Call Number: QC20.2.A1 C6
Location: PHYS
ILL: 2367623 **Maxcost:** 0

DateReq: 12/11/2002 X Yes
Date Rec: 12/13/2002 No
Borrower: UUM Conditional
LenderString: UBY,*LDL,AZU,ORU,OKS

13-46 F

Title: Computer physics communications.
Author:
Edition:
Imprint: Amsterdam, North-Holland Pub. Co.
Article: Lapenta; Control of the number of particles in...

Vol: 87 **No.:** 1-2 **Pages:** 139-eoa **Date:** May 2, 1995

Borrowing Notes: M; F; ARIEL. BCR/AMIGOS/ARL/BIG12+

Fax: --PLEASE SEND ARIEL: 128.110.38.119 --FAX:(80

```
ILL: 2367623      :Borrower: UUM      :ReqDate: 20021211 :NeedBefore: 20030110
:Status: IN PROCESS 20021212          :RecDate:          :RenewalReq:
:OCLC: 966945     :Source: OCLCILL    :DueDate:          :NewDueDate:
:Lender: UBY,*LDL,AZU,ORU,OKS
:CALLNO:  *Lender's OCLC LDR: 1- 1969-    :TITLE:  Computer physics
communications.   :IMPRINT: Amsterdam, North-Holland Pub. Co.   :SERIES:
Europhysics journal   :ARTICLE: Lapenta; Control of the number of particles in...
:VOL: 87     :NO: 1-2     :DATE: May 2, 1995          :PAGES: 139-eoa
:VERIFIED: OCLC ISSN: 0010-4655 [Format: Serial]   :PATRON:  Banerjee, Biswajit
:SHIP TO:  INTERLIBRARY LOANS/MARRIOTT LIBRARY/UNIVERISTY OF UTAH/295 S 1500  E
SALT LAKE CITY, UT 84112-0860   :BILL TO:  Same   :SHIP VIA: ARIEL FAX/UPS/1ST
:MAXCOST: 0          :COPYRT COMPLIANCE: CCL   :FAX: --PLEASE SEND ARIEL:
128.110.38.119 --FAX:(801)581-4882   :E-MAIL: ILL-REQ@LIBRARY.UTAH.EDU   :BORROWING
NOTES: BCR/AMIGOS/ARL/BIG12+   :LENDING CHARGES:                :SHIPPED:
:SHIP INSURANCE:          :LENDING RESTRICTIONS:    :LENDING NOTES:    :RETURN
TO:    :RETURN VIA:
```

ShipVia: ARIEL FAX/U

NeedBy: 1/10/2003

Return To:
ILL Office Love Library
13th R ST.
University of Nebraska-Lincoln
Lincoln, NE 68588-4103

Ship To:

INTERLIBRARY LOANS
MARRIOTT LIBRARY
UNIVERISTY OF UTAH
295 S 1500 E SALT LAKE CITY, UT 84112-0

ILL: 2367623 **Borrower:** UUM
Req Date: 12/11/2002 **OCLC #:** 966945
Patron: Banerjee, Biswajit
Author:
Title: Computer physics communications.
Article: Lapenta; Control of the number of particles in...

Vol.: 87 **No.:** 1-2
Date: May 2, 1995 **Pages:** 139-eoa
Verified: OCLC ISSN: 0010-4655 [Format: Serial]
Maxcost: 0 **Due Date:**
Lending Notes:
Bor Notes: M; F; ARIEL. BCR/AMIGOS/ARL/BIG12+

ELSEVIER

Computer Physics Communications 87 (1995) 139–154

Computer Physics Communications

Control of the number of particles in fluid and MHD particle in cell methods

Giovanni Lapenta, J.U. Brackbill

Los Alamos National Laboratory, T-3, MS B216, Los Alamos, NM 87545, USA

Received 11 July 1994; revised 6 December 1994

Abstract

We describe an algorithm to control the number of particles in fluid, particle-in-cell calculations. In problems with large variations in mass density or grid spacing, the ability to increase or decrease the number of particles in each cell of the mesh becomes essential. Here, a cell-by-cell replacement algorithm which preserves grid data and positivity of the particle data, where appropriate, is described. The algorithm preserves contact discontinuities, introduces little diffusion, and adds little or nothing to the cost of a calculation.

1. Introduction

We describe an algorithm to replace the particles in a particle-in-cell (PIC) method with minimal loss of information. The need for replacement arises in calculations where the number of particles grows in time, or where their distribution over the domain must respond to changing conditions. For example, an adaptive grid may cause the zones in regions of strong gradients to become very small. An initially uniform distribution of particles will result in few or no particles in the small zones, with consequent loss of accuracy where it is most desired.

Several techniques for replacing particles have been described. Nordmark, for example, developed a rezoning algorithm for vortex-blob calculations [1]. His algorithm transfers data from a set of particles to a grid by interpolation, and then from the grid to a regularly spaced array of particles, of which there is one per grid cell. The particle replacement is applied intermittently, and only when the accuracy of the vortex-blob calculation deteriorates below a preset value. This algorithm corresponds to the "lumped-mass-matrix" formulation described below and in Burgess et al. [2]. In Eastwood, ephemeral particles are created each cycle to model convection in fluid flow [3]. The particles are the endpoints of characteristics, along which the solution is integrated over one time step. The particles are located at regularly spaced points, thus reducing the cost of inverting a "mass-matrix" to obtain a solution on a grid. Similarly, the semi-Lagrange method integrates along selected characteristics [4,5]. (Bermejo has shown the equivalence of this method to a finite-element formulation [6].)

Here, we extend and apply ideas outlined in a mass-matrix formulation of the particle-in-cell method to develop a new method for particle rezoning. Through this approach, we seek to reduce the error introduced by rezoning, and to employ locality to increase the flexibility and selectivity of the method. The resulting method can be applied to particle-in-cell calculations where particles persist from cycle to cycle, and where one wishes to retain PIC's special ability to preserve history data over long times and to resolve contact

0010-4655/95/$09.50
SSDI 0010-4655(94)00180-4

discontinuities.

The principal elements of our method are cell-by-cell particle replacement, a variational formulation that defines the properties of the new particle set, a technique for preserving positive definiteness of the particle data, and the ability to selectively apply particle replacement to maintain accuracy and preserve contact discontinuities. In outline, we describe the local approach, give a general formulation of the method for particle replacement, present the principle of maximum uniformity and the algorithm for positive definite variables, and apply the algorithm to the FLIP formulation of PIC [7,8]. To illustrate the properties of the algorithm, we include the results of calculations of a Kelvin–Helmholtz instability in a magnetized plasma, and of the interaction of the very-local-interstellar-medium with the heliosphere.

2. Local approach

To be able to devise an algorithm to control the number of particles per cell in fluid PIC methods let us start from a brief review of the rôle played by the particles.

Diverse PIC methods have been proposed, starting from the original method by Harlow [9]. In this work we are mainly concerned with fluid codes that use the particles to model convection by moving the particles through the system and interpolating particle data onto the grid. The main example of this technique is given by the FLIP code [7], but most of the methods that will be presented here apply to other codes as well [10].

When particles are regarded as a means to model convection, an algorithm to control the number of particles per cell can be found by replacing a given set of particles by another one that gives exactly the same contributions to the grid.

In the fluid PIC codes of the type described above, we say that two sets of particles that produce the same data on the grid are equivalent. This equivalence allows us to replace a set of particles by a new set containing a different number of particles without altering the correct evolution of the system.

This idea is the founding step for particle control in fluid codes. It is important to recognize that this approach is indeed quite different from that recently proposed for kinetic codes [11], where particles carry more information than the grid can collect. In kinetic codes, the particles sample the velocity distribution function and cannot be replaced by other particles that merely conserve grid data.

The idea of replacement among equivalent sets of particles can be developed in principle along two different routes.

