A Magnetohydrodynamic Particle Code for Fluid Simulation of Plasmas

J. N. LEBOEUF, T. TAJIMA, AND J. M. DAWSON

Department of Physics, University of California, Los Angeles, California 90024

Received November 4, 1977; revised September 19, 1978

A novel Lagrangian type of MHD code has been developed by treating elements of the fluid as finite-sized particles. The particle quantities, i.e., position, mass, momentum and temperature, are pushed in a Lagrangian way, while the magnetic fields are advanced in an Eulerian manner. The fully Lagrangian fluid is represented by a distribution of Gaussian-shaped particles. Drag forces between particles with different velocities in the same cell prevent extensive multi-streaming from developing. A combination of finite differences, to calculate the magnetic field, and fast Fourier transforms, to evaluate the pressure gradient term, guarantees momentum and magnetic flux conservation. The use of particles eliminates many difficulties often associated with Eulerian codes such as, for example, negative densities. The method also means that any particle code of which there are many can be converted to a fluid code. The codes have been extensively tested with the propagation of sound waves, Alfven and magnetosonic waves among others. Applications of the codes to hydrodynamics and magnetohydrodynamics in one, two and three dimensions, in cartesian as well as in toroidal geometry, are further discussed.

I. INTRODUCTION

In the past two decades two paths have primarily been followed in plasma physics to simulate complicated plasma phenomena. The first models the plasma as a fluid and leads to the traditional fluid codes while the second reconstructs the plasma from individual particles and gives rise to the more recent particle codes. Although fluid codes lack facilities to incorporate self-consistent transport coefficients crucial to the study of plasmas, they have been extensively used because realistic geometries are handled with relative ease and slow MHD time scales can be readily followed. Since particle codes treat the plasma dynamics rigorously, they prove very helpful in the study of difficult nonlinear problems. However it is almost never possible to carry in the computer memory as many particles as are involved in the actual physical problem of interest nor is it possible to cover the wide range of time scales that occur in controlled thermonuclear research (CTR) situations within feasible simulation times. A great leap forward to tackle the first difficulty was brought about by the finite size particle codes [1–4]. This method to simulate collisionless plasmas significantly relaxed the condition on the number of particles required for meaningul simulation. Due to the finite extent of the particles only a tractable number of these macroparticles is needed to effectively suppress excessive noise due to short range

0021-9991/79/060379-30$02.00/0

Coulombic collisions and yet long range forces and thus collective phenomena are correctly represented. The second difficulty with particle codes has been harder to overcome to date. The fundamental time step used in particle pushing is restricted by the fastest occuring phenomenon whose frequency is usually either the plasma, electron cyclotron, or the photon frequencies. This restriction is extremely severe for CTR applications, since such plasmas most often encompass short time scale (10^{-12} second) phenomena up to confinement time scale (10^0 second) phenomena. Various schemes have been tried to bridge this many orders of magnitude gap in the time scales. The following are just a few among many, if piecemeal, attempts: magnetostatic codes [5], which neglect a fast radiation term to enlarge the time step, hybrid (electrons as fluid-ion particles) codes [6] and quasi-neutral codes [7]. The time step taken in the last two is generally the ion plasma period.

The MHD code we present in this article is in line with this effort to lengthen the time step as much as possible and simulate long time scale phenomena and as such is at the end of the line, since the largest time scale relevant to plasma physics is the MHD time scale. The present code exploits many of the techniques developed in particle codes. We employ finite size particles to represent elements of a single neutral fluid. Their shape factor now represents a mass distribution rather than the charge cloud of the electrostatic codes. In addition to the quantities that are carried in particle codes, such as position and velocity, particles can be assigned other quantities, such as temperature. A spatial mesh is superposed on the particles for computational convenience and fluid quantities are naturally accumulated as averages at mesh points from the particle quantities. Field quantities, such a magnetic fields, are only assigned at the mesh points. Fast Fourier transform techniques can be used to calculate forces at the grid points from the accumulated quantities. The particle pressure gradient is calculated from the density in Fourier space as is the electric field in the particle codes [8]. The magnetic force is calculated through the magnetic stress tensor which is evaluated at grid points. A difference from the magnetostatic or electromagnetic particle codes is that the self-consistent current is given as a grid quantity. In some cases fluid models tend to develop multistreaming of particles within one cell (particles in the same cell have quite different velocities) such as with shock formation. In applications of this code to hydrodynamics, it is not tolerated that particles may have significantly different velocities while occupying the same region of space, since a classical fluid is collisional in nature. Drag forces are then introduced between particles in the same cell to damp out any local anaomalies should they occur and allow particles to "communicate" and in this way insure a smooth flow. One might consider such drag forces arising from local turbulence which would occur if multistreaming were to arise. Phenomenological coefficients could be used to model these effects if they are known.

From the standpoint of fluid codes, we can look upon our scheme as a new method to solve the Lagrangian fluid equations. Traditionally two methods of extreme sort have been used to simulate fluid dynamics: the Lagrangian and Eulerian schemes. Conventional Lagrangian codes [9] create cells moving with fluid elements. For more than one dimension such an approach encounters serious difficulties in keeping track of the cells since the fluid can entangle in a very complicated way. In the Eulerian approach [10] the mesh of cells is fixed in a frame of reference at rest with the observer and the fluid streams by across the mesh. Its advantage is that the calculation proceeds without difficulty when there are large distortions in the fluid. Apart from being overly diffusive [11], this scheme has serious flaws, most notably the fluid density can go negative. The main difficulty lies in the advective terms of the Eulerian equations. To cure this, some numerical terms like, for example, additional diffusion are introduced in the continuity equation. Recently the elaborate techniques [12] of flux corrected transport have been introduced and have successfully handled these problems; however, they extract an expanse in terms of the speed and complexity of the code. Since in the present code, "particles" carry the quantities associated with fluid elements across the fixed background mesh, the advective terms do not appear and the main difficulty of the Eulerian scheme is readily removed. Their finite size provides efficient coarse-graining of the particle quantities to obtain fluid quantities.

This development parallels two updates, GAP [13] and PAL [14], of the particle in cell (PIC) [15] method, the pioneer in fluid simulation using particles. PIC distributes fluid quantities on a mesh and has the particles carry mass weighted portions of these quantities from cell to cell. This is unfortunately just as diffusive [14] as Eulerian hydrodynamics codes except for mass. Both GAP and PAL modify PIC to fully represent a Lagrangian fluid and use the mesh only as calculational convenience, which procedure is in our case a natural consequence of applying finite size particles. Both GAP and PAL employ the rather cumbersome area weighting method to give particles their spatial extent and conventional finite differences to calculate pressure gradients.

The organization of the paper is as follows. In Section II, extensive description of the present model and its conservative properties are presented. Section III is devoted to a descrption of the codes developed according to our model in one, two and three dimensions in cartesian as well as in toroidal geometry. Code checks with respect to the theoretical dispersion relation of sound waves and Alfven waves are also given. Applications of the codes to shock waves, the Kelvin–Helmoltz instability and an obstacle in a flow are detailed in Section IV. Conclusions drawn from the results are given in Section V.

II. Fluid Model with Finite Size Particles

A system of magnetohydrodynamics is appropriately described by an equation of motion (momentum) and an equation of temperature (sometimes, an equation of state) coupled with Maxwell's equations without the displacement current. The present code exploits the following feature: equations of motion (momentum and temperature) can be expressed in a Lagrangian fashion, while the field equations (Maxwell's equations) can be expressed in an Eulerian way. The task of solving Lagrangian equations is most conveniently carried out by introducing a finite number of finite-sized particles carrying physical quantities which obey the Lagrangian equations.

First of all, this scheme readily removes many difficulties associated with conventional Eulerian codes, such as negative densities and advective numerical instabilities. Secondly, this scheme can overcome the need for a large number of particles to suppress large statistical fluctuations which are not associated with Eulerian fluid codes. The finite-size particles coupled with the fast Fourier transform method provide adequate and well-tested tools to eliminate the need for excessive particle numbers. Another feature of the Lagrangian equations in the code is the introduction of Krook type drag terms. This prohibits excessive multi-streaming and still insures momentum conservation. The equation of motion is organized in such a way that the total momentum of the particles (as well as the fluid momentum) is rigorously conserved before and after particle pushing. For the equation of temperature one can resort to similar measures and retain conservation of energy.

On the other hand, the field equations are most easily treated using an Eulerian scheme. So we assign field quantities to grid points. The equation to update the magnetic field is cast in such a finite difference form that the total magnetic flux is rigorously conserved before and after field pushing. One is then guaranteed magnetic charge is not generated during a certain period of time. For a bounded plasma, we introduce boundary conditions so as to preserve the above conservative code properties.

A. *Ideal MHD Scheme*

Our finite size particle method effectively solves the continuity equation by correctly simulating the dynamics of the particles which make up the fluid. An appropriate number of Gaussian-shaped finite size neutral particles are thus initialized in the calculation at position $\mathbf{r}_i(0)$ at time $t = 0$. Their later position is found at time t by integrating their orbit equations forward, namely

$$\frac{d\mathbf{r}_i(t)}{dt} = \mathbf{v}_i(t), \tag{1}$$

given that the velocity of each particle is known as a function of time through the equation of motion.

In the ideal MHD case, the equation of motion for each particle is written as follows

$$\frac{d\mathbf{v}_i(t)}{dt} = -\frac{1}{\rho}\nabla P - \frac{1}{8\pi\rho}\nabla B^2 + \frac{1}{4\pi\rho}\nabla \cdot (\mathbf{BB}), \tag{2}$$

where the magnetic force terms are cast into momentum conservative form given that the divergence of the magnetic field is zero.

The left hand side of Eq. 2 is the total derivative of the i-th particle velocity while the right hand side only involves macroscopic quantities i.e., quantities defined at the mesh points of the fixed background grid and the mass density ρ. This part of the calculation performed on the frame of the particles is Lagrangian.

While the magnetic force terms are evaluated by finite difference methods, the particle pressure gradient is calculated efficiently and accurately using a Fast Fourier Transform algorithm. This parallels the charge density and field calculation in the electrostatic finite size particle codes.

