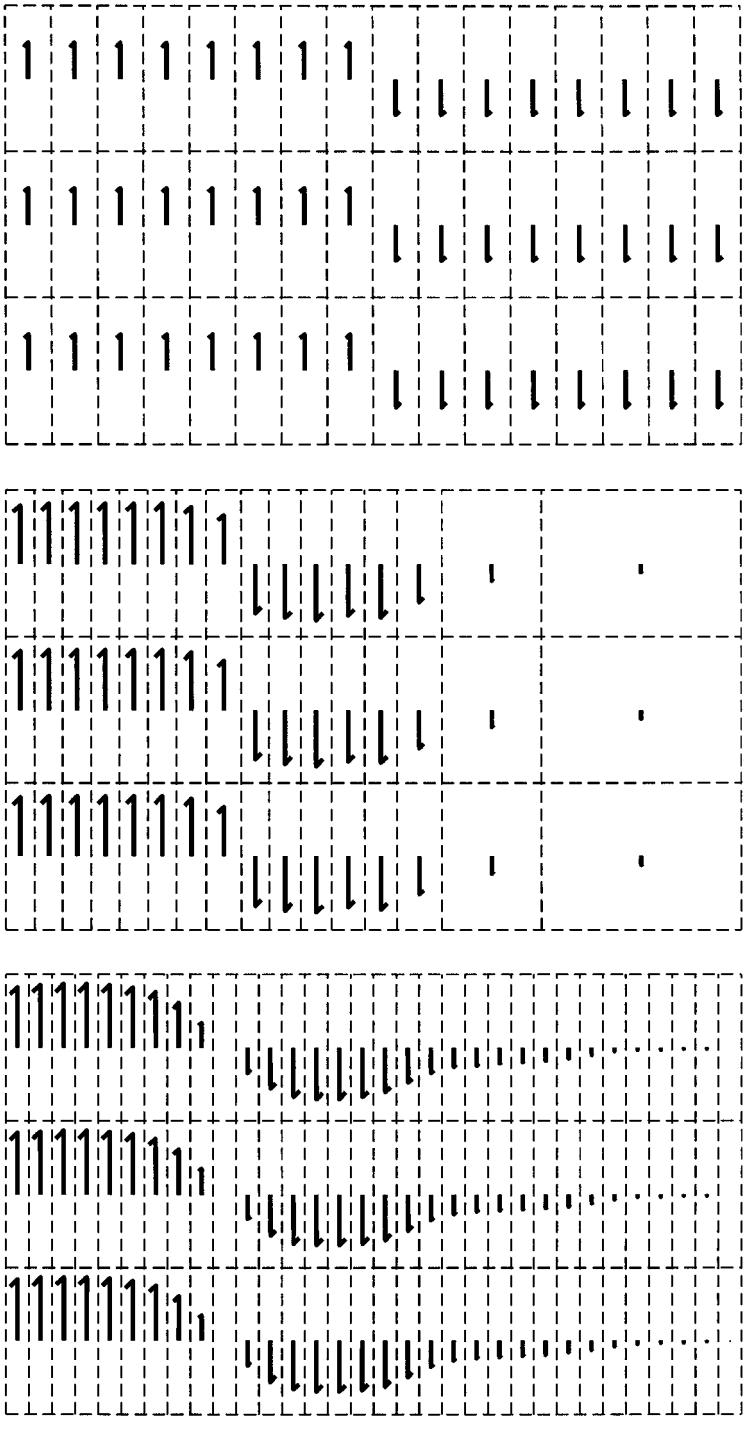

FIG. 10. Compressing antiparallel magnetic fields at cycle 0 (top), after 200 cycles in pure Lagrangian mode (middle), and after 200 cycles in pure Eulerian mode (bottom). The relative size of each vector indicates relative field strength.

