

# Particle Rezoning for Multidimensional Kinetic Particle-In-Cell Simulations

Giovanni Lapenta

*Plasma Theory Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545*  
E-mail: lapenta@lanl.gov

Received August 15, 2001; revised June 3, 2002

---

The adaptation of PIC methods requires the ability to change the number of particles during the calculation. For PIC methods it is not sufficient to adapt the computational grid. It is also necessary to control the local number of particles per cell (particle rezoning) by increasing or decreasing its value to control the local accuracy. In the present paper, we describe some general theoretical considerations regarding the accuracy of various particle rezoning methods. Four algorithms are derived and applied to 1D and 2D PIC simulations. The merits and drawbacks of the algorithms are discussed. Particle rezoning is then applied to 1D studies of collisionless shocks and the 2D simulations of charging of dust immersed in a plasma. © 2002 Elsevier Science (USA)

---

## 1. INTRODUCTION

Methods to adapt particle-in-cell (PIC) kinetic plasma calculations are very valuable in the study of multiple-length-scale problems. Typically, multiple-length-scale problems present small regions of stronger gradients embedded in large systems. Under such conditions, computational efficiency is achieved best by focusing attention on the regions of interest.

In PIC methods it is not sufficient to use adaptive grids with finer spacing in the regions of interest. It is also necessary to rezone the number of particles. Particle rezoning is defined as the operation of increasing the number of particles in regions where high accuracy is required, and of reducing the number of particles where lower accuracy can be tolerated. Finer grid spacing leads to a better description of the electromagnetic fields, but particle rezoning is needed to gain a better description of the plasma dynamics and a reduction of noise [1].

Particle rezoning is also useful for increasing selectively the accuracy in specific velocity ranges. For example, high-energy particles in collisionless shocks can play a role more important than their relative abundance would suggest [2]. In this case, the number of high-energy particles can be increased selectively.

In the present work, we extend previous methods [1] developed for 1D simulations to 2D and 3D PIC codes and we analyze their accuracy. In Section 2, we discuss the general properties of the techniques for particle rezoning, and we present methods to control the number of particles. In Sections 3 and 4, we discuss the conservation properties of the methods presented in Section 2. Section 5 summarizes the properties of the methods and gives a self-consistent description of four algorithms to increase or decrease the number of particles. Sections 6 and 7 describe, respectively, the application of particle rezoning to electromagnetic collisionless shocks and to the charging of a spherical dust particle immersed in a plasma. These applications are used to investigate the performance of the various algorithms presented here.

It is worth mentioning that Sections 3 and 4 are important in evaluating fully the merits and defects of the various algorithms. However, readers not interested in the derivation of the methods can find a self-consistent description of the algorithms in Section 5 and can skip Sections 3 and 4.

## 2. GENERAL CONSIDERATIONS

Particle rezoning is needed to increase the number of particles in regions where high accuracy is required, and to reduce the number of particles where lower accuracy can be tolerated. The primary effect of increasing the number of particles is to reduce the variance of the statistical description of the distribution function. In a PIC simulation this increases the accuracy defined as typical in Monte Carlo methods, i.e., as the variance of the simulation.

Particle rezoning must be in effect throughout the calculation to constantly keep the local required accuracy. In multiple-length-scale problems, the region of interest can move, and particle rezoning must follow the motion to keep the focus where it is needed. The approach followed here is to use adaptive grids to follow the evolution of the system [3, 4] and particle rezoning to keep the number of particles per cell constant. This approach leads to finer grid spacing in the region of interest and, automatically, to a higher density of computational particles in that region.

The problem of particle rezoning can be formulated as the replacement of a set of  $N$  particles with position  $\mathbf{x}_p$ , velocity  $\mathbf{v}_p$ , charge  $q_p$ , and mass  $m_p$ , with a different set of  $N'$  particles with position  $\mathbf{x}_{p'}$ , velocity  $\mathbf{v}_{p'}$ , charge  $q_{p'}$ , and mass  $m_{p'}$ . The criterion for replacement is the *equivalence* between the two sets, defined as the requirement that the two sets must represent the same physical system, with a different accuracy. This generic definition of equivalence between two sets is given practical bearing by specifying two rules for equivalence.

Two sets of particles are considered equivalent under the following conditions:

1. The two sets are indistinguishable on the basis of their contributions to the grid moments.
2. The two sets of particles sample the same velocity distribution function.

The first criterion concerns the moments of the particle distribution used to solve the field equations. The moments are defined at the grid points  $\mathbf{x}_g$  as

$$M_g = \sum_p S(\mathbf{x}_g - \mathbf{x}_p) q_p \mathbf{F}(\mathbf{v}_p), \quad (1)$$

where  $S$  is the assignment function [5, 6]. In general, when nonuniform grids are used,  $\mathbf{x}$  is

the natural coordinate, i.e., the system of coordinates where the spacing between consecutive points is uniform and unitary in all directions [3]. The function  $\mathbf{F}$  of the particle velocity characterizes the moment. In explicit electrostatic codes, only the charge density is required; i.e.,

$$\rho_g = \sum_p S_g(\mathbf{x}_p) q_p, \quad (2)$$

derived from (1) using  $\mathbf{F}(\mathbf{v}_p) = 1$  and using a short notation for  $S_g(\mathbf{x}_p) = S(\mathbf{x}_g - \mathbf{x}_p)$ . Electromagnetic and implicit codes [7] require higher order moments like the current density

$$\mathbf{J}_g = \sum_p S_g(\mathbf{x}_p) q_p \mathbf{v}_p \quad (3)$$

and the pressure tensor

$$\Pi_g = \sum_p S_g(\mathbf{x}_p) q_p \mathbf{v}_p \mathbf{v}_p. \quad (4)$$

The first criterion requires the two sets of particles to give the same moments relevant to the field equations. Note that if this criterion is satisfied exactly total energy and momentum are also automatically conserved. The second criterion is more difficult to apply in a quantitative fashion. In a previous work [1], it was proposed to use the  $\chi^2$  test or the Kolmogorov and Smirnov test to verify that the particle distribution is preserved. In practice, this is not easily achieved.

In fluid PIC codes, the first criterion is the only one to be applied, and general schemes for particle rezoning can be derived [8]. In kinetic PIC codes, the computational particles sample the real plasma velocity distribution, and the second criterion must also be imposed. In the kinetic case the choices are more limited. For this reason, a simpler approach is followed [1]. To increase the number of particles per cell, a given particle is split into two or more new particles displaced in space but all sharing the same speed. The weights and displacements can be chosen to conserve exactly the grid moments, and the velocity distribution is not altered because all the particles have the same velocity.

Another approach can be considered [2]. A particle can be split in the velocity space. The daughter particles have the same position but different velocity. The advantage of this method is that the charge density is not affected. However, the higher order moments (current density and energy) cannot be all preserved. Furthermore, the velocity distribution is altered.

To decrease the number of particles, the splitting operation can be inverted to coalesce two particles into one. The difficulty is that, in general, it is impossible to find two particles with the same velocity. For this reason, particles with different velocity have to be coalesced. To minimize the perturbation of the velocity distribution, the particles to be coalesced must be chosen with similar velocity. An alternative approach is to coalesce three particles into two, which allows one to conserve both energy and momentum [9].

In the following sections, we will provide general techniques to analyze the different methods introduced above and to enforce the two conservation laws introduced above.

### 3. CONSERVATION OF THE GRID MOMENTS

The first condition for particle rezoning methods is to conserve the grid moments, like charge or current density, defined in Eq. (1). When a set  $\mathcal{S}_N$  with  $N$  particles (labeled  $p$ ) is replaced with a set  $\mathcal{S}_{N'}$  with  $N'$  particles (labeled  $p'$ ), the condition is

$$\sum_{p \in \mathcal{S}_N} \mathbf{F}(\mathbf{v}_p) S_g(\mathbf{x}_p) q_p = \sum_{p' \in \mathcal{S}_{N'}} \mathbf{F}(\mathbf{v}'_{p'}) S_g(\mathbf{x}_{p'}) q_{p'}, \quad (5)$$

where  $\mathbf{x}$  is the natural coordinate that maps the physical grid (in general assumed to be nonuniform but logically rectangular) into a uniform grid.

As discussed above, it is very difficult to impose (5) when  $\mathbf{v}_p$  and  $\mathbf{v}_{p'}$  are unspecified. For this reason, in the method considered here all the particles in set  $\mathcal{S}_N$  and in set  $\mathcal{S}_{N'}$  are assumed to have the same velocity:  $\mathbf{v}_p = \mathbf{v}_{p'} = \mathbf{v}_o$  (exactly for splitting methods and approximately for coalescences). Under this hypothesis, all the moments on the grid are preserved if

$$\sum_{p \in \mathcal{S}_N} \epsilon_p S_g(\mathbf{x}_p) = \sum_{p' \in \mathcal{S}_{N'}} S_g(\mathbf{x}_{p'}) \epsilon_{p'}, \quad (6)$$

where  $\epsilon_p$  ( $\epsilon_{p'}$ ) is the fraction of the total charge in  $\mathcal{S}_N$  ( $\mathcal{S}_{N'}$ ) allotted to particle  $p$  ( $p'$ ); i.e.,  $q_p = Q \epsilon_p$  where  $Q = \sum_p q_p$  is conserved in the exchange between sets.