The first can be called *global*, in which one replaces all of the particles of the system with the properties of the new particles calculated to match the old grid data at every point on the mesh.

The second can be called *local* where one replaces the particles in one cell at a time with the properties of the new particles calculated to match the old grid data only at those points within the support of the particles.

The second approach is, in our opinion, to be preferred for the following reasons. First, *better accuracy* can be achieved by the local approach because not only the grid data but also the partial contribution coming from each cell is preserved. Second, the local approach is more *flexible*, as one can act only in those cells where control is actually needed, without having to deal with the entire grid all at once. One can easily see that a local approach can lead to *memory savings*, as any additional storage required by the algorithm will have the small size given by the number of grid points within the support of the particles in a single cell and not by the entire grid. The local approach can also lead to a substantial saving of *CPU time* as it acts only when and where it is necessary, leaving everything else untouched. Finally, as far as *parallelization* is concerned a local approach is to be preferred as it will minimize interprocessor communication.

3. General formulation of the algorithm for particle control

As stated in the previous paragraph, the algorithm to control the number of particles is based on the complete replacement of all the particles located in a given cell by a new set of particles whose positions can be chosen freely.

In a typical fluid PIC code the position in the physical space x_p is supplemented by the position in a natural space ξ_p that is introduced by mapping each grid cell onto a unit cube. In the present paper a general 3D geometry is assumed in the derivation of the method

but the examples will be taken from a 2D code. In a 3D geometry the grid is composed of tetrahedral cells having vertices x_v and centers x_c. With these assumptions the relationship between natural and physical coordinates is given by:

$$\mathbf{x} = \sum_v \mathbf{x}_v b_1(\xi^1 - \xi_v^1) b_1(\xi^2 - \xi_v^2) b_1(\xi^3 - \xi_v^3) , \quad (1)$$

where b_1 is the b-spline of order 1 [12].

It is very convenient to give the new particles the same regular spacing in natural coordinates in every cell. (The physical coordinates will follow from Eq. (1)). As an example Fig. 1 gives a uniform lattice of 9 particles in a 2D cell, but other patterns can be used as well. The use of the same pattern everywhere increases the number of operations common to all the cells that one can perform only once and for all with a considerable saving in computational cost.

The properties of the new particles, other than position, are to be found by imposing the conservation of the grid data. In PIC codes any given particle property q_p is related to a grid quantity Q_g by a relationship of the following type [7]:

$$Q_g = \sum_p q_p S_{gp} , \quad (2)$$

where the index g labels grid points and S is an assignment function.

As an example, the FLIP-MHD code [8] used in the result section below assigns to the particles a mass, a velocity, an energy and a magnetic moment. These quantities are then interpolated to the grid through a formula similar to Eq. (2) to define respectively: fluid density, momentum, energy and magnetization.

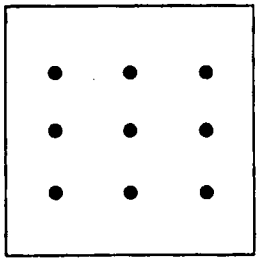


Fig. 1. Sketch of a possible disposition of nine new particles in the natural space.

The assignment function S_{gp} can differ from code to code but is often expressed in terms of b-spline b_l of order $l \leq 2$:

$$S_{gp} = b_l(\xi_p^1 - \xi_g^1) b_l(\xi_p^2 - \xi_g^2) b_l(\xi_p^3 - \xi_g^3) . \quad (3)$$

Some codes assume different interpolation functions and different grid points for different quantities. As an example FLIP-MHD uses linear interpolation to the grid vertices for mass and momentum and quadratic interpolation to the center of the cells for energy and magnetization. Thus g can either refer to cell centers c or vertices v.

With these definitions it is now possible to state exactly the central idea of our method: *replace the entire set of particles in a given cell c by a new set formed by particles having a position chosen freely (inside the cell c) and properties q_p (e.g. mass m_p, velocity $\mathbf{u}_p$, energy e_p and magnetic moment $\mathbf{b}_p$) chosen in order to preserve the old contribution to the grid as given by:*

$$Q_g^c = \sum_{p'(c)} q_{p'} S_{gp'} = \sum_{p(c)} q_p S_{gp} , \quad (4)$$

where the old particles are labeled by p', and the new by p. Note that the summations in Eq. (4) are over the particles in cell c only; Q_g^c represents the partial contribution to the grid point g coming from the particles in cell c.

The problem is therefore reduced to the solution of a linear system of equations of the type in Eq. (4) for each moment defined on the grid. A few considerations are in order.

First, Eq. (4) enforces the conservation both of the partial contribution to the grid data coming from the particles in the cell in which the replacement is performed and the global contribution. Consider the partial contribution Q_g^c to the grid point g coming from the $p(c)$ particles in cell c:

$$Q_g^c = \sum_{p(c)} q_p S_{gp} . \quad (5)$$

The global contribution due to all the particles of the system, $Q_g = \sum_c Q_g^c$ is:

$$Q_g = \sum_c \sum_{p(c)} q_p S_{gp} . \quad (6)$$

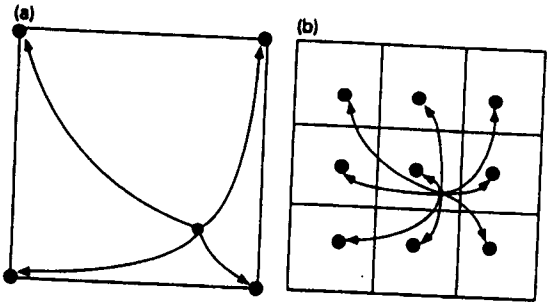


Fig. 2. Support of the assignment function S_{gp} for linear (a) and quadratic (b) interpolation.

Therefore conservation of the partial contributions Q_g^c guarantees global conservation.

Second, Eq. (4) affects only the grid points within the support of S_{gp}. This limits the domain of Eq. (4) to very few points. In Fig. 2 is shown a sketch of the number of points affected by the particles in a given cell for the case of linear and quadratic interpolation as used in FLIP. With linear interpolation, only the 4 vertices of the cell where the particles are located are affected. With quadratic interpolation the 9 centers of the cells contiguous to the particles are affected. In any case the problem size is very small: there are N_p unknowns, one for each particle, and N_g equations, one for each grid point affected.

Third, Eq. (4) is linear and therefore one can apply the well known results of linear algebra regarding the existence and uniqueness of the solution. Loosely speaking one can state that if $N_p \geq N_g$ a solution exists; however when $N_p > N_g$ the solution is not unique and an infinite set of solutions exists. Since it is usual in PIC codes that the number of particles exceeds the number of grid points, one must choose from among the solutions the one with the best properties.

Finally it is important to recognize that some of the particle and grid data (namely mass and energy) must be positive. Unfortunately the solution of a linear system is not guaranteed to be positive. Even when an infinite number of solutions exist, none of them may be positive. This circumstance is the toughest challenge in the development of a reliable control algorithm.

In the following we first find a solution for the generation of the new particle properties that are not defined positive (velocity and magnetic moment) and tackle the positive definite quantities later.

4. Principle of maximum uniformity

As noted above for non-positive definite quantities, there are solutions when $N_p \geq N_g$, but they are not unique when $N_p > N_g$. The problem is therefore underdetermined. One can formulate a variational principle to select among the set of possible solutions. A natural choice is the *principle of maximum uniformity*, which is consistent with a well known property of continuity of a fluid: Contiguous portions of a fluid have properties that vary continuously from point to point. If translated to particle codes this implies that contiguous particles, which supposedly represent contiguous portions of the fluid, should have properties that are not too different from each other. This can be formulated mathematically by requiring that the calculated value of any particle property q_p minimizes the variance of the distribution of particle properties among the particles in a given cell:

$$\sum_p \left(q_p - \bar{q}\right)^2 = \text{minimum}\,, \tag{7}$$

where $\bar{q} = \sum_p q_p/N_p$ is the mean value of the q_p among the particles in the cell.

