JOURNAL OF COMPUTATIONAL PHYSICS 115, 213–227 (1994)

Dynamic and Selective Control of the Number of Particles in Kinetic Plasma Simulations

GIOVANNI LAPENTA*

Dipartimento di Energetica, Politecnico di Torino, C.so Duca degli Abruzzi 24, I-10129 Torino, Italy

AND

JEREMIAH U. BRACKBILL

Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received September 7, 1993

An algorithm for the dynamic control of the number of particles in a particle-in-cell (PIC), plasma simulation is presented. The algorithm selectively splits and coalesces particles to control the number of particles in each grid cell. It is designed for multiple-length scale problems where an adaptive grid can be applied and for PIC simulations on parallel computers where a constant number of particles per cell is useful for computational efficiency. The algorithm preserves the charge assignment at grid points while splitting one particle into two, or while coalescing two particles into one. Errors in momentum or energy conservation are controlled by a simple, *a priori* test. The accuracy of the algorithm is demonstrated in several simulations in one dimension, including a collisionless slow-shock, where an adaptive grid calculation with dynamic number control gives comparable accuracy to a uniformly zoned calculation without dynamic control with just 50% of the effort.

1. INTRODUCTION

Plasma simulation is an important and widely used tool in understanding complex, nonlinear phenomena such as collisionless shocks [1] and laser plasma interaction [2]. Nevertheless, there are constraints on the dynamic range of variables that can be modeled that severely limit its usefulness. These constraints become more apparent as one extends simulations to inhomogeneous plasmas or to multiple-length scale problems. When there are sheaths or collisionless shocks, there is a need either to increase the number of mesh points everywhere or to use an adaptive mesh that clusters zones in regions of high gradients. When the zone sizes vary, there will result large variations in the number of particles per zone. This variation may result in too few particles to resolve the velocity distribution in the smallest zones. As difficult problems are addressed with greater variations in length scales or in plasma density than can be accommodated by standard simulation methods, one needs dynamic control of the number and weight of particles. (Parker *et al.* [3] addressed a different aspect of multi-scale problem, namely the use of different time steps in different regions of the mesh.)

Dynamic particle control refers to changing the number of particles representing the plasma during the course of a simulation. Usually, the number of particles in a simulation remains constant from beginning to end unless there is injection or absorption at the boundaries. With dynamic control, the particles may be split or coalesced when more or fewer particles are required.

Dynamic particle control becomes even more desirable with the availability of parallel computers. With the prospect of greater computational power, researchers have begun to simulate tokamak experiments in realistic geometry [4] and to model the interaction of the solar wind with planetary magnetospheres in two and three dimensions [5]. When calculations on this scale are done, one must use computer resources efficiently. That is, the load must be shared equally among the processors so that there can never occur an interval when one processor is busy while the rest are idle. To avoid this, a method for domain decomposition has been proposed [6]. We propose to go a step further toward dynamic control by providing the ability to change the number of particles in each cell, so that, for example, every cell has the same number of particles.

Several schemes for dynamic control have been proposed. In Monte Carlo studies of the linear transport of neutral particles in scattering media, e.g., neutron transport in fission reactors [7], particle splitting is useful in maintaining a constant number flux of particles where there is significant attenuation of the physical flux. In one-dimensional plasma simulations, Denavit [8] describes a method for the periodic reconstruction of the distribution function. The reconstruction involves the simulta-

* Present Address: Los Alamos National Laboratory, Los Alamos, NM 87545.

0021-9991/94 $6.00

neous replacement of all particles. The desired effect is the reduction of noise and the suppression of instabilities. For incompressible fluid flow in two dimensions, Beale and Majda [9] propose a similar reconstruction for the vortex–blob method. They seek to reduce roughness and maintain the order of accuracy of the solutions over long times. For the simulation of collisionless shocks, Quest [10] selectively splits particles to increase the ability to model minority species. By splitting, he increases the number of particles representing that component of the plasma that has been accelerated to high energies by a collisionless shock. Without splitting, there are too few particles to represent accurately the accelerated particles. With splitting, there are enough for good statistics.

The approach we present differs from Quest's in two ways. The criterion for changing the number of particles in our method is a comparison with a target number of particles per cell rather than the number of particles in a portion of the velocity distribution. Thus, some particles may be split to increase the number of particles in a cell, while other particles may be coalesced to reduce the number of particles. In addition, our method creates new particles with different positions, rather than different velocities.

The development of the algorithm proceeds through a discussion of the general framework for particle splitting, a discussion of the special properties of the b-spline, a description of the method for controlling the number of particles, and concludes with three numerical tests of the method. The first compares the original distribution with the distribution that results from coalescing particles until the number of particles is halved; the second illustrates the ability to maintain a constant number of particles when mesh points are clustered to resolve an electrostatic sheath; and the third applies the method to the calculation of a slow shock.

2. GENERAL FRAMEWORK OF THE DYNAMICAL CONTROL OF THE NUMBER OF PARTICLES

In a typical particle simulation code, either fluid or kinetic, a discretization of the spatial variable is introduced. For the sake of simplicity we consider the one-dimensional case (but 3D in the velocity space). The extension to 2D and 3D problems is straightforward. In one-dimensional simulations the region occupied by the system is the interval of the spatial axis x bounded on left by x_l and on right by x_r. It is divided into M cells $[x_i, x_{i+1}]$; $i = 1, \ldots, M$, where $x_1 = x_l$, $x_{M+1} = x_r$.

The cell width $\Delta_i = x_{i+1} - x_i$ is a function of the index i and of time when adaptive grids are used. However, one can define a natural coordinate ξ by mapping each cell in physical space onto a unit interval in the space of the natural coordinates, which are independent of time. The cell vertices x_i are mapped into $\xi_i = i$ and any point $x \in [x_i, x_{i+1}]$ is mapped into a value of $\xi \in [i, i + 1]$.

The particle motion in phase space is calculated from Newton's equations. The forces are calculated by solving Maxwell's equations for the electromagnetic fields on the spatial grid.

The solution of these equations requires a knowledge of the charge, current, density, and pressure of every species present in the system on the grid. The calculation of these moments from the particle distribution requires interpolation to transfer the information from the particles to the grid [11] for which there are many different techniques. We will focus our attention on a very general class of interpolation techniques based on the b-spline [12].

In this approach we can, for example, express the moments $\mathbf{M}$ (where boldface stands for the possible tensor nature of the moment, as is the case for the pressure) of the particle distribution of species s at any point x knowing the position ξ_p, the speed u_p, and the charge q_p of the set of N particles used to sample the distribution, as

$$\mathbf{M}(\xi) = \sum_{p=1}^{N} q_p \mathbf{T}(\mathbf{u}_p) s_l(|\xi - \xi_p|), \tag{1}$$

where $\mathbf{T}$ is the generating function for the various moments and $s_l(\xi)$ is the b-spline function of order l. Different values of l give different interpolation schemes characterized by different widths of the particles. For example, for $l = 0$ (nearest-grid-point approximation, or NGP) a given particle contributes only to the cell center in which it is located, while for $l = 1$ it contributes to two adjacent cells and for $l = 2$, to three cells [12]. In general, a particle located in the ith cell affects a restricted subset U_i of neighbour cells. The subset for $l = 2$ is $U_i = \{i - 1, i, i + 1\}$, and for $l = 1$: $U_i = \{i - 1, i\}$ if $i \leq \xi_p < i + \frac{1}{2}$; $U_i = \{i, i + 1\}$ if $i + \frac{1}{2} < \xi_p \leq i + 1$; $U_i = \{i\}$ if $\xi_p = i + \frac{1}{2}$.

In the following derivations we are interested in replacing a set of particles located in a given cell by another set. The replacement set may have a different number, which may be larger or smaller depending on the target number of particles per cell, but the new particles should correspond to the same distribution in velocity as the old particles. It is also required that the new particles have positive mass and that the fields the new particles produce be the same as the old. As suggested in Ref. [13], one can derive a method that preserves the values of the moments on the grid. In the solution of the Vlasov–Maxwell equations, the relevant moments are the charge and current density ρ and $\mathbf{J}$, which appear as sources in Maxwell's equations. For implicit methods, the pressure tensor, $\mathbf{\Pi}$, may be required also [15]. (As will become clear, it is necessary to preserve these moments, but it is not sufficient.)

This approach has the advantage that, from the point of view of the dynamic equations, the original set of particles and the new set are equivalent; the same fields will be calculated from the new set of particles as from the old. However, the subsequent evolution of the new set of particles may be different from the old set. This effect does not alter the evolution of the system immediately, but over longer times alterations in the evolution may occur. However, from the results of the tests in Section 5 with careful replacement, the correct evolution of the system

appears to be preserved and the effects of replacing particles appear to be small. Note that this approach preserved momentum. The forces among the particles of the two sets will be different, but overall momentum is preserved. The problem can then be stated as follows.

Given $N_{s,i}$ particles of the species s in cell i with masses $m_p > 0$, velocities $\mathbf{u}_p$ and positions ξ_p ($\xi_i \le \xi_p \le \xi_{i+1}$), a second set of $N'_{s,i}$ particles can be generated with masses $m_{p'} > 0$, velocities $\mathbf{u}_{p'}$ and positions $\xi_{p'}$ ($\xi_i \le \xi_{p'} \le \xi_{i+1}$) such that, for $j \in [1, M]$, it holds that

$$\rho_s(\bar{\xi}_j) = \sum_{p=1}^{N_{s,i}} q_p s_l(|\bar{\xi}_j - \xi_p|) = \sum_{p'=1}^{N'_{s,i}} q_{p'} s_l(|\bar{\xi}_j - \xi_{p'}|), \tag{2}$$

$$\mathbf{J}_s(\bar{\xi}_j) = \sum_{p=1}^{N_{s,i}} q_p \mathbf{u}_p s_l(|\bar{\xi}_j - \xi_p|) = \sum_{p'=1}^{N'_{s,i}} q_{p'} \mathbf{u}_{p'} s_l(|\bar{\xi}_j - \xi_{p'}|), \tag{3}$$

$$\mathbf{\Pi}_s(\bar{\xi}_j) = \sum_{p=1}^{N_{s,i}} m_p \mathbf{u}_p \mathbf{u}_p s_l(|\bar{\xi}_j - \xi_p|) = \sum_{p'=1}^{N'_{s,i}} m_{p'} \mathbf{u}_{p'} \mathbf{u}_{p'} s_l(|\bar{\xi}_j - \xi_{p'}|), \tag{4}$$

where $q_p/m_p = \alpha_s$ is the charge to mass ratio characteristic of the species s, and $\bar{\xi}_j = \xi_j + \frac{1}{2}$.