A pair of calculations was performed to compare the behavior of the new flux transport algorithm to that of pure Lagrangian motion. The Lagrangian calculation has 16 cells in the direction of the flow. The Eulerian calculation has twice that number of cells, with uniform width, so that the cell size is approximately equal to the size of the Lagrangian cells at the time of peak compression. The simulation time step is held constant and is selected so that peak compression occurs after 200 computational cycles. At that time, the magnetic vector field is illustrated in the bottom two frames of Fig. 10. The total flux in both the Eulerian and Lagrangian problems remains zero to machine accuracy. The ratio of the maximum value of the field to the initial magnitude after 200 cycles is 1.655 for the Lagrangian calculation and 1.74 for the Eulerian calculation. The transition from positive to negative field occurs across a single cell in the Lagrangian, but it is distributed over approximately three cells in the Eulerian calculation. Nevertheless, the Eulerian calculation compares favorably to the Lagrangian simulation for the center of the transition region is essentially the same for both cases. The two calculations have three cells in the vertical direction to illustrate that there is no unphysical variation of the magnetofluid in the direction transverse to the flow.

F. *Formation of a Diamagnetic Cavity*

The magnetic field lines of an infinitely long current-carrying wire form concentric circles around the wire, and the magnitude of the field falls inversely with the radial distance from the wire. If the wire is a perfect conductor and excludes magnetic field, the current will flow only on the surface of the conductor. If it explodes from, for example, Ohmic heating and remains a perfect conductor that continues to exclude magnetic flux, a cylindrical diamagnetic cavity will be formed as the explosion propagates outward in the radial direction.

A numerical model of this process is a good demonstration of a code's flux convection technique. As the wire expands, we expect the magnetic field to be pushed outward so that at any moment, there is no magnetic flux in the volume presently occupied by wire material. This is the case in the simulations where the inner boundary of the diamagnetic cavity corresponds to the interface between wire material and that material which is external to the wire. In Fig. 11, the magnetic state after 300 computational cycles for two different grids is illustrated by superimposing magnetic flux lines on the computational grid that supports the physical state. The grid occupies a single quadrant, located above and toward the side edges of the page from the origin, and extends to 10 times the initial radius of the wire. A symmetry boundary condition is applied to the magnetic field at the two boundaries that intersect at the origin. The calculation illustrated on the left on Fig. 11 is performed on what is essentially a rectangular mesh for which the grid lines are not aligned with the flow lines of the magnetofluid. The calculation on the right of this figure, however, is performed on a more suitable mesh for this radial expansion problem. It is satisfying that, as shown in Fig.11, the magnetic field lines for both calculations are essentially the same after 300 cycles with the differences in location of a particular isocontour never exceeding half of the width of a cell.

V. TIME-DEPENDENT SIMULATIONS OF COMPACT TOROID TRANSLATION

A simplified version of the compact toroid translation portion of Fig. 1 is formed by the volume between a pair of nested electrically-conducting cones that have a common vertex. To demonstrate the utility of the techniques described in Section III for substantial MHD

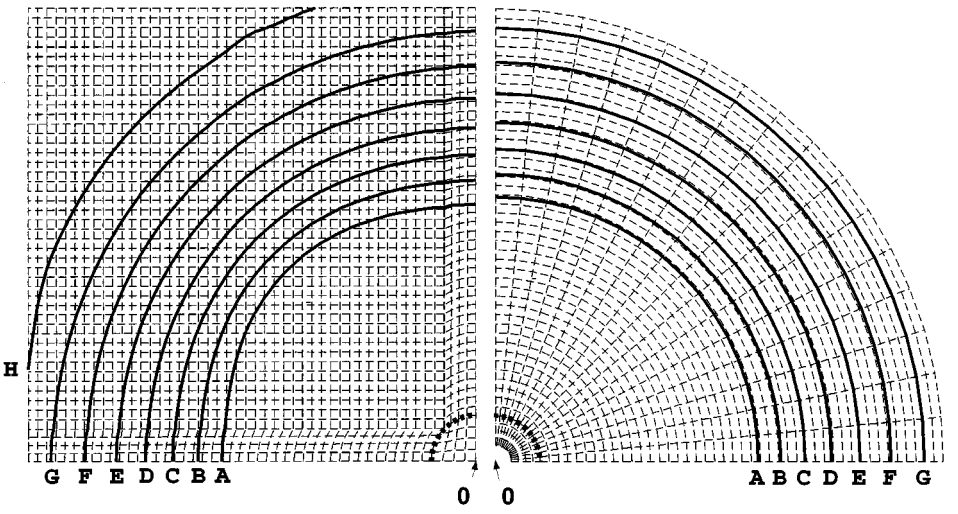

FIG. 11. Magnetic flux lines for a perfectly conducting plasma from an infinitely long exploding wire centered at the origin, 0, with current flowing out of the page after 300 computational cycles are superimposed on a fixed grid for two types of meshes in the Cartesian xy plane. On the left, the mesh is essentially rectangular, and on the right, the mesh is circular. The magnetic flux isocontours, illustrated via heavy solid curves, are uniformly distributed in increments of 0.03 Wb between $A = 0.03$ Wb and $H = 0.24$ Wb. The region between the A contour and the origin is free of magnetic flux. The original location of the outer edge of the wire is illustrated with a heavy dotted curve centered at 0.