The mathematical problem is now well posed: the solution of the minimization problem (7) has to be found such that q_p respects the linear constraint given by the conservation Eq. (4).

This problem can be solved easily by applying the method of Lagrange multipliers. With trivial algebraic manipulations one finds that the new properties q_p are given by:

$$q_p = \sum_g S_{gp}\lambda_g\,, \tag{8}$$

which gives the particle quantities as a function of N_g Lagrange multipliers. The multipliers are given by inverting a square $N_g \times N_g$ matrix:

$$\sum_{g'} \lambda_{g'} T_{gg'} = Q_g\,, \tag{9}$$

where Q_g are the grid data to be preserved (the superscript c has been dropped). The matrix of the linear system in Eq. (9) is symmetric and is given by:

$$T_{gg'} = \sum_p S_{gp} S_{g'p}\,. \tag{10}$$

The matrix $T_{gg'}$ is known as a *mass matrix*. The solution just found has some interesting properties. First, the mass matrix that is derived here from a variational problem is already used in PIC codes [2] and more generally in numerical methods [13]. The derivation just given, besides extending the application of the mass matrix to the control of the number of particles per cell, gives an explanation of the reason why mass matrix tends to give uniform results. Second, the mass matrix $T_{gg'}$ depends only on the natural coordinates of the new particles inside the cell. If they are always chosen in the same way this allows one to precompute the inverse mass matrix once for the entire grid. Precomputation reduces the computational cost of solving the linear problem (9) for the Lagrange multipliers. Finally, the solution of the problem is quite efficient (both in terms of memory and CPU) because it is expressed as a function of N_g Lagrange multipliers, where N_g can be considerably smaller than the number of particles N_p.

The problem of generating q_p when q_p is not required to be positive is therefore solved if $N_p \geq N_g$. In the less frequent case with $N_p < N_g$, no solution exists for the local replacement and the new particles cannot exactly preserve grid data. However one can solve the overdetermined problem approximately. If one minimizes the mean square error:

$$\sum_g \left(\sum_p q_p S_{pg} - Q_g \right)^2 = \text{minimum}, \qquad (11)$$

the solution can be easily obtained. In numerical calculations, errors in the conservation of the grid quantities are small even when $N_p < N_g$.

Our attention will be now focused to the study of the generation of positive definite quantities when $N_p \geq N_g$ holds.

5. Positive definite variables

The application of the algorithm of maximum uniformity as developed above may lead to the generation of negative masses or energies. Negative particle masses or energies are not automatically a sign of the failure of the method. As long as the grid quantities which should be positive are kept positive, the presence of negative particle properties is harmless. However, over time, one must anticipate that negative particles will result in negative grid quantities. For example if negative particles congregate in a cell, negative grid values may result. This process is made even more likely by the use of adaptive grids that may happen to focus around a negative particle, leading to negative grid quantities.

For these reasons, effort has been spent in understanding when the occurrence of negative masses or energies is more likely and in finding ways to overcome the problem. The problem can be attacked in two ways. One can study analytically a few simple classes of configurations, and one can apply the principle of maximum uniformity to many different configurations. The results of the analysis can be summarized by stating that negative masses or energies are most often generated when the positions of the new set of particles is very different from the old one.

One important example of the appearance of negative particles is given by cells where an interface between two fluids is present. If the set of new particles of either of the two fluids contains particles located on the wrong side of the interface, the principle of maximum uniformity leads to negative results. For this reason, it is better not to control the number of particles per cell in those cells where an interface is located.

On the other hand there are many situations where there is no apparent reason to forbid control and yet the new and old sets of particles are different enough to give negative particles. In this situation the application of the principle of maximum uniformity is simply inadequate and further measures are needed to ensure particle positivity.

One can enforce positivity by replacing the matrix $T_{gg'}$ in Eq. (9) with a *lumped mass matrix* [2]:

$$T^L_{gg'} = \delta_{gg'} \sum_{g''} T_{gg''} . \qquad (12)$$

In this case the Lagrange multipliers become $\lambda_g = Q_g / T^L_g$ where $T^L_g = \sum_{g'} T_{gg'}$. The results are therefore certainly positive but grid data Q_g is no longer exactly preserved. In practice, the exclusive use of $T^L_{gg'}$ leads to unacceptably high local errors [14].

Nevertheless this approach leads to a method that has been applied successfully to the problems shown

in the following section. One can replace $T_{gg'}$ with the *partially lumped mass matrix*:

$$T^*_{gg'} = (1 - \epsilon_{g'})T_{gg'} + T^L_{gg'}\epsilon_{g'}\,, \tag{13}$$

obtained by blending (by column, with coefficients $\epsilon_{g'}$ different from column to column) the original mass matrix with the lumped mass matrix.

Some thought on the use of the partially lumped mass matrix is in order. First, an optimum choice of $\epsilon_{g'}$ may allow one to enforce positive results without introducing significant error. This goal is achieved by calculating the minimum value of $\epsilon_{g'}$ that gives positive results. Second, although small local errors are introduced, grid variables are still conserved globally. By summing over g the equation for the Lagrange multipliers:

$$\sum_{g'} \lambda_{g'} T^*_{gg'} = Q_g \tag{14}$$

that replaces Eq. (9) when $T^*_{gg'}$ is used, one finds that $\sum_g Q_g$ is exactly preserved. (Note that this result is made possible by the definition of $T^*_{gg'}$ through a blending by column. A blending by row, e.g., $T^\dagger_{gg'} = (1 - \epsilon_g)T_{gg'} + T^L_{gg'}\epsilon_g$, would not give global conservation.)

The application of the partially lumped mass matrix is satisfactory in all of the applications shown in the following section. Nevertheless, another approach has been tried that can generate positive results without introducing any local error.

This alternative approach is based on the principle of *maximum entropy*, widely used in many physical applications [15]. The method maximizes the following expression:

$$S = -\sum_p q_p \log q_p\,, \tag{15}$$

which can be interpreted to be the entropy (or information content) of the system of configurations q_p. The expression for q_p is derived by applying the technique of Lagrange multipliers to Eq. (15), with the constraints given by Eq. (4). The particle properties are then given by:

$$q_p = \frac{1}{Z}\exp\Big(-\sum_{g=1}^{N_g-1} \lambda_g S_{gp}\Big)\,, \tag{16}$$

where Z is the partition function. The Lagrange multipliers are obtained by solving the following system of $N_g - 1$ non-linear equations:

$$Q_g = \frac{\partial}{\partial \lambda_g}\log Z\,, \tag{17}$$

in which the grid data has been normalized so that $\sum_g Q_g = 1$ to allow a direct interpretation in terms of probability theory.

Eq. (17) for the Lagrange multipliers is non-linear and quite expensive to solve, especially in the context of particle control where the operation has to be repeated for many cells and many time steps.

However, this approach gives the exact result whenever a solution exists. Unfortunately as mentioned before the solution to the problem for positive definite quantities may not exist. In this case the approach of the partially lumped mass matrix is still able to generate a result, with a small local non-conservation of the grid quantities. Similarly the maximum entropy can be applied loosely by solving Eq. (17) in a mean square sense, introducing again a small local error. Due to this effect, the local error introduced by maximum entropy is only marginally smaller than the error arising from the partially lumped mass matrix. Lower cost definitely favors the use of a partially lumped mass matrix [14].

6. Results

The algorithm to control the number of particles has been implemented in the FLIP-MHD particle-in-cell code, in two dimensions [8]. In all the simulations below the control algorithm is applied to generate the masses m_p, velocities $\mathbf{u}_p$, energies e_p and magnetic moments $\mathbf{b}_p$ for the new particles.