Equations (2) through (4) are nontrivial when at least one of $s_l(|\bar{\xi}_j - \xi_p|)$ is nonzero. For particles located in cell i, this limits the range of j to the subset $V_i = \bigcup_{p=1}^{N_{s,i}} U_i$. For $l = 1$, the subset V_i depends on the positions of the particles inside the cell; for particles distributed over the whole cell, it is $V_i = \{i - 1, i, i + 1\}$. (Note that the new particles will have different locations than the old. Thus, this method differs from Quest's [10], in which the new particles have the same locations but different velocities.)

3. SOME PROPERTIES OF THE B-SPLINE INTERPOLATION

The problem of controlling the number of particles per cell, can be attacked by looking carefully at the properties of the b-spline functions used to interpolate between the particles and the grid.

Consider a single particle located in cell i with natural coordinate ξ_p ($\xi_i < \xi_p < \xi_{i+1}$). Where $S_{jp} = s_l(|\bar{\xi}_j - \xi_p|)$, it holds that [12]

$$\sum_{j \in U_i} S_{jp} = 1. \tag{5}$$

Further, consider the linear and quadratic b-splines ($l = 1, 2$).

The set $\{S_{jp}\}_{j \in U_i}$ can be interpreted as a vector $\mathbf{S} = \{S_m = S_{i+m-2,p}\}_{m=1,3}$ in the three-dimensional space $\mathfrak{R}^3$ (for $l = 1$ either $S_{i+1,p} = 0$ or $S_{i-1,p} = 0$ depending on the position ξ_p inside the cell). Equation (5) restricts the vector $\mathbf{S}$ to lie in the plane

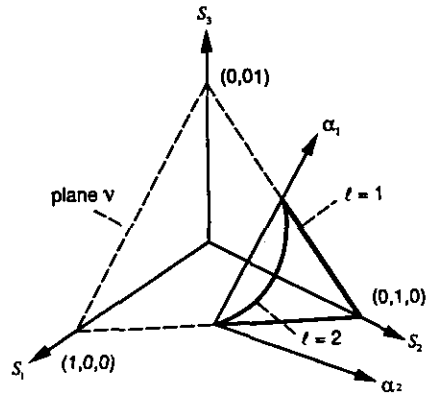

FIG. 1. The plane $\mathcal{V}$ defined by the equation $S_1 + S_2 + S_3 = 1$ is shown. Superimposed are drawn the axes α_1 and α_2 defined in Section 3.

defined by solutions to the equation $S_1 + S_2 + S_3 = 1$ which can be mapped easily onto the usual $\mathfrak{R}^2$ space. This simple mapping allows one to visualize the geometrical properties of the vector $\mathbf{S}$ and to derive some useful properties from which a simple but very effective scheme for controlling the number of particles per cell is developed. We need, first, to characterize the mapping between the plane in the 3D space to the 2D space. The following result is derived easily.

PROPOSITION 1. *The set $\mathcal{V} \subset \mathrm{P}^3$ of vectors $\mathbf{S}$, having the property $\sum_{m=1}^{3} S_m = 1$, can be mapped onto the two-dimensional space $\mathfrak{R}^2$. We choose to define the following mapping $\mathcal{G}$: $\{S_m\}_{m=1,3} \to \{\alpha_k\}_{k=1,2}$ as*

$$\mathbf{S} = \mathbf{c} + \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2, \tag{6}$$

where $\mathbf{c} = (\frac{1}{2}, \frac{1}{2}, 0)$, $\mathbf{v}_1 = (-\frac{1}{2}, 0, +\frac{1}{2})$ and $\mathbf{v}_2 = (-\frac{1}{8}, \frac{1}{4}, -\frac{1}{8})$.

Proof. The derivation of Proposition 1 is straightforward. The set $\mathcal{V}$ is a subset of $\mathfrak{R}^3$ whose dimensionality is 2, due to the relationship $\sum_{m=1}^{3} S_m = 1$; it can be regarded as the plane defined by the equation $S_1 + S_2 + S_3 = 1$ of the three-dimensional space. It is therefore possible to map $\mathcal{V}$ onto the usual $\mathfrak{R}^2$ plane. We choose to map $\mathcal{V}$ by introducing in $\mathfrak{R}^3$ a frame of reference with its center at the point $\mathbf{c} \in \mathcal{V}$ that, for $l = 1$, 2, can be regarded as the outcome of an interpolation from a particle centered on the left vertex of the cell. One axis (α_1) is chosen parallel to $\mathbf{v}_1$, i.e., the direction parallel to the segment intercepted by the plane $\mathcal{V}$ with the coordinate plane (S_1, S_3) (see Fig. 1). A second axis (α_2) is chosen to be orthogonal to the first axis and in the plane $\mathcal{V}$. This axis can be shown to be directed along $\mathbf{v}_2$. The third axis is immaterial to our analysis and can be chosen freely. Equation (6) follows immediately from expressing a point $\mathbf{S} \in \mathcal{V}$ within the new frame of reference.

The linear mapping $\mathcal{G}$ from the original point $\mathbf{S}$ to the 2D vector $\boldsymbol{\alpha}$ is now uniquely defined. The expression for the mapping $\mathcal{G}$ can be rewritten conveniently as

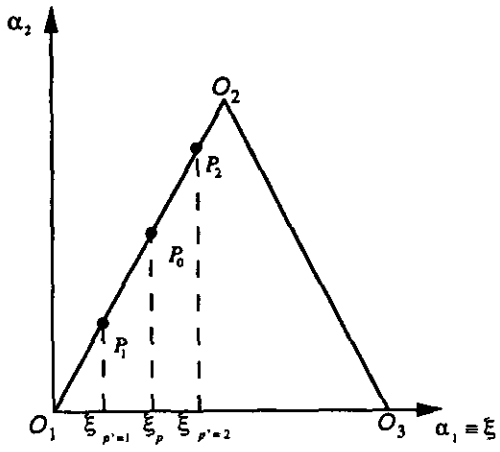

FIG. 2. With linear b-splines, any two points P_1 and P_2 lying on different sides of P_0, but on the same side of the triangle, correspond to particles with positive weight and with the same "footprint" on the grid.

$$S_1 = \frac{1}{2} - \frac{\alpha_2}{2} - \frac{\alpha_2}{8}$$
$$S_2 = \frac{1}{2} + \frac{\alpha_2}{4} \tag{7}$$
$$S_3 = \frac{\alpha_1}{2} - \frac{\alpha_2}{8}.$$

These equations can be inverted easily to yield

$$\alpha_1 = 2(S_3 + \alpha_2/8)$$
$$\alpha_2 = 4(S_2 - 1/2). \tag{8}$$

The set of vectors $\mathbf{S}$ resulting from the interpolation of a single particle is further reduced to a subset of $\mathcal{V}$ by the relationship with the b-splines. Indeed the b-spline interpolation acts as a nonlinear mapping from the one-dimensional ξ_p space onto a line in the 3D space. In this context the following result can be derived.

PROPOSITION 2. *The graph of the function* $\mathbf{S}(\xi_p) = \{S_m(\xi_p) = s_i(|\xi_{i+m-2} - \xi_p|)\}$, $[i, i+1] \subset \Re \to \mathcal{V} \subset \Re^3$ *is a line in* $\Re^3$ *whose parametric equations are, for* $l = 1$,

$$S_1 = \begin{cases} 1 - |\frac{1}{2} + \xi_p - i| & \text{if } \xi_p < \xi_{i+1/2} \\ 0 & \text{otherwise} \end{cases}$$
$$S_2 = 1 - |\tfrac{1}{2} - \xi_p + i| \tag{9}$$
$$S_3 = 1 - S_1 - S_2,$$

and for $l = 2$,

$$S_2 = \tfrac{3}{4} - (\tfrac{1}{2} - \xi_p + i)^2$$
$$S_3 = (\xi_p - i)^2/2 \tag{10}$$
$$S_1 = 1 - S_1 - S_2;$$

$\mathcal{G}$ *maps the graph of* $\mathbf{S}$ *into the line in* $\Re^2$ *of equations*

$$\alpha_2 = 2 - |4(\alpha_1 - \tfrac{1}{2})|$$
$$\alpha_1 = \xi_p \tag{11}$$

for $l = 1$ *and*

$$\alpha_2 = -4(\alpha_1^2 - \alpha_1)$$
$$\alpha_1 = \xi_p \tag{12}$$

for $l = 2$.

Proof. These results easily follow using the analytic expressions for the b-splines and using Eq. (6) for mapping the line from $\mathcal{V}$ to $\Re^2$.

The result just stated has an important geometrical meaning. As a particle moves from the left vertex of the cell toward right the vector $\mathbf{S}$ moves along the line of Eq. (9) (or Eqs. (10)) on the plane $\mathcal{V} \subset \Re^3$. This line can be visualized much more easily in $\Re^2$: the evolution of $\mathbf{S}$ is mapped by $\mathcal{G}$ onto the triangle ($l = 1$) of Fig. 2 or the parabola ($l = 2$) of Fig. 3. This simple picture of the evolution of the vector $\mathbf{S}$, representing the interpolation between particles and grid, allows very simple reasoning about the control of the number of particles per cell.

Before that, two other results are needed.

PROPOSITION 3. *Given* N *vectors* $\mathbf{S}_n \in \mathcal{V}$ *and the corresponding vectors* $\boldsymbol{\alpha}_n \in \Re^2$, *linear combinations* $\mathbf{S} = \sum_n a_n \mathbf{S}_n$ *are mapped by* $\mathcal{G}$ *onto linear combinations (l.c.)* $\boldsymbol{\alpha} = \sum_n \boldsymbol{\alpha}_n$ *if and only if* $\sum_n a_n = 1$.