In the isothermal case ∇P becomes

$$\nabla P = T\nabla n. \tag{3}$$

This density is in turn expressed as

$$n(\mathbf{r}) = \sum_i f(\mathbf{r} - \mathbf{r}_i), \tag{4}$$

where the summation runs over all particles. The quantity $f(\mathbf{r} - \mathbf{r}_i)$ represents, in a sense, the spatial distribution of mass about the central point $\mathbf{r}_i$ of the particle. In the present paper we adopt a normalization convention for spatial dimensions such that the particle number and number density are mutually interchangeable, i.e., the grid spacing Δ is equal to unity. Without this assumption, the equations would be overly complex with such a factor as Δ^d for the different dimensions d (one, two or three). In this way, the equations are the same regardless of the dimensionality of the particular version of the code. The explicit form we use for f is a Gaussian of radius a, namely (in the most general three-dimensional case)

$$f(\mathbf{r} - \mathbf{r}_i) = \frac{1}{(2\pi a^2)^{3/2}} \exp[-(x - x_i)^2/2a^2 - (y - y_i)^2/2a^2 - (z - z_i)^2/2a^2] \tag{5}$$

However any other convenient form for f could be used. The particle positions $\mathbf{r}_i$ are determined in terms of the nearest grid point $\mathbf{r}_g$ and the density then has the form

$$n(\mathbf{r}) = \sum_g f(\mathbf{r} - \mathbf{r}_g)\, \rho_{\mathrm{NGP}}(g). \tag{6}$$

The summation is over all points $g = (l, m, n)$ on the three dimensional grid and ρ_{NGP} is expressed as the number of particles in cell g i.e.,

$$\rho_{\mathrm{NGP}}(g) = \sum_{i\in g} 1. \tag{7}$$

What we shall describe are nearest grid point codes; however, it should be possible to introduce corrections similar to the dipole expansion used in electrostatic models. The Fourier transform $(n\mathbf{k})$ can now be obtained by performing a single three-dimensional Fourier transform. Therefore,

$$n(\mathbf{k}) = (L_x L_y L_z)^{-1} \exp(-k^2a^2/2) \sum_g \exp(-i\mathbf{k} \cdot \mathbf{r}_g)\, \rho_{\mathrm{NGP}}(g), \tag{8}$$

or

$$n(\mathbf{k}) = \exp(-k^2a^2/2) \cdot FFT[\rho_{\mathrm{NGP}}(g)]. \tag{9}$$

The quantities L_x, L_y, L_z are the dimensions of the (x, y, z) grid while the term $\exp(-k^2a^2/2)$ is just the form factor of the finite size Gaussian shaped particles. The density gradient is written as

$$(\nabla n)_k = i\mathbf{k}n(\mathbf{k}). \tag{10}$$

The force in real space is subsequently obtained by an inverse Fourier transform of the above quantity.

In actuality, Eq. (2) should explicitly contain a particle drag term between particles in a single cell to prevent multistreaming; including such a term the equation for the velocity in finite difference form is written

$$v_i^{n+1/2} = v_i^{n-1/2} + \Delta t\, F_g{}^n/n_g{}^n - \nu(v_i^{n-1/2} - v_{\text{fluid}}^{(n)}), \tag{11}$$

where n stands for $t = n\,\Delta t$, $F_g{}^n$ represents the right hand side of Eq. (2) and ν is an adjustable viscous drag coefficient. The fluid velocity is defined as the average velocity of all particles in a given cell before updating.

$$v_{\text{fluid}}^{(n)} = \langle v\rangle^{(n)} = \sum_{i\in g} v_i^{n-1/2}\Big/ n_g{}^n. \tag{12}$$

The denominator $n_g{}^n$ represents the number of particles in cell g at time $t = n\,\Delta t$. It should be noted that $\sum_{i\in g} v_i$ in the continuum limit goes to nv. Eq. (12) guarantees that there appears no net force from the drag term in particular cell (see also Appendix). If $\nu = 0$, the velocity of each particle is leap-frogged; if $\nu = 1$, the fluid velocity of each cell $\langle v\rangle^{(n)}$ is leap-frogged since, when the velocity is updated, the particle positions and thus the fluid density are unchanged. Because of this leap-frogging we expect only slight numerical diffusion. This may not be the case for the GAP[10] code, since it is essentially a first order scheme.

Now let us prove that Eq. (11) is momentum conservative: the sum over all particles on the left hand side should vanish in a periodic system. After summation, one obtains

$$\begin{aligned}\sum_{\forall i} m(v_i^{n+1/2} - v_i^{n-1/2})/\Delta t &= -\sum_g \sum_{i\in g} \frac{1}{n_g{}^n}(\nabla P)^n - \sum_g \sum_{i\in g} \frac{(\nabla B^2)^n}{8\pi n_g{}^n}\\ &+ \sum_g \sum_{i\in g} \frac{(\nabla\cdot\mathbf{BB})^n}{4\pi n_g{}^n} - \nu m\Big(\sum_g n_g{}^n\langle v\rangle^{(n)} - \sum_g n_g{}^n\langle v\rangle^{(n)}\Big),\end{aligned} \tag{13}$$

where we used the definition of fluid velocity given by Eq. (12).

In the finite difference scheme, every grid quantity on the right hand side of Eq. (13) appears twice with a different sign. Thus for any mesh quantity ∇A, the summation of ∇A over all cells is zero for our uniform spatial mesh.

Therefore, given that, when summed over all particles, the drag term is identically zero, Eq. (13) yields

$$\sum_{\forall i} m(\mathbf{v}_i^{n+1/2} - \mathbf{v}_i^{n-1/2}) = 0. \tag{14}$$

In other words, total momentum is rigorously conserved.

The induced magnetic field is determined from a partial differential equation in an Eulerian frame, i.e.

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}). \tag{15}$$

In the above equation $\mathbf{v}$ stands for the fluid velocity previously defined from the particle velocities and the magnetic field B on the right hand side is the total field, i.e., the initial plus induced magnetic fields. Since v and B are grid quantities, the field pushing equation appears in its Eulerian form. It is readily integrated in time by the conservative Lax method [16], known to be stable as long as the Courant–Friedrichs–Lewy (CFL) condition on the time step is met. For a periodic system, our scheme is flux conservative. Indeed, finite differencing of the right hand side of Eq. 15 and summation over all grid points yields

$$\frac{\partial \Phi}{\partial t} \equiv \sum_g \frac{\partial \mathbf{B}_g}{\partial t} = \sum_g [\nabla \times (\mathbf{v} \times \mathbf{B})]_g = 0, \tag{16}$$

because of the ∇ operator in front of the $v \times B$ term. Here a grid cell has unit area.

In principle particles can carry any quantity θ besides their positions and velocities as long as the governing equation for that particular quantity, can be cast into the form

$$d\theta/dt = L, \tag{17}$$

where L only involves grid quantities. In particular the temperature is one such quantity and in our code we take temperature as an additional quantity carried by the particles. In subsection B and in the Appendix more discussion is given on the temperature equation.

B. *Resistive MHD Scheme*

When the MHD fluid is resistive i.e., non ideal, some transport coefficients such as resistivity and heat conductivity must be included in our model.

While the orbit equation and the equation of motion are unchanged, the field equation now becomes

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (v \times \mathbf{B}) - \frac{c^2}{4\pi} \nabla \times \boldsymbol{\eta} \cdot (\nabla \times \mathbf{B}), \tag{18}$$

where the resistivity $\boldsymbol{\eta}$ is in general a tensor. Note that the field pushing equation is again flux conservative for a periodic system.

Because of the conductivity and heat diffusivity the temperature equation, obtained from the second moment of the kinetic equation, can be written as follows

$$\frac{dT_i}{dt} = -\frac{1}{n}\left[(\mathsf{P} \cdot \nabla) \cdot \mathbf{v} - \mathbf{J} \cdot \left(\mathbf{E} + \frac{1}{c}\mathbf{v} \times \mathbf{B}\right) + \rho_e(\mathbf{E} \cdot \mathbf{v}) + \nabla \cdot \mathbf{W}\right]. \tag{19}$$

Here ρ_e is the charge density, effectively zero in the ideal MHD case, and the electric field is expressed through Ohm's law as

$$\mathbf{E} = -\frac{\mathbf{v} \times \mathbf{B}}{c} + \eta \cdot \mathbf{J}; \qquad \mathbf{J} = \frac{c}{4\pi} \nabla \times \mathbf{B}. \tag{20}$$

$\mathbf{W}$ is the heat flux due to collisions, namely,

$$\mathbf{W} = -\kappa \cdot \nabla T, \tag{21}$$

where κ is the heat conductivity tensor due to collisions. The first term in Eq. (19) is the work accompanied by the expansion or compression of the fluid. Equation (19) then yields

$$\frac{dT_i}{dt} = -\frac{1}{n}(\mathsf{P} \cdot \nabla) \cdot \mathbf{v} + \frac{c^2}{(4\pi)^2 n} [(\nabla \times \mathbf{B}) \cdot \eta \cdot (\nabla \times \mathbf{B})] + \frac{1}{n}(\nabla \cdot \kappa \cdot \nabla) T. \tag{22}$$

Again all quantities on the right hand side are grid quantities. Due to the Joule heating term, the second term in Eq. (22), temperature is not conserved.

The heating term in Eq. (22) is always positive definite but the others are not. Thus one possible difficulty could arise if we encountered negative temperatures in this case; however one way to overcome such a difficulty, if it occurred, would be to spread the heat over a few cells, (i.e. add additional heat diffusion), as is done for the pressure gradient force term in the shock wave application to the presented later. The energy conservation is discussed in the Appendix.

C. *Bounded System*

For hydrodynamic situations, a rigid but slippery wall is introduced in one direction. At the walls, we then enforce

$$\begin{aligned} \langle v \rangle_\perp &= 0, \\ P_\perp^{\text{out}} &= P_\perp^{\text{in}}, \end{aligned} \tag{23}$$

with particles reflected elastically. To calculate the pressure gradient force, fast Fourier transform techniques can still be used in the periodic directions while finite-differencing is employed in the bounded direction.

In the magnetohydrodynamic case a rigid, slippery and metallic wall is implemented in one direction, while the others remain periodic. In addition to the above two boundary conditions and specular reflection of the particles at the walls, we now have

$$\begin{aligned} B_\perp &= 0, \\ E_\parallel &= (v \times B)_\parallel = 0, \\ B_\parallel^{\text{out}} &= B_\parallel^{\text{in}}. \end{aligned} \tag{24}$$

The particle pressure calculation is the same as in the hydrodynamics case. The flux and momentum are still conserved.