Table 1 reports for each quantity the algorithm used and the conservation rule that is applied. The new mass and energy are calculated with partial lumping to preserve positivity. To avoid creating particles with very small or very large mass or energy, a minimum value is defined as a fraction, α, of the average,

$$q_{\min} = \frac{\alpha}{N_p}\sum_g Q_g\,. \tag{18}$$

The new momentum and magnetic moments, which

Table 1
Summary of the methods used for generating the new particle properties

Particle property	Control algorithm	Conservation
mass m_p	partial lumping	mass: $R_g = \sum_p S_{gp} m_p$
energy e_p	partial lumping	energy: $I_g = \sum_p S_{gp} e_p$
velocity $\mathbf{u}_p$	maximum uniformity	momentum: $\mathbf{P}_g = \sum_p S_{gp} m_p \mathbf{u}_p$
magnetic moment $\mathbf{b}_p$	maximum uniformity	magnetization: $\mathbf{M}_g = \sum_p S_{gp} \mathbf{b}_p$

are not required to be positive, are calculated from the principle of maximum uniformity without lumping. The results indicate that this mixed approach does not result in large particle velocities.

Below, we give results that demonstrate the ability to control the number of particles in a kinematic and in a fully dynamic problem. The kinematic problem provides a non-trivial test, the calculation of uniform flow through a non-uniform grid, where comparison with an exact solution can be made. The problem also illustrates the necessity for particle control. The fully dynamic problem, a Kelvin–Helmholtz instability on an adaptive mesh, illustrates the effective use of cell-by-cell control in multi-fluid problems with mixing. The particle control method presented above has been also applied to fluid simulations of glow discharges for plasma processing devices. Results and considerations relative to this application are reported in Ref. [16].

6.1. *Heliospheric flow*

Our first example is motivated by a recent study of the interaction of the heliosphere with the very local interstellar medium (VLISM). In a fully dynamic calculation, the inner boundary of the polar grid, Fig. 3, is the outer boundary of the heliosphere, and the bottom edge is an axis of symmetry. On the inner boundary, flow conditions representing solar wind flow just outside the heliospheric termination shock are prescribed. On the outer boundary, the VLISM flow is modeled by prescribed inflow on the upstream portion of the boundary and continuative outflow on the downstream portion.

To illustrate particle control, we have shrunk the inner radius and replaced the prescribed heliospheric inflow by a no-slip boundary condition ($\mathbf{v}_g = 0$). In addition, we have prescribed the pressure $p = 0$ everywhere, so that the flow is uniform in space and time. Thus, the computational result should be a uniform flow-velocity and density everywhere but at the inner boundary.

The initial conditions are shown in Fig. 3. The grid, as noted above, is a polar grid with a radial spacing that is largest in the outer zones. The particles are initially loaded with uniform spacing in natural coordinates, with 9 particles per cell. The initial flow velocity (not shown) is uniform and corresponds to a transit time across a diameter equal to 9000 in problem time units. The density, also not shown, is uniform and equal to 0.03 problem units.

Without control of the particle number, the results are shown in Figs. 4a and 5a. If one compares Figs. 3 and 4a, the passage of time has caused the original particles to be displaced leftward with their relative position and spacing preserved. They have been replaced on the upstream side by new particles that have been injected at the upstream boundary. The original and new particles are clearly distinguished by the dif-

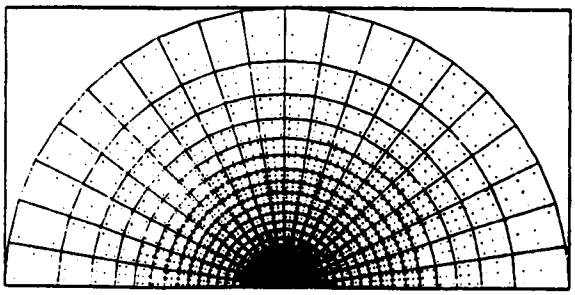


Fig. 3. Heliospheric flow: Initial condition for a kinematic test of a heliospheric flow. The computational grid is overlayed on the initial locations of the particles.

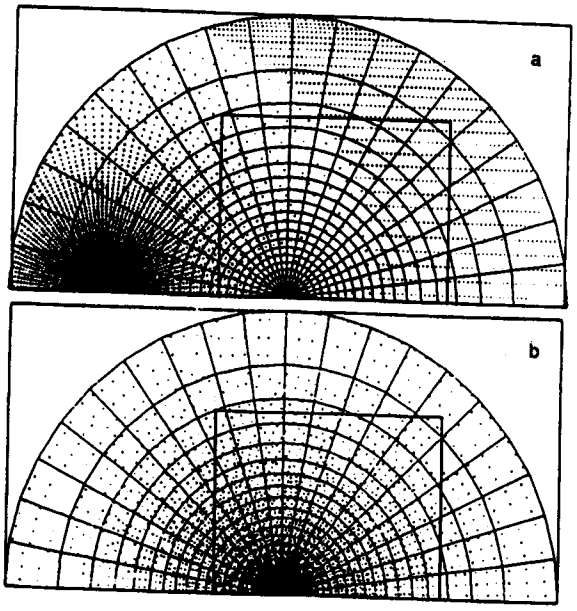


Fig. 4. Heliospheric flows: Location of the particles at the end of the simulation, $t = 3 \times 10^3$ in the units of the problem. Case a is the uncontrolled run. In case b the particles have been controlled. An almost uniform distribution is achieved. Note that the square boxes are blown up in Figs. 6 and 7.

ference in spacing. The flow velocity, Fig. 5a, appears to be uniform; but closer examination reveals that the vectors corresponding to many grid points are missing. A magnified plot of the particles and grid, Fig. 6, indicates the reason. In some of the smaller cells, there are no particles at all and in many there is only one. The number of particles per cell, Fig. 8a, indicates there are more than 200 particles in the large cells and none at all in the smaller ones.

With control of the number of particles, the results are shown in Figs. 4b and 5b. The particles, Fig. 4b, have been redistributed among the cells; and a magnified plot of the particles and grid, Fig. 7, shows that no cell is without particles. As a result, there are no grid points where the velocity is zero, Fig. 5b. The number of particles per cell, Fig. 8b, provides further evidence that control has worked. The number per cell varies between 5 and 12.

In this calculation, particle replacement is attempted every time step in every cell where $|n_p - n_{po}| > (n_{po})^{1/2}$. n_{po} is the initial and target number density and n_p is the present value. The reduced precision required of control reduces the cost, while the frequent application prevents any cell from emptying in the interval between control.

Of course, control of the number of particles is not the only objective. The grid data must be preserved while control is exercised.

The density without control, Fig. 9a, is compared with the density with control, Fig. 9b. Detailed comparison is difficult because of the difference in scale, but the maximum and minimum values for the density in the two cases are nearly equal. There is, however, more short wavelength variation in the controlled case. As remarked earlier, control preserves grid data at the time of replacement but may cause differences in subsequent time steps. The variation in Fig. 9b may be a manifestation of those differences.

Note that in the controlled run the improved description of the fluid flow allows us to use considerably larger time steps (more than a factor 2, Fig. 9). This beneficial effect is an indirect effect of particle control

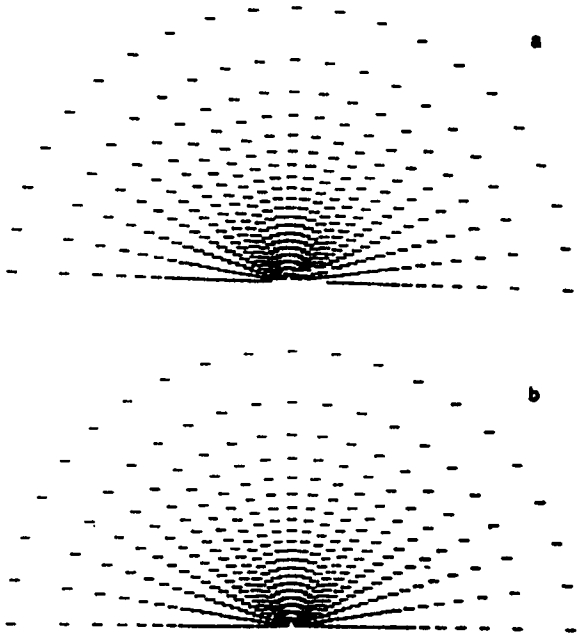


Fig. 5. Heliospheric flow: Velocity plot at the end of the simulation ($t = 3 \times 10^3$). The velocity at each grid vertex is represented by an arrow oriented along the local velocity vector and with a length proportional to the modulus of the velocity vector. Case a is the uncontrolled run. In case b the particles have been controlled.

