JOURNAL OF COMPUTATIONAL PHYSICS 13, 455–482 (1973)

An Implicit, Almost-Lagrangian Algorithm for Magnetohydrodynamics*

J. U. BRACKBILL AND W. E. PRACHT

University of California, Los Alamos Scientific Laboratory, Los Alamos, New Mexico 87544

Received January 7, 1973

An implicit numerical algorithm for the solution of transient, two-dimensional magnetohydrodynamic flows is presented. The algorithm is formulated for an arbitrary Lagrangian–Eulerian computation mesh. This allows the use of variable resolution meshes and reduces computational diffusion of the magnetic field so that flows with a magnetic Reynolds number of the order of hundreds may be calculated. In addition, the implicit formulation makes possible the efficient calculation of low speed flows. Sample calculations illustrating the properties of the algorithm are presented.

INTRODUCTION

There are two serious difficulties in the numerical calculation of magnetohydrodynamic flow in even the simplest cases [1]. First, if the Alfvén speed is very large compared with the maximum fluid velocity, stability conditions severely limit the maximum time step in an explicit calculation. Boris [2] has suggested a solution to this problem that makes the speed of light an adjustable parameter in a relativistic formulation of the equations of magnetohydrodynamics. In the first approximation, the mass-density is replaced with a generalized mass-density tensor that includes a term proportional to the Maxwell stress tensor, and inversely proportional to the square of the speed of light. With this replacement, the maximum Alfvén speed is equal to c, the speed of light. Since c is an adjustable parameter, its value can be chosen so that a reasonable computation time step can be used. This is a valid approach so long as the results are independent of the value chosen for c.

Second, computational diffusion often prevents the accurate resolution of fluid–fluid or fluid–field interfaces. Some diffusion is necessary to cancel destabilizing truncation errors in the difference approximations to the convection terms. It is usually added through artificial viscosity or through diffusionlike truncation errors

* This work was performed under the joint auspices of the United States Atomic Energy Commission and the Defense Nuclear Agency.

introduced by donor cell differencing of the convection terms. Since the destabilizing error is proportional to the velocity gradients [3], it is worse the greater the zone to zone variation of velocity. Artificial viscosity bounds this variation to an acceptable level for a given spatial resolution. However, the need for it is reduced only as spatial resolution is increased, or as higher order corrections to the difference equation are added. Boris has proposed an approach which bounds zone-to-zone variation of the data by a flux correction method [4]. This may prove to be an acceptable and generally applicable method, but it requires the imposition of an additional constraint on the numerical solutions which may not always be appropriate. On the other hand, a Lagrangian formulation reduces the need for artificial viscosity by removing the necessity for differencing convection terms. If the formulation is implicit, the Alfvén problem is also solved.

An implicit formulation of the dynamical equations is unconditionally stable. Furthermore, when the equations include all of the relevant physics, their iterative solution to a prescribed accuracy ensures that the calculation will resolve all important fluid motions. This point will be discussed in greater detail in the section on the computational method.

A Lagrangian calculation reduces the need for artificial viscosity by eliminating the convection terms. In addition, Lagrangian zoning can provide resolution where it is needed, follow material interfaces, and allow the tailoring of zoning to the problem to be solved. However, the solution of implicit equations is sometimes so cumbersome and Lagrangian zoning so prone to instability in flow fields with shear that an implicit Lagrangian formulation is simply incapable of solving any practical problem. The ICED–ALE technique (Implicit Continuous-Fluid Eulerian–Arbitrary Lagrangian-Eulerian) recently reported by Hirt and Amsden [6, 7] extends an earlier implicit Eulerian technique [5] for the numerical solution of fluid flow problems. It provides an economical algorithm for the solution of implicit Lagrangian equations and a provision for automatic rezoning which permits the use of almost-Lagrangian zoning. We will discuss the extension and application of the ICED–ALE technique to magneto-flow problems.

In Section I, we discuss the formulation of composite solution of the equations of magnetohydrodynamics in a coordinate system which may be translating and distorting. In Section II we give the algorithm by which the equations of Section I are to be solved, and discuss the changes in the convergence and stability properties of the modified ICED–ALE equations. In Section III, we give comparisons among Eulerian, Lagrangian, and almost Lagrangian numerical calculations of the same problem, and give the results of the application of this technique to the solution of a z-pinch problem.

I. The MHD Equations

The following equations are a mathematical model for a charge-neutral, but fully ionized, fluid with isotropic resistivity and pressure which are functions of the thermodynamic variables. The evolution of this model in time obeys four conservation laws. It obeys the three ordinary ones for a fluid, those for mass, momentum, and energy, plus an additional conservation law for magnetic flux given by Faraday's law,

$$\partial \mathbf{b}/\partial t = \nabla \times \{(\mathbf{u} \times \mathbf{b}) - \eta \mathbf{j}\}, \tag{1}$$

where $\mathbf{b}$ is the magnetic field intensity, $\mathbf{u}$ the fluid velocity, η the resistivity, and $\mathbf{j}$ the current density. This equation states that the magnetic flux, defined by the integral,

$$\phi = \int_{a(S)} \mathbf{b} \cdot d\mathbf{a},$$

is constant on any open surface S whose boundary is moving with the fluid velocity and on which the resistivity is zero. That is, the divergence of the field is time independent. Since it is always zero initially, it will remain zero.

Most numerical solutions of the MHD equations in two dimensions have been obtained with a fixed Eulerian computation mesh [1, 8]. We have found it useful to solve the dynamical equations on a Lagrangian mesh, and rezone as necessary in a separate step. The equations solved in the Lagrangian step are (in mks units),

$$(d\rho/dt) + \rho(\nabla \cdot \mathbf{u}) = 0, \tag{2}$$

$$d\mathbf{u}/dt = -(1/\rho)\, \nabla \cdot (\mathbf{\Pi} + \mathbf{M}) + \mathbf{g}, \tag{3}$$

$$d\mathbf{b}/dt = (\mathbf{b} \cdot \nabla)\, \mathbf{u} - \mathbf{b}(\nabla \cdot \mathbf{u}) - \nabla \times \eta \mathbf{j}, \tag{4}$$

and

$$d\bar{\imath}/dt = -(1/\rho)((\mathbf{\Pi} + \mathbf{M}) \cdot \nabla) \cdot \mathbf{u}, \tag{5}$$

where $\mathbf{g}$ is the gravitational acceleration, $\bar{\imath}$ the sum of the specific internal and magnetic field energy, $\mathbf{\Pi}$ the fluid stress tensor, and $\mathbf{M}$ the Maxwell stress tensor. The current density $\mathbf{j}$ is given by Ampère's law,

$$\mathbf{j} = (1/\mu)(\nabla \times \mathbf{b}), \tag{6}$$

where μ is the permeability. The Cartesian components of the fluid stress tensor, $\mathbf{\Pi}$, are given by,

$$\Pi_{ij} = -((p/3) + \lambda(\partial u_m/\partial x_m))\, \delta_{ij} + \mu'((\partial u_i/\partial x_j) + (\partial u_j/\partial x_i)), \tag{7}$$

where p is given by the equation of state, λ and μ' are the coefficients of bulk and shear viscosity, and δ_{ij} is the Kronecker delta symbol. The components of the Maxwell stress tensor are given by,

$$M_{ij} = (1/\mu)(b_i b_j - (1/2)\, b_i b_i \delta_{ij}). \tag{8}$$

The equation for the magnetic field energy given by

$$(d/dt)(\mathbf{b} \cdot \mathbf{b}) = -2\mu(\mathbf{M} \cdot \nabla) \cdot \mathbf{u} - 2\eta \mathbf{j} \cdot \mathbf{j},$$

is combined with the equation for the internal energy,

$$\rho(d/dt)(i) = -(\mathbf{\Pi} \cdot \nabla) \cdot \mathbf{u} + \eta \mathbf{j} \cdot \mathbf{j},$$

to form Eq. (5).

In the rezone step, the intensive time derivative of each quantity is calculated at the moving grid point. Where the relative velocity of the fluid and grid is given by,

$$\mathbf{u}_{\text{rel}} = \mathbf{u} - \mathbf{u}_g, \tag{9}$$

and $\mathbf{u}_g$ is the assigned grid velocity, the intensive time derivative is given by,

$$\partial\psi/\partial t = (d\psi/dt) - (\mathbf{u}_{\text{rel}} \cdot \nabla)\, \psi, \tag{10}$$

where ψ is some representative variable. If $\mathbf{u}_g$ is equal to $\mathbf{u}$, the intensive time derivative and the Lagrangian derivative are equal and the end result of steps one and two is a Lagrangian calculation. If $\mathbf{u}_g$ is equal to 0, the intensive derivative is equal to the Eulerian derivative and the end result of steps one and two is an Eulerian calculation. However, $\mathbf{u}_g$ is arbitrary and the calculation need not be either pure Eulerian or pure Lagrangian.