III. Code Implementation and Testing

The scheme which couples the Lagrangian finite-sized particle pushing with the Eulerian field updating described in the previous section has been implemented in various dimensions (1, 2, $2\frac{1}{2}$ and 3), various dynamics (hydrodynamic, MHD, isothermal, full temperature dynamics etc.), various geometries and boundary conditions (cartesian, toroidal, with or without surfaces etc.). More than 20 of these varieties of working codes have been implemented on the UCLA-IBM360/91 and on the Livermore CDC7600. These codes have been extensively tested on numerical and physical grounds. Alternate codes, such as the FFT version (i.e., FFT for the pressure gradient calculation with finite differencing for the magnetic fields) and the conventional finite difference version are checked against each other. They fare quite well and give essentielly the same results. In the following we describe in detail scheme implementation and the simulation results of code checking.

A. 1-*D Code*

A one-dimensional (1-D) hydrodynamics code, with periodic boundary conditions was implemented on the Livermore *A* machine and on the IBM 360-91 at U.C.L.A. Typically 15 particles per cell are used on a 128 cell grid. The particle half-width is taken to be one grid spacing $\varDelta$. Particle quantities such as velocity and position are effectively leap frogged. Particles are given small (typically 0.01 to $0.1C_s$) random initial velocities. The dispersion ıelation for sound waves is then obtained from a correlation of density over 4000 time steps ($\varDelta t = 0.20C_{s\varDelta}^{-1}$). The results are in good agreement with the linear dispersion curve given by theory ($\omega = k_{\parallel}C_s$), as shown in Fig. 1a, at least for modes one through more than twelve. In fact, the slope of the simulational dispersion curve shoots slightly above that predicted by theory. This is due to the fact that apart from the "fluid temperature" the random motion of the macroparticles constitutes an additional source of temperature. The actual slope is thereby given by $\tilde{v}_T = (C_s^2 + \delta v^2)^{1/2}$, where δv is the average "thermal" velocity of the macroparticles.

It is interesting to compare how square or triangular shaped waves propagate under force free conditions in the present code and in a conventional Eulerian code. The latter shows a severe deformation of square or triangular shapes. Even the recent flux corrected transport Eulerian [12] code greatly alters triangular waves. On the other hand, the present code induces no deformation, as expected.

B. 2-*D Codes*

Two distinct versions of the scheme described in Section 2 were implemented on the IBM 360-91 at UCLA as well as the Livermore *A* machine, one hydrodynamic version and one ideal magnetohydrodynamic configuration.

The two-dimensional hydrodynamic (2-D hydro) code uses 16 particles per cell in a 64 × 16 periodic system in cartesian geometry. Effective particle size is again one grid spacing $\varDelta$. As in the 1-D code, the standard leapfrog algorithm is used to advance particles and velocities in time and thermal random excitation of the particles is implemented at $t = 0$. A correlation of the density over 2000 time steps ($\Delta t = 0.2C_s^{-1}\Delta$) yields a sound wave dispersion relation which agrees favorably with theoretical predictions (Fig. 1b). This is so for the first few modes until Brillouin zone effects [17] or the finite size particle effects set in at short wavelengths. The time to perform a particle push in the standard Fortran code is 35 μsec on the IBM 360/91 computer.

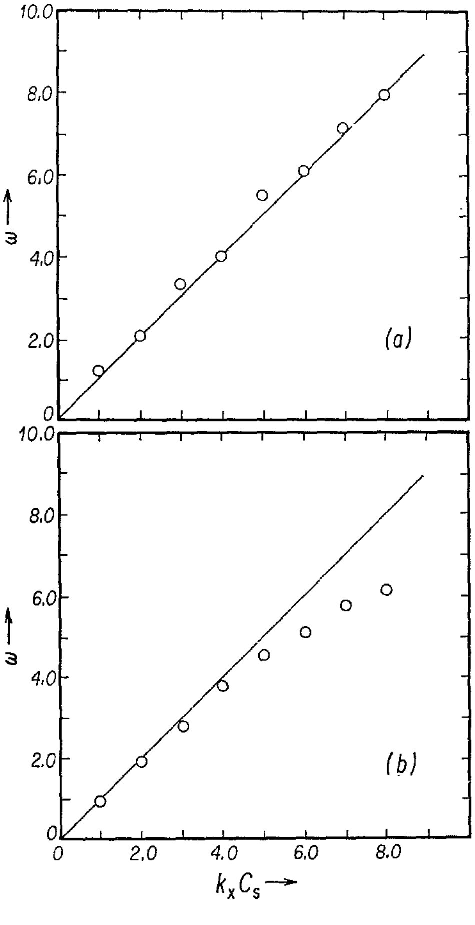

FIG. 1. Sound wave dispersion relation. a) 1-D code results. b) 2-D code results. Circles represent the simulation results, the full line theory.

The 2-D ideal MHD code employs 4 particles per cell in a 128 × 32 periodic system in cartesian geometry. Particle size is also one cell. The constant magnetic field is along the longer x direction (in principle it can have an arbitrary orientation in the x–y plane). The magnetic field strength used yields an Alfven speed of $5C_s$. Particles are initially given small random velocities. Correlations of the induced magnetic fields in the x and y directions over 2000 time steps ($\Delta t = 0.1 C_s^{-1}\Delta$; smaller time steps must be used in order to comply with the CFL condition for $V_A = 5C_s$). These yield dispersion relations for the Alfven waves $\delta B_y(k_x, 0)$ and magnetosonic waves

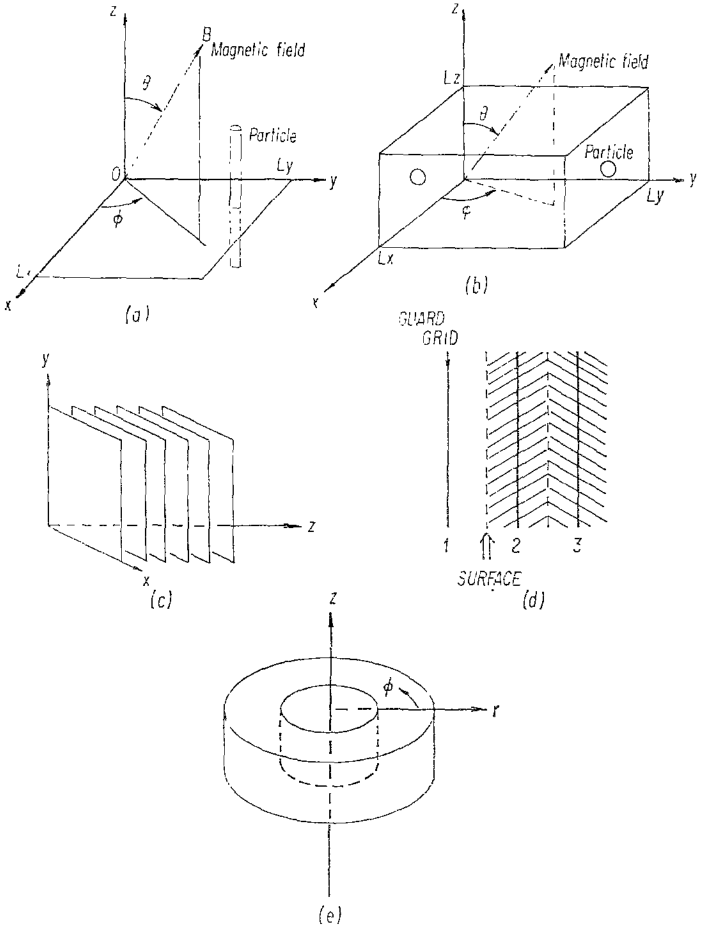

FIG. 2. Configurations of the models. a) 2-1/2 D MHD model. The magnetic field can point in any direction and the cylindrical rod particles can move in all directions. b) 3-D MHD model with spherical or ellipsoidal particles. c) The 3D surface code grid arrangement d) The 3D surface code guard grids, surface position and particles placing. e) The toroidal configuration is modeled as an annulus with metallic walls. The system is periodic in the z direction.

$\delta B_x(0, k_y)$ identical to those of Fig. 3 obtained with the $2\frac{1}{2}$ D version of the MHD scheme. Note that fewer particles are needed in the MHD case in order to obtain a sharp frequency spectrum for the eigenmodes, in low or moderate β plasmas. The divergence of the magnetic field remains zero within machine precision throughout the run both on the IBM and CDC computers, $\sim 10^{-22}$ for the CDC machine. Such good divergence values are important especially for low β plasma simulations invoving some density fluctuation activity, because this is related to magnetic activity:

$$\frac{\delta n}{n} \sim \frac{B_0\,\delta B}{B_0^2}\,\frac{1}{\beta}, \tag{25}$$

where δn is the error density fluctuation created by an error magnetic perturbation due to nonzero divergence of the magnetic field B.

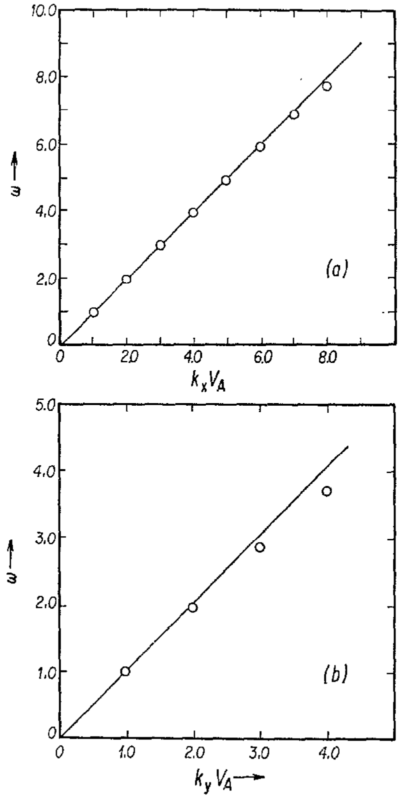

FIG. 3. Dispersion relation for Alfven (a) and magnetosonic (b) waves in the 2-1/2 D code. $V_A = 5C_s$ and the external magnetic field is in the x direction. Circles represent the simulation results, the full line theory.

C. $2\frac{1}{2}$ *MHD Code*

For the test runs of the $2\frac{1}{2}$-D MHD code, the particle number and size as well as the time step size were taken to be the same as above. The magnetic field and velocity now have a third component, although spatial variations are limited to the x–y plane. Particles are infinite cylindrical rods moving in any of three directions. The magnetic field can have an arbitrary orientation in the x–y–z coordinate system.