problems, MACH2 was run in pure Eulerian mode to simulate the interesting and difficult problem of compact toroid translation and compression. In this section, we perform a numerical experiment by beginning a simulation with interesting initial conditions in a closed system that may be difficult to achieve in the laboratory. Simulations of a more physical system are discussed in Ref. [12]. The initial magnetic field distribution has all three components of the magnetic field arranged in a force-free configuration with nonzero magnetic helicity. Such a configuration is called a Woltjer–Taylor state [13, 14]. For our purposes, it is sufficient to know that a compact toroid plasma in such a state is relatively stable and the force exerted by the plasma currents on the plasma vanishes because J is everywhere parallel to B. The initial mass density and magnetic field distributions for a geometry in which the radius of the outer cone falls by a factor of nine over the length of the cone is illustrated in Fig. 12.

The 1 mg, $\Gamma = 5/3$ fully ionized hydrogen plasma is given an initial speed of 10^5 m/s in the direction toward to the intersection of the two cones. The initial speed is somewhat less that the initial average Alfvén speed of $\sim 2 \times 10^5$ m/s. The plasma mass is initially distributed so the density isocontours are essentially aligned with the poloidal magnetic flux lines, but this is not an important aspect of the problem. As the plasma is compressed between the converging electrodes, a force is generated by image currents in the electrodes that acts in the direction to expel the plasma from the broad end of the inter-electrode region and tends to slow the plasma. In the process, kinetic energy is converted to magnetic energy. Furthermore, the internal energy of the plasma increases with compression. The initial magnetic energy is chosen to be 2 kJ so that, with the initial kinetic energy of 5 kJ, the plasma will reach at a later time a state of maximum compression in this particular geometry with essentially zero kinetic energy; then it will reverse direction and accelerate

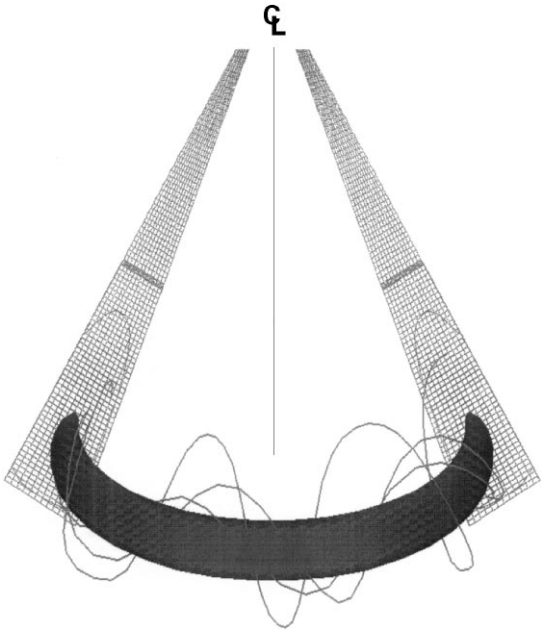

FIG. 12. The initial MACH2 mass density and force-free magnetic field distributions for a compact toroid geometry in which the radius of the outer cone falls by a factor of nine over its length. Sample magnetic flux lines wrap around an isosurface of plasma mass density. The computational mesh is also illustrated.

back to the broad end of the electrode gap where it originated. The 2D grid has 24 cells in the transverse direction, and 192 cells in the longitudinal direction.