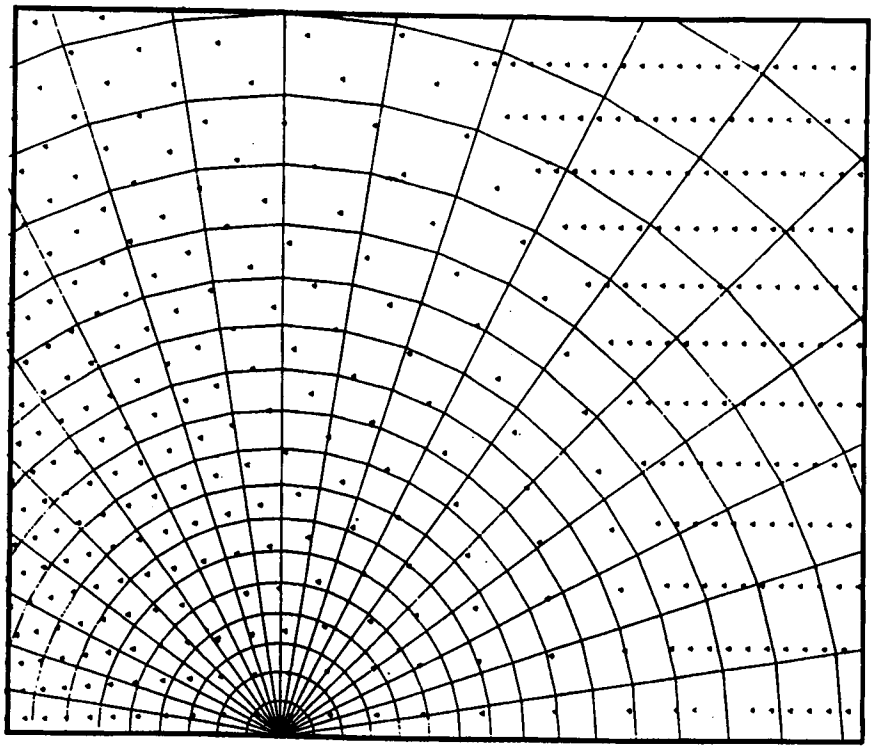


Fig. 6. Heliospheric flow: Blow up of the detail in the box of Fig. 4a. The particle positions are shown with respect to the grid. Empty cells are present.

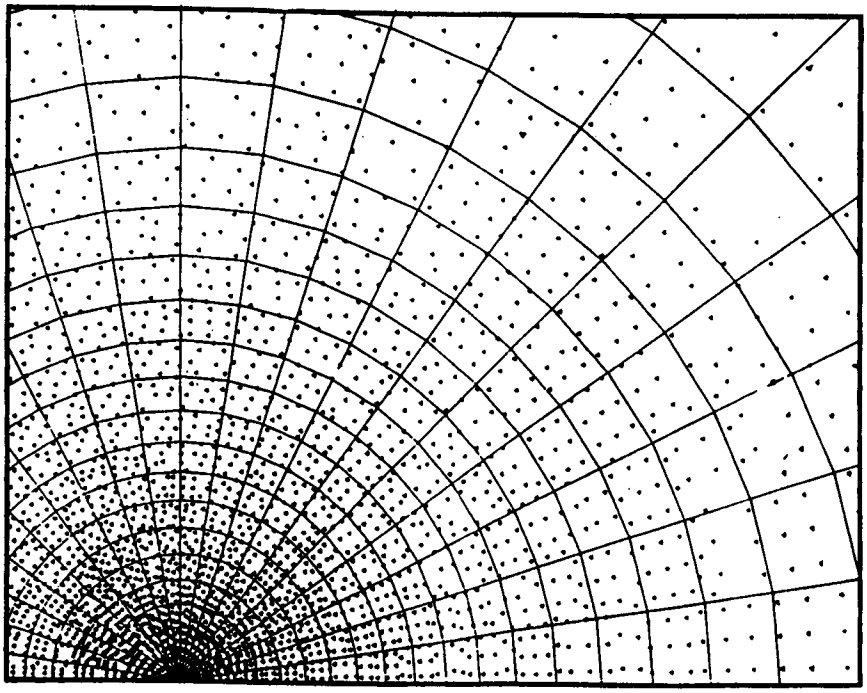


Fig. 7. Heliospheric flow: Blow up of the detail in the box of Fig. 4b. The particle positions are shown with respect to the grid. No empty cell is present.

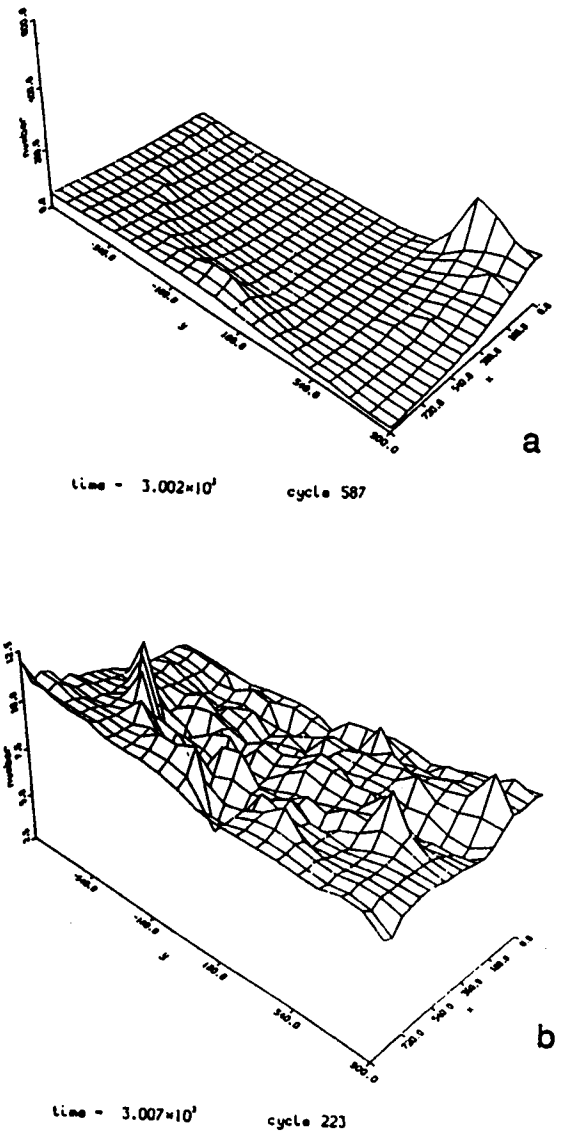


Fig. 8. Heliospheric flow: Plot of the number of particles per cell at the end of the simulation, without control (a) and with control (b).

that avoids unphysical zero values of the fluid velocity. Similar improvements have been observed in connection with the use of particle control in other applications, namely in the study of glow discharges. [16]

The single cell on the left side of Fig. 7 with 4 particles suggests further work on the modality of application of the control algorithm might be useful. It appears that the injected particles in the upstream cells have been controlled more aggressively than desirable, for these particles will soon move into smaller zones where a higher density of particles will be needed. Evidently, some way not only to measure present needs but also to anticipate future needs is required.