This is depicted in Fig. 2a. For the testing cases it was, however, taken along the longer x direction. The dispersion relation of Alfven waves ($\delta B_y(k_x, 0)$ and $\delta B_z(k_x, 0)$) as well as of magnetosonic waves ($\delta B_x(0, k_y)$), obtained from a 2000 time steps correlation of the various magnetic field components in the thermal runs are compared to theory [$\omega = k_x V_T$ and $\omega_{MS} = k_y(V_A^2 \times C_s^2)^{1/2}$]. As is shown in Figs. 2a and 2b excellent agreement is obtained for Alfven waves and magnetosonic waves until Brillouin zone effects or the finite size particle effects set in. The divergence of the magnetic field is again zero within machine precision. Time to perform a particle push in the standard Fortran code is 60 μsec on the IBM 360/91 computer. and magnetosonic modes with β less than one) and the "particle modes" (sound wave and its related modes), it is clear that the "mesh modes" have much sharper resolution than the "particle modes" when the degree of freedom (such as numbers of grids and particles) are the same in both cases. This was anticipated; although in the present codes the particles play a vital role in order to guarantee fluid density conservation and fluid momentum conservation, modes of magnetic waves in low to moderate β plasmas are transmitted primarily through the magnetic stress tensor which is a grid quantity as if the grids were an ether medium. On the other hand, modes of sound waves or sonic shock waves are transmitted through the medium which is represented on the mesh by an ensemble of particles.

D. 3-*D Codes*

Several distinct ideal and nonideal fully three-dimensional MHD codes are at present in operation. Here we describe scheme implementation and code testing results of the 3-D cartesian ideal MHD codes with periodic boundary conditions and with metallic boundary conditions. We also present a scheme for implementation for toroidal geometry.

A 3-D cartesian code with periodic boundary conditions is implemented as follows. A full three-dimensional cartesian mesh is used to accumulate the fluid quantities from the particle quantities. This code is tested by comparing simulation results on the IBM computer to the theoretical linear dispersion relation of Alfven and magnetosonic waves. Eight particles per unit cubic cell are used in a 32 × 8 × 8 system for Alfven waves or an 8 × 32 × 8 one for magnetosonic waves, the constant initial magnetic field being in the x direction in both cases. This insures a satisfactory number of modes in both directions in turn. A sketch of the model is shown in Fig. 2b. Two polarizations in the induced field are included, namely $\delta B_y(k_x, 0, 0)$ and $\delta B_z(k_x, 0, 0)$ for Alfven waves and $\delta B_x(0, k_y, 0)$ and $\delta B_z(0, k_y, 0)$ for magnetosonic waves; we have sacrificed the third branch of the magnetosonic waves for testing purposes. Dispersion curves obtained from the correlation over 1000 time steps ($\Delta t = 0.1 C_s^{-1} \Delta$) of the relevant magnetic field components are in excellent agreement with theoretical predictions [18] as is shown in Figs. 4a and 4b. The divergence of the magnetic field is zero within machine precision. Two distinct methods have been utilized to calculate the pressure gradient, the conventional finite difference method and the fast Fourier transform technique employed in the lower dimensions. In addition to the purpose of checking out the accuracy of two independent codes one against the other, this gave us experience with a pure finite difference code which method can easily be applied to systems of a bounded plasma or to non-cartesian geometries. Finite differencing has an advantage here in contrast to the ordinary charged particle codes in that we do not have the difficulty arising from excessive noise due to Coulombic collisions of particles, because the particles in the present codes are neutral. In fact, both versions of the codes show excellent agreement up to the fourth decimal point of any physical quantity for the MHD case. The test results shown are those of the FFT version. In the hydrodynamic case, however, the finite difference code is weakly unstable. It is subject to a lattice instability. This instability is readily removed by either implementing the temperature equation (Eq. (22)) or by introduction of a magnetic field. This is because the unrestrained relative lattice vibration of the grids is suppressed by introduction of binding forces between the cells, e.g., the last term in Eq. (22) for temperature.

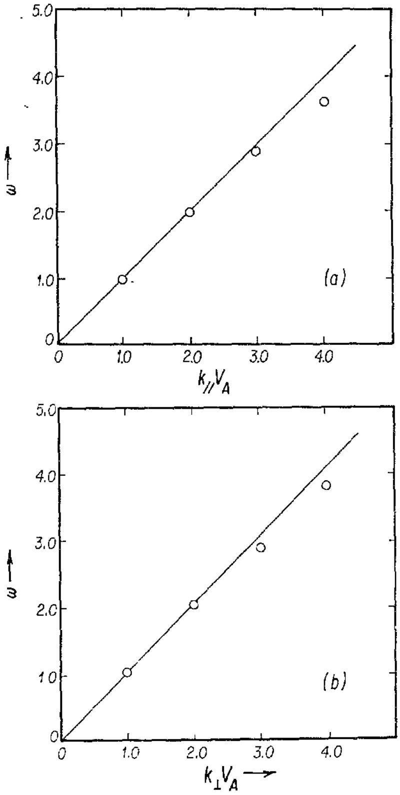

FIG. 4. Dispersion relation for Alfven (a) and magnetosonic (b) waves in the 3D periodic code. $V_A = 4C_s$ and the external magnetic field is in the x direction.

A cartesian 3-D code with rigid metallic walls is implemented in the following way.

The system is now bounded in the z direction and periodic in the other two. The configuration consists of a series of planes at each grid point in the bounded z-direction, as shown in Fig. 2c, on which the relevant quantities are treated as in the 2D codes, since periodicity is retained in the x and y direction. Finite differences between these planes are carried out in the z-direction to calculate the pressure force and the magnetic fields. The boundary conditions described in Section IIC are incorporated. Two guard grids are introduced and values of magnetic field and pressure are fixed either by the boundary conditions or determined by symmetry. Particles are loaded [illegible] quantities are determined from the known real system quantities as follows.

$$\begin{aligned} v_z(1) &= -v_z(2), \\ B_z(1) &= -B_z(2), \\ E_\parallel(1) &= (v \times B)_\parallel(1) = -E_\parallel(2) = -(v \times B)_\parallel(2), \end{aligned} \tag{26}$$

thereby insuring that $v_\perp$, $B_\perp$ and $E_\parallel$ are zero on the actual surface. Furthermore

$$B_\parallel(1) = B_\parallel(2). \tag{27}$$

Identical relationships hold at the other wall. In this way, all quantities appearing in the finite differences are fully defined from known real system quantities.

With such boundary conditions, this scheme is again flux conservative and the divergence of the total magnetic field is indeed zero within machine precision. With 8 particles per unit cell, in an $8 \times 8 \times 32$ system with an initial constant magnetic field along the z direction, correlations of δB_x and δB_y for fixed k_z are taken over 1000 time steps ($t = 0.1C_s^{-1}\Delta$) and compared with theoretical predictions for Alfven wave dispersion. Fig. 5 shows reasonable agreement between the two values.

A toroidal code with rigid metallic boundaries in the r direction is implemented in the following fashion. A sketch of our toroidal model is shown in Fig. 2e. We settle upon cylindrical geometry with a hole from $r = 0$ to $r = r_{\min}$. There are two metallic walls at $r = r_{\min}$ and $r = r_{\max}$. Periodic boundary conditions are used in the z and ϕ directions while the previously discussed metallic boundary conditions are imposed in the r direction. Fourier transforms serve to calculate the pressure force in the periodic direction while NGP weighting and finite differences are implemented in the r direction.

To have a uniform initial density, particles must be loaded in a nonlinear way. Let $r(i)$ be the ith particle radial coordinate, $r_{\min}$ and $r_{\max}$ respectively the inner and outer toroidal radii. The formula for $r(i)$ reads as follows

$$\begin{aligned} r(i) = \tfrac{1}{2}\{r(1) &+ r_{\text{wall}} - r_{\min} + [(r(1) + r_{\text{wall}} - r_{\min})^2 \\ &+ 4(r_{\max}\,\Delta r(I-1) + r(1)\cdot(r_{\min} - r_{\text{wall}}))]^{1/2}\}. \end{aligned} \tag{28}$$

Here $r(1)$ represents the position of the first particle, r_{wall} that of the first wall and Δr stands for the grid spacing.

In order to have force balance a toroidal field of the form $B_\phi \alpha\ 1/r$ is imposed. Because of the particular geometry of the model, extra forces must be taken into account in the equation of motion for the ϕ and r components of the velocity (the centrifugal force and half the Coriolis force)

$$dv_i^\phi/dt = [-\nabla p - \mathbf{B} \times (\nabla \times \mathbf{B})]^\phi - \langle v\rangle^r \langle v\rangle^\phi/r, \tag{29}$$

$$dv_i^r/dt = [-\nabla p - \mathbf{B} \times (\nabla \times \mathbf{B})]^r + [\langle v\rangle^\phi]^2/r. \tag{30}$$

Because of the geometrical factors r and $1/r$ associated with the differential operators ∇, ∇x and ∇ as well as the last terms in Eqs. (29) and (30), the equations of motion and field pushing equation are not anymore cast in the momentum and flux conservative form. For this reason, when tests of the code were performed on the IBM computer, the divergence of B was of the order of 10^{-5}. The scheme was however numerically stable.

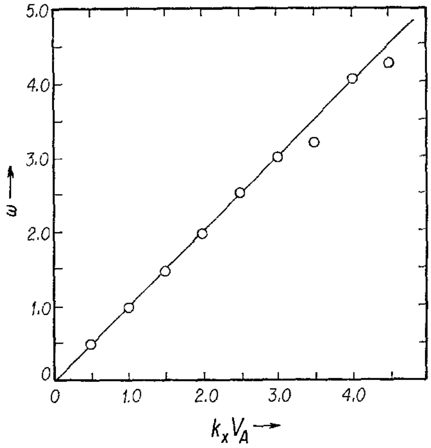

FIG. 5. Alfven wave dispersion relation in the 3D surface code. The metallic surfaces are located at $z = 0.5\Delta$ and $z = 32.5\Delta$ and the x and y directions are periodic. $V_A = 4C_s$ and the external magnetic field is in the z direction.

IV. APPLICATIONS

We shall demonstrate various applications of the present code to fluid systems which involve abrupt discontinuities, highly nonlinear evolution of an instability and sharp plasma profiles including a vacuum in hydrodynamics and magnetohydrodynamics. With slight accommodations the following problems are readily simulated in our code: shock wave propagation on the 1-D hydrodynamic code, the Kelvin–Helmholtz instability and its nonlinear evolution and a blunt projectile in a fluid on both the 2-D hydrodynamic and the 2½-D magnetohydrodynamic codes. All studies were performed on both the UCLA IBM 360-91 and the Livermore CDC 7600 computers.