A simple study of the conditions required for the satisfaction of Eq. (6) can be developed if the function  $S_g(\mathbf{x}_p)$  is expanded in Taylor series,

$$S_g(\mathbf{x}_p) = S_g(\mathbf{x}_o) + \sum_{\alpha} \left. \frac{\partial S_g}{\partial x_{\alpha}} \right|_{\mathbf{x}_o} \Delta x_p^{\alpha} + \frac{1}{2} \sum_{\alpha\beta} \frac{\partial^2 S}{\partial x_{\alpha} \partial x_{\beta}} \Delta x_p^{\alpha} \Delta x_p^{\beta} + \dots, \quad (7)$$

where  $x_p^{\alpha}$  is the  $\alpha$ -component of  $\mathbf{x}_p$ ,  $\Delta x_p^{\alpha} = x_p^{\alpha} - x_o^{\alpha}$ , and  $\mathbf{x}_o$  is chosen for convenience.

Using Eq. (7), Eq. (6) can be reformulated, imposing the equality of each term of the series expansion for the two sets,

$$\sum_p \epsilon_p = \sum_{p'} \epsilon_{p'}, \quad (8)$$

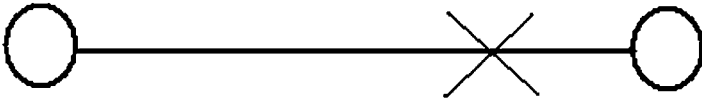
$$\sum_p \epsilon_p \Delta x_p^{\alpha} = \sum_{p'} \epsilon_{p'} \Delta x_{p'}^{\alpha}, \quad (9)$$

$$\sum_p \epsilon_p \Delta x_p^{\alpha} \Delta x_p^{\beta} = \sum_{p'} \epsilon_{p'} \Delta x_{p'}^{\alpha} \Delta x_{p'}^{\beta}, \quad (10)$$

for  $\alpha, \beta = 1, 2, 3$ , where the set is truncated at the second order.

Equations (8)–(10) are relative to the zeroth, first, and second order; higher order equations are similar. Equations (8)–(10) can be used to determine the properties of the new set  $\mathcal{S}_{N'}$ . A few comments on Eqs. (8)–(10) are in order.

First, the assignment function  $S_g(\mathbf{x}_p)$  is usually a  $b$ -spline of order  $\ell$ :  $S_g(\mathbf{x}_p) = b_{\ell}(x_p - x_g) b_{\ell}(y_p - y_g) b_{\ell}(z_p - z_g)$ ;  $b_{\ell}$  has only  $\ell - 1$  continuous derivatives, and all derivatives of order  $m > \ell$  vanish. Therefore, the series expansion (7) is valid everywhere only to order



**FIG. 1.** Particle rezoning in 1D. One particle (cross) is split in space or two particles (circles) are coalesced.

$(\ell - 1)$ , and to order  $\ell$  in the region where the  $\ell$ th derivative is continuous. For example,

$$b_1(\xi) = \begin{cases} 1 - |\xi| & \text{if } |\xi| < 1, \\ 0 & \text{otherwise,} \end{cases}$$

where  $\xi = x_p^\alpha - x_g^\alpha$ .  $b_1$  is continuous everywhere but its derivative is discontinuous in the cell center. As a consequence, Eq. (8) is always valid, but Eq. (9) is valid only if all the particles are in the same half of a cell [1].

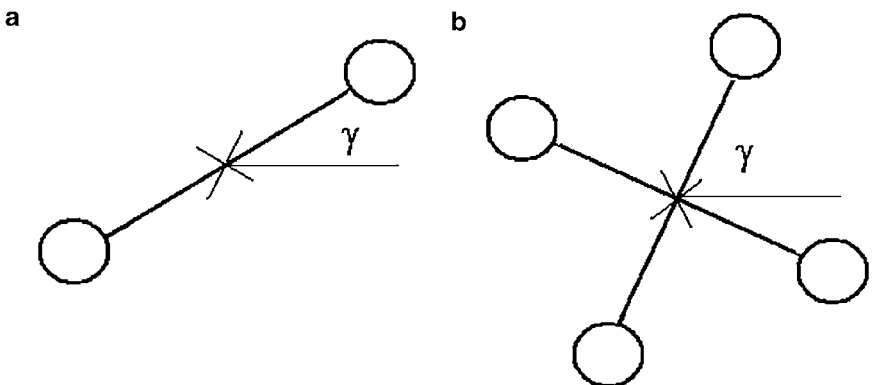
Second, the equations to be considered are only up to the order of the  $b$ -spline. For example, in case of  $\ell = 1$ , Eqs. (8) and (9) need to be considered, but Eq. (10) is relevant only for  $\alpha \neq \beta$ .

Finally, if, for a given order of interpolation  $\ell$ , only  $m < \ell$  equations such as Eqs. (8)–(10) are considered, the first truncated order will give an estimation of the error.

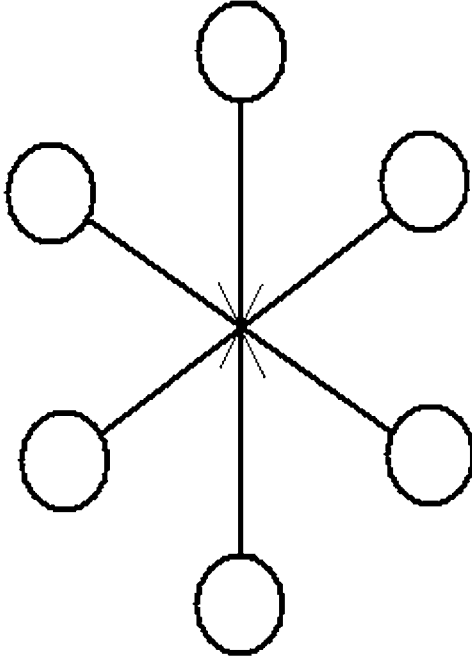
Equations (8)–(10) will now be used to derive various particle control schemes in one, two, and three dimensions.

### 3.1. Binary Schemes

Binary schemes are based on splitting one given particle in  $N$  new particles displaced in space. The inverse operation is used to coalesce particles. In 1D, two new particles can be generated, displaced in opposite directions (equally or unequally) from the parent particle (see Fig. 1). In multidimensional problems, the same approach can be used, splitting the particle in two, along any given direction (see Fig. 2a). A more symmetric scheme is to split one particle in four (in 2D), or six (in 3D), in an orthogonal configuration (see Figs. 2b



**FIG. 2.** Particle rezoning in 2D. One particle (cross) can be split into two (circles) along any direction on the plane (a) or it can be split into four (circles) in an orthogonal configuration directed along any direction (b). The angle  $\gamma$  is with respect to the horizontal axis.



**FIG. 3.** *Particle rezoning in 3D.* The 2D configurations in Fig. 2 can be used in 3D as well. In 3D one can also consider an orthogonal configuration to split one particle (cross) into six (circles).

and 3). To derive the new particle properties, Eqs. (8) and (9) are used:

$$\begin{cases} \sum_{p'} \epsilon_{p'} = 1 \\ \sum_{p'} \epsilon_{p'} \Delta x_{p'}^\alpha = 0. \end{cases} \quad (11)$$

These are derived from Eqs. (8) and (9), recalling that  $\mathcal{S}_N$  has just one particle and setting  $x_o = x_p$ .

Infinite solutions exist. A natural choice is to select the new particles to have the same weights  $\epsilon_{p'} = 1/N_{p'}$ . In that case, Eqs. (11) are satisfied if all particles have the same displacement (in absolute value but opposite in sign). As noted above, Eqs. (11) give exact conservation of the grid moments only if two conditions are satisfied.

First, all the particles should be in the domain where the series expansion is valid, which is the same half of the cell in 1D (quarter in 2D and octant in 3D).

Second, higher order equations are not considered, and an error is introduced. In 1D, no error is present for interpolation orders  $\ell \leq 1$ , but an error is present for  $\ell \geq 2$ . An error estimate is given by the first truncated order:

$$\delta S_g = \frac{\partial^2 S_g}{\partial x^2} \left( \sum_{p'} \epsilon_{p'} \Delta^2 x_{p'} \right). \quad (12)$$

The error is of the second order in  $\Delta x_{p'}$ , and it does not depend on  $x_o$  ( $\partial^2 S_g / \partial x^2$  is a

constant). In 2D and 3D, the error is present for  $\ell \geq 1$  and can be estimated as

$$\delta S_g = \frac{1}{2} \sum_{\alpha\beta} \frac{\partial^2 S_g}{\partial x^\alpha \partial x^\beta} \Big|_{x_o} \left( \sum_{p'} \epsilon_{p'} \Delta x_{p'}^\alpha \Delta x_{p'}^\beta \right). \quad (13)$$

For  $\ell = 1$ , only  $\alpha \neq \beta$  is present and the error vanishes when the sum in Eq. (13) is zero. For example in 2D, the scheme in Fig. 2a has zero error if the angle with the horizontal axis is  $\gamma = k\frac{\pi}{2}$  (with integer  $k$ ), while the scheme in Fig. 2b always has zero error. For  $\ell = 2$ , an error is always present and is again of the second order.