The boundaries of the simulation domain are assumed to be rigid walls that exclude magnetic flux. Therefore, the magnetic flux through the boundary of the problem domain should be identically zero for all time. Figure 13 shows a sequence of snapshots of poloidal

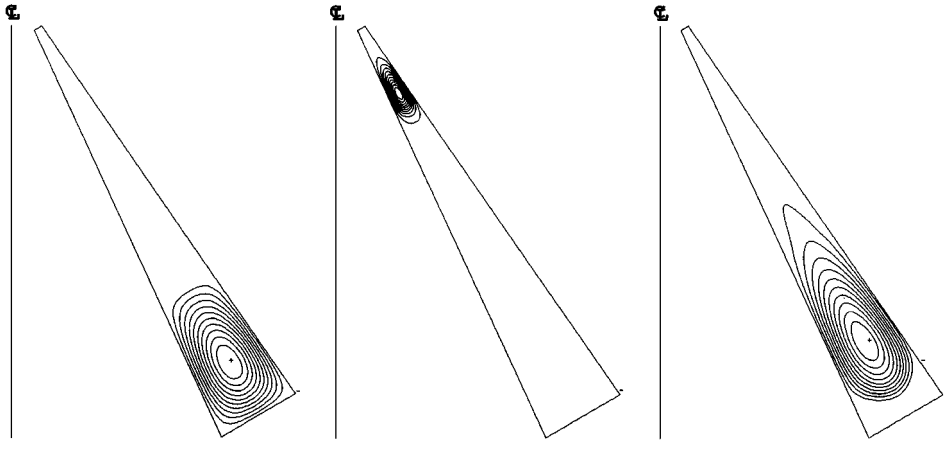

FIG. 13. Poloidal magnetic flux lines for three different times of a MACH2 calculation of the compression of a CT given an initial speed parallel to the intersection of the two cones. From left to right: the initial state at $t = 0$, the state of maximum compression at $t = 10.5\,\mu$s, the state at $t = 20\,\mu$s.

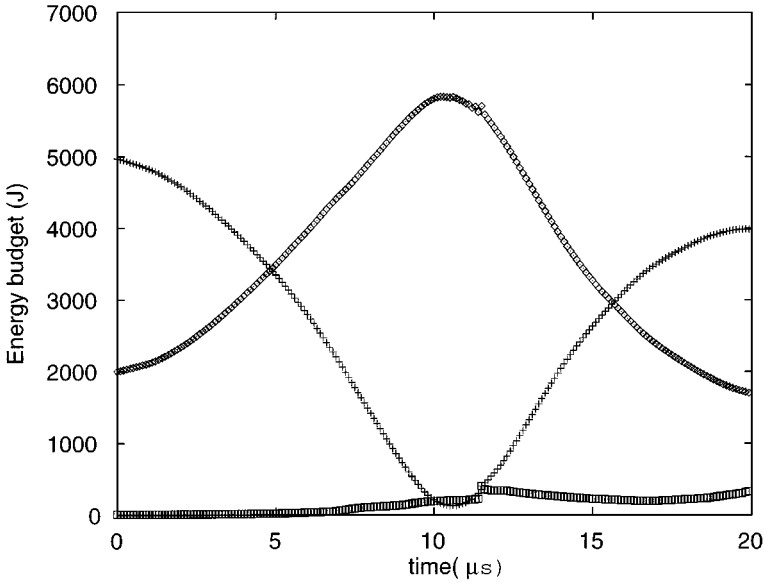

FIG. 14. The energy budget for the compact toroid compression simulation with MACH2 is shown vs time. The different curves are: kinetic energy (plus signs), magnetic energy (diamonds), and internal energy (boxes).

magnetic flux contours in the rz plane of a cylindrical coordinate system at three different times from the simulation. Only that portion to the right of the axis of symmetry is shown. Isocontours of toroidal magnetic field look essentially like the poloidal flux lines, although the toroidal component of the magnetic field is strongest where the poloidal component is weakest, and vice versa. The distribution of the plasma mass at any t is similar to that of the poloidal magnetic field at the same time.

The kinetic energy reaches its lowest value of 2.5% of its initial value at approximately 10.5 μs at which time the magnetic energy has its greatest value of 5.8 kJ. The plasma internal energy never exceeds 400 J. Ideally, and neglecting nonadiabatic effects, we expect the initial and final states to be time symmetric. However, numerical diffusion causes the dynamics to be somewhat time asymmetric. The energy distribution as a function of time for this simulation is shown in Fig. 14. Quantitatively, the total energy falls monotonically in time by approximately 14% over the 8400 cycles it takes the simulation to advance the dynamical equations by 20 μs. More importantly, however, over the course of the 8400 cycles the magnetic flux-conserving algorithm produces an error of 0.1% in the size of the total flux through the walls—which ideally should remain identically zero—relative to the maximum value of the flux function at the initial time, t.