A. *Shock waves*

Shock wave formation due to an abrupt density jump has been simulated using the present code in 1-D. The code is slightly modified to accommodate a sharp discontinuity, mainly to avoid excess short scaled density oscillations due to the finite number of particles and the finite grid. The simulation shows propagation of shock fronts and formation of rarefaction waves.

Fifteen particles per cell on the average are loaded on a 128 cell system so as to produce the square wave type density profile shown in Fig. 6a. As isothermal fluid model is used. The density of the dense part is 20, while that of the dilute part is 10. Particles are initially at rest. The pressure and fluid velocity are averaged over three grid points to insure a smooth flow across the discontinuity, apart from the smoothing due to the viscous drag coefficient. This produces a minimum shock thickness of 3 grid spacings. Figure 6 shows the time development of the density profile for this free expansion shock. The observed shock thickness is somewhat larger at 10Δ than the expected three grid spacings. This is probably due to the coarse NGP accumulation of the particle quantities at the grid points to obtain the fluid quantities. The fluid's evolution is characterized by the rapid formation of a density plateau between rare faction and shock front regions. The plateau is the density level reached when the two shocks collide before recurrence of the initial state. The gas attains its maximum velocity in this same plateau region as is clear from the fluid velocity frames of Fig. 6.

A schematic time evolution of the density profile is shown in Fig. 7, with the initial state in Fig. 7a, plateau formation at intermediate times in Fig. 7b and the state of the gas before collision at late times. With the aid of Fig. 7b, conservation of mass is written

$$\rho_p(u_s - v_f) = \rho_l u_s , \tag{31}$$

where ρ_p and ρ_l represent the plateau density and the density level of the dilute part of the gas while u_s and v_f stand for the shock front speed and the fluid velocity in the plateau region. Conservation of momentum reads

$$P_p - P_l = \rho_l u_s v_f . \tag{32}$$

Here P_p and P_l are the gas pressure in the plateau and dilute regions. For the isothermal gas under consideration $P = C_s^2 \rho$, with C_s being the sound speed. Normalizing all velocities with respect to C_s, the above two equations yield

$$v_f = u_s - 1/u_s , \tag{33}$$

$$\rho_p/\rho_l = u_s^2. \tag{34}$$

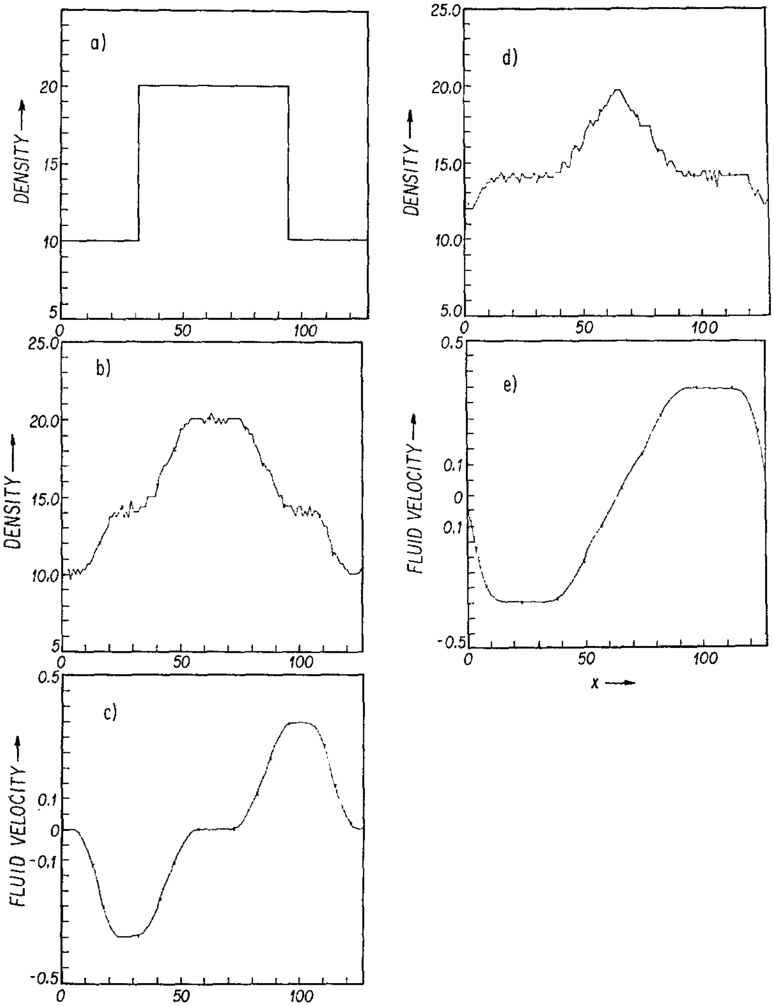

FIG. 6. Free expansion shock in the 1-D hydrodynamic code. Initially the density at $x = (32\Delta, 96\Delta)$ is twice the background density. a) The density profile at $t = 0$. b) The density profile at $t = 20C_s^{-1}\Delta$. c) The fluid velocity at $t = 20C_s^{-1}\Delta$. d) The density profile at $t = 30C_s^{-1}\Delta$. e) The fluid velocity at $t = 30C_s^{-1}\Delta$.

From Figs. 7a and 7c, where the various w's represent the length of a particular density region, conservation of energy is written

$$\begin{aligned} &w_h{}^0\rho_h \ln \rho_h + w_l{}^0\rho_l \ln \rho_l \\ &\quad = w_p\rho_p \ln \rho_p + w_l\rho_l \ln \rho_l + \tfrac{1}{2}w_p\rho_p v_f{}^2. \end{aligned} \tag{35}$$

In the above equation, all $w\rho \ln \rho$ stand for the potential energy of the various regions of the isothermal gas while the last term stands for the kinetic energy of the still

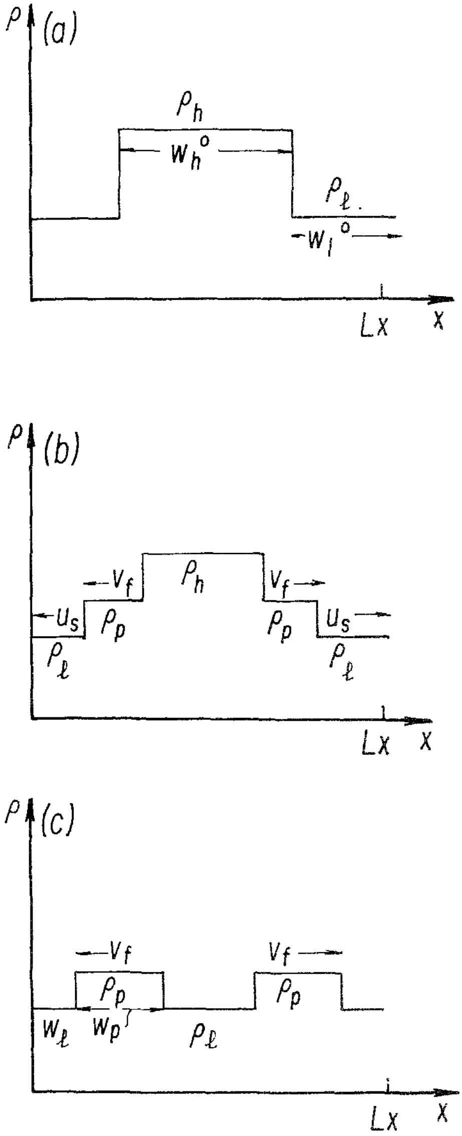

FIG. 7. Schematic evolution of the density profile for the free expansion shock. a) The initial state. b) The plateau formation at intermediate times. c) The late state.

expanding gas. The quantity ρ_h is the density level of the dense part of the gas. Using conservation of mass

$$w_h{}^0\rho_h + w_l{}^0\rho_l = w_p\rho_p + w_l\rho_l \,, \tag{36}$$

and Eqs. (33) and (34), the following equation is obtained from Eq. (35)

$$(u_s^2 - 1)\ln(\rho_h/\rho_l) - u_s^2\left(1 - \frac{\rho_l}{\rho_h}\right)\left[\ln u_s^2 + \frac{1}{2}\left(u_s^2 - 2 + \frac{1}{u_s}\right)\right] = 0. \quad (37)$$

This determines u_s, hence ρ_p and v_f, in terms of the only known quantity in the problem, the initial density ratio ρ_h/ρ_l.

Solution of Eq. 37 for the experimental ratio $\rho_h/\rho_l = 2$ yields $u_s^2 = \rho_p/\rho_1 = 1.41$ [illegible]

also observed for a density ratio of 3 to 1.

Small scale oscillations in front of the shock as well as behind it, are due to dispersion present in the model through the use of finite size particles and grids. They can be suppressed, to some extent, by increasing the number of particles.

B. *Kelvin–Helmholtz Instability*

The Kelvin–Helmholtz instability due to a flowing fluid in contact with a stationary fluid has been simulated using the 2-D and 2-½D codes for the hydrodynamics case and the MHD case, respectively. We observed formation of vortex structures as the instability develops in the hydrodynamic case. With a strong enough magnetic field the instability is stabilized as expected.

Both a subsonic flow case and a supersonic one are given for the hydrodynamic cases. Sixteen particles per cell are loaded uniformly throughout the 32 × 32 system. The middle layer of these paiticles is given a streaming velocity of $0.3C_s$ in the subsonic case and $1.5C_s$ in the supersonic one. A slight perturbation of the form $y = \epsilon \sin(k_x x)$, where ϵ is typically 0.01, is initially imposed on the streaming part of the fluid to start off the Kelvin–Helmholtz instability. Figure 8a depeicts the initial x-y space pattern for the streaming particles. Subsequent frames in Fig. 8 show the parallel development of this x-y-pace, at equivalent times, for subsonic flow on the left and supersonic flow on the right. In the former, small vortices formed at the contact surface between stationary and streaming particles slowly propagate inward, an indication of the gentle linear growth of the Kelvin–Helmholtz instability. In the latter, violent early distortion of the flow pattern rapidly evolves into two great vortices which subsequently rotate around each other. These vortices survive for a lont time and provide the system with a rather marked structure. The last frame in Fig. 8 is a collage of two such flow patterns to clearly demonstrate this. Analysis of the simulation results in the subsonic case, yields a growth rate for the Kelvin–Helmholtz instability in agreement with theory, i.e., $\gamma \sim \frac{1}{2}k_x V_{\text{flow}}$. More precisely for $V_{\text{flow}} = 0.3C_s$ and a mode 1 perturbation ($k_x = 0.196$), theorery predicts $\gamma = 2.95 \times 10^{-2}$ while simulation yields $\gamma \sim 0.03$.