It is worth noting that the error for  $\ell = 2$  cannot be removed by considering the equation for the second order, Eq. (10). In fact, for  $\alpha = \beta$ , Eq. (10) becomes

$$\sum_{p'} \epsilon_{p'} (\Delta x_{p'}^\alpha)^2 = 0, \quad (14)$$

which clearly cannot be satisfied for  $\Delta x_p^\alpha \neq 0$  and  $\epsilon'_p > 0$ . This result was obtained also in Ref. [1], using geometric arguments.

### 3.2. Ternary Schemes

Ternary schemes remove the impossibility expressed in Eq. (14) to obtain exact rezoning schemes for  $\ell = 2$ . These schemes involve the coalescence of three particles into two. The inverse operation (splitting) is also possible. The scheme can be worked out in 1D, giving exact conservation. Choosing  $x_o$  as the center of mass, the equations to be satisfied are

$$\sum_{p'} \epsilon_{p'} = 1, \quad (15)$$

$$\sum_{p'} \epsilon_{p'} \Delta x_{p'} = 0, \quad (16)$$

$$\sum_{p'} \epsilon_{p'} (\Delta x_{p'})^2 = M_{xx}, \quad (17)$$

where  $M_{xx} = \sum_p \epsilon_p (\Delta x_p)^2$ . Equation (16) follows from Eq. (9), using  $x_o$  as the baricenter.

It is easy to verify that Eqs. (15)–(17) are all satisfied if the two new particles have equal weights  $1/2$  and are displaced by  $\delta = \sqrt{M_{xx}}$  in the opposite sides of the baricenter. The method has no error for interpolation orders  $\ell \leq 2$ , provided the particles are on the same half of the cell (for interpolation of order  $\ell = 1$ ). The method cannot be extended to 2D. Indeed, for a system of two particles, there is a simple relationship between  $M_{\alpha\beta} = \sum_p \epsilon_p \Delta x_p^\alpha \Delta x_p^\beta$ ,

$$M_{xx} M_{yy} = (M_{xy})^2, \quad (18)$$

which, in general, is not satisfied by a system of three particles. Only approximate ternary schemes can be derived for coalescences in 2D or 3D. More-complex schemes are required to preserve the moments exactly when more dimensions are involved and when higher order interpolation schemes are used.

## 4. CONSERVATION OF THE VELOCITY DISTRIBUTION FUNCTION

The second condition for particle rezoning methods is to conserve the local velocity distribution function  $f(v)$ . This requirement is difficult to impose and verify. In a previous work, the  $\chi^2$  and Kolmogorov and Smirnov tests [1] were applied to verify the preservation of velocity distributions. However, the simplest approach is to replace sets of particles with the same velocity. Evidently, the replacement of particles with velocity  $v_o$  and cumulative statistical weight  $w_o$  with new particles with the same velocity  $v_o$  and cumulative weight  $w_o$  (but a different number of particles) does not alter the distribution. This requirement determines where binary and ternary schemes ought to be used.

### 4.1. Binary Schemes

When a particle is split in  $N$  particles displaced in space, the conservation of  $f(v)$  is exact. However, if the alternative approach to displace the  $N$  new particles in the velocity space and not in the coordinate space [2] is used the velocity distribution is altered. In Section 7, it will be shown that this can lead to a failure of the splitting in velocity. The splitting in space is to be preferred.

When two particles are coalesced, it is not possible, in general, to find two particles displaced in space and not in velocity. For this reason, one has to select in a cell two particles close in velocity and space. As a matter of fact, the two selected particles will, in general, differ in velocity, introducing a slight perturbation of  $f(v)$ .

The new particle velocity can be selected in order to preserve the overall momentum or energy. However, it is not possible to preserve both; i.e.,

$$\begin{aligned} m_1 v_{1\alpha} + m_2 v_{2\alpha} &= m_A v_{A\alpha}, \\ m_1 v_1^2 + m_2 v_2^2 &= m_A v_A^2, \end{aligned} \quad (19)$$

where 1 and 2 label the old particles and  $A$  the new coalesced particle, and where  $\alpha$  labels the components of the velocity ( $x$ ,  $y$ , or  $z$ ). The error introduced by coalescences is, in loose terms, less relevant than a corresponding error generated by splitting, as the regions where splitting is performed are the regions of maximum interest, where the highest accuracy is desired.

### 4.2. Ternary Schemes

As discussed above, this method deals with coalescing three particles into two. Evidently, this method complicates the search of particles with similar velocities when compared to binary schemes. Moreover, the three selected particles have a larger spreading than the two particles selected in binary coalescences, increasing the perturbation of the velocity distribution. However, this method has the advantage of allowing preservation of both overall momentum and energy; i.e.,

$$\begin{aligned} m_1 v_{1\alpha} + m_2 v_{2\alpha} + m_3 v_{3\alpha} &= m_A v_{A\alpha} + m_B v_{B\alpha}, \\ m_1 v_{1\alpha}^2 + m_2 v_{2\alpha}^2 + m_3 v_{3\alpha}^2 &= m_A v_{A\alpha}^2 + m_B v_{B\alpha}^2, \end{aligned} \quad (20)$$

where 1, 2, and 3 label the old particles, and  $A$  and  $B$  label the new particles. Note that strictly only total energy needs to be conserved, but we choose to preserve energy in each



direction to obtain a closed system. Equations (20) can be satisfied simultaneously; i.e.,

$$v_{A,B\alpha} = \frac{\mathcal{M}_\alpha \pm \sqrt{\mathcal{E}_\alpha - \mathcal{M}_\alpha^2}}{m_A + m_B}, \quad (21)$$

where  $\mathcal{M}_\alpha$  is the total momentum (in the direction  $\alpha$ ) and  $\mathcal{E}_\alpha$  the total energy in the direction  $\alpha$ . Note that the masses  $m_A$  and  $m_B$  were derived in the previous section on the basis of the conservation of grid moments. The ternary scheme is not suitable for splitting because it introduces a perturbation of the velocity distribution while the binary scheme is exact.

## 5. SUMMARY OF ALGORITHMS FOR PARTICLE REZONING

In the previous sections, we derived all the required blocks to build algorithms to change the number of particles in any given cell. In the following, we provide a precise algorithmic description of two methods to increase the number of particles per cell and of two methods to decrease the number of particles per cell.

**ALGORITHM S1.** Given a cell  $g$  with  $N_p$  particles in a 1D, 2D, or 3D system, any chosen particle (labeled  $o$ ) with charge  $q_o$  (and mass obtained from the charge-to-mass ratio for the species), position  $\mathbf{x}_o$  (in natural coordinates), and velocity  $\mathbf{v}_o$  can be replaced by  $N'$  particles, labeled  $p' = \{1, 2, \dots, N'\}$ . In 1D,  $N' = 2$  and the new properties are  $q_{p'} = q_o/2$ ,  $x_{p'} = x_o \pm 1/N_p$  (where the cell size is unitary),  $\mathbf{v}_{p'} = \mathbf{v}_o$ . In 2D,  $N' = 4$  and the new properties are  $q_{p'} = q_o/4$ ;  $x_{1,2} = x_o \pm 1/N_p$ ,  $x_{3,4} = x_o$ ;  $y_{1,2} = y_o$ ,  $y_{3,4} = y_o \pm 1/N_p$ ;  $\mathbf{v}_{p'} = \mathbf{v}_o$ . In 3D,  $N' = 6$  and the new properties are  $q_{p'} = q_o/6$ ;  $x_{1,2} = x_o \pm 1/N_p$ ,  $x_{3,\dots,6} = x_o$ ;  $y_{1,2,5,6} = y_o$ ,  $y_{3,4} = y_o \pm 1/N_p$ ;  $z_{1,\dots,4} = z_o$ ,  $z_{5,6} = z_o \pm 1/N_p$ .

Note that the choice of the particle  $p = o$  in the set of  $N_p$  particles in the cell  $g$  is free. In the result sections, we choose the particle with the largest energy:  $m_p \mathbf{v}_p^2$ . Algorithm S1 preserves exactly the velocity distribution function and grid moments. However, for quadratic assignment functions the grid moments are only approximately preserved (see Section 3).

**ALGORITHM S2.** Given a cell  $g$  with  $N_p$  particles in a 1D, 2D, or 3D system, any chosen particle (labeled  $o$ ) with charge  $q_o$  (and mass obtained from the charge-to-mass ratio of the species), position  $\mathbf{x}_o$  (in natural coordinates), and velocity  $\mathbf{v}_o$  can be replaced by  $N' = 2$  new particles, labeled  $p' = \{1, 2\}$ . The properties of the new particles are  $q_{p'} = q_o/2$ ,  $\mathbf{x}_{p'} = \mathbf{x}_o$ , and  $\mathbf{v}_{p'} = \mathbf{v}_o \pm \delta \mathbf{v}$ .