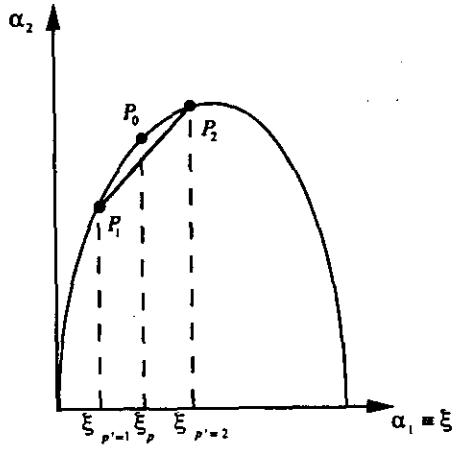

FIG. 3. With quadratic b-splines, particles at P_1 and P_2 have positive weight but yield the same "footprint" only as their distance from P_0 approaches zero.

Proof. This result can be understood easily by noting that if $\sum_n a_n \neq 1$ then $\mathbf{S} = \sum_n a_n \mathbf{S}_n \notin \mathcal{V}$ because $\sum_j S_j \neq 1$. On the other hand, recalling the expression (6) for the mapping $\mathcal{G}$, we have:

$$\sum_n a_n \mathbf{S}_n = \sum_n a_n(\tfrac{1}{2}, \tfrac{1}{2}, 0) + \sum_n a_n \alpha_{n1} \mathbf{u}_1 + \sum_n a_n \alpha_{n2} \mathbf{u}_2, \quad (13)$$

where the need for the constraint on the sum of the a_n is obviously required to map the linear combinations into each other.

The final goal of these calculations is to be able to compare different sets of particles; therefore, the contributions from different particles must be combined. Proposition 3 allows one to operate on linear combinations in the 2D space as if they were performed directly on the original vector **S**. One must only take care to respect the condition $\sum_n a_n = 1$.

An important result can be derived concerning the interpolation from two different sets of particles, the first with just one particle and the second with L particles, where $L > 1$.

PROPOSITION 4. *For linear b-spline interpolation ($l = 1$), given a particle located at ξ_p and any two particles ($L = 2$) located in the same half of the cell as ξ_p, i.e., with $\Pi_{p'=1,2}(\xi_{p'} - i - \frac{1}{2}) > 0$, but on different sides of ξ_p, i.e., with $\Pi_{p'=1,2}(\xi_{p'} - \xi_p) < 0$, we have:*

$$S_{jp} = \sum_{p'=1}^{2} \varepsilon_{p'} S_{jp'}, \forall j \in U_i, \quad (14)$$

with

$$\varepsilon_2 = \frac{\xi_p - \xi_{p'=1}}{\xi_{p'=2} - \xi_{p'=1}} > 0, \quad \varepsilon_1 = 1 - \varepsilon_2 > 0. \quad (15)$$

Proof. This result follows from Propositions 1 through 3. First we have that if Eq. (14) holds, $\sum \varepsilon_{p'} = 1$ easily follows by summing over $j \in U_i$. This allows us to use Proposition 3. From Proposition 2, S_{jp} is mapped onto a point P_0 on the triangle of Eq. (11) (see Fig. 2). Any two points P_1 and P_2 on the same side of the triangle, where P_0 lies but on opposite sides of P_0 can be linearly combined, with positive factors, to regain P_0. The weights are given by elementary geometry. These weights, thanks to Proposition 3, are the $\varepsilon_{p'}$ of Eq. (15).

A few comments on Proposition 4 are in order. First, if one of the new positions is chosen to coincide with the old one, we get a trivial result, namely the old particle is recovered and the other particle has zero weight. Second, if the points P_1 and P_2 are on different sides of the triangle, no linear combination of them can give any point on the triangle (apart from the trivial case of P_1 and P_2 themselves) and therefore not P_0. Third, $L > 2$ is allowed, but the weights $\varepsilon_{p'}$ are then not uniquely defined. Fourth, the inequalities, Eq. (15), guarantee that the particles have positive weights.

A final comment concerns the extension of the previous result to the case $l = 2$. The exact extension is indeed impossible as any linear combination of points on the parabola of Eq. (12), with positive factors, results in a point *inside* the polygon $\mathcal{P}$ defined by these points. This polygon is inscribed inside the parabola and none of the points on the parabola can be inside $\mathcal{P}$ (apart from the vertices of the polygon themselves). Therefore we cannot reconstruct a point P_0 with other points different from P_0. However, if two points P_1 and P_2 are chosen close to P_0, the parabola can be approximated linearly; and the results of Proposition 4 still hold (see Fig. 3).

However, the results for $l = 1$ do extend exactly to the case $l = 0$, because in this case U_i reduces to cell i alone, and Eq. (14) reduces to $1 = \sum_{p'} \varepsilon_{p'}$, since $S_{ip} = 1$. Any set of L particles lying in cell i trivially satisfy the condition, provided that for each set, the weights sum to 1.

4. A METHOD TO CONTROL THE NUMBER OF PARTICLES

Proposition 4 derived in Section 3 provides the basis for a simple technique for controlling the number of particles per cell. In the following we detail the application of the results derived in Section 3 to splitting and coalescing particles. This ability can be used successfully in kinetic PIC simulations to reduce the computational burden required in many important physical situations.

4.1. *Splitting*

The result of Proposition 4 can be used immediately to construct an algorithm to split any given particle into two new particles. The algorithm is: *Replace one particle with position ξ_p, velocity $\mathbf{u}_p$, charge q_p, and mass m_p, by $L = 2$ new particles with positions $\xi_{p'}$, velocities $\mathbf{u}_{p'} = \mathbf{u}_p$, charges $q_{p'} = \varepsilon_{p'} q_p$, and masses $m_{p'} = \varepsilon_{p'} m_p$, where*

$$\varepsilon_2 = \frac{\xi_p - \xi_{p'=1}}{\xi_{p'=2} - \xi_{p'=1}} > 0, \quad \varepsilon_1 = 1 - \varepsilon_2 > 0, \quad (16)$$

are the weights as derived in Proposition 4.

Note that Eq. (14) of Proposition 4 assures that this replacement preserves the contributions of particle p to all moments of the distribution with linear interpolation,

$$\sum_L q_{p'} S_{ip'} \mathbf{T}(\mathbf{u}_{p'}) = q_p \mathbf{T}(\mathbf{u}_p) S_{jp}, \quad (17)$$

and that the particles have positive mass. This result is valid even if ρ and **J** are assigned to different grids. Thus, one can use this algorithm to control the number of particles in codes that use staggered grids, such as CELEST1D [15].

Moreover, because the new particles have different positions in space, their evolution in time will be different even though their velocities are equal. This property of the splitting algorithm guarantees that the increase in the number of particles results in a real increase in the capability of describing a feature of the system and in a reduction in the level of noise in the results of the simulation. Further, it is argued that the velocity distribution varies very little over the distance a particle can travel in a time step, typically a cell. Thus, separating the particles in space by a fraction of a cell width introduces little or no error.

This algorithm, which is based on Proposition 4, is valid only when the interpolation scheme uses NGP ($l = 0$) or linear ($l = 1$) b-splines. In principle, it cannot be used for the PIC codes in which interpolation schemes with $l = 2$ are used. However, as previously remarked, the results of Proposition 4 can be considered a good approximation for $l = 2$, especially when the positions of the two new particles are sufficiently close. We suggest choosing two new positions by the rule:

$$\xi'_{p'=1} = \begin{cases} \xi'_p & \text{if } \xi'_p < \frac{1}{2}, \\ \frac{1}{2}(\xi'_p - \frac{1}{2}) + \frac{1}{2} & \text{if } \xi'_p \geq \frac{1}{2}, \end{cases} \tag{18}$$

$$\xi'_{p'=2} = \begin{cases} \xi'_p + \frac{1}{2}(\frac{1}{2} - \xi'_p) & \text{if } \xi'_p < \frac{1}{2}, \\ \xi'_p + \frac{1}{2}(1 - \xi'_p) & \text{if } \xi'_p \geq \frac{1}{2}, \end{cases} \tag{19}$$

where $\xi'_p \equiv \xi_p - i$, that is to say in the midpoints of the two segments $O_1 - P_0$ and $P_0 - O_2$ if $\xi'_p < \frac{1}{2}$ or $O_2 - P_0$ and $P_0 - O_3$ if $\xi'_{p-i} \geq \frac{1}{2}$ (see Fig. 2).

Note that this rule causes particles very close to the center of the cell to be replaced by two new particles whose distance from the original particle are unequal. This gives rise to two new particles with very different weights. To maintain roughly uniform weights, we avoid splitting particles near the center of the cell.

We have also generated new particles with the positions $\xi_{p'} = \xi_p \pm \delta$, where δ is a fixed displacement whose value can be, for example, assigned to $\delta = 1/N$, where N is the number of particles per cell. In this case the two new particles have the same weight. However particles with $\xi_p - i < \delta$ (i.e., close to the left boundary of the cell) or $\xi_p - i > 1 - \delta$ (i.e., close to the right boundary) cannot be split.

The results reported in Section 5 are obtained with CELEST1D, which uses b-splines with $l = 0, 1, 2$ simultaneously for different moments [15]. The results show that both these schemes for generating the new particles can be very satisfactory and that the approximation underlying the application of the algorithm for $l = 2$ is accurate.

Note finally that it would be possible to split one particle into more than two particles, $L > 2$, but in this case the weight of the new particles is not uniquely determined by Proposition 4. If more than one new particle is needed, one should split several different initial particles into two rather than one particle into more than two. The former avoids the creation of particles with charges much smaller than the charge of the original particle.

4.2. *Coalescence*

Even though the splitting technique can be inverted exactly, in principle, to obtain a method to coalesce $L = 2$ particles into one, it is never possible to find two particles in a given cell with exactly the same velocity. Still this technique can be used, at least approximately, if one coalesces only those particles with similar velocities.

With this approximation, we can use the results given in Proposition 4 to coalesce two particles (labeled $p = 1, 2$) with positions ξ_p, velocities $\mathbf{u}_p$, charges q_p, and masses m_p into one (labeled *new*) with charge $q_{\text{new}} = q_1 + q_2$, mass $m_{\text{new}} = m_1 + m_2$, and position $\xi_{\text{new}} q_{\text{new}} = \xi_1 q_1 + \xi_2 q_2$. This last result follows from Proposition 4 using Eq. (15) with $\varepsilon_1 = q_1/q_{\text{new}}$ and $\varepsilon_2 = q_2/q_{\text{new}}$.