It has been known that a constant magnetic field along the flow direction stabilizes the Kelvin–Helmholtz instability if $V_A > V_{\text{flow}}$, while it has no effect for stabilization if it is transverse to the flow direction [19]. To observe this we use the 2-½D MHD code described previously. The constant magnetic field is along the x direction and

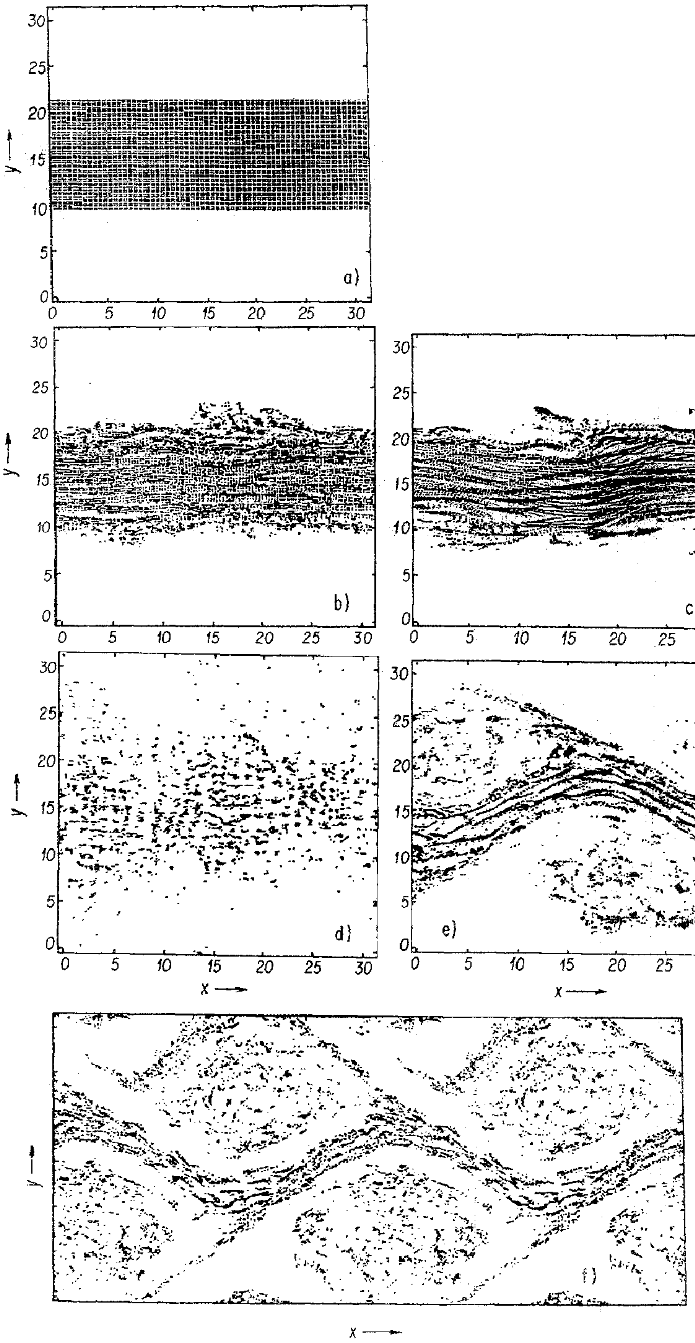

FIG. 8. Onset and nonlinear evolution of the Kelvin-Helmholtz instability. Instantaneous frames extracted from a movie show the streaming particles in $x - y$ space. Initially the streaming particles are situated at $y = (9.5\Delta, 21.5\Delta)$ with a velocity of $0.3C_s$ in the subsonic case and $1.5C_s$ in the supersonic one. a) The initial state. b) The intermediate stage ($t = 200\ C_s^{-1}\Delta$) for supersonic flow. c) The intermediate stage ($t = 40\ C_s^{-1}\Delta$) for supersonic flow. d) The late stage ($t = 600\ C_s^{-1}\Delta$) for subsonic flow. e) Vortex formation at $t = 120\ C_s^{-1}\Delta$ for supersonic flow. f) Late stage ($t = 200\ C_s^{-1}\Delta$) for supersonic flow. Two frames have been juxtaposed in the x direction.

the middle layer of the fluid is given a velocity V_{flow} of $1.5C_s$ in the x direction in addition to initial random velocities in the x and y direction. We expect a broad spectrum of modes to be unstable due to the Kelvin–Helmholtz instability. With relatively weak magnetic field ($V_A = C_s < V_{\text{flow}}$), the instability develops in spite of the existence of the magnetic field. For $V_A = V_{\text{flow}} = 1.5C_s$, the flow is still weakly unstable while for values of V_A in excess of V_{flow} the situation is stable.

C. *Obstacle in a Flow/Projectile in a Fluid*

Flow patterns and bow shock formation due to an "obstacle" in a flow have been simulated on the 2-½D magnetohydrodynamic code. The system is equivalent to a projectile in a stationary fluid by a Galilei transformation. The obstacle creates a laminar pattern or a turbulent drag pattern when it is placed in a subsonic flow with or without magnetic field. On the other hand, the obstacle can create a shock "cone" (bow chock) with its apex at the obstacle and a low density region in the shadow of the obstacle when the flow is superalfvenic or supersonic with or without the magnetic field.

We employ 64 × 64 particles in all on a 32 × 32 cell system with flow velocity in the x direction. The static magnetic field, when present, is oriented in the z direction. Our "obstacle" is represented by an attractive or repulsive force potential placed at the center of the system. In the present runs, the force is repulsive:

$$\mathbf{F} = (\mathbf{F}_0 \cdot (\mathbf{r} - \mathbf{r}_0)/b^2) \exp[-(r - r_0)^2/2b^2]. \tag{38}$$

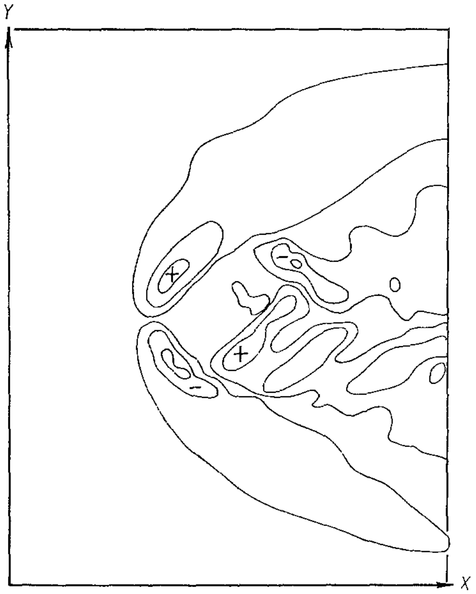

FIG. 9. Hydrodynamic supersonic flow around an obstacle. The Mach number is 2 and the obstacle is located at (16Δ, 16Δ). The contours of the fluid velocity in the y direction at $t = 30\ C_s^{-1}\Delta$ are shown.

Here F_0 is an adjustable normalization constant for the force, b represents the half-width of the force potential and $\mathbf{r}_0$ is the force center. The range of the obstacle force is effectively 1 grid length. In order to simulate an infinitely long flow (nonperiodic), a particle which leaves one side of the system in the x direction reenters from the other with a random position in y and initial velocities at $t = 0$ for v_x and v_y. This way the recirculating particles in the x direction have no memory of the perturbation when they reenter. In the y direction the system is periodic. However, magnetic perturbations are not so easily eliminated and to some extent they reenter the system.

Figure 9 shows a result of the simulation of a hydrodynamic supersonic flow with an obstacle. The Mach number with respect to the sound speed is 2. The shock cone is formed with its apex at the obstacle as is expected. A rather large shadow of vacuum is created behind the flow. The shock angle α is given [20] by $\alpha = \sin^{-1}(1/M)$, which is 30° in the simulation.

Figure 10 exhibits the case of a magnetohydrodymanic flow with an obstacle. The Mach number with respect to V_A is 2.5. The shock structure in the induced magnetic field is conspicuous. The flow velocity patterns also suggest the shock formation.

When a vacuum is created, we have to take into account the fact that the magnetic field in vacuum is not infinitely rigid either. By definition there is no current in vacuum. Therefore the algorithm (Eq. (15)) implies that the magnetic field in vacuum does not change. But in reality the MHD fluid can bend the magnetic field in vacuum adjacent to the fluid. For cases where we expect to or actually have vacuum, we therefore

$$\nabla^2 \mathbf{A} = -\frac{4\pi}{c}\mathbf{J}, \tag{39}$$

where $\mathbf{A}$ is a vector potential of the magnetic field $\mathbf{B}$. We adopted the Coulomb gauge. Equation (39) is not redundant when the system involves a vacuum. It plays a role to extrapolate the magnetic field from the plasma region into the current free region. The vector potential and thereby the magnetic field instantaneously adjusts to the plasma current. The current is calculated by $\mathbf{J} = (c/4\pi)\, \nabla \times \mathbf{B}$ in the region where there is plasma. In a periodic system the homogeneous solution to Eq. (32) can be set equal to zero. We thus calculate the special solution to Eq. (32) through the FFT, given that $\mathbf{J}$ is expressed as $\nabla \times \mathbf{B}$ in Fourier space.