To satisfy the conservation relations, the intensive derivatives for mass, momentum and total energy are written in integral form,

$$\frac{\partial}{\partial t} \int_V \rho \, dV = - \int_{s(V)} \{(\mathbf{n} \cdot \mathbf{u}_{\text{rel}})\, \rho\}\, ds, \tag{11}$$

$$\frac{\partial}{\partial t} \int_V \rho \mathbf{u} \, dV = \int_V \rho \frac{d\mathbf{u}}{dt} \, dV - \int_{s(V)} \{(\mathbf{n} \cdot \mathbf{u}_{\text{rel}})\, \rho \mathbf{u}\}\, ds, \tag{12}$$

and

$$\frac{\partial}{\partial t} \int_V \epsilon \, dV = \frac{d}{dt} \int_V \epsilon \, dV - \int_{s(V)} \{(\mathbf{n} \cdot \mathbf{u}_{\text{rel}})\, \epsilon\}\, ds, \tag{13}$$

where V is a subvolume of the computation region, and $s(V)$ is its bounding surface with outward directed normal vector $\mathbf{n}$. The total specific energy, ϵ, is defined by

$$\epsilon \equiv (1/2)(\mathbf{u} \cdot \mathbf{u} + \mathbf{b} \cdot \mathbf{b}/\mu\rho) + i.$$

The intensive derivative for the magnetic field is similarly given by

$$\frac{\partial}{\partial t}\int_V \mathbf{b}\, dV = \frac{d}{dt}\int_V \mathbf{b}\, dV - \int_{s(V)} \{(\mathbf{n} \cdot \mathbf{u}_{\text{rel}})\, \mathbf{b}\}\, ds. \tag{14}$$

When the magnetic field energy is much larger than the internal energy, greater accuracy in the internal energy can be achieved by direct calculation of the internal energy. It has also been found [6] that when the flow is far-subsonic and the internal energy is high, an implicit solution of the internal energy equation is preferable.

II. The Numerical Solution of the MHD Equations

Hirt and Amsden's [6, 7] composite algorithm, ICED–ALE, for fluid-flows is extended to magneto fluid-flows by adding an implicit Lagrangian difference equation for Faraday's law, a rezone equation for the magnetic field, and an explicit calculation of resistive diffusion. In addition, a method for mesh relaxation is applied which allows a computation to proceed with nearly Lagrangian zoning, but without Lagrangian instabilities.

A complete set of difference equations is given for a computation mesh composed of arbitrary quadrilateral zones in which the thermodynamic variables, density, internal energy, magnetic field and viscous stress terms are defined at cell centers; and the dynamic variables, position and velocity, are defined at cell vertices. The boundary of the computation mesh is assumed to be a rigid, frictionless, and infinitely conducting wall.

A. *The Lagrangian Phase*

The greater efficiency of the ICED–ALE technique, relative to an explicit difference formulation, comes from making certain terms in the dynamical equations implicit. This allows the use of a larger time step than the stability condition for an explicit formulation would allow.

In the ICED–ALE hydrodynamic calculation, the mass equation is solved by successive overrelaxation with pressure as the iteration variable. Changes in the pressure in one cell are coupled to changes in adjacent cells through the equations of motion. Economy in the iteration is achieved by computing only those changes that occur in an adiabatic process implicitly. The viscous stresses, those proportional to μ' and λ in Eq. 7, are treated explicitly in a separate calculation. The internal energy is not included in the iteration. A simple proportionality between changes in pressure and changes in density given by the adiabatic sound speed gives the necessary equation of state. Additional economy comes from updating the equations of motion only in those cells in which the calculated changes in pressure exceed the convergence criterion. Furthermore, for small resistivity, explicit difference equations for the resistive diffusion of the magnetic field are satisfactory.

The explicit difference equations for the resistive diffusion terms are solved before commencing the iteration. Their solution requires three steps. First, Ampère's law is solved to obtain the current density at each vertex; second, the field diffusion caused by current circulation is calculated; third, the Ohmic heating is calculated and added to the internal energy.

Ampère's law is solved for the current at each vertex by path integration around a subdivision of the computation mesh called a momentum control volume. The sides of a momentum control volume are bounded by the four vertices adjacent to the one at which the current is being calculated. Where x's denote cell centers, and circles cell vertices, a subregion of the computation mesh is labeled as shown in Fig. 1. The solid lines connecting vertices are cell boundaries. The dashed lines are the boundaries of the momentum control volume surrounding vertex 1. The mesh is cylindrically symmetric, with the axis of symmetry lying to the left of the subregion shown. The coordinates of vertex 1 are r and z. The magnetic field lies in the r–z plane, and its components, b_r and b_z, are stored at cell centers labeled a through d. The current flow is in a direction perpendicular to the r–z plane, and is given by

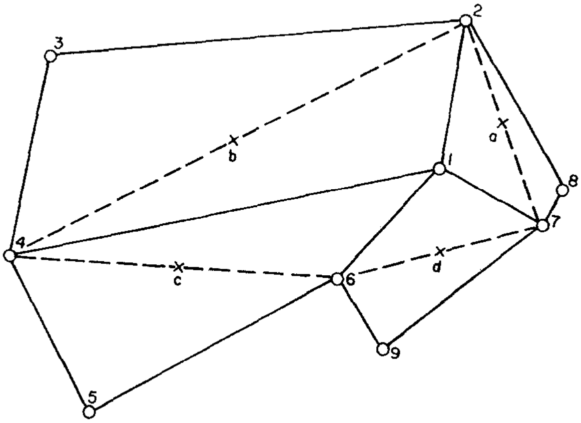

FIG. 1. The dashed line surrounds the momentum control volume corresponding to vertex 1.

$$-\mu j_1 A_1 = [b_{r_a}(r_7 - r_2) + b_{z_a}(z_7 - z_2) + b_{r_b}(r_2 - r_4) + b_{z_b}(z_2 - z_4) + b_{r_c}(r_4 - r_6) + b_{z_c}(z_4 - z_6) + b_{r_d}(r_6 - r_7) + b_{z_d}(z_6 - z_7)], \quad (15)$$

where A_1 is the area of the momentum control volume,

$$A_1 = (1/2)[(r_2 - r_6)(z_4 - z_7) - (z_2 - z_6)(r_4 - r_7)]. \quad (16)$$

The magnetic field diffusion caused by current flow is calculated by approximating the volume integral for the current circulation within a subvolume of the computation mesh,

$$\frac{d}{dt}\int_S \mathbf{b}\, dV = \int_S (\mathbf{n} \times \eta \mathbf{j})\, dS.$$

The indicated surface integral is performed over the surface of a computation cell. The value of the current density on a side is equal to the average of the current densities at the vertices bounding that side. The contribution of the current flow at vertex 1 to the magnetic field diffusion in cell a is a typical example of the equations we obtained. It is given by,

$$-\delta b_{r_a} = \eta j_1[(r_7^2 - r_2^2)]\, \delta t/4V_a\,, \tag{17}$$

and

$$-\delta b_{z_a} = \eta j_1[(r_2 + r_1)(z_2 - z_1) + (r_1 + r_7)(z_1 - z_7)]\, \delta t/4V_a\,, \tag{18}$$

where V_a, the volume of cell a is given by,

$$V_a = (1/8)(r_1 + r_2 + r_8 + r_7)[(r_7 - r_2)(z_8 - z_1) - (z_7 - z_2)(r_8 - r_1)] \tag{19}$$

and δt is the time step. At the boundaries of the computation mesh, the current is defined by a path integration over a path which lies completely inside the plasma. The assumption that magnetic field gradients are zero between boundary cell centers and boundaries determines the current density at the boundary.

Finally, the contribution of the Ohmic heating to the internal energy is calculated. It can be shown that if the heating rate in each cell is proportional to the average of the square of the current densities at the four neighboring vertices, energy is conserved to $O(\delta t^2)$. Thus, the contribution of the current at vertex 1 to the Ohmic heating in cell a is given by the equation,

$$\delta i_a = (1/4)\, \eta j_1^2\, \delta t/\rho_a\,, \tag{20}$$

where ρ_a is the mass density in cell a.