Note that, as in S1, the choice particle  $o$  is free. Algorithm S2 preserves exactly the grid charge density  $\rho_g$  (for any choice of the assignment functions), but the velocity distribution (and consequently the higher order moments on the grid) are perturbed. The direction of  $\delta \mathbf{v}$  is free.

**ALGORITHM C1.** Given a cell  $g$  with  $N_p$  particles in 1D, 2D, or 3D systems, choose  $N = 2$  particles,  $p = \{1, 2\}$ , close to each other in the phase space. Their properties are  $q_p$ ,  $\mathbf{x}_p$ , and  $\mathbf{v}_p$ . The two chosen particles can be replaced by one particle (labeled  $A$ ) with  $q_A = q_1 + q_2$ ,  $\mathbf{x}_A = (q_1 \mathbf{x}_1 + q_2 \mathbf{x}_2)/q_A$ ,  $\mathbf{v}_A = (q_1 \mathbf{v}_1 + q_2 \mathbf{v}_2)/q_A$ .

Algorithm C1 preserves the overall charge and momentum and the charge density  $\rho_g$  but perturbs the velocity distribution. Note that one can choose  $\mathbf{v}_A$  to preserve the energy, but it is not possible to preserve energy and momentum together. The crucial point of algorithm C1

is to choose two particles close in velocity and space. A pair search of the two particles closest in velocity is usually too expensive. For this reason, we perform a diatomic search that sorts the particles into two bins and selects the largest bin. The binning is repeated in sequence for each spatial direction and component of the velocity. The binning is continued until the number of particles in the largest bin is small enough to use a pair search.

**ALGORITHM C2.** Given a cell  $g$  with  $N_p$  particles in a 1D system, choose  $N = 3$  particles,  $p = \{1, 2, 3\}$ , close to each other in the phase space. Their properties are  $q_p, \mathbf{x}_p, \mathbf{v}_p$ . The chosen particles can be replaced by two particles (labeled  $A, B$ ) with  $q_{A,B} = \frac{1}{2}Q$ ,  $x_{A,B} = x_{cm} \pm \sqrt{M_{xx}}$ ,  $v_{A,B,\alpha} = \mathcal{M}_\alpha \pm \sqrt{\mathcal{E}_\alpha - \mathcal{M}_\alpha^2}$ , where  $Q = \sum_p q_p$ ,  $M_{xx} = \sum_p q_p (x_p - x_{cm})^2$ ,  $x_{cm}$  is the baricenter of the original set,  $\mathcal{M}_\alpha = \sum_p Q_p v_{p\alpha}$ , and  $\mathcal{E}_\alpha = \sum_p q_p v_{p\alpha}^2$ . The index  $\alpha$  labels the components of the velocity.

Algorithm C2 preserves the overall charge, momentum, and energy. Algorithm C2 preserves also  $\rho_g$  but neither the velocity distribution nor the higher order moments on the grid. Algorithm C2 is not directly extended to 2D and 3D. The choice of the particles to coalesce is similar to C1, except that the three closest particles are chosen in the final bin. In the following, algorithms C1, C2, S1, and S2 will be tested in 1D and 2D problems to evaluate their performance.

## 6. 1D—COLLISIONLESS SHOCKS

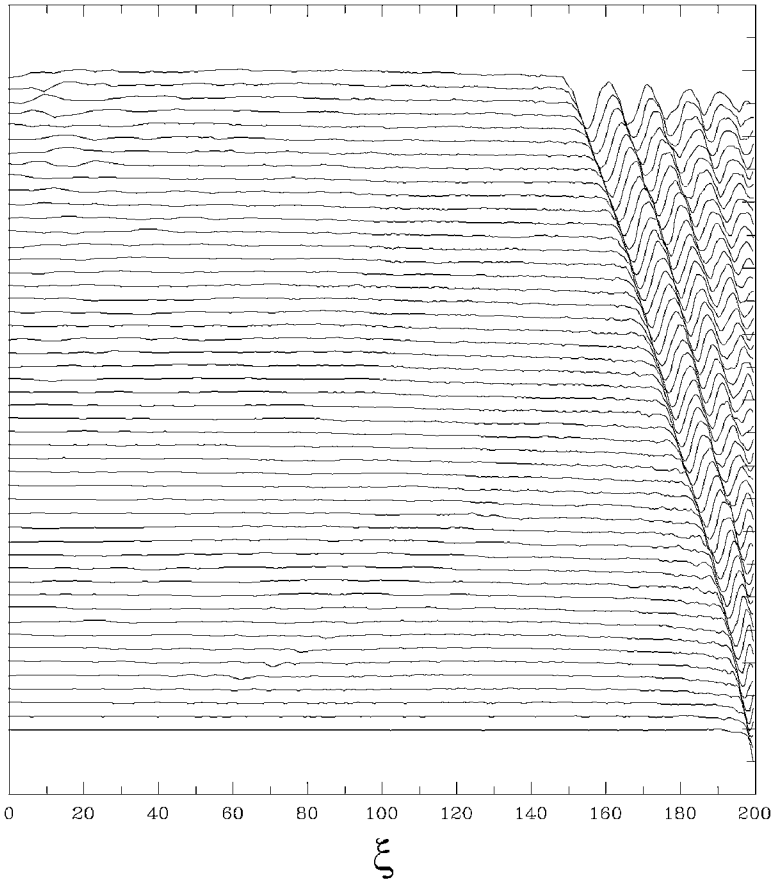
Simulations of collisionless shocks provide a sensitive test of the accuracy of particle rezoning methods [1]. In the slow shock calculations considered here, a magnetized plasma is flowing toward a piston that reflects the particles. A switch off slow shock is considered, and the component of the magnetic field perpendicular to the normal of the piston is set to zero.

We consider here the same conditions reported by Brackbill and Vu [10]. The initial configuration is chosen according to the Rankine–Hugoniot conditions. The initial 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 ratio of the upstream ion cyclotron and ion plasma frequencies is  $\omega_{ci}/\omega_{pi} = 0.01$ , and the shock normal angle, with respect to the magnetic field, is  $\psi = 75^\circ$ . The size of the simulation region is  $L = 200 c/\omega_{pi}$ , and the shock is followed until  $\omega_{ci}t = 50$ . Particles are injected at the right boundary to simulate a flowing plasma.

The simulations are performed using CELESTE1D [11], a 1D implicit PIC code, suitably modified by the author to include particle control.

As a reference, we repeated the calculation in [10] with a uniform grid and without particle rezoning. The grid has 400 cells giving a uniform spacing with  $\Delta x = 0.5 c/\omega_{pi}$ ; 256 electrons and 64 ions per cell are used. Figure 4 shows the stack plot of  $B_z$  as a function of the position at 50 equally spaced time intervals between  $t = 0$  and  $\omega_{ci}t = 50$ .

The reference results are compared with a calculation where particle rezoning is performed using Algorithm S1 for splitting and C1 for coalescences. The computation uses an adaptive grid with finer spacing in the shock region ( $\Delta x \approx 0.5 c/\omega_{pi}$ ) and coarser outside. The region of fine spacing expands in time to follow the motion of the shock. The grid spacing in the region of the shock is kept fixed, and, consequently, the grid spacing in the coarser region grows to keep the number of grid points constant and equal to 300. Figure 5