In practice calculation of the vector potential $\mathbf{A}$ is also useful to plot the magnetic field lines. In a two dimensional system the equation for the field line is given by

$$\frac{dx}{B_x} = \frac{dy}{B_y}. \tag{40}$$

Equation (40) is recast in terms of the vector potential into $dA_z = 0$. Therefore plotting of equi-contour lines of A_z yields the field lines plot of the magnetic field. Two field lines plots are shown in Fig. 11, one for a supersonic superalfvenic flow

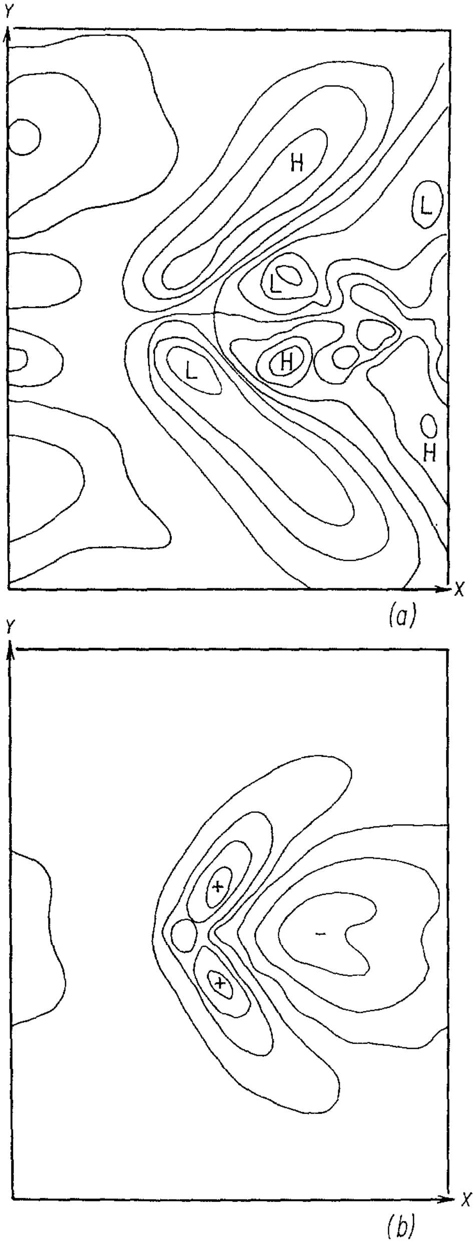

FIG. 10. Magnetohydrodynamic supersonic flow around an obstacle. The Mach number is 2 and the Alfven number A (V_{flow}/V_A) is 2.5. Again the obstacle is situated at the center of the frames. a) The induced magnetic field contour at $t = 16\ C_s^{-1}\Delta$. b) The induced magnetic field contours in the z direction with the external magnetic field also in the z direction at $t = 16\ C_s^{-1}\Delta$.

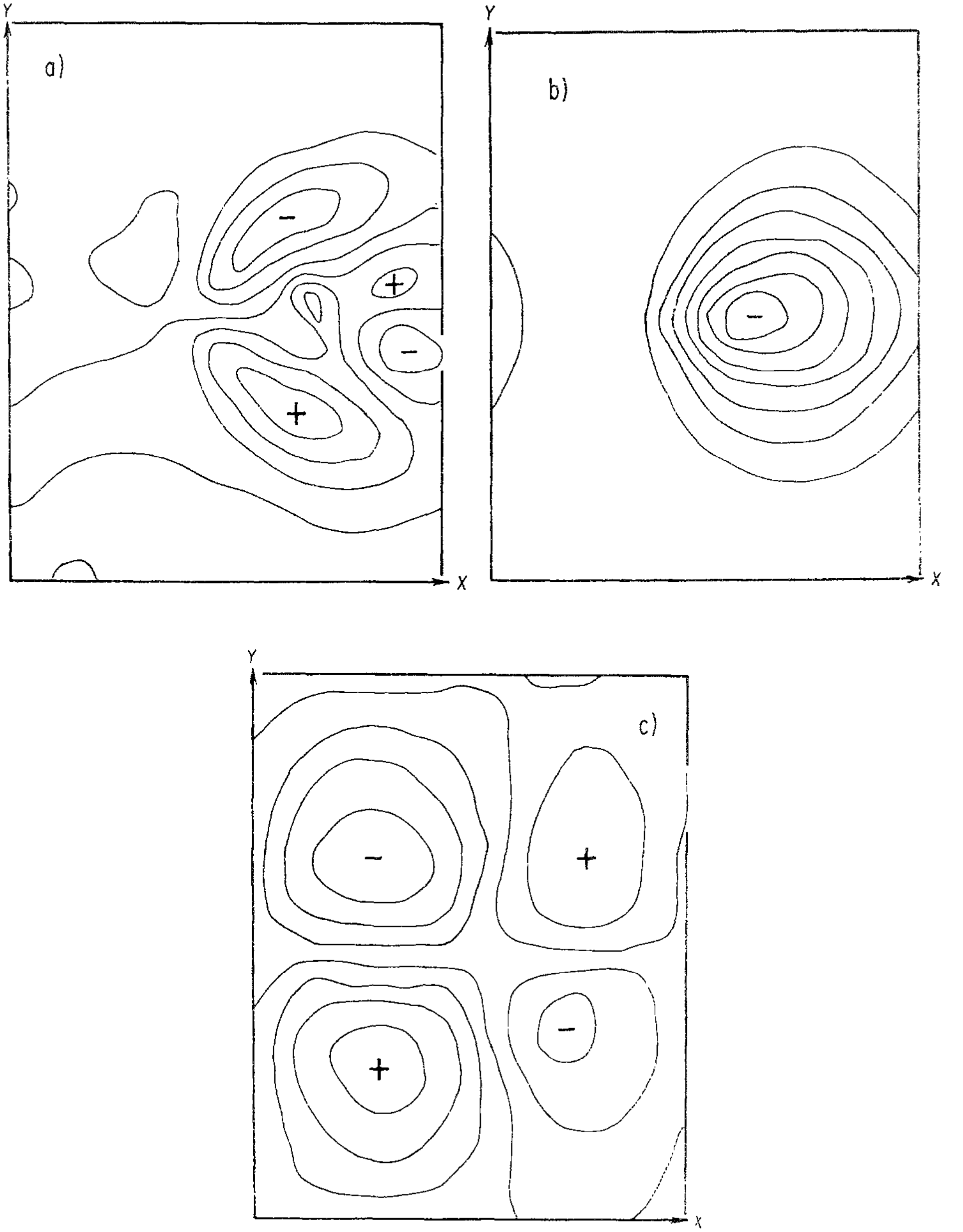

FIG. 11. Magnetic field line patterns created by an obstacle in a flow. a) The flow is supersonic ($M = 2$ and $A = 2.5$) with the external magnetic field in the x direction at $t = 8\ C_s^{-1}\Delta$. b) The flow is the same as a) with the external magnetic field in the y direction at $t = 8\ C_s^{-1}\Delta$. c) The flow is subsonic ($M = 0.9$ and $A = 0.6$) with the external field in the x direction at $t = 32\ C_s^{-1}\Delta$.

(Alfven Mach number of 2.5) and the other for a subsonic, subalfvenic flow (Alfven Mach number of 0.6). The former exhibits a solenoidal pattern in the induced magnetic field, while the latter gives a quadrupole structure for the magnetic field.

V. Conclusions

We have developed a type of Lagrangian MHD simulation codes utilizing the finite size particle method. The codes are relatively simple and straightforward in algorithm, numerically very stable and versatile. One can run on this simulation code hydrodynamic or magnetohydrodynamic problems with or whithout the temperature equation, with or without boundaries and in various dimensions. Since the code is still stable in cases of a plasma with very sharp variation of density and/or magnetic field and even for the formation of a local vacuum, it is expecially useful for the study of a relatively strong nonlinear regime of MHD instabilities. It is also readily applicable to shock waves and associated phenomena such as flow of the solar wind around the earth or other bodies, implosion/explosion in laser fusion and aerospace problems as well as ample applications to turbulent plasmas involved in CTR problems. With slight modifications, one can also simulate on the code meteorological problems or oceanographic problems. In fact, the equation of motion for a fluid in a rotating frame of reference is rather similar to that of MHD:

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{nm}\nabla P - \nabla[\tfrac{1}{2}(\mathbf{\Omega} \times \mathbf{x})^2 + \Phi] + 2\mathbf{v} \times \mathbf{\Omega}, \tag{41}$$

where the second term represents the centrifugal force and the potential force and the last term the Coriolis force. The Coriolis term resembles the Lorentz force. Any other fluids with other force laws could also be handled by implementing the appropriate forces such as, for example, the Yukawa force. Coding for a multi-species MHD fluid is also straight-forward in our scheme. Thus one should be able to exploit the code for a wide range of problems in plasma physics or fluid dynamics, as well as related areas. Moreover, one should be able to mix Vlasov particle models for some hot collisionless components while retaining fluid equations for other components (for example electrons) of the system. The similarity in treatment between the particle and fluid models should make hybrid models relatively easy to construct.

APPENDIX. Time-Centered Scheme and Energy Conservation

We discuss the time-centering scheme and the energy conservation of the code. In the text we described the leap-frog scheme for the particle positions and velocities coupled with the conservative Lax method for the magnetic field. In this appendix we introduce a fully time-centered scheme which is especially useful for the case with temperature equation. Although the scheme in the text was, in principle a first order scheme in the magnetic field pushing, the accuracy was quite satisfactory as was discussed earlier. For instance, the value of divergence B and the sharp spectrum of the magnetic waves have demonstrated this. The scheme to be discussed is bulkier than the one adopted in the text. But this scheme has an attractive property, energy conservation, in addition to the second order accuracy guaranteed by full time-centering.

The system of MHD equations with the temperature equation can be schematically written as follows,

$$\frac{dv_i}{dt} = F_v(T, B, n), \tag{A-1}$$

$$\frac{dx_i}{dt} = v_i ,$$

$$\frac{dT_i}{dt} = F_T(T, B, n, v), \tag{A-3}$$

$$\dot{B} = F_B(B, v), \tag{A-4}$$

where the quantities appearing in the arguments of the RHS of all equations, except Eq. (A-2), are fluid quantities. Inspection of Eqs. (A-1) to (A-4) quickly tells us that there is no intrinsic time-centering scheme, because a quantity evaluated on the left hand side should also appear on the right hand side staggered by a half-time step. As a result, a few quantities are entangled. However, if we store old values one full step back, we can extrapolate new values at the desired time step. A desirable full time-centering can be written as

$$v_i^{n+1/2} - v_i^{n-1/2} = F_v(T^{n*}, B^{n*}, n^n), \tag{A-5}$$

$$x_i^{n+1} - x_i^{n} = v_i^{n+1/2}, \tag{A-6}$$

$$T_i^{n+1/2} - T_i^{n-1/2} = F_T(T^{n*}, B^{n*}, n^n, v^{n*}). \tag{A-7}$$

$$B^{n+1/2} - B^{n-1/2} = F_B(B^{n*}, v^{n*}), \tag{A-8}$$

where integers or half integers as superscripts represent the time step and the asterisk stands for extrapolated values. For example, the extrapolation may be given as

$$n^{n*} = \tfrac{3}{2} n^{n-1/2} - \tfrac{1}{2} n^{n-3/2}. \tag{A-9}$$

We will show that with this assignment of time-steps for the quantities the total energy of this sytem is exactly conserved before and after pushing. In addition the code still preserves the conservative properties of total momentum and the flux, as was discussed in the text.