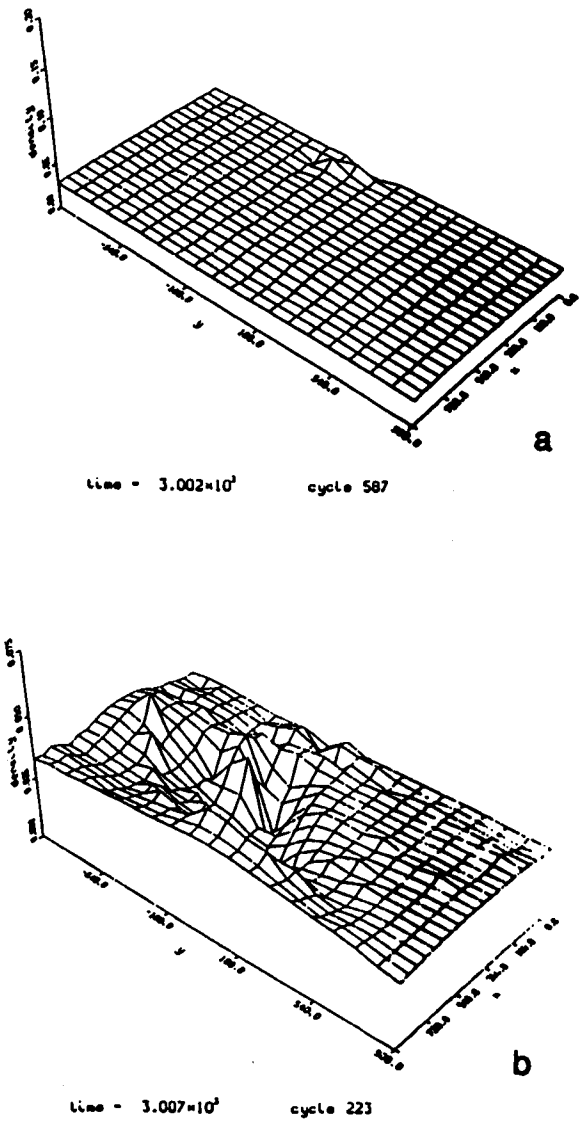


Fig. 9. Heliospheric flow: Fluid density at the end of the simulation for the uncontrolled (a) and controlled (b) run.

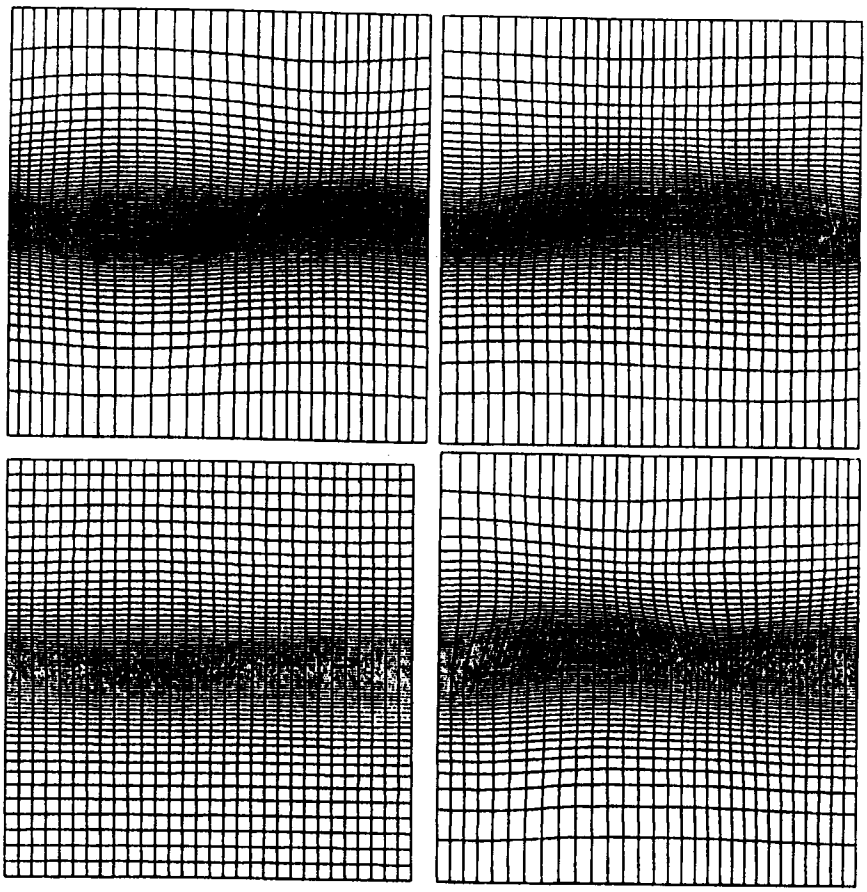


Fig. 10. Kelvin–Helmholtz instability: Plot of the coordinate lines of the adaptive grid used. Four different times are shown: $t = 25$ (a, bottom-left), $t = 50$ (b, upper-left), $t = 75$ (c, bottom-right), and $t = 100$ (d, upper-right).

6.2. *Kelvin–Helmholtz instability*

The influence of particle control on a dynamical calculation of the Kelvin–Helmholtz instability in two dimensions is illustrated by a comparison between calculations with and without control. In these calculations on a square domain of side L, a time-dependent adaptive grid with 60 zones in y and 30 in x resolves the shear layer between two counterflowing streams. Initially, there are 16 particles per cell. The system is composed of two counterstreaming fluids. The velocity of the two fluids is opposite, and particles belonging to each fluid are labeled by a variable, called color, that allows us to tell the two fluids apart. Periodicity is imposed in x, and free-slip boundary conditions are applied at $y = 0$ and $y = L$. The discontinuity in the flow velocity across the shear layer is $\Delta u = a$, where a is the sound speed. The flow transit time, L/u, is equal to 120 problem time units.

Without control, the adaptive grid, Figs. 10a–d, refines the shear layer by clustering grid lines along the center line. The location of the line of maximum clustering changes over time in response to changes in the vorticity. Figs. 11a–d show the location of the particles at four different times during the simulation: $t = 25, 50, 75, 100$. Note that only particles that were initially in the lower fluid (below the interface) are plotted. The particles depict the development of large amplitude vortices. Note the coalescence of several small vortices into one large vortex between $t = 50$, Fig. 11b, and $t = 100$, Fig. 11d. The number of particles, Figs. 12a–d, varies between $\sim$3–4 in the shear