The new velocity can be defined to preserve each component of the particle momentum

$$q_1 \mathbf{u}_1 + q_2 \mathbf{u}_2 = q_{\text{new}} \mathbf{u}_{\text{new}} \tag{20}$$

or to preserve the energies of each degree of freedom of particle motion,

$$q_1 u_{1,\alpha}^2 + q_2 u_{2,\alpha}^2 = q_{\text{new}} u_{\text{new},\alpha}^2 \tag{21}$$

but not both simultaneously ($\alpha = 1, 2, 3$ indicates the three components of the velocity vector). However the results of several tests (see Section 5) shows very clearly that either choice, Eq. (20) or Eq. (21), gives acceptable results. In practice when $|\mathbf{u}_1 - \mathbf{u}_2| \ll |\mathbf{u}_1| + |\mathbf{u}_2|$ (i.e., the velocities are similar) the fulfillment of any one of Eqs. (20) and (21) results in a rather good approximation to the other.

Note that the extension of this algorithm to $l = 2$ is once again only approximate. This technique is indeed less precise than the splitting. The error in the approximation underlying the extension of Proposition 4 to $l = 2$ is added to the averaging of the velocities. Averaging alters the particle distribution. However, as shown in the results of Section 5 if the particles are chosen to be close in the phase-space (and in particular in the velocity space) the degradation in the description of the particle distribution is low.

5. RESULTS

The algorithm for controlling the number of particles per cell developed in the previous Section has been extensively tested in several applications using an implicit, adaptive-grid, electromagnetic code, CELEST1D, for plasma simulation in one dimension [15].

The tests probe two aspects of the algorithm. First, we show

that the technique to control the number of particles does not alter significantly the particle distribution function and the evolution of the system; no nonphysical processes are introduced and the correct evolution is conserved. Second, we show clearly in two relevant physical situations that the adoption of an adaptive grid and the control of the number of particles can indeed greatly improve the performance of PIC simulation codes by significantly reducing the CPU time needed to obtain the required results.

5.1. *Test* 1: *Particle Distribution*

Before trying to apply the control of the number of particles per cell in a kinetic code, it is important to check how, given a set of particles sampled from a known distribution function, particle coalescence alters the original distribution function. We test only coalescence, because splitting does not change the particle distribution at all since the new and the old particles share the same velocity. To resemble as closely as possible typical PIC simulations, good statistics of the particle distribution are achieved by generating a system of 200 cells with 64 particles each. The particle distribution function is obtained by superimposing two warm beams with Maxwellian distributions. The beams are given equal temperatures corresponding to a Debye length, $\lambda_{De} = 0.4c/\omega_{pe}$, and with beam velocities, $u_x/c = -0.5$ and $u_x/c = 0.5$. We limit our attention here to the x-component, u_x, of the velocity.

By repeatedly coalescing particles in each cell, we reduce the number of particles from 64 per cell to 32. The pairs of particles to be coalesced are chosen to have a difference between the two velocities,

$$u^{(1)} = \left|\frac{\mathbf{u}_1 q_1 + \mathbf{u}_2 q_2}{q_1 + q_2}\right|; \quad u^{(2)} = \left(\frac{u_1^2 q_1 + u_2^2 q_2}{q_1 + q_2}\right)^{1/2} \tag{22}$$

(the notation of Section 4.2 is used, and the index x is dropped) as close as possible to a given value:

$$\varepsilon = \frac{|\mathbf{u}^{(1)} - \mathbf{u}^{(2)}|}{|\mathbf{u}^{(1)}| + |\mathbf{u}^{(2)}|} 200 = \varepsilon_{\text{target}}. \tag{23}$$

Note that the definition (23) of the distance in the velocity space is more useful than the natural distance $|\mathbf{u}_1 - \mathbf{u}_2|$ as it is directly related to the violation of the conservation of energy and momentum. Indeed $\mathbf{u}^{(1)}$ is the velocity of the new particle as required by conservation of momentum and $\mathbf{u}^{(2)}$ by conservation of energy; the difference between the two is, therefore, related to the inability to satisfy both the conservation relations.

The aim is to assess how large an error is introduced by the difference in velocity between the coalesced particles. This goal is achieved by applying the χ^2 test (with 30 bins) and the Kolmogorov–Smirnov (K–S) [14] test to the particles before and after coalescence.

TABLE I

$\varepsilon_{\text{target}}$	ε_{av}	ε_{max}	K-S	χ^2
Before coalescence	—	—	100%	95%
10^{-1}	9.999×10^{-2}	1.060×10^{-1}	100%	95%
1	1.000	1.023	100%	94%
3	3.000	3.032	100%	91%
5	4.998	5.048	96%	89%
10	10.000	10.124	57%	32%

Table I reports the results of the tests as a function of the assigned $\varepsilon_{\text{target}}$. As a matter of fact the two particles coalesced in each cell do not have exactly a value of ε equal to $\varepsilon_{\text{target}}$; therefore, Table I reports the maximum and average value of ε among the cells.

The results show that when the particles are close in velocity, the degradation of the information is quite low; and when the particles are not close in velocity, the distribution of the particles is considerably altered and the probability of agreement with the original distribution, as evaluated by the χ^2 and K–S tests, is low.

It must be stressed that the results of this test show the reliability of the coalescing method for low values of ε, but they cannot be used to find a threshold value of ε. This is due to the intrinsic limitation of the tests performed. They only give a statistical answer regarding the agreement between the two distributions. Further, the results of this test are very noisy. This is especially true for the χ^2 test where the results depend also on how the particles are grouped into bins.

A limit value for ε can only be found in relation to the specific problem at hand. Indeed different limit values of ε have to be expected depending on the structure of the particle distribution and on the importance of the zones where the particles are coalesced. We return to this problem below.

Note, finally, that, as anticipated in Section 3, the results in Table I are not affected by the procedure used to average the old velocities to generate the velocity of the new particle (Eq. (20) or Eq. (21)). This is a strong hint that the approximation underlying the coalescing method is indeed extremely good.

5.2. *Test* 2: *Sheath Problem*

The actual benefits that can be gained by the control of the number of particles are shown in problems where localized phenomena are present. Two examples are the electrostatic sheath problem and the electromagnetic slow-shock results that follow.

When a Maxwellian plasma of finite size is surrounded by perfectly conducting walls at zero potential, the different mobility of electrons and ions gives rise to a potential barrier which slows the flow of electrons toward the walls [16]. The presence of thin sheath regions with strong gradients of the electrostatic potential suggests that the use of a grid with finer spacing in

the sheaths and coarser spacing in the bulk of the plasma can be very effective.

We simulate a one-dimensional plasma composed by ions and electrons with mass ratio $m_i/m_e = 100$. Initially the electron temperature corresponds to $\lambda_{De} = 0.1c/\omega_{pe}$, and the ratio of the ion and electron temperatures is given by $\lambda_{De}/\lambda_{Di} = 1.0$. No magnetic field is present, and the electrostatic potential is constrained to be 0 at the boundary of the system. An open boundary condition is assumed for the particles so that, when a particle hits the wall, it is simply removed from the simulation. Note that no steady state can be reached.

The simulation box is $L = 4c/\omega_{pe}$ wide, and the evolution is followed for $\omega_{pe}t = 20$. As the simulation proceeds, we adapt the grid. We generate the adaptive grid from an elliptic diffusion-like equation suggested by Winslow [19],

$$\frac{d}{d\xi} w \left(\frac{dx^{n+1}}{d\xi} \right) = w\tau(x^{n+1} - x^n), \tag{24}$$

where x^n (x^{n+1}) is the old (new) grid as a function of the natural coordinate ξ, and τ is a time constant controlling the delay in the response of the grid to changes in the weight function w. The weight w in this problem is related to the x-component of the electric field E_x. If the computed electric field were used directly as a weight function, the noise usually present in PIC simulations would result in an equally noisy grid. For this reason the weight w is related to E_x by a smoothing operator [17],

$$\frac{d^2w}{d(x^n)^2} - \frac{w}{L^2} = -\frac{|E_x|}{L^2}, \tag{25}$$

where L is the coupling distance over which the noise is effectively smoothed away. A simple Fourier analysis shows that Eq. (25) acts effectively as a low-pass filter to damp the high frequency noise. The electric field from which w is calculated is averaged over several time steps to reduce the noise further.

The control of the number of particles per cell adopts Eqs. (18)–(19) as a rule for generating the new positions when a particle is split and with the average velocity defined by Eq. (21) to preserve energy. In each time step the splitting algorithm acts in those cells where the number of particles drops below the initial value by more than one particle. Similarly the coalescing routine acts where the number of particles exceeds the target value by one. Note that the number of particles in the cell j, N_j, is defined as

$$N_j = \sum_p s_2(|\xi_p - \xi_j|), \tag{26}$$

where ξ_j is the center of the jth cell. With this definition N_j is a real number, and a fractional number of particles per cell are present.

The particles to be coalesced in a given cell are chosen to be the closest in velocity among the particles in the cell. However, there appears to be no need to put a threshold value for the difference ε defined in Eq. (23) in this calculation.

In the following, the results of four different runs are reported to give a complete view of the advantages that can be achieved by the use of an adaptive grid with a control of the number of particles.

Run a is a reference case with a very fine grid (1000 cells) and with many particles per cell (128) obtained without adapting the grid or controlling the number of particles per cell. In Run b the adaptive grid is used with the control of the number of particles described in the previous sections. A much smaller number of particles per cell is used (16) on a much coarser grid (80 cells). Run c is similar to Run b, but an adaptive grid is used wihtout controlling the number of particles so that the number of particles per cell will change quite abruptly from cell to cell and time step to time step. Finally, Run d is a standard run without an adaptive grid or control of the number of particles, but with the same number of particles and cells as in Runs b and c.

Let us start with a comparison of the spatial profile of the electron potential energy at the end of the simulation as shown in Fig. 4. It is clear that Run b is the closest to the reference results (Run a) both in the general shape of the potential, and in the actual numerical value.

Even more stringent conclusions can be drawn by comparing the velocity distribution of the ions exiting the system (Fig. 5). The poor statistics of the particle distribution are particularly evident in Run c, where an adaptive grid is used without controlling the number of particles.