Let us calculate the temporal evolution of the magnetic field energy, the internal energy and the fluid kinetic energy. In order to avoid excess superscripts we write the following equation in a differential form instead of in a difference form. Conversion from one to the other is straightforward. We also normalize quantities so that c

and 4π do not appear. First, the evolution of the magnetic field energy is calculated from Ampere's law and Faraday's law:

$$\begin{aligned} B\dot{B} + \mathbf{J}\cdot\mathbf{E} &= \mathbf{E}\cdot\nabla\times\mathbf{B} - \mathbf{B}\cdot\nabla\times\mathbf{E} \\ &= \mathrm{div}(\mathbf{B}\times\mathbf{E}). \end{aligned} \tag{A-10}$$

Utilizing Ohm's law $\mathbf{E} = \eta\mathbf{J} - \mathbf{v}\times\mathbf{B}$, Eq. (*A*-10) is cast into

$$\tfrac{1}{2}B^2 = -\eta J^2 + \mathbf{J}\cdot\mathbf{v}\times\mathbf{B} - \eta\,\mathrm{div}(\mathbf{J}\times\mathbf{B}), \tag{A-11}$$

where the first term on the RHS is the energy loss due to Joule heating and the second term is the work done by the Lorentz force. Summation over all the grids will give

$$\sum_g \tfrac{1}{2}\dot{B}^2 = -\sum_g \eta J^2 + \sum_g \mathbf{J}\cdot\mathbf{v}\times\mathbf{B} - \eta\sum_g \mathrm{div}(\mathbf{J}\times\mathbf{B}). \tag{A-11$'$}$$

The internal energy can be calculated from Eq. (22):

$$\begin{aligned} \sum_i \frac{dT_i}{dt} = &-\sum_g\sum_{i\in g} T\nabla\cdot\mathbf{v} + \sum_g\sum_{i\in g}\frac{\eta}{n}J^2 \\ &+ \sum_g\sum_{i\in g}\frac{1}{n}\nabla\cdot\varkappa\cdot\nabla T - \nu_T\sum_g\sum_{i\in g}(T_i - \langle T\rangle), \end{aligned} \tag{A-12}$$

where $\langle T\rangle = \sum_{i\in g} T/\sum_{i\in g} 1$.

Equation (A-12) becomes

$$\sum_i \frac{dT_i}{dt} = -\sum_g nT\nabla\cdot\mathbf{v} + \sum_g \eta J^2. \tag{A-13}$$

The kinetic energy of fluid motion can be calculated from Eq. (2). Multiplying by the fluid velocity v and summing over all the particles, we obtain

$$\sum_i mv\frac{d}{dt}v_i = -\sum_g \mathbf{v}\cdot\nabla nT + \sum_g \mathbf{v}\cdot\mathbf{J}\times\mathbf{B} - \nu\sum_g\sum_{i\in g}(v_i - \langle v\rangle), \tag{A-14}$$

where $\langle v\rangle = \sum_{i\in g} v_i/\sum_{i\in g} 1$. Note that $\langle T\rangle$ and $\langle v\rangle$ are different from T and v. We will come back to this point later on in more detail. The last term in Eq. (A-14) is zero. The total energy evolution of the system is given by adding Eqs. (A-11$'$), (A-13) and (A-14). Note that all the quantities on the RHS's of Eqs. (A-11$'$), (A-13) and (A-14) can be expressed in terms of quantities evaluated at time step n, because of the assignments of time steps in Eqs. (A-5) to (A-8). The last term in Eq. (A-11$'$) is zero, because of the finite difference scheme coupled with periodic boundary conditions (or metallic boundary conditions). Wiiting the superscripts explictely, the summation of Eqs. (A-11'), (A-13) and (A-14) yields

$$\begin{aligned}\sum_g \tfrac{1}{2}[(B^{n+1/2})^2 - (B^{n-1/2})^2] + \sum_g (T_i^{n+1/2} - T_i^{n-1/2}) + \sum_g m\mathbf{v}^{n*}n^n \cdot (\mathbf{v}_n{}^{+1/2} - \mathbf{v}^{n-1/2}) \\ = -\sum_g n^n T^{n*}\nabla \cdot \mathbf{v}^{n*} - \sum_g \mathbf{v}^{n*}\cdot\nabla(n^nT^{n*}) + \sum_g \mathbf{J}^{n*}\cdot\mathbf{v}^{n*}\times B^{n*} + \sum_g \mathbf{v}_n{}^{*}\cdot\mathbf{J}^{n*}\times B^{n*} \\ = -\sum_g \nabla\cdot(nT\mathbf{v})^{n*} = 0. \qquad \text{(A-15)}\end{aligned}$$

To derive Eq. (A-15), we used the vector identity $\mathbf{A}\cdot\mathbf{B}\times\mathbf{A}$ and the "surface" summation as mentioned for Eq. (A-11'). Given that $v^{n*} = \frac{1}{2}(v^{n+1/2} + v^{n-1/2})$, Eq. (A-15) is finally cast into

$$\begin{aligned}\sum_g \tfrac{1}{2}(B^{n+1/2})^2 + \sum_i T_i^{n+1/2} + \sum_g mn^n(v^{n+1/2})^2 \\ = \sum_g \tfrac{1}{2}(B^{n-1/2})^2 + \sum_i T_i^{n-1/2} + \sum_g mn^n(v^{n-1/2})^2. \qquad \text{(A-16)}\end{aligned}$$

The first, the second and the third terms on the left hand side are interpreted respectively as the field energy, the internal energy and the kinetic energy after pyshing at time step n. The corresponding ones on the right hand side are interpreted as the energies before pushing at $t = n\,\Delta t$. Thus the total energy before and after the time step n is conserved.

Some comments are in order for the time step used to calculate the drag terms as in Eqs. (A-12) and (A-14). The third quantities T and v are given by extrapolation as in Eq. (A-9).

They are therefore evaluated exactly at time step n. On the other hand, $\langle v\rangle$ and $\langle T\rangle$ are given as

$$\langle v\rangle = \sum_{i\in g} v_i^{n-1/2}\Big/n^n, \qquad \text{(A-17)}$$

$$\langle T\rangle = \sum_{i\in g} T_i^{n-1/2}\Big/n^n. \qquad \text{(A-18)}$$

Prima facie the definitions in Eqs. (A-17) and (A-18) seem to involve a strange mixture of time steps. However, this is only the scheme which makes the summation of the drag terms over a cell exactly zero:

$$\sum_{i\in g} (v_i - \langle v\rangle) = 0, \qquad \text{(A-19)}$$

$$\sum_{i\in g} (T_i - \langle T\rangle) = 0. \qquad \text{(A-20)}$$

In fact, if one takes a "time-centered" assignement such as

$$\langle\tilde{v}\rangle = 2\sum_{i\in g} v^{n-1/2}\Big/(n^{n-1} + n^n),$$

the system is numerically unstable.

Acknowledgments

The authors wish to acknowledge stimulating discussions with P. Pritchett, C. C. Lin, R. Huff, T. Kamimura, and A. T. Lin. The authors would also like to thank C. C. Wu for making available his theoretical calculations on isothermal, free expansion shocks. This work was supported in part by NSF Contract PHY74-15233 and USDOE EY-76-C-03-0010 PA26/Task III. One of us (JNL) is a National Research Council of Canada Post Doctoral Fellow.

References

1. J. M. Dawson, *in* "Methods in Computational Physics" (B. Adler *et al.*, Eds.), Vol. 9, p. 1, Academic Press, New York, 1970.
2. C. K. Birdsall, A. Bruce Langdon, and H. Okuda, *in* "Methods in Computational Physics" (B. Adler *et al.*, Eds.), Vol. 9, p. 241, Academic Press, New York, 1970.
3. R. W. Hockney, *Phys. Fluids* **9** (1966), 1826.
4. O. Buneman, *Phys. Rev.* **115** (1959), 503.
5. C. W. Nielson and H. Ralph Lewis, *in* "Methods in Computational Physics" (John Killeen, Ed.), Vol. 16, p. 367, Academic Press, New York, 1976. J. Busnardo-Neto, P. L. Pritchett, A. T. Lin and J. M. Dawson, *J. Comput. Phys.* **23** (1977), 300.
6. C. W. Nielson, D. Winske, and D. W. Hewett, *in* "Proceedings, Annual Meeting on Theoretical Aspects of CTR, Madison, Wisconsin," April 1976.
7. H. Okuda *et al.*, *Bull. Amer. Phys. Soc., Ser. II* **21** (1976), 1037. J. Byers *et al.*, LLL Preprint, UCRL-79437, 1977.
8. W. L. Kruer, *Nucl. Technol.* **27** (1975), 216.
9. C. K. Chu, *Proc. Symp. Appl. Math.* **18** (1965), 1.
10. K. V. Roberts and N. O. Weiss, *Math. Comput.* **20** (1966), 272.
11. K. V. Roberts and D. E. Potter, *in* "Methods in Computational Physics" (B. Adler *et al.*, Eds.), Vol. 9, p. 339, Academic Press, New York, 1976.
12. J. P. Boris and D. L. Book, *in* "Methods in Computational Physics" (John Killeen, Ed.), Vol. 16, p. 85, Academic Press, New York, 1976.
13. B. M. Marder, *Math. Comput.* **29** (1976), 434.
14. R. L. Morse *et al.*, *in* "7th. Amer. Conf. on Plasma Simulation," papers D113–116, 1975.
15. F. H. Harlow, *in* "Methods in Computational Physics" (B. Adler *et al.*, Eds.), Vol. 3, p. 319, Academic Press, New York, 1964.
16. R. D. Richtmyer and K. W. Morton, "Different Methods for Initial Value Problems," Wiley–Interscience, New York, 1967.
17. C. Kittel, "Introduction to Solid State Physics," p. 262, Wiley, New York, 1966.
18. L. D. Landau and E. M. Lifshitz, "Electrodynamics of Continuous Media," p. 213, Pergamon, Long Island City, N.Y., 1960.
19. S. Chandrasekhar, "Hydrodynamic and Hydromagnetic Stability," Oxford Univ. Press, London/New York, 1961.
20. L. D. Landau and E. M. Lifshitz, "Fluid Mechanics," p. 310, Pergamon, Long Island City, N.Y., 1959.