JOURNAL OF COMPUTATIONAL PHYSICS 52, 390–413 (1983)

Second-Order Fluid Particle Scheme

A. NISHIGUCHI AND T. YABE

Institute of Laser Engineering, Osaka University,
Yamada-Oka, Suita, Osaka 565, Japan

Received September 9, 1982; revised February 1, 1983

A new momentum and energy conservative fluid particle scheme (SOAP) having a second-order accuracy in the advection is developed. The scheme can be used in a nonuniformly spaced and/or moving grid system. The required particle numbers and calculation time are substantially reduced with less noise and numerical viscosity. The scheme is successfully extended to polar coordinate system without the difficulty of numerical centrifugal force.

I. INTRODUCTION

At present, there are many techniques solving the fluid equations in the multidimensions. The representative methods are the Eulerian, Lagrangian and their mixed methods, and the Finite Element method. Each method has been used according to its applicabilities.

The Eulerian formulation remains applicable for a wide range of fluid distortions. The principal difficulty is that it tends to introduce false diffusions, for example, noticeable in the large gradient regions and so on. When an element of material enters a cell, its characteristics are uniformly mixed with those of all the others in the cell. In order to avoid the numerical diffusion, some special technique such as Flux-Corrected Transport (FCT) [1] is required.

The Lagrangian approach [2, 3] has proved particularly useful for treating the system involving several fluids. The interfluid boundaries are always clearly delineated. A large number of strikingly successful calculations have been performed by several groups of workers. But it is limited to use with systems in which no large distortions of the fluid occur. In a finite-sized grid, various topological catastrophes can happen which reduce the further results to nonsense.

Besides them, there are several methods to treat the fluid equations, for example, ICEDALE [4], MFE [5] and Particle-in-Cell (PIC) method [6]. Among them, the PIC method can treat large distortions without any difficulties although it is generally somewhat extravagant in the use of memory space in computing machine because it employs the double grid system. The attractive feature of the PIC method is no mass diffusion and the ability of treating large distortions of fluid. The main defects of the PIC method are numerical noise, viscosity and heat conduction. If the numerical

0021-9991/83 $3.00

noise can be effectively reduced with fewer particles, the problems of memory capacity and calculation time can be considerably improved.

For the countermeasure of these defects, GAP [7], PAL [8] and some recent work [9] employed the full particle method which is more Lagrangian than the PIC method. Due to the collisionless nature of particle motion, they need arificial relaxation between the particle's physical quantities and cell-wise quantities to sustain a fluid nature. These artificial relaxation coefficients are required for all quantities except mass and are problem-dependent when thermal conduction, heating or ion-electron temperature relaxation is added to energy equation [10] and/or magnetic flux convection coupled with magnetic field diffusion analyzed in Ref. [9] is treated by the same method.

This article proposes a new second-order accurate fluid particle (SOAP) scheme, which has the second-order accuracy in the treatment of advective term in space and the first-order accuracy in time. Our scheme is more Eulerian than the full particle method [7–9] and close to the original PIC method. In our scheme, only the particle's mass and locations are stored in the particle's memory, and momentum and total energy (internal energy and kinetic energy) are convected in a similar way as Harlow's method but in a more accurate way. Because no relaxation coefficients are required to sustain a fluid nature and no extra memories in particle are required other than mass and locations, the extension of the scheme to a more complicated system [11] or a more general hyperbolic system is straightforward. Furthermore, momentum and total energy are conserved.

In Section II, collisional quasiparticle methods are briefly reviewed and their advantages and defects are clarified. Section III provided a basic algorithm of our SOAP scheme. Test calculations of the scheme are given in Sections IV and V, with the extension of our scheme to the polar coordinate system described in Section V.

II. Brief Review of Collisional Quasiparticle Methods

The Particle-in-Cell method was developed by F. H. Harlow and others at Los Alamos in 1955 [6]. First, we review the original description of the method and point out the defects of it as an introduction to our new scheme described in the following sections. Although the method can be applied to a variety of hyperbolic equations, our discussions will be limited to the fluid dynamics described by the following equations:

$$\partial \rho/\partial t + \nabla \cdot \rho\vec{v} = 0, \tag{1}$$

$$\partial \rho\vec{v}/\partial t + \nabla \cdot \rho\vec{v}\vec{v} = -\nabla p, \tag{2}$$

$$\partial \rho E/\partial t + \nabla \cdot \rho\vec{v}E = -\nabla \cdot p\vec{v}, \tag{3}$$

where ρ, $\vec{v}$, p and E are the density, velocity, pressure and specific total energy, respectively. The second terms in the left-hand side of Eqs. (1)–(3) represent the advection by fluid bulk motion. In many cases, this effect is very important but occasionally causes numerical instabilities in finite difference calculations. In the PIC method, the fluid bulk motion is realized by means of fluid particles. Each particle represents the fluid element and moves in a Lagrangian fashion. On the contrary, the object space is divided into a system of cells as depicted in Fig. 1. The grid point is the center of the cell. Each cell is characterized by a set of variables describing the mean values of velocity, internal energy, density and pressure inside the cell. The calculation proceeds through a series of cycles of duration Δt^n. Each cycle is further subdivided into two phases; the Eulerian and the Lagrangian phases.