Following the calculation of the resistive diffusion, the equations of motion are solved in explicit form. The resulting velocity field provides initial values for the iterative solution of the implicit equations. The explicit difference equations for vertex 1 are given by

$$\begin{aligned}\tilde{u}_1 = u_1 + \delta t[&r_1(z_2 - z_4)(2p + B_r^2 + B_z^2) \\ &+ (r_2 + r_4)\{(r_2 - r_4)(B_r B_z) - (z_2 - z_4)\, B_r^2\}]/4m_1\,,\end{aligned} \tag{21}$$

and

$$\begin{aligned}\tilde{v}_1 = v_1 - \delta t[&(r_2 + r_4)\{(r_2 - r_4)(2p + B_r^2 - B_z^2) \\ &+ (z_2 - z_4)(B_r B_z)\}]/4m_1\,,\end{aligned} \tag{22}$$

where u and v are the r and z components of $\mathbf{u}$, B_r and B_z are equal to $b_r/\mu^{1/2}$ and $b_z/\mu^{1/2}$, m_1 is half the mass enclosed by the momentum control volume, and the tilde signifies that the left hand side is an estimate of the velocity at time $t + \delta t$. The equations for vertices 2, 3, and 4 are obtained by cyclic permutation of the indices in Eqs. 21 and 22. With the solution of the equations of motion, the iteration for the solution of the implicit equations can begin.

For each component of the magnetic field, a quantity S is defined whose value is proportional to the error still remaining in the solution of the implicit equations. The quantity S_r for the r-component is defined by,

$$S_r \equiv ({}^LB_r - B_r)/\delta t - {}^LB_r\, DUDR - {}^LB_z\, DUDZ + {}^LB_r\, D, \tag{23}$$

and S_z for the z-component is defined by,

$$S_z \equiv ({}^LB_z - B_z)/\delta t - {}^LB_z\, DVDZ - {}^LB_r\, DVDR + {}^LB_z\, D, \tag{24}$$

where D is the difference approximation to the divergence of the velocity field, and $DUDR$, etc. are approximations to the derivatives $\partial u/\partial r$, etc. The superscript L denotes the solution of the Lagrangian step for time $t + \delta t$.

The iterative solution of these equations is performed by computing field variations which make the magnitude of S_r and S_z diminish, and the incremental changes in the velocity due to these field variations. The $(l + 1)$ iterate is calculated from the l-th iterate by a solution of the Newton–Raphson equations,

$$0 = S_r({}^{(l)}B_r\; {}^{(l)}B_z) + (\delta S_r/\delta B_r)\, \delta B_r + (\delta S_r/\delta B_z)\, \delta B_z\,, \tag{25}$$

and

$$0 = S_z({}^{(l)}B_r\; {}^{(l)}B_z) + (\delta S_z/\delta B_r)\, \delta B_r + (\delta S_z/\delta B_z)\, \delta B_z\,, \tag{26}$$

where the superscript (l) denotes the iterate number. The (l)-th and $(l + 1)$ iterate are related by the equation,

$${}^{(l+1)}B = {}^{(l)}B + \delta B.$$

The variations of S_r and S_z with B_r and B_z are given by the equations,

$$\delta S_r/\delta B_r = (1/\delta t) + (D - DUDR) + 2V_A{}^2\, \delta t/\delta z^2, \tag{27}$$

$$\delta S_r/\delta B_z = -DUDZ, \tag{28}$$

$$\delta S_z/\delta B_r = -DVDR, \tag{29}$$

and finally,

$$\delta S_z/\delta B_z = (1/\delta t) + (D - DVDZ) + 2V_A{}^2\, \delta t/\delta r^2. \tag{30}$$

These variations include the variation of the velocities with the magnetic field resulting in terms whose coefficient is the Alfvén speed,

$$V_A = ((B_r^2 + B_z^2)/\rho)^{1/2}. \tag{31}$$

The equations are derived for a general computation mesh, and can therefore be used for nonrectilinear Lagrangian meshes.

For the MHD calculation the iteration variable in the mass equation is the augmented pressure given by,

$$\bar{P} = P + (1/2)(B_r^2 + B_z^2), \tag{32}$$

The augmented pressure is simply the sum of the traces of the fluid and Maxwell stress tensors. The addition of the field to the fluid pressure alters the equation of state by the addition of a term proportional to the square of the density, and this term is accounted for by substituting the magnetoacoustic speed for the adiabatic sound speed in the proportionality between pressure and density changes in the iteration. This proportionality is now given by,

$$\delta\bar{P} = (C_S^2 + V_A^2)\,\delta\rho,$$

where the changes in density are given by solution of a Newton–Raphson equation derived from the mass equation [6]. Changes in the magnetic field and the augumented pressure produce changes in the velocity field which must be calculated. For example, at vertex 1, the component difference equations are,

$$\begin{aligned}\delta u_1 = {} & (\delta t/2m_1)[r_1(z_2 - z_4)\,\delta\bar{P} \\ & + (1/2)(r_2 + r_4)\{(r_2 - r_4)(B_z\,\delta B_r + B_r\,\delta B_z) - (z_2 - z_4)(B_r\,\delta B_r)\}]\end{aligned} \tag{33}$$

and,

$$\begin{aligned}\delta v_1 = {} & -(\delta t/2m_1)[(1/2)(r_2 + r_4)\{(r_2 - r_4)(\delta\bar{P} - B_z\,\delta B_z) \\ & + (z_2 - z_4)(B_z\,\delta B_r + B_r\,\delta B_z)\}].\end{aligned} \tag{34}$$

The equations for the changes in the other vertices are obtained by cyclic permutation of the indices in the equations above. When changes in the iteration variables, $\delta\bar{P}$, δB_r and δB_z in a cell become less than the tolerance level, the incremental momentum equation is bypassed. When changes become less than the tolerance level everywhere, the iteration ceases. The results are the density, pressure, magnetic field, and velocities at the advanced time. The Lagrangian step is completed by solving the internal energy equation. Before continuing, however, we will discuss the equations for the velocity derivatives in Eqs. 23 and 24.

The minimum requirements on the accuracy of the difference formulas for the velocity derivatives is that they be consistent with the differential equations being approximated. In addition, the solution of the magnetic field equations should conserve flux in the Lagrangian phase independently of the manner in which the computation mesh is rezoned. Difference formulas, for which the lowest order truncation errors are proportional to δr and δz, are given by equations of the same form as that for $DUDR$ in cell a,

$$DUDR = (1/2)\{(u_7 - u_2)(z_8 - z_1) - (u_8 - u_1)(z_7 - z_2)\}/A_a\,. \tag{35}$$

This equation is derived by expanding the differences $u_7 - u_2$ and $u_8 - u_1$ in a Taylor series, and solving the equations for $\partial u/\partial r$ and $\partial u/\partial z$ [9]. The velocity divergence D is given by,

$$D = (DUDR + DVDZ)(1 + (UOR)\,\delta t) + UOR, \tag{36}$$

where UOR in cell a, for example, is given by,

$$UOR = (u_7 + u_8 + u_2 + u_1)/(r_7 + r_8 + r_2 + r_1). \tag{37}$$

With these equations for the velocity derivatives, and a definition for the magnetic flux passing through an open surface bounded by a cell, as in cell a for example, given by,

$$\phi_a \equiv (1/8)(r_7 + r_8 + r_2 + r_1)\{B_{z_a}(r_8 + r_7 - r_2 - r_1) - B_{r_a}(z_8 + z_7 - z_2 - z_1)\}, \tag{38}$$

the error in flux conservation in each cell, each cycle is proportional to δr^2 and δz^2. The results in Section III show that the errors in flux conservation are small.

Finally, the internal energy equation is solved. The internal energy equation can be differenced so that total energy is conserved by balancing the change in the kinetic energy of the fluid against the work done by the internal degrees of freedom of the fluid. The change in the kinetic energy of a vertex can be equated to the work done by the resultant of the forces exerted on that vertex by the four neighboring cells. Conversely, the change in the internal energy of a cell can be equated to the work done by the forces it exerts on the four neighboring vertices. The work done is equal to the product of the force exerted and the displacement. If the displacement is calculated with the average of the old and time advanced velocities, the product of the resultant of the forces and the displacement is equal to the change in kinetic energy of the vertex. Thus, the sum of the changes in the internal energies of cells and kinetic energies of vertices over the entire mesh is zero and energy is conserved.