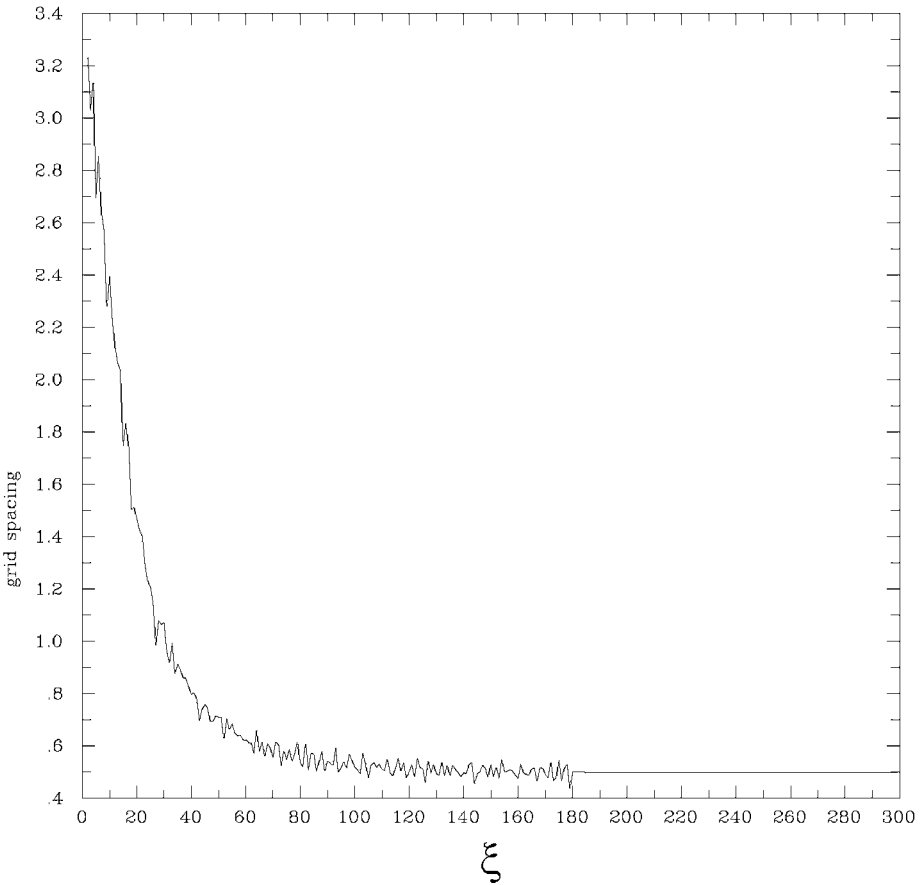


**FIG. 4.** *Slow shock.* Reference calculation with a uniform grid with 400 points. Stack plot of  $B_z$  at 50 equally spaced times from 0 to  $50 \omega_{ci}^{-1}$ . Space is normalized as  $\xi = x \omega_{pi} / c$ . The spatial profile of  $B_z$  is plotted at 50 equally spaced times between  $\omega_{ci} t = 1$  and 50.

shows the grid spacing at the end of the simulation. Fewer particles are used: 128 electrons and 32 ions. The particles are loaded with a uniform number per cell, leading to higher accuracy where the grid is finer. Particle rezoning is required to keep the uniformity of the number of particles per cell as the grid is adapted.

Figure 6 shows the stack plot of  $B_z$  for the calculation described above. Clearly, the evolution of the system is calculated correctly. In particular, the shock has traveled backward along the axis for a length of  $50 c / \omega_{pi}$ , as in the reference case (Fig. 4) and as required by the Rankine–Hugoniot conditions.

Figure 7 shows the number of particles in the calculation as a function of time. Particles are continuously injected at the right boundary to simulate a flowing plasma. Particle rezoning is able to coalesce enough particles to keep the number of particles almost constant. For comparison, if the calculation is conducted without rezoning, the number of particles increases by 40%. Furthermore, particle rezoning maintains the required local accuracy. Figure 8 shows the number of particles in each cell at the end of the simulation. The number is fairly constant, except in the first few cells. Algorithm C1 is applied in every cell where the target number of 128 electrons and 32 ions is exceeded, but only once per time step.

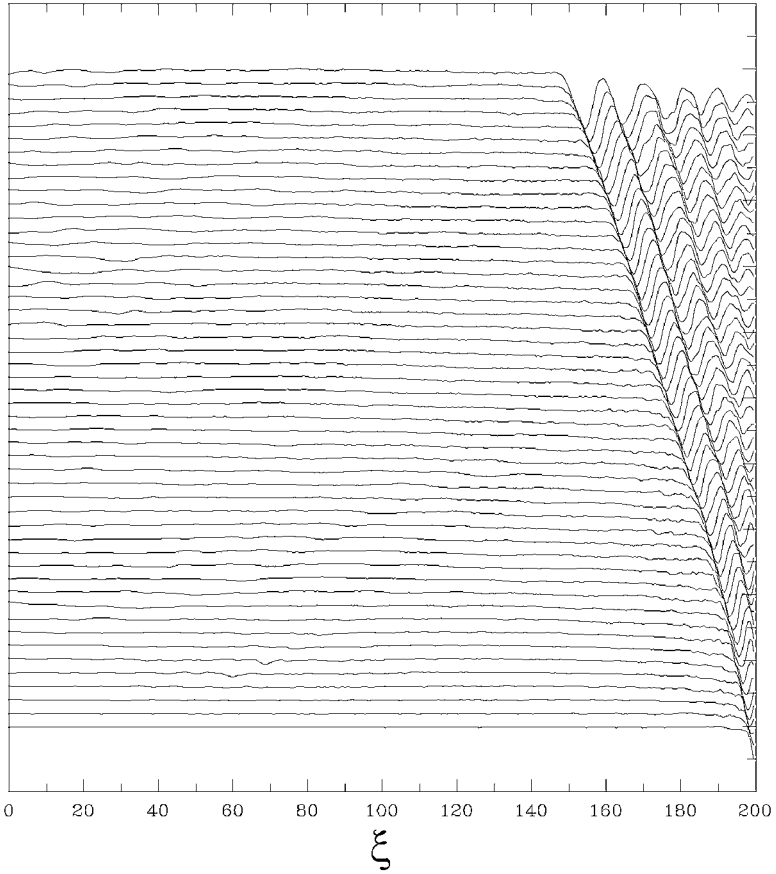


**FIG. 5.** *Slow shock.* Adaptive grid used in a calculation with rezoning. The grid spacing at the end of the simulation is shown. Space is normalized as  $\xi = x\omega_{pi}/c$ .

Evidently, in the leftmost cells the rate of growth of the grid spacing and of injection of particles would require a more aggressive strategy to keep the number of particles constant. However, even in the leftmost cells, the number of particles is still manageable, and, as noted above, the total number of particles in the system is essentially constant. However, a more aggressive coalesce strategy in the leftmost cells would lead to a modification of the distribution function of the injected particles and it is therefore not advisable.

The continuous injection of particles and the rightward flow makes the simulation of slow shocks barely suitable for testing the splitting algorithms and much more sensitive to the choice of the method to coalesce particles. We compared the results above with a simulation where Algorithm C2 is used instead of C1, all else being identical. Figure 9 shows the stack plot of  $B_z$  for such calculation. Clearly, the velocity of the shock is now altered.

For this particular case, Algorithm C2 fails where C1 was successful. The situation is paradoxical only at first sight. Algorithm C2 has better conservation properties as far as the moments are considered; it preserves both energy and momentum while C1 can only preserve one. However, as noted above, Algorithm C1 preserves the velocity distribution better. The comparison of Fig. 9 with Fig. 6 shows that, in this case, the preservation of some particular moments of the velocity distribution (energy and momentum) is less relevant than

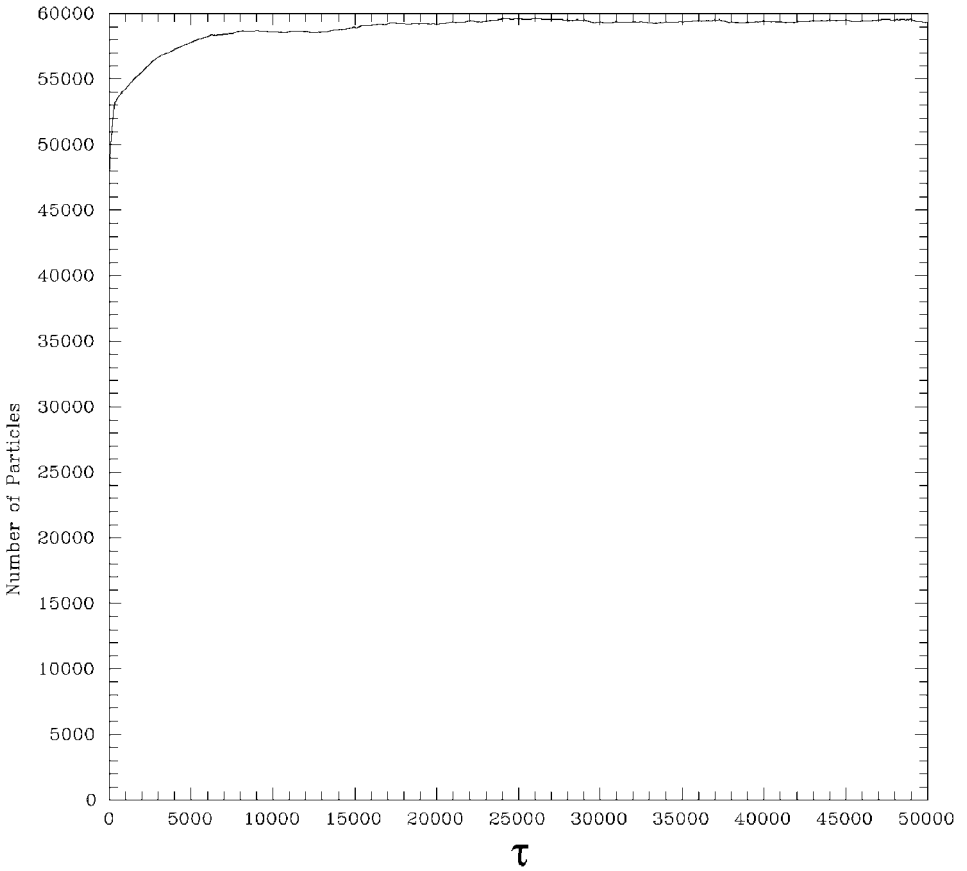


**FIG. 6.** *Slow shock.* Stack plot of  $B_z$  for a calculation with particle rezoning, using algorithms S1 and C1. Space is normalized as  $\xi = x\omega_{pi}/c$ . The spatial profile of  $B_z$  is plotted at 50 equally spaced times between  $\omega_{ci}t = 1$  and 50.

preserving the velocity distribution in a global sense. This is hardly a surprise, as kinetic systems, unlike fluid models, are concerned with the shape of the velocity distribution and not merely with its first few moments. However, in certain applications involving collisional systems, ternary schemes can prove useful in representing collision processes [9].