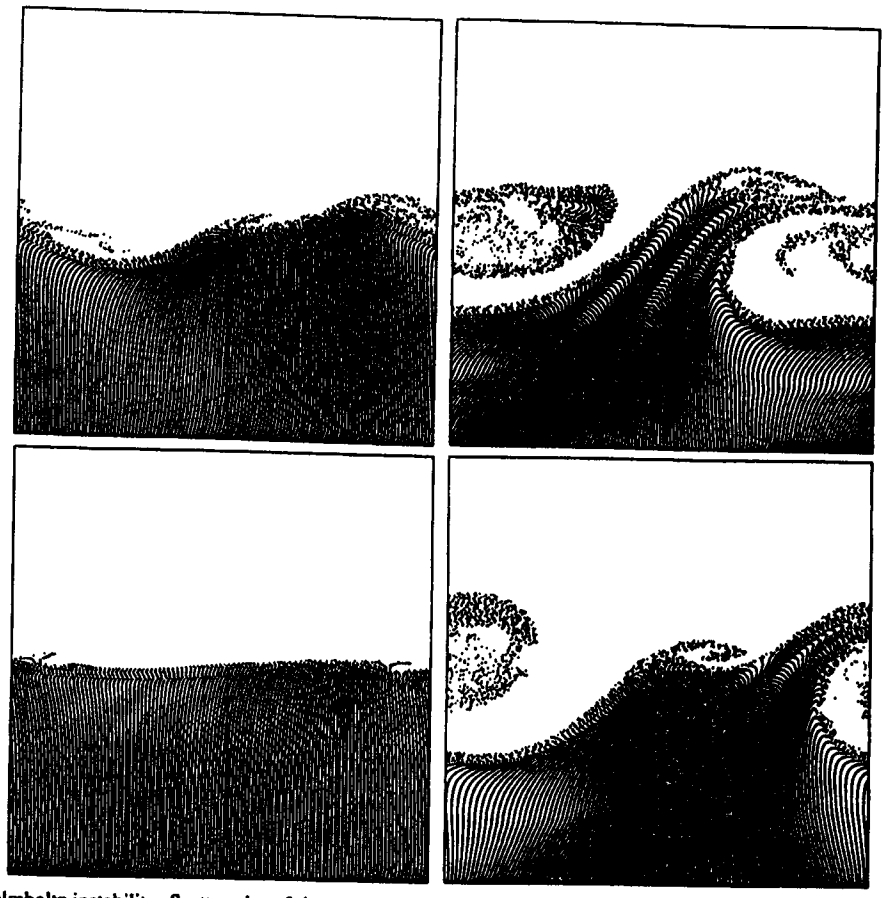


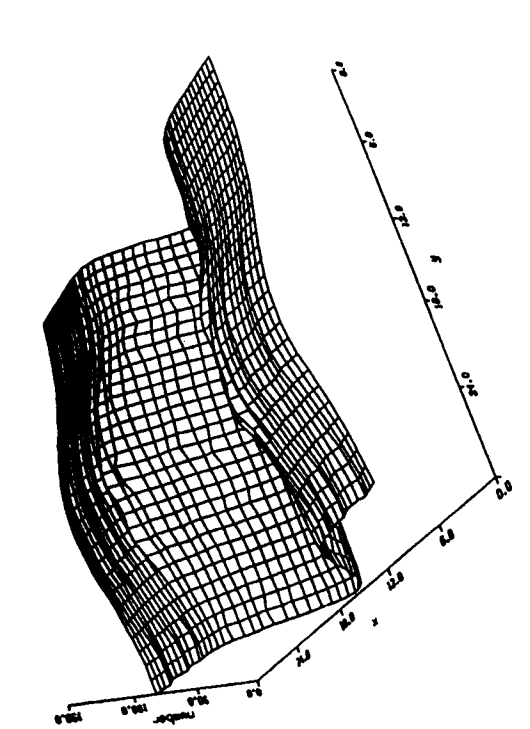
Fig. 11. Kelvin-Helmholtz instability: Scatter plot of the particles that were initially located below the interface. The same times in Fig. 10 are shown. This plot refers to a run without particle control.

layer to ~ 35 near the $x = 0$ and $x = L$ boundaries. At $t = 75$ and 100, the variation increases and the maximum number increases to ~ 100.

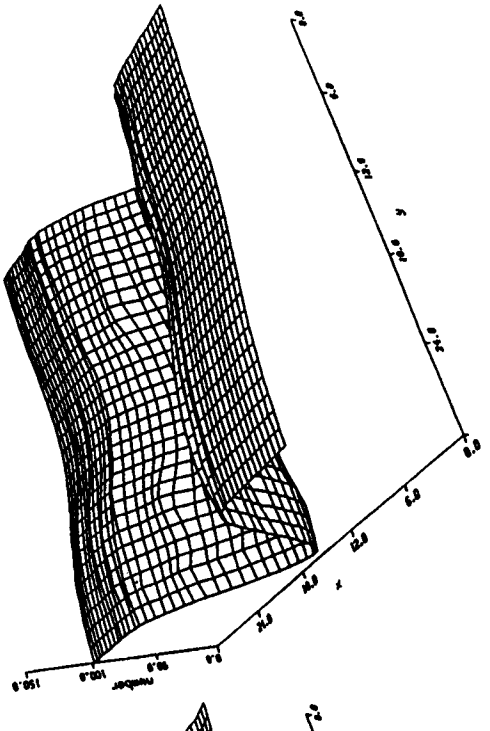
With control, the results are shown in Figs. 13 and 14. The application of particle control here is different from the kinematic case. In this example, control is imposed every 25 cycles to all cells, except mixed cells, where $n_p \neq n_{po}$. (A mixed cell is one containing particles with different color properties.) That is, the precision is high but the frequency of application is low. In Fig. 13a, the number of particles at $t = 25$ varies as much as in the uncontrolled case, Fig. 12a, because control has not yet been imposed. However, at $t = 50$, after control has been imposed once, the variation is much diminished, especially because the number of particles in cells outside the shear layer is reduced. (We note that control has been imposed at cycle 25, 17 cycles before the time shown in Fig. 13b).

Control remains effective throughout the calculation, Figs. 13c and 13d, everywhere except the shear layer itself, where mixed cells prevent the exercise of control.

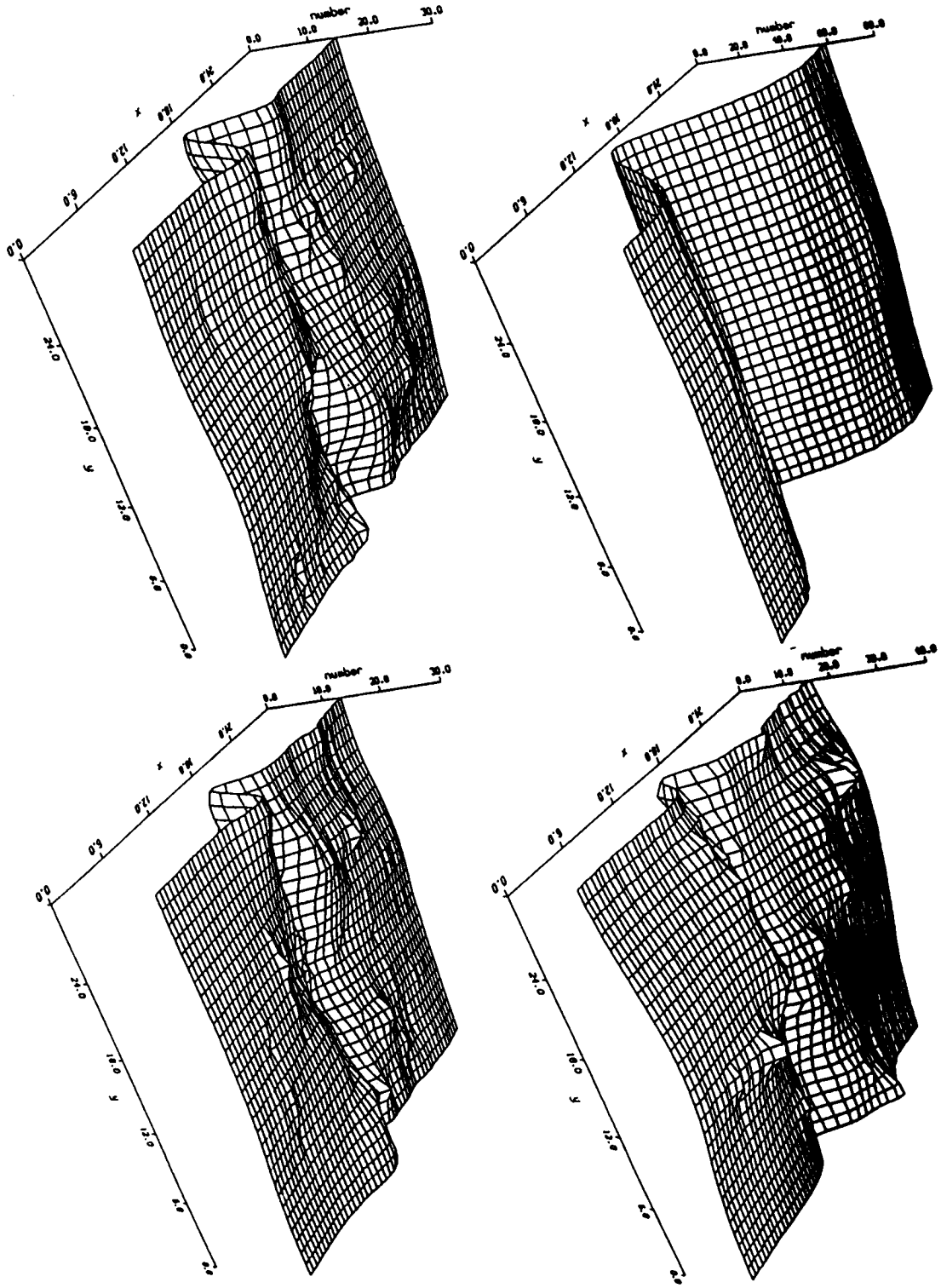
Fig. 12. Kelvin-Helmholtz instability: Plot of the number of particles per cell at the same times in Fig. 10. Run without particle control.