The change in the sum of internal and magnetic field energies is obtained by summing the work done by a cell on its four neighboring vertices, and is given by,

$${}^{L}\bar{i} - \bar{i} = \delta\bar{i}_1 + \delta\bar{i}_2 + \delta\bar{i}_3 + \delta\bar{i}_4\,, \tag{39}$$

where $\delta \bar{\imath}_1$, the work done on vertex 1 is given by,

$$\begin{aligned}\delta \bar{\imath}_1 = {} & (1/2)[({}^L u_1 + u_1)\{r_1(z_2 - z_4)\, {}^L\bar{P} \\ & + (1/2)(r_2 + r_4)((r_2 - r_4)\, {}^L B_r\, {}^L B_z - (z_2 - z_4)\, B_r\, {}^L B_r)\} \\ & - ({}^L v_1 + v_1)\{(1/2)(r_2 + r_4)((r_2 - r_4)({}^L\bar{P} - B_z\, {}^L B_z) \\ & + (z_2 - z_4)\, {}^L B_r\, {}^L B_z)\}]\, \delta t. \end{aligned} \tag{40}$$

The other contributions to $({}^L\bar{\imath} - i)$ are given by cyclic permutation of the indices in the equation for $\delta \bar{\imath}_1$. If the ratio, β, of the fluid to the magnetic field pressure is less than ϵ, the tolerance level in the iteration, errors in the calculation of the change in $\bar{\imath}$ are likely to be larger than the internal energy i. To prevent negative internal energies when β is less than ϵ, we calculate a fluid pressure from the density, ${}^L\rho$, and internal energy, i, available to us. The change in i is then calculated directly from an appropriately modified Eq. (40). When β is greater than ϵ, the internal energy, ${}^L i$, is obtained by subtracting the magnetic field energy from ${}^L\bar{\imath}$. Thus, whatever the value of β, the error in the total energy is never greater than $O(\epsilon)$.

B. *The Rezone Phase*

The rezone phase of the composite algorithm consists of two parts: the calculation of the grid motion, and the fluxing of thermodynamic variables from cell to cell due to the relative motion of the grid and fluid. The method and motivation for the calculation of grid motion for the almost Lagrangian calculation will be discussed first.

The grid motion is arbitrary. If the grid velocity is zero, the calculation is Eulerian, and if it is equal to the fluid velocity, the calculation is Lagrangian. Neither an Eulerian calculation nor a Lagrangian calculation is completely satisfactory. Difference approximations in the convection terms for an Eulerian calculation introduce computational diffusion, and distortion of the Lagrangian grid in shear flows introduces inaccuracy, and sometimes even numerical instabilities. The grid velocity can also be prescribed so that the mesh moves in a way intermediate between an Eulerian and Lagrangian mesh. The intermediate mesh, which we call an almost Lagrangian mesh, is free from Lagrangian instabilities and decreases computational diffusion by reducing the relative motion of the mesh and the fluid. The grid velocity for the almost Lagrangian mesh will be equal to the fluid velocity plus an additional, and presumably small, velocity which reduces the displacement of the computation mesh from an associated ideal mesh. The prescription for the ideal mesh will take into account the desire to reduce the average relative motion of the grid and fluid, as well as the desire for local rectilinearity. For instance, if the fluid is translating in bulk, the ideal mesh ought to translate with the same velocity. The one necessary requirement the prescription must satisfy is that it must define a unique ideal mesh for a given computation mesh.

(For example, the prescription presented later in the discussion uniquely defines an ideal mesh by solving an associated boundary value problem for which the boundary conditions are given by the computation mesh.) There is probably no universal prescription for an ideal mesh equally applicable to all problems. Certainly, the aptness of the prescription to the application will contribute to the success of an almost Lagrangian calculation.

The almost Lagrangian mesh will reduce computational diffusion more for transient flows than for steady flows. This is demonstrated by examining the properties of a model for computing the grid velocity. The grid velocity for an almost Lagrangian calculation is given by,

$$\mathbf{u}_g(\mathbf{r}, t) = \mathbf{u}_f(\mathbf{r}, t) - \boldsymbol{\delta}(\mathbf{r}, t)/\tau, \tag{41}$$

where $\mathbf{u}_g$ is the grid velocity, $\mathbf{u}_f$ the fluid velocity, $\boldsymbol{\delta}$ the displacement of the computation mesh from the ideal mesh, and τ the relaxation time for the displacement to reduce to zero if the fluid velocity were zero. Consider now a case where the ideal mesh is stationary in a frame moving with the average fluid velocity. In the moving frame the displacement of the computation mesh from the ideal mesh is,

$$\boldsymbol{\delta}(\mathbf{r}, t) = \int_0^t \mathbf{u}_g(\mathbf{r}, t')\, dt'. \tag{42}$$

Suppose the fluid velocity, $\mathbf{u}_f$, in the moving frame is equal to $\mathbf{U}(\mathbf{r})$ in the time interval $t_1 \leqslant t \leqslant t_2$, and zero otherwise. If initially $\boldsymbol{\delta}$ were zero, the grid velocity would be given by,

$$\mathbf{u}_g(\mathbf{r}, t) = \mathbf{u}_f(\mathbf{r}, 0) - \mathbf{U}(\mathbf{r}) \exp(-(t - t_2)/\tau)[1 - \exp(-(t_2 - t_1)/\tau)]. \tag{43}$$

For steady flow, that is for $t_1 = 0$ and $t_2 = t$, the grid velocity is,

$$\mathbf{u}_g(\mathbf{r}, t) = \mathbf{U}(\mathbf{r}) \exp(-t/\tau). \tag{44}$$

After a time the grid velocity becomes very small and the almost-Lagrangian mesh behaves like an Eulerian mesh. The one difference is that the mesh is displaced from the ideal mesh, which coincides with an Eulerian mesh in this case, by an amount,

$$\boldsymbol{\delta}(\mathbf{r}, t) = \mathbf{U}(\mathbf{r})\, \tau.$$

The displacement of the almost Lagrangian mesh is bounded for steady flows, and this is good, but the almost Lagrangian mesh introduces as much computational diffusion as a comparable Eulerian calculation.

On the other hand, for transient flow, the almost Lagrangian mesh reduces the relative velocity between the fluid and the grid. When the interval, $(t_2 - t_1)$, over which the fluid velocity is different from zero is shorter than τ, the relative velocity, $\mathbf{u}_{\text{rel}} = \mathbf{u}_g - \mathbf{u}_f$, satisfies the inequality,

$$| \mathbf{u}_{\text{rel}} | \leqslant | -((t_2 - t_1)/\tau) \mathbf{U} | < | \mathbf{U} | . \tag{45}$$

While the fluid velocity is nonzero, the mesh essentially follows the fluid. When the fluid velocity is once again zero, the mesh slowly relaxes back to the ideal mesh thus reducing the average relative velocity. All truncation errors in the convection terms with nonlinear dependence on the relative velocity will consequently be reduced. Therefore, the need to add explicit positive diffusion to stabilize the equations is also reduced.

The almost Lagrangian calculations presented here use a prescription for an ideal mesh developed by Browne [10]. This prescription gives a mesh with local rectilinearity, which allows adjacent zones to vary in size. Rectilinearity reduces truncation errors in the difference equations, and the freedom to use variable zone sizes allows regions where strong compressive gradients exist to be well resolved. (However, these properties also mean that the ideal mesh and the computation mesh will differ more in shearing flows than in compressional flows.) Thus, this ideal mesh will be especially useful for calculations of compressive flow where it will introduce a minimum of relative motion.

To see how this prescription is obtained, consider the integral,

$$I = \int \{(\nabla i)^2 + (\nabla j)^2\} \, dV, \tag{46}$$

where ∇ is the gradient operator with respect to the Eulerian coordinates r and z, and i and j are mesh coordinates, $i = i(r, z), j = j(r, z)$. When the boundary conditions specify j on boundaries where i is constant, and i on boundaries where j is constant, the problem of minimizing I is a Dirichlet problem. The solutions, i and j, to this problem are conjugate harmonic functions which satisfy the Cauchy conditions and thus have the property that lines of constant i and lines of constant j are perpendicular [11]. These solutions satisfy the condition of local rectilinearity and therefore define a desirable ideal mesh. The connection with the computation mesh is established below.