## 7. 2D—CHARGING OF DUST PARTICLES

Dust particles immersed in plasmas tend to acquire a negative charge. The ions and electrons of the plasma reach the surface and stick to it. If no secondary emission or photoemission is present, the equilibrium charge on the dust particle must be negative to repel the more mobile electrons and attract the ions to achieve a balance of electron and ion currents. This problem is of interest in laboratory and in space plasmas [12, 13]. We consider here the case where a plasma with an ion-to-electron-temperature ratio  $T_i/T_e = 0.05$  and ion-to-electron-mass ratio  $m_i/m_e = 1836$  is drifting relative to a spherical dust particle of radius  $a/\lambda_{De} = 0.4$ , where  $\lambda_{De}$  is the electron Debye length. The relative velocity  $w$  is expressed by the Mach number  $M = wm_i^{1/2}/(kT_e)^{1/2} = 10$ . The system is simulated using



**FIG. 7.** *Slow shock.* History of the total number of particles in the calculations reported in Fig. 6. Time is normalized as  $\tau = \omega_{pi}t$ . The initial number of particles is 48,000.

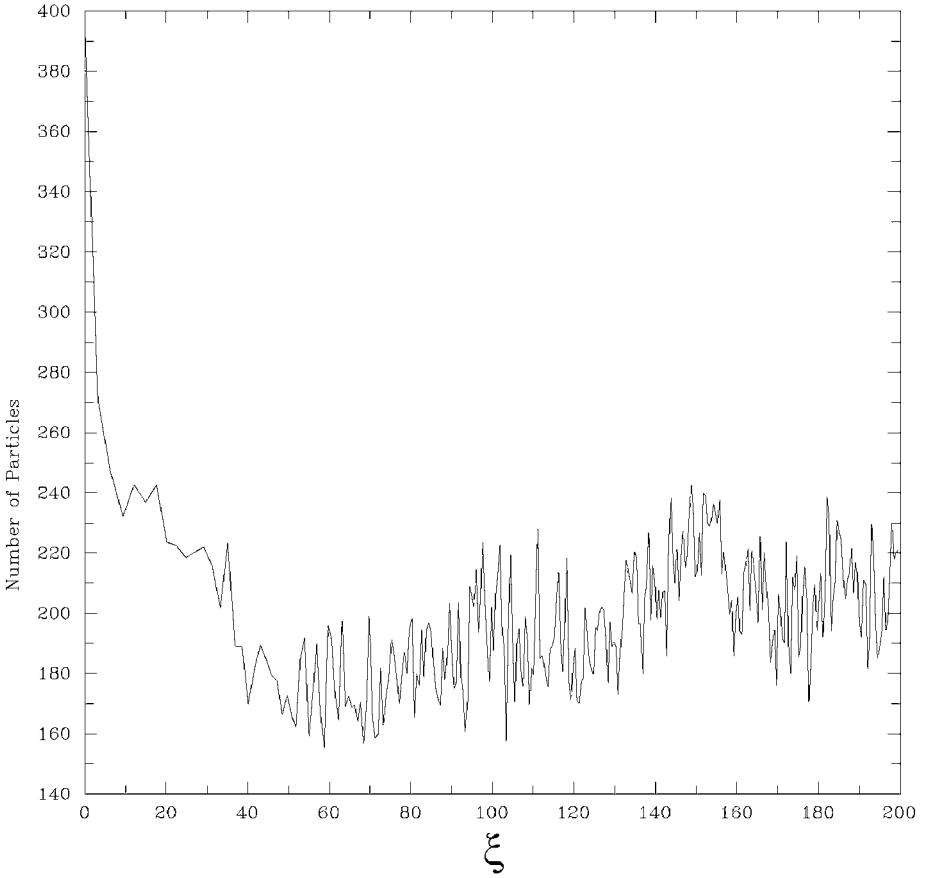
a cylindrical coordinate system with the vertical axis along the direction of the plasma flow and centered in the center of the spherical dust particle. In this configuration, the azimuthal coordinate is invariant, and the problem is 2D axisymmetric.

The interaction of the dust particle with the plasma is described with the immersed boundary method. The application of the immersed boundary method in PIC codes is described in Ref. [14] for fluid problems and in Ref. [15] for plasmas. In the present work, we will use the immersed boundary explicit PIC code DEMOCRITUS developed by the author for dusty plasma simulations [12]. A brief description of the method is given below; more details can be found in Ref. [12].

The dust particle is represented by motionless computational particles (object particles) with properties suitable to describe the macroscopic properties of the dust. Dust plasma interface conditions are treated with the immersed boundary method in two steps.

First, we assign to the object particles a susceptibility  $\chi_p$  that can be interpolated to the vertices of the grid  $\mathbf{x}_v$  to obtain a grid susceptibility,

$$\chi_v = \sum_p S_{vp} \chi_p, \quad (22)$$



**FIG. 8.** *Slow shock.* Number of particles in each cell at the end of the calculation reported in Fig. 6. Space is normalized as  $\xi = x\omega_{pi}/c$ .

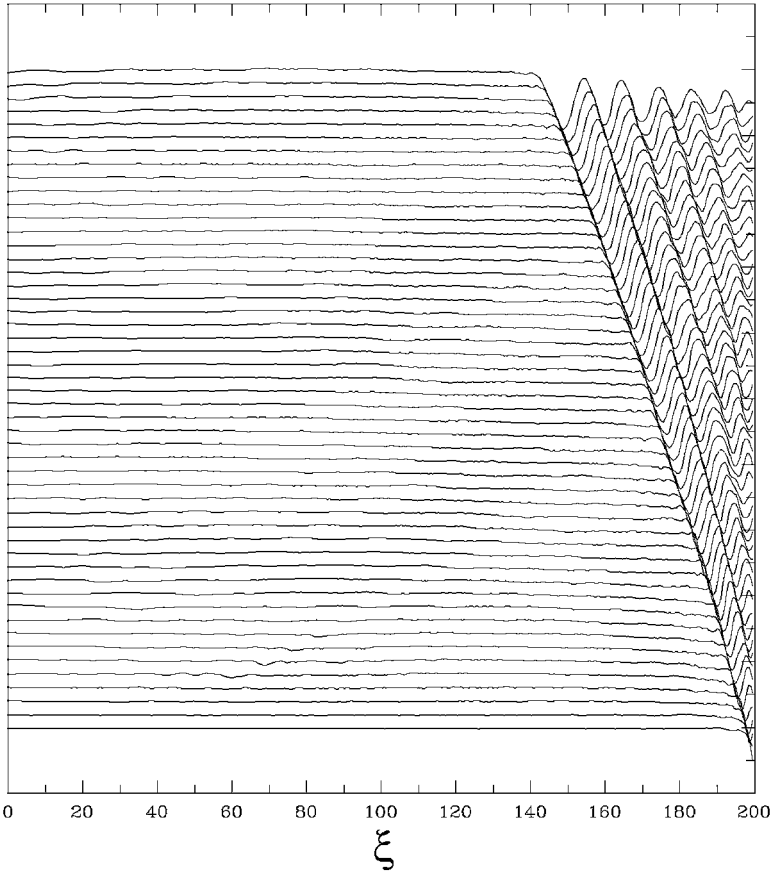
where  $S_{vp}$  are the linear assignment weights. The grid susceptibility is used to alter the Poisson's equation,

$$\mathcal{D}_{cv}(1 + \chi_v)\mathcal{G}_{vc'}\phi_{c'} = \rho_c, \quad (23)$$

where the potential  $\phi$  and the charge density  $\rho$  are defined on the cell centers  $\mathbf{x}_c$  and repeated indexes are summed. The operators  $\mathcal{D}_{cv}$  and  $\mathcal{G}_{vc}$  are a difference approximation of the divergence and gradient, respectively. As discussed in detail elsewhere [14, 15], Eq. (23) is solved everywhere, including in the interior of the dust particle. The term  $(1 + \chi_v)$  gives an approximation to the correct interface conditions for the electric field. In the present case,  $\chi_v$  is the susceptibility of dielectric dust.