5.3. *Test* 3: *Slow Shock*

The simulation of a collisionless shock is a nontrivial test of the algorithm for dynamic control. There is considerable nonuniformity of the gradient length scales, and an adaptive grid can give increased resolution without increase in the computational effort, but only if the number of particles per cell is controlled. However, the distribution in velocity is very far from equilibrium, and preserving the first and second moments is probably insufficient to maintain the accuracy of the calculation. Thus, dynamic control must avoid introducing errors in the distribution. In fact, the dynamic control algorithm assumes no knowledge of the plasma distribution, and it contains a control parameter, ε, defined by Eq. (23) to control the error that is made when coalescing particles. This parameter can be made as small as one likes, even zero, so that no error is made, but at the cost of reducing the degree of control of the particle number.

In the slow-shock calculation below, a flowing, magnetized plasma is reflected by a stationary piston where the component of the magnetic field perpendicular to the normal piston is set to zero, consistent with a switch-off slow shock [18]. The initial values for the plasma state variables and fields are set according

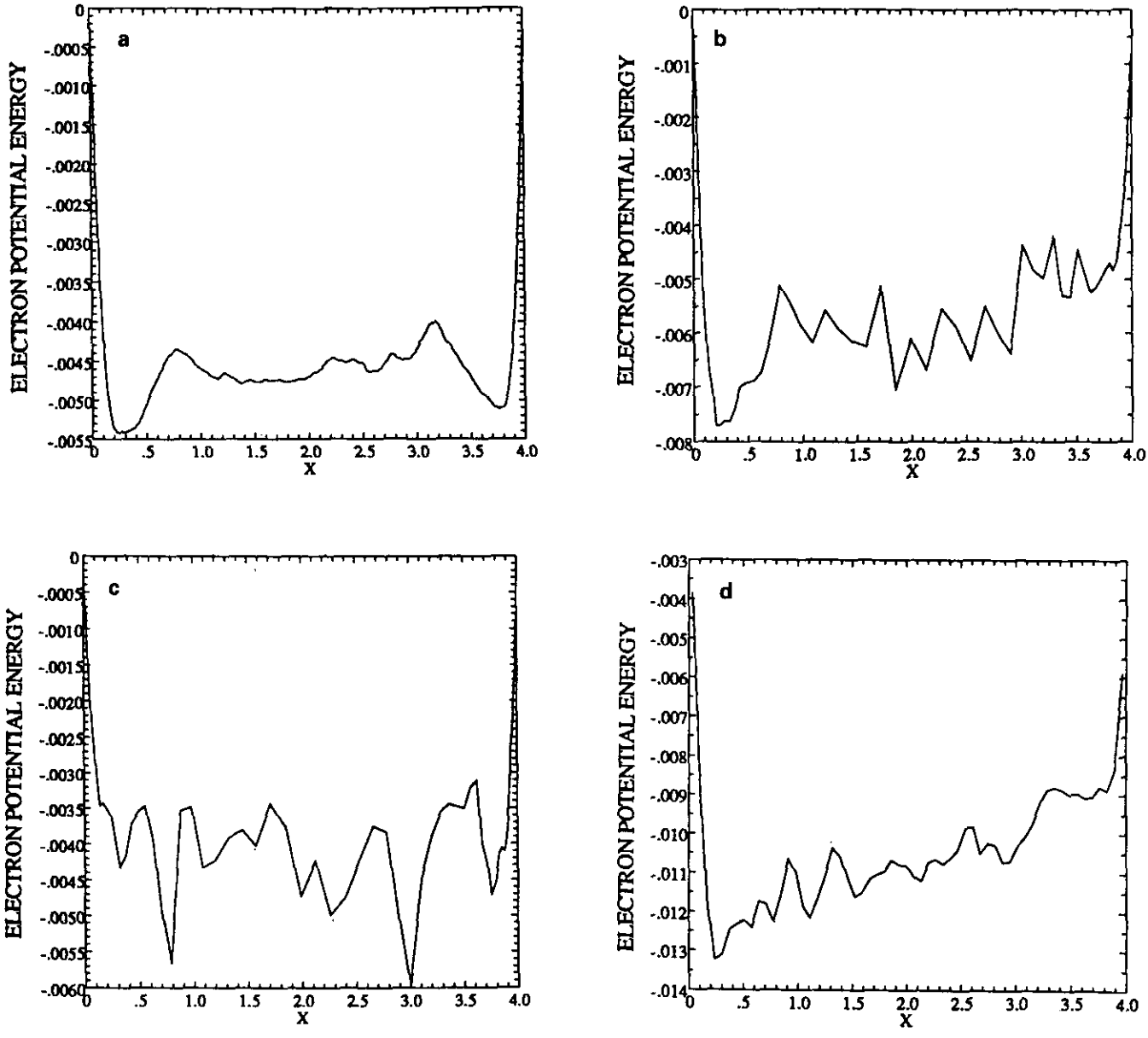

FIG. 4. The profile of the potential for the calculation of an electrostatic sheath is shown for (a) a reference calculation with 1000 zones and 128 particles per cell; (b) a calculation with an adaptive grid and dynamic particle control with 80 zones and 16 particles per cell; (c) a calculation with an adaptive grid as in (b) but without dynamic control; and (d) a calculation with a fixed grid as in (a), but with 80 zones and 16 particles per cell as in (b) and (c). Only (b) compares with (a) in the value of the potential drop across the sheath indicating that the particle flux through the sheaths is calculated inaccurately in (c) and (d).

to the Rankine–Hugoniot conditions. In the case below, initially the ratios of the electron and ion pressures to the upstream magnetic field are, $\beta_e = \beta_i = 0.01$. The ratio of ion to electron mass is $m_i/m_e = 25$, the shock normal angle with respect to the magnetic field is $\psi = 75°$, and the ratio of the upstream Alfvèn speed to the speed of light is $V_A/c = \omega_{ci}/\omega_{pi} = 0.01$. The simulation box length is $L = 2000c/\omega_{pi}$, and the shock is followed until $\omega_{ci}t = 5000$.

For reference, the results of two calculations on uniform grids are shown. The calculations are performed with 200 cells and spacing $1.0c/\omega_{pi}$, and with 400 cells and spacing $0.5c/\omega_{pi}$. In Fig. 6, a stack plot of B_z as a function of position is drawn at 50 equally spaced intervals between $t = 0$ and $\omega_{ci}t = 5000$. In Fig. 7, a hodogram of B_z as a function of B_y is plotted at $\omega_{ci}t = 5000$. The shock front, Fig. 6, coincides with the abrupt change in the magnetic field. The trailing magnetic wave

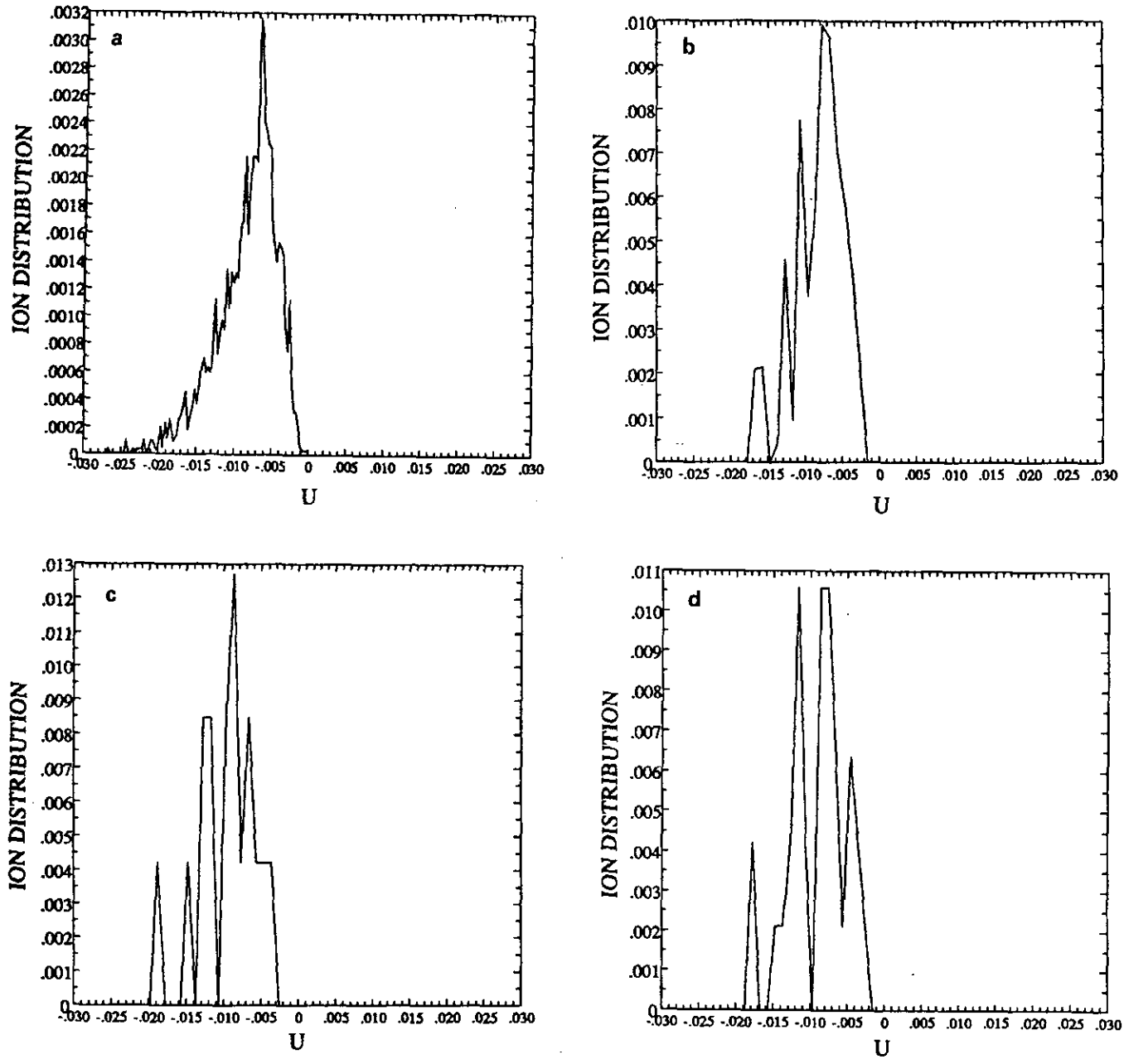

FIG. 5. The velocity distribution of ions penetrating the sheaths is plotted for cases (a)–(d) described in Fig. 4. The shape in (a) is reproduced in (b); the difference in maximum value is due to the smaller number of particles in (b) than in (a). Cases (c) and (d) indicate there are too few particles to give an accurate particle flux.