A difference approximation to Eq. (46) is obtained by using the relations between mesh coordinates and Eulerian coordinates given by Schulz [12]. For example, the variation of i with r is replaced with the variation of z with j by means of the equation,

$$\partial i/\partial r = (\partial z/\partial j)/J,$$

where J is the Jacobian of the transformation between mesh and Eulerian coordinates. When the equations are discretized, one finds the Jacobian is equal to the area of a computation cell. With the substitution of these relations into the integral, and the replacement of the integral with a sum over discrete mesh coordinates, the difference analog, $\bar{I}$, to I in Eq. (46) is obtained. It is given by

$$\bar{I} = \sum_{i,j} \frac{(\Delta_j r_i{}^j)^2 + (\Delta_i r_i{}^j)^2 + (\Delta_j z_i{}^j)^2 + (\Delta_i z_i{}^j)^2}{A_{i+1/2}^{j+1/2}}, \tag{47}$$

where $A_{i+1/2}^{j+1/2}$ is a computation cell area, $(r_i{}^j, z_i{}^j)$ are the Eulerian coordinates of points with mesh coordinates (i, j) and $\Delta_l z_l{}^k$ is the forward difference, $(z_{l+1}^k - z_l{}^k)$ of $z_l{}^k$ with respect to the repeated index l. When $\bar{I}$ is minimized with respect to one vertex at a time, the equations are linear and can be directly solved for each vertex displacement. The displacement of vertex 0 which minimizes $\bar{I}$ in the momentum control volume (shown in Fig. 2) surrounding vertex 0 is given by,

$$\delta r_0 = \sum \{(r_0 - (1/2)(r_i + r_{i+1}))/Ai_{i+1}\} \Big/ \sum (1/Ai_{i+1}), \tag{48}$$

and,

$$\delta z_0 = \sum \{(z_0 - (1/2)(z_i + z_{i+1}))/Ai_{i+1}\} \Big/ \sum (1/Ai_{i+1}), \tag{49}$$

where $i = 1, 2, 3, 4$ (cyclic), and Ai_{i+1} is given by,

$$Ai_{i+1} = (1/2)[(r_i - r_0)(z_{i+1} - z_0) - (z_i - z_0)(r_{i+1} - r_0)]. \tag{50}$$

These displacements are the displacements from the ideal mesh, defined as the mesh for which $\bar{I}$ is a minimum, and are to be substituted into Eq. (41) to calculate

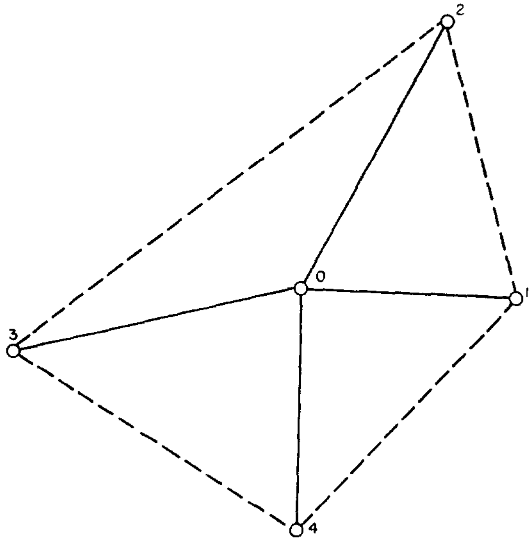

FIG. 2. The dashed line surrounds the momentum control volume corresponding to vertex 0.

the grid velocities. The boundary conditions for which $\bar{I}$ is to be minimized are given by the computation mesh from the previous cycle, and it is easily shown that the mesh for which $\bar{I}$ is a minimum with given boundary conditions is unique. Thus, for a given computation mesh there is a unique ideal mesh.

Simple test cases show that reducing the displacements calculated from Eqs. (48) and (49) makes the angles subtended by the boundaries about each vertex more nearly right angles. Thus, as the zone size tends to zero, the ideal mesh, for which these displacements are zero, will converge to the solutions to Eq. (46) and thus will share with them the property of local rectilinearity. Therefore, repeated calculation of the grid velocities necessary to reduce the displacements calculated from Eqs. (48) and (49) relaxes the computation mesh toward an ideal mesh with local rectilinearity.

In an almost Lagrangian calculation, grid velocities are calculated once each cycle from Eqs. (41), (48), and (49). The relaxation time, τ appearing in Eq. (41) should be set to the largest value consistent with preventing Lagrangian instabilities. It should be noted that for δt equal to twice τ, the displacements are reduced to zero in one time step. However, δt is typically much less than τ, so that the reduction in the grid displacement each time step depends essentially linearly on δt. Therefore, the motion over an interval of time will be nearly independent of the number of time steps in that interval.

The relative motion of the mesh and fluid in an MHD calculation requires the exchange of magnetic field between cells in addition to the exchange of material. This exchange is given by the surface integral in Eq. (16). In finite difference form, this integral will be represented by four flux terms, one for each side of a computation cell, and each of the form of a product of the change in cell volume due to the motion of the side and the value of the component of the magnetic field on that side. These flux terms can be obtained by replacing the density with the field components in the mass flux equations given by Hirt and Amsden [6, 7]. With these fluxes, the approximation to Eq. (16) is given by

$$^{R}B_{z0} = {}^{L}B_{z0}\,{}^{L}V/{}^{R}V + \sum \phi_{zll+1}, \tag{51}$$

and

$$^{R}B_{r0} = {}^{L}B_{r0}\,{}^{L}V/{}^{R}V + \sum \phi_{rll+1}, \tag{52}$$

where l are the labels of the vertices of a computation cell, $l = 1, 2, 3, 4$ (cyclic), ϕ_{ll+1} is the flux of the magnetic field across the boundary $ll + 1$, and the superscript R means evaluated after completion of the rezone. The coordinates after rezone are given by,

$$^{R}r = r + u_g\,\delta t, \tag{53}$$

and

$$^{R}z = z + v_g\,\delta t. \tag{54}$$

These values are substituted into Eq. (19) to evaluate ${}^{R}V$ with appropriate changes of indices. All other thermodynamic variables are rezoned as given by Hirt and Amsden [6, 7], including provisions for upstream differencing.

There are two cases where the accuracy of flux conservation can be checked. When $\mathbf{u}_g$ is equal to the fluid velocity, the rezone phase is inoperative and the flux conservation properties of the Lagrangian phase determine the flux conservation properties of the composite algorithm. When $\mathbf{u}_g$ is zero, the composite algorithm is Eulerian and the difference equations for the field may be compared with those obtained by differencing Eq. (1) in conservation form. As has already been mentioned, the Lagrangian phase conserves flux to $O(\delta r^2, \delta z^2)$ in each cell, each cycle. By comparison, a composite Eulerian scheme ($\mathbf{u}_g = 0$) conserves flux to $O(\delta r, \delta z)$. The principal source of error is the interpolation to find the value of the field on a cell boundary. The interpolation could be improved, but the present version is economical and the results, as shown, are quite acceptable when the almost Lagrangian mode is used.

C. *Convergence of the Iteration*

The implicit dynamical equations are solved by the method of successive overrelaxation, for which an algorithm is given in Section A. Detailed discussion of the convergence properties of the algorithm are beyond the scope of this paper. However, the algorithm is at least as rapidly convergent as a Jacobi, or simultaneous, iteration whose properties are well understood. To gain insight into the special advantages in calculational efficiency of an implicit formulation of the dynamical equations, the properties of a Jacobi iteration, specifically the rate at which numerical information is propagated over the mesh, will be assumed for the solution algorithm.

The Jacobi iteration increases the domain of dependence of the numerical solution by one cell in each direction each iteration. The iteration can converge only if the numerical domain of dependence is larger than the physical domain of dependence. Thus the Courant number, the number of computation cells a physical signal can propagate in one time step, determines the number of iterations necessary to reach convergence to a specified precision.

In a given problem with many modes of signal propagation there are important and unimportant signals. An unimportant signal is one that is indistinguishable from allowed error, when that error is small enough that decreasing it does not alter the qualitative results of a calculation. Thus, an effective Courant number can be defined as the number of computation cells important physical signals can propagate in a time step. The effective Courant number is often much less than the ordinary Courant number, which is defined with the maximum possible signal speed, especially in very low speed flows where only very weak acoustic or magnetoacoustic waves are generated. The equations are complete, however, and do allow for acoustic waves if they should develop. Thus, for a given number of iterations, say that number which corresponds to an effective Courant number of one, the time step for an implicit calculation can be much larger than for a comparable explicit calculation for low speed flows. For high speed flows, those for which the Mach number is greater than 0.1 perhaps, the time step for explicit and implicit calculations will be comparable.