Second, the object particles exert a friction on the plasma particles, via a slowing property  $\mu_p$  that is interpolated to the grid, as in Eq. (22), to produce a grid quantity  $\mu_v$  used to introduce a damping term to the equation of motion of the plasma particles

$$\frac{d\mathbf{v}_p}{dt} = \sum_v \mathbf{E}_v S_{vp} - \mathbf{v}_p \sum_v S_{vp} \mu_v. \quad (24)$$



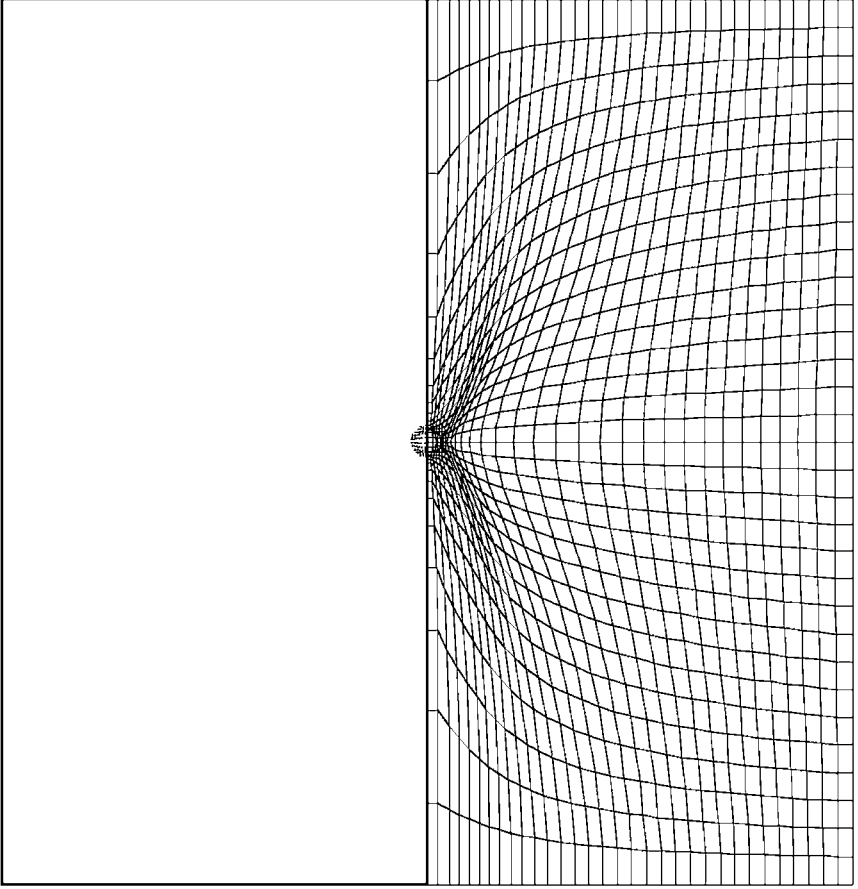
**FIG. 9.** *Slow shock.* Stack plot of  $B_z$  for a calculation similar to the one shown in Fig. 6, but with algorithm C2 instead of C1. Space is normalized as  $\xi = x\omega_{pi}/c$ . The spatial profile of  $B_z$  is plotted at 50 equally spaced times between  $\omega_{ci}t = 1$  and 50.

The second term in Eq. (24) can be as big as desired to stop the plasma particles on the surface of the dust. The damping term is zero everywhere outside the region occupied by the dust. Equations (23) and (24) allow one to treat the field and particle boundary conditions on the surface of the dust.

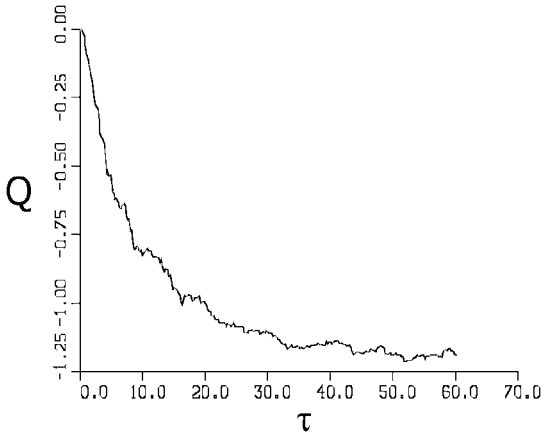
Figure 10 shows the configuration of the grid and of the dust particle for the problem considered here. Note that a nonuniform (but constant in time) grid is used to describe better the sheath around the dust particle. The distance of the dust particle from the boundaries is  $20 \lambda_{De}$ . The plasma species are initially loaded according to a drifting maxwellian distribution with a downward vertical net flow velocity corresponding to a Mach number  $M = 10$ . The ion-to-electron-mass ratio is  $m_i/m_e = 1836$  and the temperature ratio is  $T_i/T_e = 0.05$ . To reach an equilibrium, particles that flow out of the lower boundary are replaced by particles injected at the top boundary [12].

Figure 11 shows the history of the net charge accumulated on the dust particle. In this case, particle rezoning (Algorithms S1 and C1) was used to ensure the accuracy of the calculation. The particles are loaded, initially, with a constant number of particles per cell, leading to a higher concentration around the dust particle where the cells are smaller. However, the plasma flow tends to empty the region around the dust, reducing the accuracy. Splitting

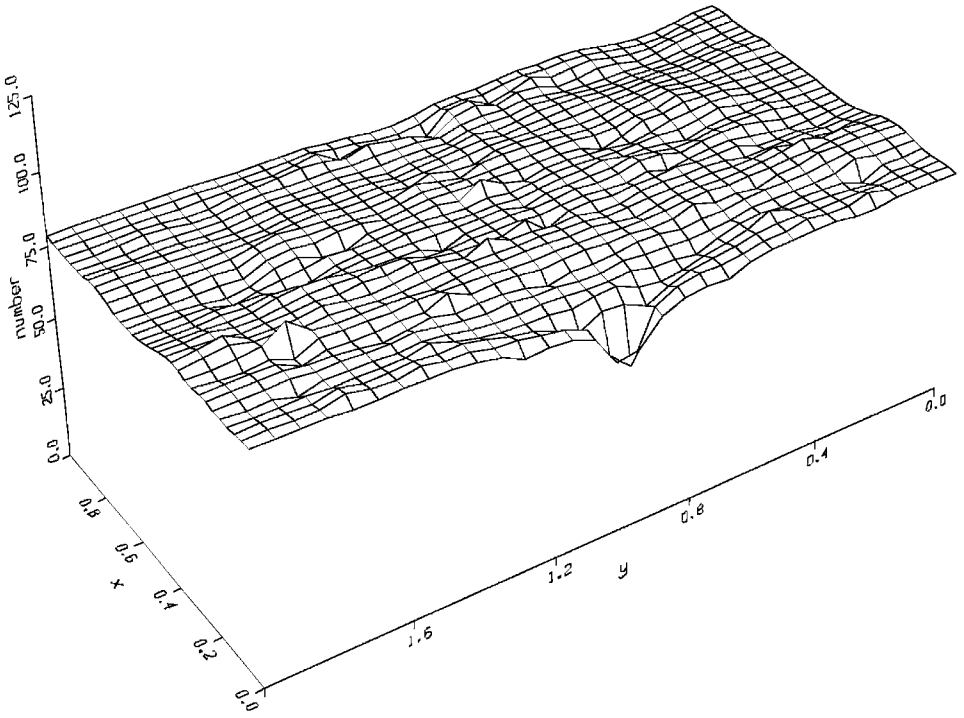




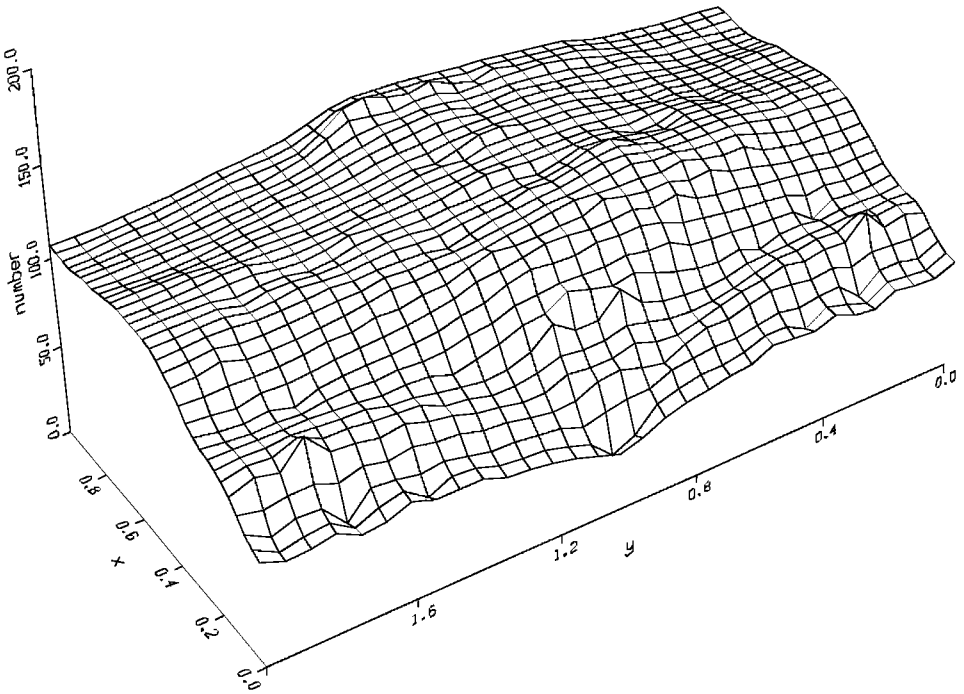
**FIG. 10.** *Dust charging.* Configuration of the system. The dust particle is spherical and is on the axis of symmetry at the origin of a cylindrical system. A nonuniform grid is used.