(TMW) behind the shock is a monochromatic, circularly polarized wave whose characteristic signature is the spiral trace in Fig. 7. In both runs, the shock speed is the same, but the TMW wavelength is one-half as large with 400 cells as with 200. The wavelength is sensitive, evidently, to the cell spacing. To demonstrate convergence, even smaller spacing is required than that given by the 400 cell mesh.

To increase resolution, adaptive grids are generated using Eq. (24), but with a prescribed weight function to give finer gridding in the region of the shock. The weight evolves in time to follow the shock, which progresses at a known, steady speed, and is given by

$$w = 1 + \tanh\left(\frac{x - x_c}{\mathscr{L}}\right) + \sigma. \tag{27}$$

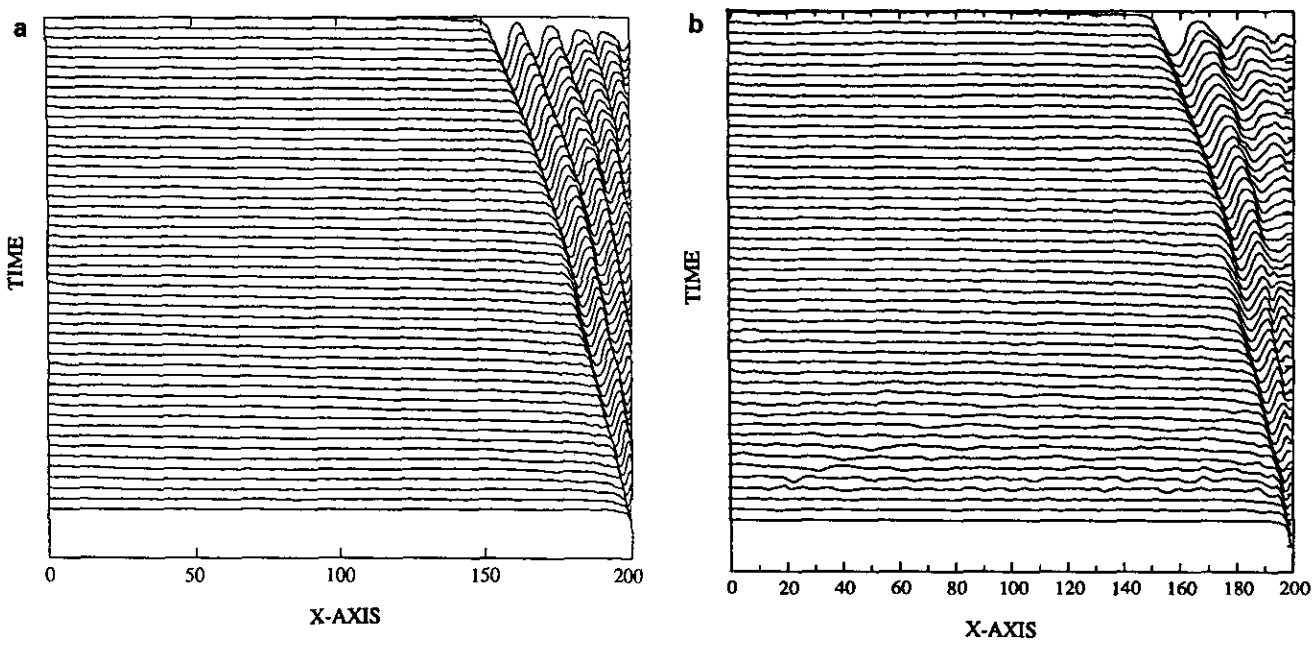

FIG. 6. The spatial profile of B_z is plotted at 50 equally spaced times between $\omega_{ci}t = 0$ and $\omega_{ci}t = 5 \times 10^3$: (a) with 400 cells and (b) with 200 cells.

In the simulations, $L = 30\ c/\omega_{pi}$ and x_c varies linearly in time from $x_c = 120\ c/\omega_{pi}$ to $x_c = 60\ c/\omega_{pi}$. The value of σ varies throughout the calculation either to keep the width of the largest cell constant and equal to a prescribed value as in the 400-cell calculation below, or to keep the smallest cell spacing constant and equal to a prescribed value as in the 260-cell calculation below. For the grid generated by the weight in Eq. (27), the largest cell is the leftmost cell. The variation in cell size for grids with 400 and 260 cells are shown in Figs. 8a and b. The 400-cell adaptive grid has a grid spacing in the shock region approximately equal to $0.25\ c/\omega_{pi}$ at the end of the calculation, Fig. 8a, double the resolution obtained with a uniform grid and 400 cells. The 260-zone calculation has a grid spacing of $0.5\ c/\omega_{pi}$ in the shock region, Fig. 8b. Note the size of the largest zone in Fig. 8b. The ratio of the largest to the smallest zone is nearly 50!

As noted above, one must be more careful to preserve information on the particle distribution when coalescing particles

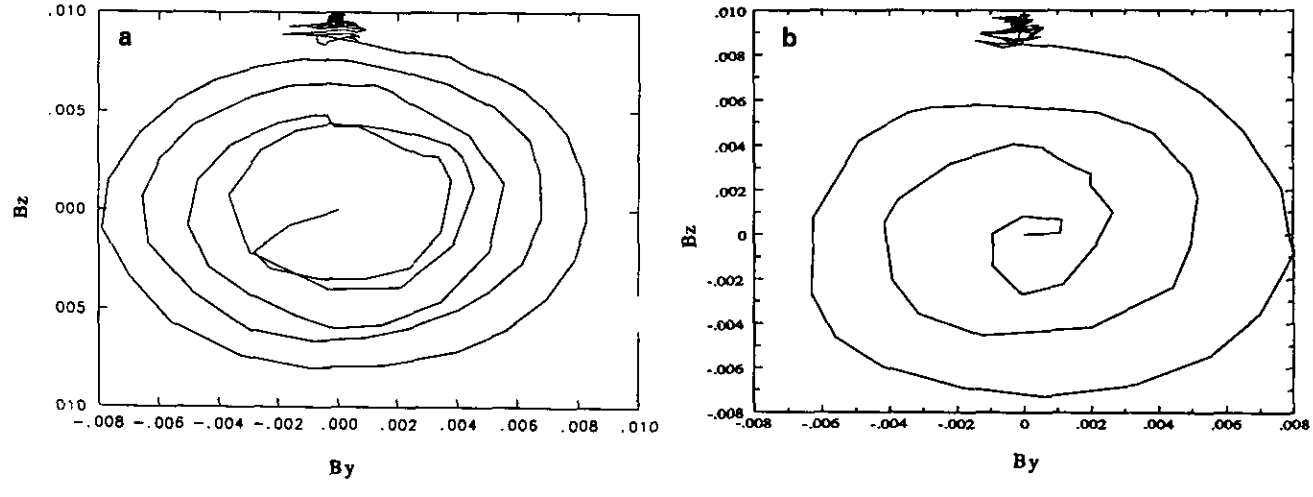

FIG. 7. The hodograms corresponding to Fig. 6 are shown. At the upstream boundary, $\mathbf{B} = (0., 0., 0.01)$. At the downstream boundary, $\mathbf{B} = 0$. With increased resolution, the wavelength is decreased.

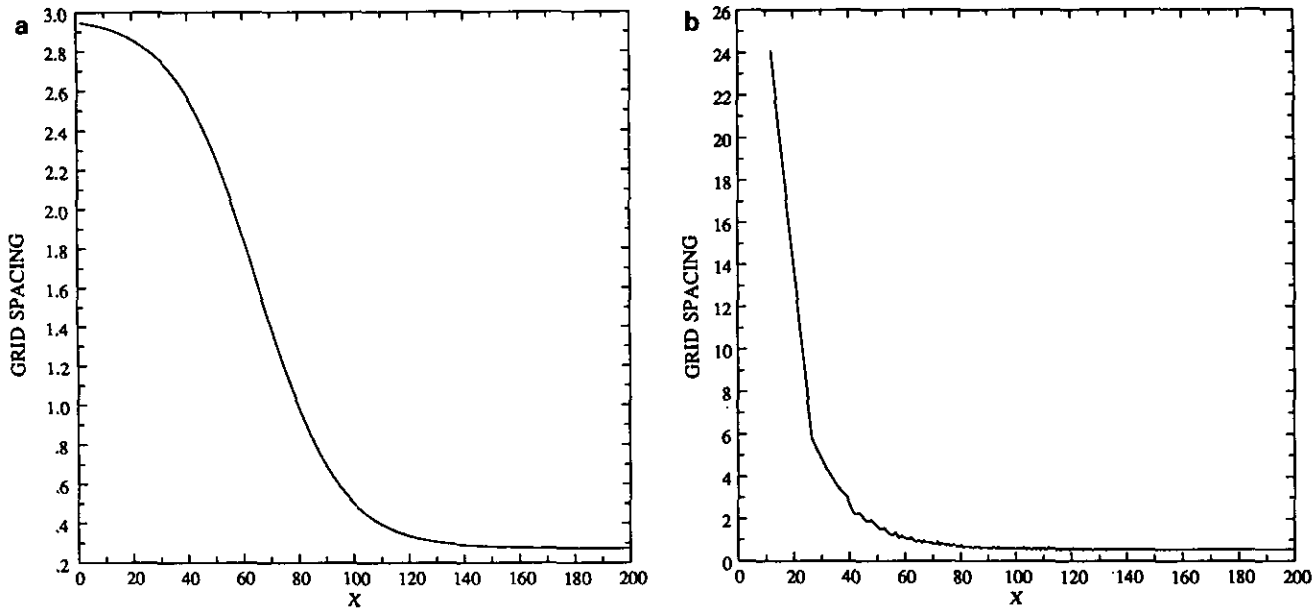

FIG. 8. The grid spacing is plotted for (a) 400 cells and (b) 260 cells. Note the cell-to-cell variation in spacing caused by jiggling in (b).

in the slow shock calculation than in previous cases. The effect of coalescence is here more subtle than before, because in the shock problem the mass density increases through the shock. Therefore, the number of particles in this region is increased by the combined effect of splitting particles and by physical compression, and the number of particles per cell will rise above the threshold, triggering a coalescence that can introduce errors in the most interesting zones. To reduce errors, the control parameter ε, defined by Eq. (23), is set to a small value, and a pair of particles to be coalesced is required to be within one-half the grid spacing. In addition, the tolerance in the maximum number of particles is increased so that the number can exceed the target value by 50% before coalescence begins. If the number falls below the target value, particles are split, with new positions chosen according to the rule $\xi_{p'} = \xi_p \pm 1/N_p$.