Results

We illustrate the properties of the composite algorithm by presenting examples of its application to several sample problems. In the first example, the axisymmetric expansion of a hot, highly conducting spherical pellet in a magnetic field is calculated, and the effect of various zoning techniques on the computational diffusion of the magnetic field is demonstrated. The initial conditions for the pellet problem are given in Table I. The initial magnetic field is the result of the superposition of a uniform axial field and the dipole field that makes the field inside th pellet exactly zero [13]. The field outside the pellet is given by,

TABLE I

Data for the Pellet Expansion Calculations

Initial Conditions (in arbitrary units)		
	Pellet	Background Plasma
Density, ρ	10.0	0.01
Internal Energy, i	10.0	1.0
Pellet Radius, a	1.0	
General Data		
Ambient Field Intensity, B_0 — 1.0 (arbitrary units)		
Magnetic Reynolds Number, Rm — 200.0		
Resistive Diffusion Coefficient, η/μ — 0.05 (arbitrary units).		
Convections Terms — upstream differencing		

$$\mathbf{B} = \hat{\mathbf{z}} B_0[1 + a^3((1/2)\, r^2 - z^2)/(r^2 + z^2)^{5/2}] - \hat{\mathbf{r}} B_0[3/2rza^3/(r^2 + z^2)^{5/2}], \quad (55)$$

where $\hat{\mathbf{r}}$ and $\hat{\mathbf{z}}$ are radial and axial unit vectors, B_0 is the intensity of the uniform field, and a is the radius of the pellet. The pellet and initial field are shown in Fig. 3a, where contours of constant magnetic flux and fluid marker particles are

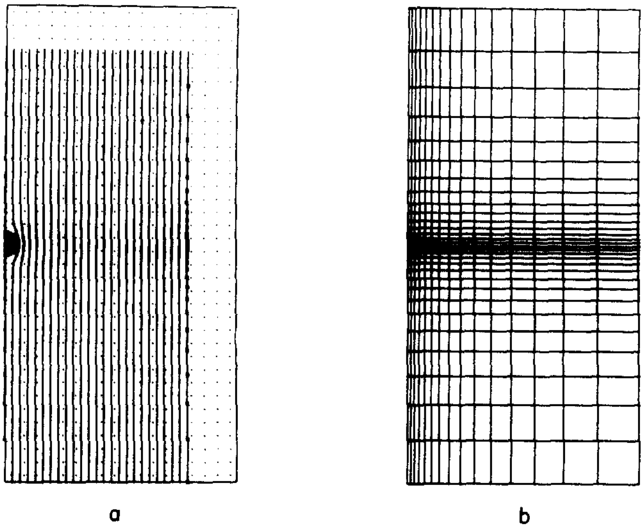

FIG. 3. The initial conditions for the pellet expansion calculation are shown. Figure 3a shows the magnetic field lines, and 3b the computation mesh.

plotted. The left border is the axis of symmetry of the cylindrical coordinate system. On it is the small, dense circle of particles which mark the pellet. The rectangular array of widely separated particles serves as a raster. The square root of the flux is plotted, so that uniform field intensity results in equally spaced contours. The same contour values are shown in all figures.

The initial computation mesh is shown in Fig. 3b. There are 15 zones radially and 30 axially. The smallest zones, at the center of expansion, have the dimensions $\delta r/a$ and $\delta z/a$ equal to 0.25, and increase both radially and axially in a geometric sequence with the common factor 1.2.

Although the fluid pressure inside the pellet-field interface is isotropic, the magnetic field back pressure outside is not. It is zero in the axial direction at the poles, and nonzero in the radial direction at the equator. Thus, the expansion becomes aspherical very rapidly. In fact, the pellet collapses radially and expands axially until it reaches its steady state, an infinitely long filament along the axis of symmetry. A computation will give the correct steady state only if the effective diffusion time $\tau_D \cong \mu D^2/\eta_{\text{eff}}$, where D is the diameter of the pellet and η_{eff} is the effective resistivity, is long compared with the expansion time. An effective magnetic Reynolds number, which measures the relative importance of diffusion and convection terms in Faraday's law, is defined by,

$$R_m \equiv uD\mu/\eta_{\text{eff}}, \tag{56}$$

where u is a characteristic fluid speed. This number measures the differences due to computational diffusion among the calculations with the various zoning options. Four such calculations are shown in Figs. 4 and 5 at a time of one arbitrary time unit. Figure 4a shows an Eulerian mesh. Figure 4b shows a Lagrangian mesh in which the mesh vertices move with the fluid velocity. Figure 4c shows an almost-Lagrangian calculation, called ALE1. Figure 4d shows an almost Eulerian calculation, called ALE2. The relaxation factor, γ, which is equal to $\delta t/\tau$, (cf. Eq. (41)) is set to 0.1 in ALE1 and 2. in ALE2. Increasing γ makes the zones progressively

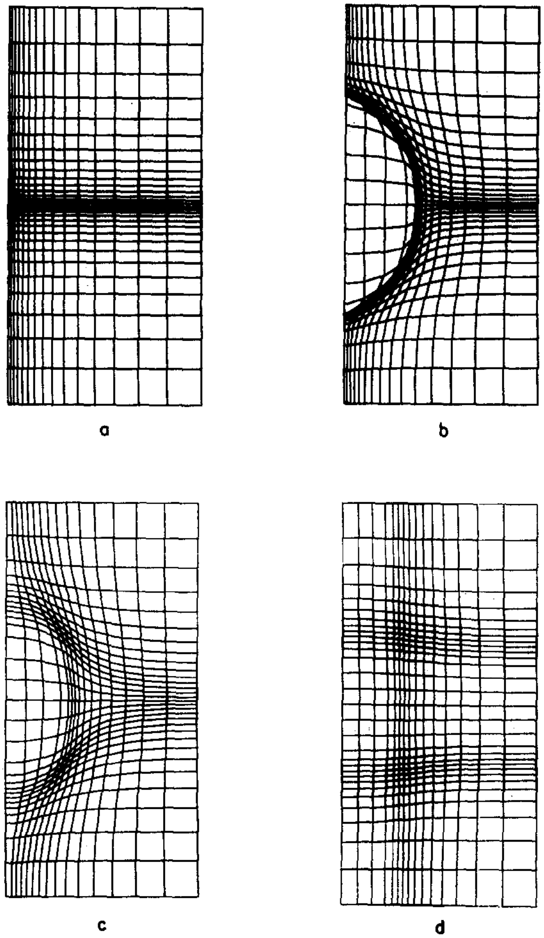

FIG. 4. The computation meshes for the pellet expansion calculation at one time unit are shown. Figure 4a shows the Eulerian mesh, Fig. 4b the Lagrangian mesh, Fig. 4c the ALE1 mesh, and Fig. 4d the ALE2 mesh.

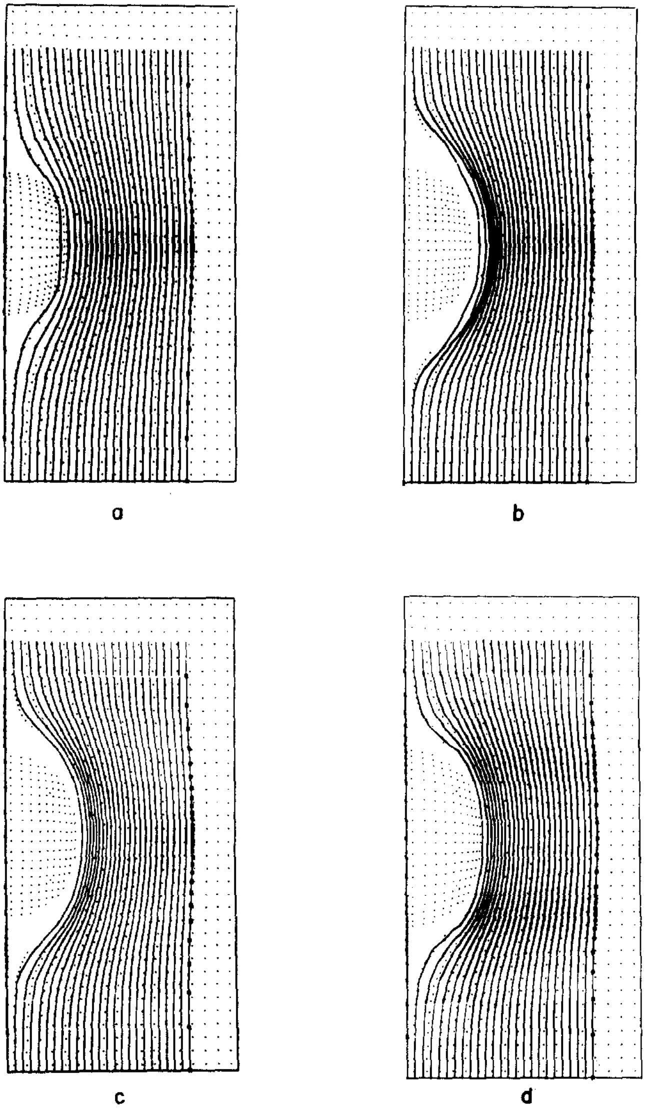

FIG. 5. The magnetic field plots corresponding to the computation meshes in Fig. 4 are shown.

more rectilinear. However, since any rectilinear mesh is a solution to Laplace's equation, even when γ equals 2.0 the relaxer allows cells to compress and dilate. Figure 5 shows the corresponding magnetic flux plots. The effects of computational diffusion are most clearly seen in the Eulerian calculation in which pellet marker particles have crossed the innermost flux contour. A quantitative comparison of the calculations is given in Table II, where the effective Reynolds number is given.