time - 2.514×10¹
cycle 21

time - 5.044×10¹
cycle 46


time - 7.526×10¹
cycle 102

time - 1.002×10²
cycle 135


number
x
y

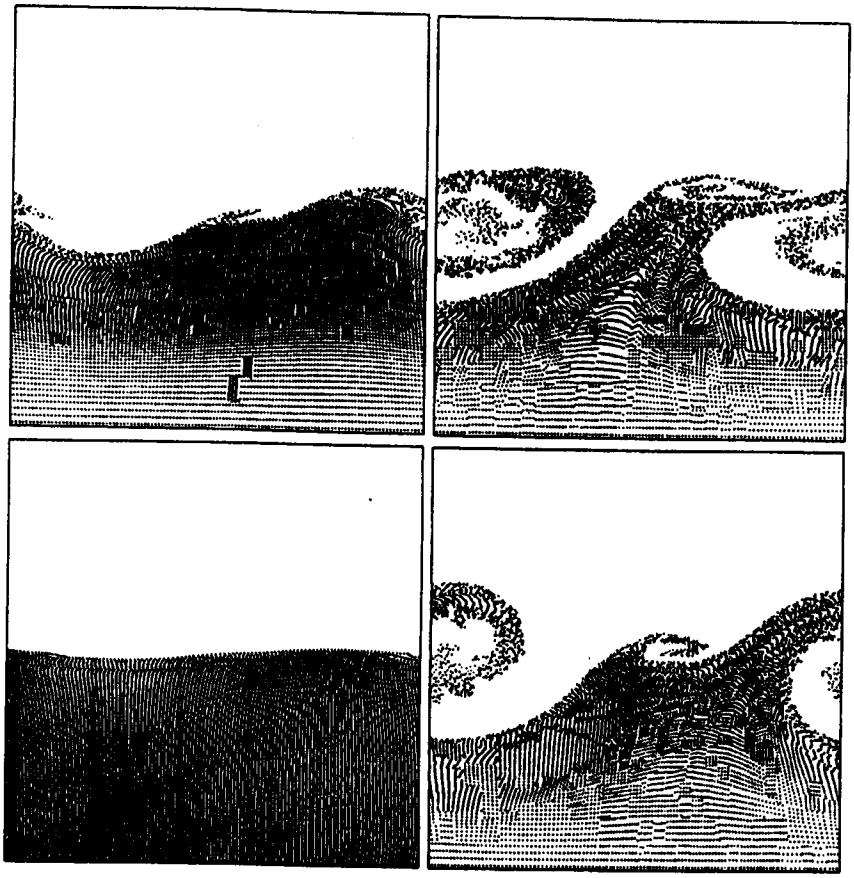


Fig. 14. Kelvin–Helmholtz instability: Plot of the particle locations. Same conditions as in Fig. 11 and same times plotted. Particle control is used.

The particle plots for the controlled case, Fig. 14, depict a similar development of the instability to those shown in Fig. 11. However, the patchy appearance is caused by failure of the control algorithm to find a new particle set that satisfies both the constraint of positivity and the required accuracy. (Less than 2.5% local error is allowed in the calculation.) At later times, most of the dense patches have disappeared. At t = 100, note that the original particles persist in mixed cells at the material interface.

The comparison of Figs. 11 and 14 shows convincingly that an almost uniform number of particles per cell is achieved without introducing any non-physical alteration of the correct behavior of the system. In fact, the small differences may be due principally to a different, average time step (100 cycles to completion with control, 135 without).

Fig. 13 Kelvin–Helmholtz instability: Plot of the number of particles per cell. The parameters of the simulation are as in Fig. 12. The same times are shown also, but particle control is now used.

7. Conclusions

A method for rezoning the particles in a fluid PIC method has been presented. The method can replace a set of particles with a new one with the desired number of particles. Exact conservation can be imposed in most cases. For positive definite quantities, the introduction of a relatively small diffusion may be needed to avoid negative values.

The results presented show that the rezoning of the particles can be achieved with high efficiency. Moreover, the particle replacement appears to preserve very accurately the correct evolution of the system. The method can be applied to a wide range of problems where the adaptivity of the grid or the wide variance of the properties of the system causes conventional PICs to fail.

Acknowledgements

We gratefully acknowledge support from the DOE Office of Fusion Energy; NASA; the Politecnico di Torino, Turin, Italy; and the Institute for Geophysics and Planetary Physics, Los Alamos. This research was performed in part using the resources located at the Advanced Computing Laboratory of Los Alamos National Laboratory.

References

[1] H.O. Nordmark, J. Comput. Phys. 97 (1991) 366.
[2] D. Burgess, D. Sulsky and J.U. Brackbill, J. Comput. Phys. 103 (1992) 1.
[3] J.W. Eastwood, Comput. Phys. Commun. 44 (1987) 73.
[4] A. Robert, J. Meteor. Soc. Japan 60 (1982) 319.
[5] A. Robert, T.L. Yee and H. Ritchie, Mon. Wea. Rev. 113 (1985) 388.
[6] R. Bermejo, Mon. Wea. Rev. 118 (1990) 979.
[7] J.U. Brackbill and H. M. Ruppel, J. Comput. Phys. 65 (1986) 314.
[8] J.U. Brackbill, J. Comput. Phys. 96 (1991) 163.
[9] F.H. Harlow, Los Alamos National Laboratory Report No. LA-2301 (1959) (unpublished).
[10] J.U. Brackbill and J.J. Monaghan, Particle Methods in Fluid Dynamics and Plasma Physics (North-Holland, Amsterdam, 1988).
[11] G. Lapenta and J.U. Brackbill, J. Comput. Phys. 115 (1994) 213.
[12] J.J. Monaghan, SIAM J. Sci. Stat. Comput. 3 (1982) 422.
[13] O.C. Zienkiewicz and K. Morgan, Finite Elements and Approximation (Wiley, New York, 1983).
[14] G. Lapenta, Simulation Methods for Multiple Length Scale Problems in Thermonuclear Plasmas, Ph.D. Thesis, Politecnico di Torino, Torino (1994) (in Italian).
[15] R.M. Bevensee, Maximum Entropy Solutions to Scientific Problems (Prentice-Hall, Englewood Cliffs, NJ, 1992).
[16] G. Lapenta, F. Iinoya and J.U. Brackbill, Particle in cell simulation of glow discharges in complex geometries, IEEE Trans. Plasma Science, accepted.

ELSEVIER

Abstract

This paper c
software suite
relativistic cha
important, and
the numerical s
parallel compu
for large scale

1. Introducti

The softwa
erful new cap
modelling. It v
and three-dim
tubes and mic
tion of electr
flow is import
be applied to c
tackled by fi
codes. The so
Vlasov set of
and particle m
ume; this is ap
acteristic time
of light across
timescale, elec

[1] E-mail: jim.ea

0010-4655/95/$0
SSDI 0010-465