With an adaptive grid and dynamic control in the 400-cell case, the total number of particles increases rapidly at first, and much more slowly later, Fig. 9a. There are increases in a uniform grid calculation without dynamic control, because particles are injected and there is a density increase through the shock. The relative increase with 200 cells and a uniform grid, Fig. 9b, is nearly as large as in Fig. 9a; but the rate of increase is uniform throughout the calculation. The number of particles per cell, Fig. 10a, is nearly as constant as in a uniform grid case for $x > 40\ c/\omega_{pi}$, even though the cell spacing varies. It appears that dynamic control is successful in maintaining a constant number of particles per cell in a nonuniform mesh. However, the control appears to fail in the left-hand zones, where the number of particles in the leftmost cell climbs to 3800, almost 10 times as many as in the zones to the right. The principal reason is the rapid growth in the largest zone in the first few cycles of the computation. The grid is uniform initially, and the nonuniform grid is turned on at $t = 0$. This causes the number of particles in the leftmost cell to increase rapidly. (The cells to the right decrease in size, and the dynamic control responds by splitting particles, which causes the total number of particles to increase.) The coalescence routine in this calculation is limited to coalescing at most one pair of particles each time step, and this is not sufficient to reduce the number of particles in the largest zones to the target number. An obvious cure for this problem is to generate the particles initially on a nonuniform grid. Then the large initial increase in the number would be avoided. This is the preferable solution, because it is much less costly.

With the adaptive grid and dynamic control, the resolution in the shock region is doubled without increasing the number of cells or the number of particles. Figures 11 and 12 display a comparison between calculations with 400- and 260-cell adaptive grids. In Fig. 11, a stack plot of B_z as a function of position is drawn at 50 equally spaced intervals between $t = 0$ and $\omega_{ci}t = 5000$. In Fig. 12, a hodogram of B_z as a function of B_y is plotted at $\omega_{ci}t = 5000$. As above, the shock front, Fig. 11, coincides with the abrupt change in the magnetic field, and the TMW behind the shock is a monochromatic, circularly polarized wave whose characteristic signature is the spiral trace in Fig. 12. In both runs, the shock speed is the same, and the TMW is very similar.

Consider first the 400-cell adaptive grid calculation, Figs. 11a and 12a, compared with the 400-cell uniform grid calculation, Figs. 6a and 7a. There is one significant difference between the two runs. Upstream from the shock, the adaptive grid calculation with 400 cells is significantly more noisy than the uniform

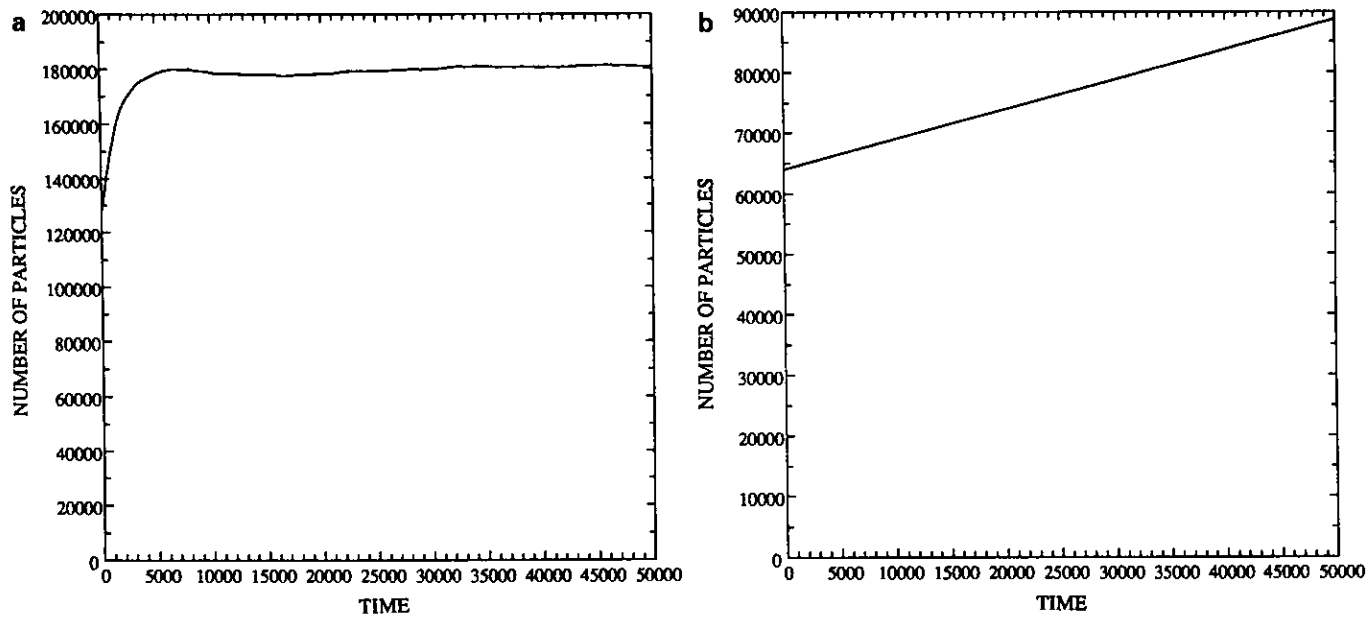

FIG. 9. The total number of particles is plotted against time for (a) a 400-cell adaptive grid with dynamic control and (b) a 200-cell uniform grid without control. The initial rise in (a) is caused by rapid changes from a uniform grid initially. The rise in (b) is due to injection.

grid case. Especially in Fig. 11a, where waves grow and then disappear, there are upstream waves that seem to have no connection to the shock. One clue to their origin is given by the ratio of the Debye length to the cell spacing, Fig. 8a, which varies from $\lambda_{De}/\Delta x = 2.8 \times 10^{-3}$ with a cell spacing of 0.25 c/ω_{pi}, to $\lambda_{De}/\Delta x = 2.4 \times 10^{-4}$ with a cell spacing of 3.0 c/ω_{pi}. Both values are small enough that the simulation should be unstable to the finite-grid instability. One author shows that for $\lambda_{De}/\Delta x < 0.3$, the finite-grid instability should occur [20]. Its absence may be explained by the reduction in the growth rate of the instability with implicit differencing and higher order interpolation [21, 22]. However, when σ is decreased so that

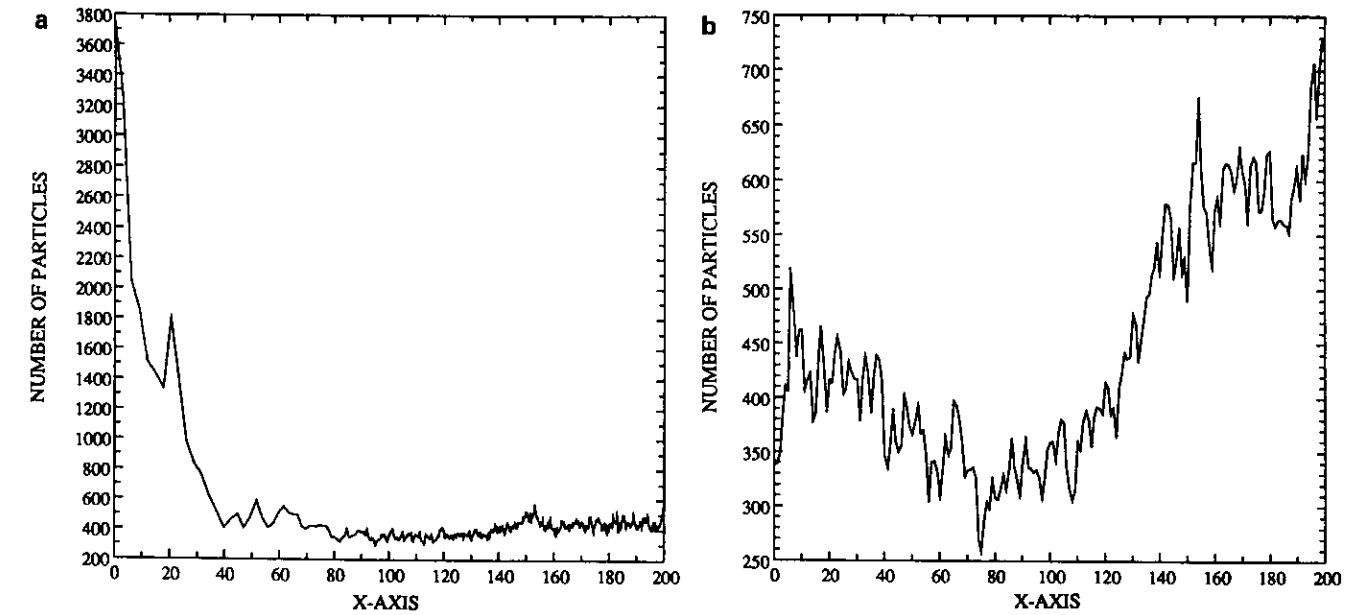

FIG. 10. The number of particles per cell on (a) a 400-cell adaptive grid with dynamic control is compared with (b) a 200-cell uniform grid without control. The large number in the left cell in (a) indicates that the control is not keeping pace with the rapid growth of the largest cells.