TABLE II

Dependence of Computational Diffusion on the Zoning Mode

Zoning mode	Relaxation coefficient, γ	Effective diffusion coefficient, η_{eff}/μ	Magnetic reynolds number, $uD\mu/\eta_{\text{eff}}$
Eulerian	—	1.2	8.0
Lagrangian	—	0.05	200.0
Almost-Lagrangian	0.1	0.05	200.0
	1.0	0.213	47.0
	2.0	0.225	44.0

The effective diffussion coefficient is obtained by measuring the thickness of the current sheet separating the interior and exterior of the pellet at 0.8 time units. Computational diffusion introduced by the Eulerian calculation reduces R_m from the intended value 200 to 8, but the diffusion introduced by relaxing the mesh with γ equal to 0.1 leaves the Reynolds number at its intended value.

The effect of the computational diffusion on the pellet's approach to steady state is illustrated in Fig. 6, where the pellet's expansion at 2.0 time units, as computed

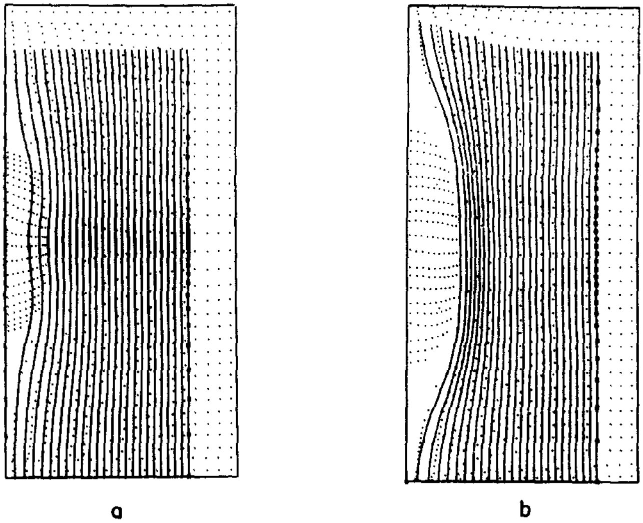

FIG. 6. The magnetic field plots for the Eulerian and the ALE1 calculations are shown at two time units in Figs. 6a and 6b, respectively.

with an Eulerian mesh, is compared with its expansion, as computed with the ALE1 mesh. Because of the greater ability of an undiffused field to continue to drive the axial expansion of the pellet, the axial expansion is clearly greater in ALE1 than in the Eulerian calculation.

Another significant difference among the calculations with the various zoning options is the size of the time step. Figure 7 shows the time step plotted against the time for all four calculations. (The difference in computing time per cycle among the calculations is less significant. Moving the mesh in an almost-Lagrangian calculation increases the computing time from 1.0 to 1.2 milliseconds per cell per cycle on the CDC 7600 from a comparable Eulerian calculation.) The time step is determined by the more restrictive of two fundamental stability conditions; the positive volume condition,

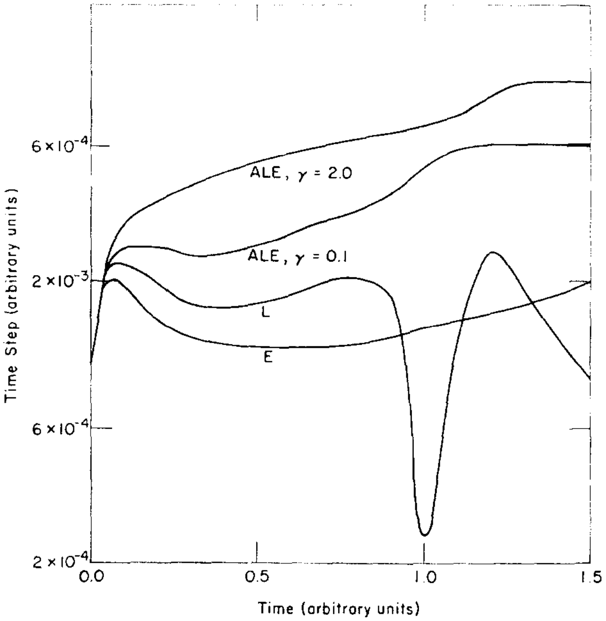

FIG. 7. A comparison of the size of the time step for the various calculations of the pellet expansion is given by this plot of the time step against the time.

$$\max_{i,j}\{u\ \delta t/\delta x\} \leqslant 1, \qquad (57)$$

where $\max_{i,j}$ means the maximum over the entire mesh, and the stability condition for the explicit viscous diffusion calculation,

$$\max_{i,j}\{(\lambda + 2\mu')\ \delta t/\rho\ \delta x^2\} \leqslant 1. \qquad (58)$$

(We have neglected the resistive diffusion because it is small.) In these calculations, the kinematic viscosity, $(\lambda + 2\mu')/\rho$, has been replaced by a variable viscosity. The viscosity is calculated to cancel destabilizing truncation errors in the momentum equation, which occur even when donor cell mass convection is used [3]. This variable viscosity is given by,

$$(\lambda + 2\mu') = ((1/2)\, u^2\, \delta t + 0.7 u \delta x)\, \rho. \tag{59}$$

It is possible to calculate the quantity $u\delta t/\delta x$ from the ratio of the time step determined by the inequality (57) to that determined by inequality (58). The ratio lies between 1.15 and 1.20 for all four calculations indicating that δt goes directly as $\delta x/u$. The ALE2 time step is largest simply because the zones are larger in regions of fluid flow in ALE2 than in any other calculation. ALE1 is seventy per cent as fast. That is, the cost of minimizing γ to reduce computational diffusion is an increase in the computing time of thirty per cent.

The second example is the calculation of the compression of a dynamic z-pinch [14, 15]. Four features of the algorithm are illustrated by this calculation. First, the algorithm applies equally well to cylindrical coordinates and Cartesian coordinates. Second, the algorithm may be used with a nonrectilinear mesh. Such a mesh, for one quadrant of a pinch, has been generated by mapping a 10 × 10 rectangular mesh bounded by the lines $x = -\infty$, $x = \ln(10)$, $y = 0$ and $y = \pi/2$ onto a quandrant of a circle of by means of a transformation given by,

$$x' = \exp(x) \cos y,$$

and

$$y' = \exp(x) \sin y.$$

The mesh for this two-dimensional calculation is shown in Fig. 8a. The axis of the pinch is in the lower left corner. Third, the mesh relaxer may be applied to nonrectilinear meshes. This is shown by Fig. 8c, where the computation mesh at 0.1 time units problem time is clearly of the same form as the initial mesh even though the outer zones have dilated and the inner zones have compressed. Since the mapping function is analytic, the transformed mesh is also a solution of Laplace's equation. Thus, the action of the relaxer is to drive the mesh toward a mesh of similar form as the fluid causes dilation and compression of the computation zones. Fourth, the calculation is stable even when the Courant condition is violated. In the outer region of the mesh, farthest from the axis, the density is equal to 1×10^{-5}. This gives a ratio of signal speed to maximum explicit mesh signal speed, $V_A \delta t/\delta x$, approximately equal to ten, instead of a value less than one necessary to satisfy the Courant condition. The absence of numerical instabilities in this calculation is evidence that the implicit formulation has solved the Alfvén problem.

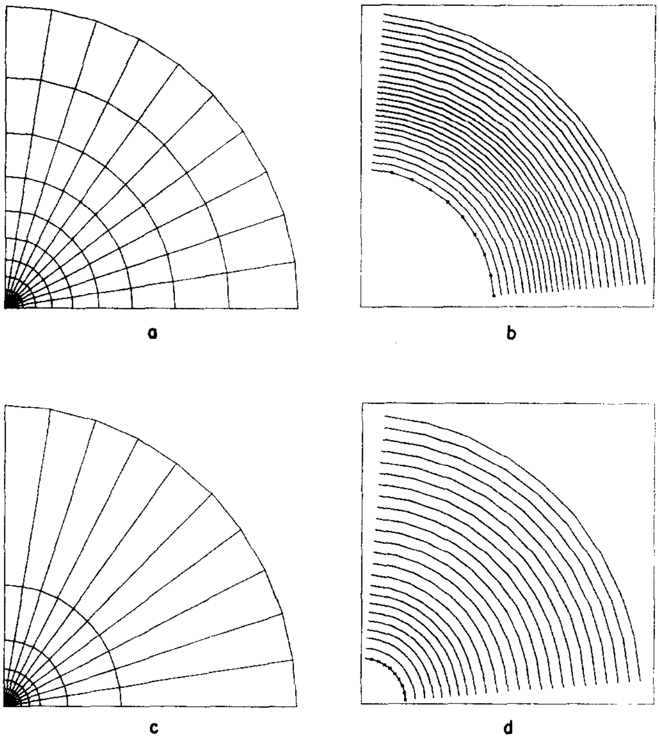

FIG. 8. The computation mesh and magnetic field plots are shown for zero and one tenth time units. The abscissa is the x-coordinate, the ordinate the y-coordinate. The length units are arbitrary.