The energy loss occurs because there is no dynamical equation for the total energy, and the conservative upwind numerical convection scheme for momentum and magnetic flux only conserves energy in the limit of infinitesimal cell size. We could impose energy conservation by accumulating the local energy loss during each cycle of numerical convection and adding this energy to the internal energy of the plasma, but this would not decrease the degree of time asymmetry for this simulation. It is interesting to note that, because the initial rotational velocity is zero, it should remain so as long as the plasma is force free ($\boldsymbol{J}$ parallel to $\boldsymbol{B}$). Hence, the degree to which the net rotational energy differs from zero is then a reflection of

the accuracy of the numerical scheme for convecting the magnetic field. For this simulation, the ratio of the rotational energy to the total energy never exceeds 1%.

VI. CONCLUSIONS

We have developed a new technique for the numerical convection of the magnetic vector field in an arbitrary coordinate ALE code and implemented the technique in the two-dimensional MHD code, MACH2, and the three-dimensional MHD code, MACH3. The power of this technique was illustrated by performing six relatively simple test problems and a single complex problem of compact toroid compression and translation between a pair of conical electrodes.

The technique is perfectly flux conserving and has general applicability to codes in which all components of the magnetic field are co-located.

ACKNOWLEDGMENTS

The authors have benefited from lengthy and detailed discussions with Professor Norman F. Roderick. This work was supported by the Air Force Office of Scientific Research. We are particularly indebted to Dr. Charles Holland for his encouragement and support.

REFERENCES

1. J. H. Degnan, R. E. Peterkin Jr., G. P. Baca, J. D. Beason, D. E. Bell, M. E. Dearborn, D. Dietz, M. R. Douglas, S. E. Englert, T. J. Englert, K. E. Hackett, J. H. Holmes, T. W. Hussey, G. F. Kiuttu, F. M. Lehr, G. J. Marklin, B. W. Mullins, D. W. Price, N. F. Roderick, E. L. Ruden, C. R. Sovinec, P. J. Turchi, G. Bird, S. K. Coffey, S. W. Seiler, D. Gale, J. D. Graham, M. Scott, and W. Sommars, *Phys. Fluids B* **5**, 2938 (1993).
2. M. H. Frese, National Technical Information Service Document No. ADA 192285, Mission Research Corporation Report AMRC-R-874, 1987. [Copies may be obtained from the National Technical Information Service, Springfield, VA 22161. The price is $14.95 plus a $3.00 handling fee. All orders must be prepaid]
3. U. Shumlak, T. W. Hussey, and R. E. Peterkin Jr., *IEEE Trans. Plasma Sci.* **23**, 83 (1995).
4. J. U. Brackbill and W. E. Pracht, *J. Comput. Phys.* **13**, 455 (1973).
5. S. I. Braginskii, in *Reviews of Plasma Physics*, Vol. 1, edited by M. A. Leontovich (Consultants Bureau, New York, 1965), p. 205.
6. C. R. Evans and J. F. Hawley, *Astro. Phys. J.* **332**, 659 (1988).
7. See for example, D. Mihalas and B. W. Mihalas, *Foundations of Radiation Hydrodynamics* (Oxford Univ. Press, London, 1984), p. 58.
8. J. M. Stone, J. F. Hawley, C. R. Evans, and N. L. Norman, *Astro. Phys. J.* **388**, 415 (1992).
9. B. van Leer, *J. Comput. Phys.* **23**, 276 (1977).
10. J. U. Brackbill and D. C. Barnes, *J. Comput. Phys.* **35**, 426 (1980).
11. D. C. Barnes and J. U. Brackbill, *Nucl. Sci. Eng.* **64**, 18 (1977).
12. R. E. Peterkin Jr., *Phys. Rev. Lett.* **74**, 3165 (1995).
13. L. Woltjer, in *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 44 (National Academy of Sciences, Washington, DC, 1958), p. 489.
14. J. B. Taylor, *Phys. Rev. Lett.* **33**, 1139 (1974).