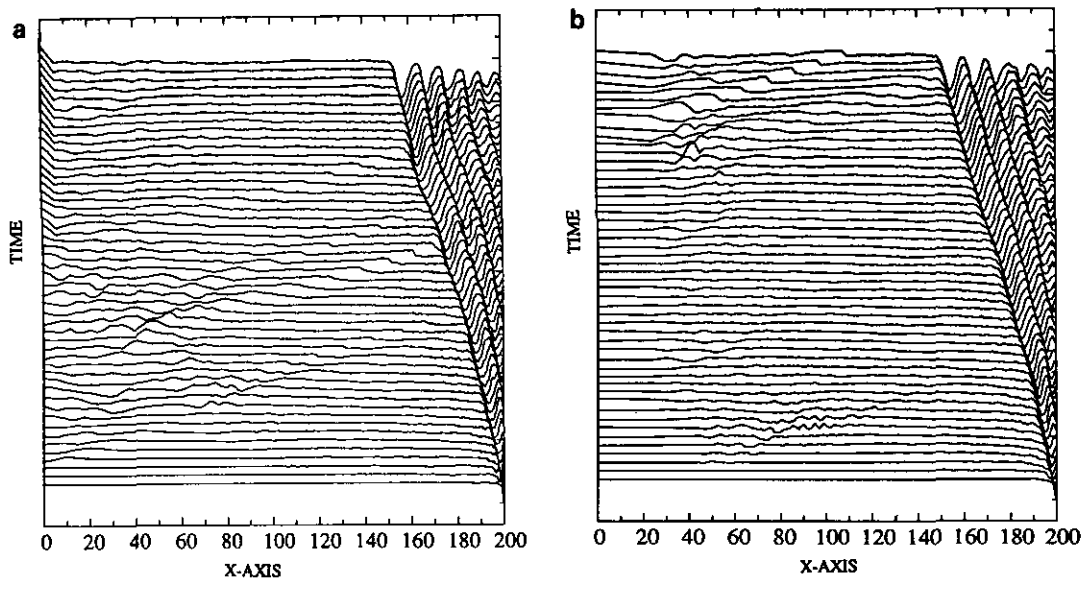

FIG. 11. Stack plots of B_z for (a) a 400-cell and (b) a 260-cell adaptive grid are shown.

the largest zone is larger than 3.0 c/ω_{pi}, the threshold even for the implicit calculation is crossed and a very strong instability occurs.

This serious constraint on the adaptive grid can be made less stringent using a modification of a technique suggested by Chen *et al.* [23]. The growth rate of the finite-grid instability is reduced by "jiggling" the mesh [24]. With jiggling and an adaptive grid, dynamic control, and 260 cells, the results, Figs. 11b and 12b, are virtually identical with those for a 400-cell uniform grid, Figs 8a and 9a, and nearly as good as for a 400-cell adaptive grid. The degree to which the grid is jiggled, i.e., given a random displacement each time step, can be seen Fig. 8b. The cell-to-cell variation in cell spacing is relatively small.

Finally, the relative change in the total number of particles is small in Fig. 9a, but the number of particles in the largest zone is astounding, Fig. 10a. Yet, there is not a correspondingly large increase in the total number of particles. The reason may be found in the data in Table II, where the variation in the size

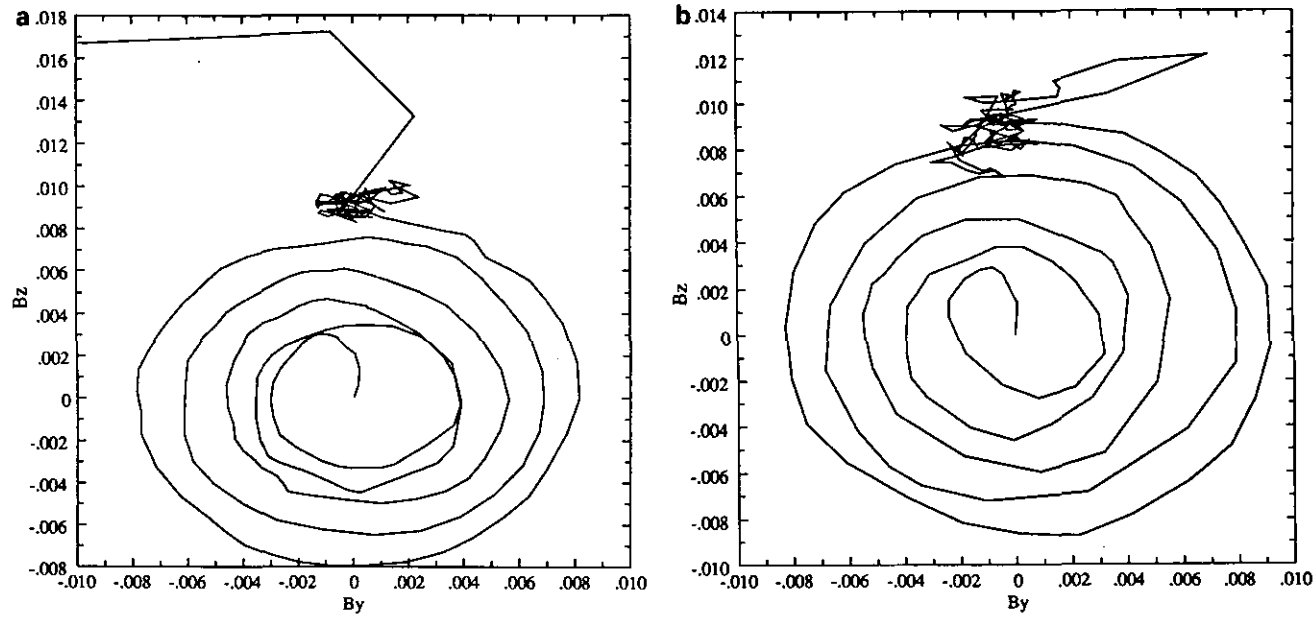

FIG. 12. Hodograms corresponding to Fig. 11 are plotted.

TABLE II

Time (ω_{ci}^{-1})	$\Delta_{max}(c/\omega_{pi})$	N_{max}	$N/\Delta x_{max}$
1000	1.6	630	394
2000	2.2	760	345
3000	3.1	1150	370
4000	17.0	5000	294
5000	24.0	7700	321

of the largest cell and the number of particles in that cell are listed. Obviously, the coalescence routine is not keeping up with the increase in the size of the largest zone late in the calculation. The number density drops but not rapidly enough to keep the number per cell constant. Perhaps a change to the coalescence routine to allow more than one pair of particles per time step to be coalesced would be useful in this case to give greater control of the particle number, but it is probably not a good idea to allow a cell to grow so rapidly.

6. CONCLUSIONS

A selective algorithm for splitting and coalescing particles has been presented. The algorithm allows the number of particles per cell to be controlled dynamically and should be especially useful when used with an adaptive grid or to maintain a level load for parallel computing.

The algorithm is approximate for the coalescence of particles, but an error tolerance can be maintained by a simple check. In cases where there is a significant deviation from local thermodynamic equilibrium, as in the slow-shock, the error tolerance must be strict. With less strict tolerances, the accuracy of the slow-shock calculations is decreased significantly.

The advantage of the algorithm presented here, besides its selectivity, is that it can be applied without knowledge of the specific distribution. It is not necessary, for example, to characterize the distribution by the computation of its velocity moments. Rather, from a very general consideration of the interpolation to the grid and a test of the accuracy of the preservation of momentum and energy, there results a simple test to determine when it is possible to coalesce and which pair of particles should be coalesced. Other reasonable considerations, such as when one should relax the requirement that there should be an equal number of particles per cell or when one should require that particle weights not be too disparate, are seen to work even in complex problems like the slow shock.

Finally, the real value of this algorithm will probably be found in the increased range of problems to which plasma simulation can be applied through its use. For its potential to be realized fully, the algorithm must be extended to two or three dimensions, and it must be proven in a wide range of problems. Nevertheless, the gains in speed and accuracy that have been demonstrated in one dimension indicate that this is a fruitful direction to follow.

ACKNOWLEDGMENTS

We gratefully acknowledge support from D.O.E. Office of Fusion Energy and Associazione per lo Sviluppo Scientifico e Tecnologico del Piemonte.

REFERENCES

1. K. B. Quest, "Simulation of Quasi-Parallel Collisionless Shocks," in *Collisionless Shocks in the Heliosphere* (Am. Geophys. Union, Washington, DC, 1985).
2. W. L. Kruer, *The Physics of Laser Plasma Interactions* (Addison–Wesley, Redwood City, CA, 1988).
3. S. E. Parker, A. Friedman, S. L. Ray, and C. K. Birdsall, *J. Comput. Phys.* **107,** 388 (1993).
4. J. U. Brackbill, in *First Engineering Research Power Supercomputer Users Symposium,* Gaithersburg, Maryland, S. C. Staff, 1991.
5. V. A. Thomas and D. Winske, *Geophys. Res. Lett.* **17,** 1247 (1990).
6. P. C. Liewer and V. K. Decyk, *J. Comput. Phys.* **85,** 302 (1989).
7. R. E. Alcouffe and R. D. O'Dell, in *CRC Handbook of Nuclear Reactor Calculations* (CRC Press, Boca Raton, FL, 1986).
8. J. Denavit, *J. Comput. Phys.* **9,** 75 (1972).
9. J. T. Beale and A. Majda, *J. Comput. Phys.* **58,** 188 (1985).
10. K. B. Quest, "Particle Acceleration in Cosmic Plasmas," in *Particle Acceleration in Cosmic Plasma,* edited by G. P. Zank and T. K. Gaisser (Am. Inst. Phys., New York, 1992).
11. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw–Hill, New York, 1985).
12. J. J. Monaghan, *Comput. Phys. Rep.* **3,** 71 (1985).
13. J. U. Brackbill, "Alternative Approaches to the Modeling of Tracer Transport," in "Proceedings, WCRP Scientific Symposium on Global Tracer Transport Models, December 1990," Los Alamos Report LA-UR-90-4234 (unpublished).
14. J. D. Gibbons, *Nonparametric Statistical Inference* (McGraw-Hill, New York, 1971).
15. H. X. Vu and J. U. Brackbill, *Comput. Phys. Commun.* **69,** 253 (1992).
16. R. J. Procassini, C. K. Birdsall, and E. C. Morse, *Phys. Fluids B* **2,** 3191 (1990).
17. J. U. Brackbill, *J. Comput. Phys.* **108,** 38 (1993).
18. H. X. Vu, J. U. Brackbill, and D. Winske, *J. Geophys. Res.* **97,** 13839 (1992).
19. A. M. Winslow, Lawrence Livermore National Laboratory Report UCID-19062, 1981 (unpublished).
20. C. K. Birdsall and N. Maron, *J. Comput. Phys.* **36,** 1 (1980).
21. J. U. Brackbill and D. W. Forslund, *J. Comput. Phys.* **46,** 271 (1982).
22. H. Abe, N. Sakairi, R. Itatani, and H. Okuda, *J. Comput. Phys.* **63,** 247 (1986).
23. L. Chen, A. B. Langdon, and C. K. Birdsall, *J. Comput. Phys.* **14,** 200 (1974).
24. J. U. Brackbill and G. Lapenta, *J. Comput. Phys.* **114,** 77 (1994).