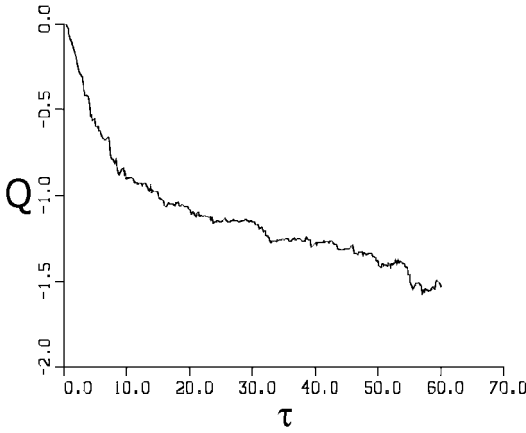
**FIG. 11.** *Dust charging.* History of the net charge accumulated on the dust particle. Calculation with particle rezoning, using algorithms S1 and C1. Charge is normalized as  $Q = q_{dust}/2\pi n e \lambda_{De}^3$  and time as  $\tau = t\omega_{pe}$ .



**FIG. 12.** *Dust charging.* Number of particles per cell at the end of the calculation shown in Fig. 11. Space is normalized as  $\xi = x\omega_{pe}/c$ .



**FIG. 13.** *Dust charging.* Number of particles per cell at the end of a calculation without particle rezone. Space is normalized as  $\xi = x\omega_{pe}/c$ .

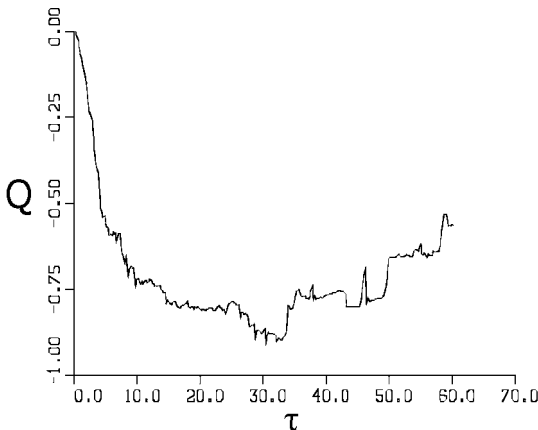


**FIG. 14.** *Dust charging.* History of the net charge accumulated on the dust particle calculation without particle rezoning. Charge is normalized as  $Q = q_{dust}/2\pi n e \lambda_{De}^3$  and time as  $\tau = t\omega_{pe}$ .

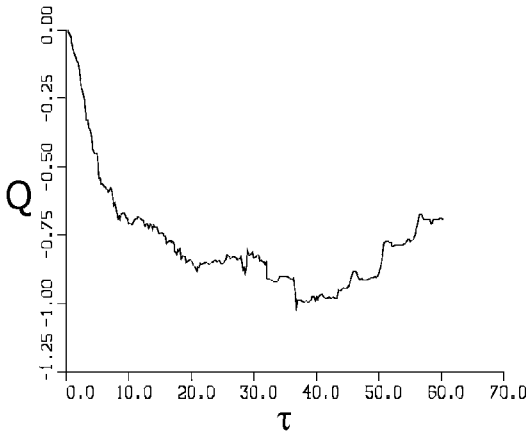
the particles moving toward the dust and coalescing the particles moving away from it is desirable to keep the number of particles per cell and the accuracy constant. Figure 12 shows the number of particles per cell at the end of the calculation. Clearly, particle rezoning is effective in achieving uniformity.

If the calculation is repeated without particle rezoning, the accuracy worsens in time as the region around the dust becomes less populated. Two effects lead to decreased accuracy around the dust particle: the particles originally present are in part captured by the dust and in part just simply flow away according to their average downward velocity of Mach  $M = 10$ . The new particles that replace them are flowing from regions of larger cells and are less numerous, leading to a decrease of accuracy.

Figure 13 shows the number of particles at the end of a calculation without particle rezoning. The region around the dust is sampled poorly by the particles, decreasing severely



**FIG. 15.** *Dust charging.* History of the net charge accumulated on the dust particle. Calculation with particle rezoning, using algorithms S2 (with  $\delta\mathbf{v} = \mathbf{v}_o/10$ ) and C1. Charge is normalized as  $Q = q_{dust}/2\pi n e \lambda_{De}^3$  and time as  $\tau = t\omega_{pe}$ .



**FIG. 16.** *Dust charging.* History of the net charge accumulated on the dust particle. Calculation with particle rezoning, using algorithms S2 (with  $\delta\mathbf{v} = \mathbf{v}_o/100$ ) and C1. Charge is normalized as  $Q = q_{dust}/2\pi n e \lambda_{De}^3$  and time as  $\tau = t\omega_{pe}$ .

the accuracy. Figure 14 shows the history of the dust charge. At the beginning, the evolution is correct, but as the number of particles decreases around the dust, the accuracy worsens and the calculation loses any physical meaning: the dust charge does not even reach a steady state. Clearly, in this case, particle rezoning is crucial to obtaining the correct result.

The calculation described above is repeated using Algorithm S2 to test the effects of the splitting in velocity on the particle distribution function. Figure 15 shows the charge history in this case. Evidently, the particle distribution function is altered to an unacceptable degree. The charge on the dust is very sensitive to the electron velocity distribution function [12], which determines the fraction of electrons reflected by the sheath. In Fig. 15, the splitting is done with Algorithm S2 using  $\delta\mathbf{v} = \mathbf{v}_o/10$  in the direction of  $v_x$ . If  $\delta\mathbf{v} = \mathbf{v}_o/100$  is used, the alteration is still very relevant (see Fig. 16). Evidently, in the case of dust charging, Algorithm S2 fails where S1 was successful.

## 8. CONCLUSIONS

We have formulated algorithms for particle rezoning in multidimensional kinetic PIC simulations. The present work extends to 2D and 3D methods previously developed for 1D systems [1].

The best algorithm for increasing the number of particles has been found to be the splitting of one particles into various particles (two in 1D, four in 2D, and eight in 3D) displaced in space.

The best algorithm for decreasing the number of particles has been found to be the coalescence of two particles close to each other in phase space.

We have applied the algorithm with success to 1D studies of collisionless shocks and to 2D problems involving the self-consistent charging of a dust particle immersed in a plasma.

## ACKNOWLEDGMENTS

The author wishes to acknowledge the help of Jerry Brackbill in the form of the many discussions on the application of the immersed boundary method and of particle control within the kinetic PIC method. Jerry Brackbill

has also very kindly provided the code CELESTE2D used as a framework to develop DEMOCRITUS. The code CELESTE1D was provided kindly by Jerry Brackbill and Huang Vu.

This work was supported by the United States Department of Energy under Contract W-7405-ENG-36, by NASA under the “Sun Earth Connection Theory Program,” and by the European Union under Contract HPRN-CT2000-00140.

Financial support from the Italian Istituto Nazionale per la Fisica della Materia (INFN) and Gruppo Nazionale per la Fisica Matematica of the Italian Istituto Nazionale D’Alta Matematica (GNFM–INDAM) has been used by the author to support partially his visits to Los Alamos during the time frame of development of the present work.

## REFERENCES

1. G. Lapenta and J. U. Brackbill, *J. Comput. Phys.* **115**, 213 (1994).
2. 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).
3. J. U. Brackbill, *J. Comput. Phys.* **108**, 38 (1993).
4. G. Lapenta, Variational grid adaptation based on the minimization of local truncation error: Time independent problems, submitted for publication.
5. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Hilger, Bristol, 1988).
6. C. K. Birdsall and A. B. Langdon, *Plasma Physics Via Computer Simulation* (McGraw–Hill, New York, 1985).
7. H. X. Vu and J. U. Brackbill, *Comput. Phys. Commun.* **69**, 253 (1992).
8. G. Lapenta and J. U. Brackbill, *Comput. Phys. Commun.* **87**, 139 (1995).
9. D. J. Cooperberg, V. Vahedi, and C. K. Birdsall, in *15th International Conference on the Numerical Simulation of Plasmas, Valley Forge, 1994*, Paper 3B21, Princeton Plasma Physics Laboratory, Princeton, NJ, 1994.
10. J. U. Brackbill and H. X. Vu, *Geophys. Res. Lett.* **20**, 2015 (1993).
11. H. X. Vu and J. U. Brackbill, *Comput. Phys. Commun.* **69**, 253 (1992).
12. G. Lapenta, *Phys. Plasmas* **6**, 1442 (1999).
13. G. Lapenta, *Phys. Rev. Lett.* **75**, 4409 (1995).
14. D. Sulsky and J. U. Brackbill, *J. Comput. Phys.* **96**, 339 (1991).
15. G. Lapenta, F. Inoya, and J. U. Brackbill, *IEEE Trans. Plasma Sci.* **23**, 769 (1995).