Figure 9 shows the radial variation of the plasma density and magnetic field intensity at various times. (The radial variation was observed to be independent of azimuth.) Figure 9a shows the initial conditions. The succeeding plots show the three phases of a z-pinch [15]. In the first phase, the inward motion of the plasma-field interface drives a cylindrical shock toward the axis. The shock is shown in Fig. 9b, where the density and the field intensity are plotted at a problem time of 0.05 units. In the second phase, the shock reflects off the axis, causing a very large, momentary compression at the axis. This is shown in Fig. 9c, where a density maximum of 34.4 at the axis can be seen. (Also shown are the curves from an Eulerian calculation. The maximum density at the axis is only a tenth as large. Not only is there less resolution at the axis in the Eulerian calculation, but there is also more diffusion of the field into the plasma, and both tend to reduce the maximum compression.) In the third phase the outward moving shock intercepts the still inward moving plasma interface and drives it outward and a rarefaction propagates

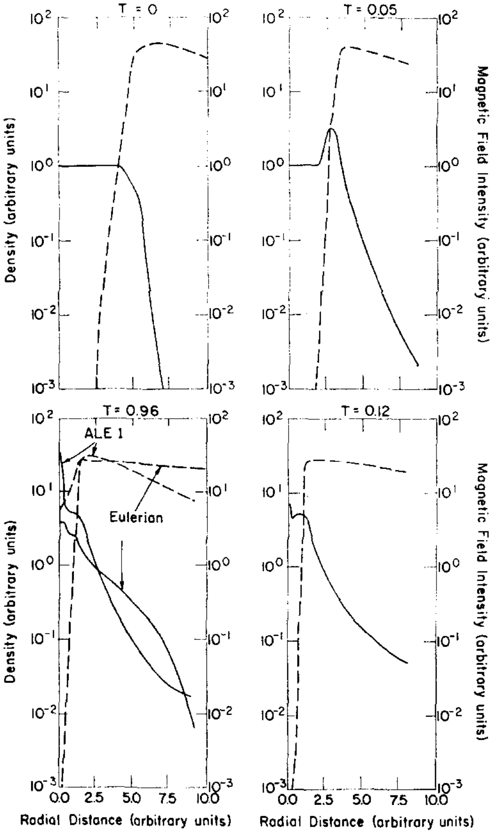

FIG. 9. The radial variations of the density and magnetic field are shown at 0.0, 0.05, 0.096, and 0.12 time units in Figs. 9a, 9b, 9c, and 9d, respectively. The solid curves correspond to the density, the dashed curves to the magnetic field intensity.

back into the plasma. The density profile at the time of interception is shown in Fig. 9d at a time of 0.12 units, but the outward moving shock cannot be seen. This is not an unexpected result, for the plasma never occupies more than eight radial zones, and the considerable momentum diffusion due to the action of the alternate node coupler discussed in Refs. [6 and 7] damps the outward moving shock. The result of the artificially high momentum diffusion is that the apparent equilibrium radius of the column, about 1.33 problem units, is larger than it should be as shown in Fig. 10. That is, the density is lower and the temperature is higher than would result from a less momentum diffusive calculation. For reference, the equilibrium radius would be 0.611 units if the compression were adiabatic.

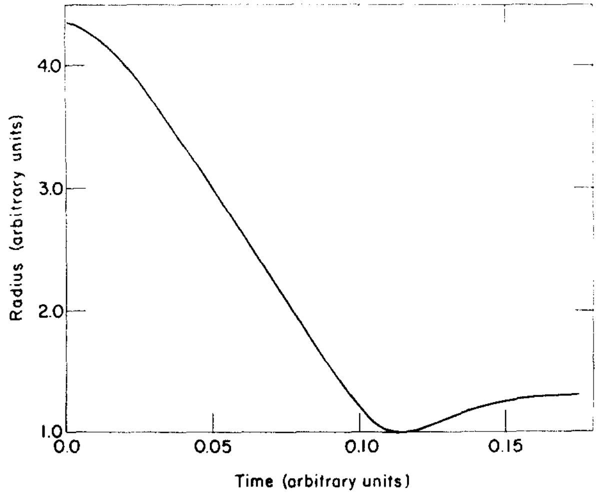

FIG. 10. The radius of the plasma column is plotted against the time for the z-pinch calculation.

Figures 8b and 8d show the magnetic flux contours for problem times of 0.0 and 0.1 time units. To the outermost contour in Fig. 8b is assigned the value 105.58, and to the outermost contour in Fig. 8d is assigned the value 105.30, giving direct evidence that the algorithm conserves magnetic flux very well.

In these calculations, the time step is determined by the size of the smallest zones, which lie on the axis. When the converging shock reaches the axis, the axial zones become very small and make the time step small. The result is that this calculation with only 100 zones, required 1400 computation cycles at a cost of three minutes on the CDC 7600. In cases such as this, the calculation can be made more economical by setting a minimum zone size.

CONCLUDING REMARKS

We have reported the extension of the ICED–ALE technique to magnetohydrodynamics, and the application of Browne's mesh relaxing algorithm [10] to allow the use of almost Lagrangian zoning. The sample calculations presented illustrate that the algorithm solves the Alfvén problem and reduces computational diffusion. The algorithm can be applied to many other problems as well. For example, it is easy to add anisotropic heat and electrical conductivity because, with Lagrangian zoning, cell boundaries can be made to follow field lines, and a complex geometry problem is replaced by a trivial indexing problem. More generally, Lagrangian zoning can also follow fluid–fluid interfaces as easily as fluid–field interfaces.

There is one problem for which we can offer no completely satisfactory solution. As we have pointed out in the discussion of the z-pinch calculation, the algorithm is momentum diffusive. Part of the diffusion is due to the node coupler which suppresses the alternate node or coasting instability [6]. The diffusion introduced by the node coupler is proportional to δr^2 and δz^2, and thus can be reduced by finer zoning.

Finally, the first order time differencing used for the dynamical equations introduces diffusionlike truncation errors which can be quite large. However, the intent in this paper has been to address the problem of diffusion caused by fluxing material from cell to cell, and the calculational examples included were chosen to demonstrate reduction of diffusion for cases where fluxing errors are large. Other problems may be encountered where first order accuracy is not sufficient, and for these problems second order accurate differencing in time must be used. The scheme presented here is easily modified to give second order accuracy in the time step.

Acknowledgments

The authors thank members of Groups T-3 and J-10 at the Los Alamos Scientific Laboratory for their interest and assistance. We especially wish to thank P. Browne, B. J. Daly, and C. W. Hirt for their substantial contributions to this work. This work was performed under the auspices of the United States Atomic Energy Commission.

References

1. Keith V. Roberts and D. E. Potter, *Meth. Comp. Phys.* **9** (1970), 339.
2. J. P. Boris, "Physically Motivated Solution of the Alfvén Problem," unpublished.
3. C. W. Hirt, *J. Comp. Phys.* **2** (1968), 339.
4. J. P. Boris, *J. Comp. Phys.* **11** (1973), 38.
5. F. H. Harlow and A. A. Amsden, *J. Comp. Phys.* **8** (1971), 197.
6. C. W. Hirt and A. A. Amsden, An Arbitrary Lagrangian–Eulerian Computing Method for All Flow Speeds, submitted for publication to *J. Comp. Phys.*
7. A. A. Amsden and C. W. Hirt, "YAQUI: An Arbitrary Lagrangian–Eulerian Computer Program for Fluid Flows at All Speeds," Report No. 5100, Los Alamos Scientific Laboratory, Los Alamos, NM, 1972.
8. I. Lindemuth and J. Killeen, Proceedings of the Fourth Conference on Numerical Simulation of Plasmas, U. S. Naval Research Laboratory, Washington, D. C., 1970.
9. Leroy Amunrud and S. Robert Orr, "A Note on Inverted Centers of Pressure and Crossed Mass Points in Two-Dimensional Hydrodynamics Calculations," unpublished report, Los Alamos Scientific Laboratory, Los Alamos, NM, 1958.
10. P. Browne, unpublished notes, Los Alamos Scientific Laboratory, Los Alamos, NM.

11. Bernard Epstein, "Partial Differential Equations," Chaps. 6 and 7, McGraw–Hill, New York, 1962.
12. William D. Schultz, *Meth. Comp. Phys.* **3** (1964), 1.
13. John David Jackson, "Classical Eletrodynamics," Chap. 5, p. 166, Wiley, New York, 1962.
14. L. A. Artsimovich, "Controlled Thermonuclear Reactions" (A. C. Kolb and R. S. Pease, Eds.), Chap. 5, Gordon and Breach, New York, 1964.
15. G. H. A. Cole, *Sci. Progr.* **47** (1959), 437.