In the Eulerian phase, we neglect the advection terms and the required finite difference equations in the two-dimensional Cartesian formulation are those of momentum and energy;

$$\rho^n_{i,j}\frac{\tilde{u}_{i,j}-u^n_{i,j}}{\Delta t^n}=-\frac{1}{\Delta x}(p^n_{i+1/2,j}-p^n_{i-1/2,j}), \tag{4}$$

$$\rho^n_{i,j}\frac{\tilde{v}_{i,j}-v^n_{i,j}}{\Delta t^n}=-\frac{1}{\Delta y}(p^n_{i,j+1/2}-p^n_{i,j-1/2}), \tag{5}$$

$$\rho^n_{i,j}\frac{\tilde{I}_{i,j}-I^n_{i,j}}{\Delta t^n}=-p^n_{i,j}\left(\frac{\bar{u}_{i+1/2,j}-\bar{u}_{i-1/2,j}}{\Delta x}+\frac{\bar{v}_{i,j+1/2}-\bar{v}_{i,j-1/2}}{\Delta y}\right). \tag{6}$$

The specific internal energy I is given by

$$I=E-(u^2+v^2)/2.$$

Quantities labeled with integer-plus-one-half subscript are obtained as an average from the two adjacent cells. The superscript n represents the time cycle counting. The velocities distinguished by a bar mean the average between the old and tentative new values denoted by tilde. In the Eulerian phase only the cell-wise quantities are changed.

In the Lagrangian phase, the particles are forced to move with the tilde velocity, where an effective velocity is calculated by an area-wise velocity weighting among the tilde velocities at the four nearest cell centers and the particle position, such as,

$$\vec{V}_p=A_1\vec{V}_{i,j}+A_2\vec{V}_{i+1,j}+A_3\vec{V}_{i,j+1}+A_4\vec{V}_{i+1,j+1}, \tag{7}$$

where A_1, A_2, A_3 and A_4 are the fractional areas of overlap of a particle-centered cell onto the four adjacent cells (the areas are normalized as $\sum A_n=1$). With this effective velocity, the new position of the particle $\vec{x}^{n+1}_p$ is calculated to be

$$\vec{x}^{n+1}_p=\vec{x}^n_p+\vec{V}_p\Delta t^n. \tag{8}$$

If a particle moves across a cell boundary, the particle mass, momentum and energy are simply added to the new cell and subtracted from the old cell. The energy and momentum are assigned to the particle only when it moves across a cell boundary. So those quantities are not stored in each particle's memory.

The successful running of numerous calculations has demonstrated many times that the PIC method is useful for the calculation of problems in fluid dynamics involving large distortions of the fluid. The method has been useful for a number of problems for which other methods have been hard to handle. But the PIC method has certain limitations and some defects as listed below.

(i) The numerical noise caused by using the finite numbers of fluid particles. If many particles are employed in order to reduce the numerical noise, it wastefully requires much computer memory capacity and calculation time.

(ii) The range of variation in density is limited by initial particle number in one cell. This is the direct source of the numerical noise as well. The continuous change of physical quantities is desirable.

(iii) The numerical viscosity and thermal diffusion caused by the averaging effects in cells are the severe defects as the case may be. These arise from the treatment of the distributions of the velocity and internal energy in a cell. The internal energy distribution in mixed cells was fairly discussed, but the best treatment of them is not settled down.

(iv) The spatial resolution is limited to a certain degree because of using the Eulerian grid. For best resolution, one must use numerous grids if fixed uniformly spaced grids are used. It is rather wasteful.

To overcome these defects, some workers [7–9] have modified the original PIC method. In the PIC method, fluid quantities such as momentum and energy are stored on grids, and particles carry mass-weighted portions of these quantities from cell to cell. On the contrary, the later developments modify the PIC to fully represent a Lagrangian fluid and to use grids only for field calculations. In this form, they can suppress numerical noise and diffusions. But due to the collisionless nature of particle motion, they tend to produce local anomalies such as multistreaming instabilities and/or multitemperature problem. These effects are not intrinsic in original hyperbolic equations. To suppress these effects and to sustain a fluid nature, they need artificial relaxations between particles' physical quantities and cell-wise quantities. If the relaxation coefficients are poorly settled, local anomalies grow intolerably or numerical diffusions cause oversmoothing. The best selections of the relaxation coefficients are complicated because, for example, in the calculation of internal energy the relaxation coefficients should include the effects of fluid kinetics, thermal conduction, energy deposition and so on [10]. Furthermore the problem becomes more complicated in the case of the convection of nonfluid quantities such as magnetic flux convection coupled with diffusion [9].

Fully Lagrangian fluid particle method needs much computer memory capacity when it is used in two- or three-dimensional problems and in a complicated problem which includes convections of fluid, electron temperature, ion temperature, multicomponent of magnetic flux, nuclear fusion products, ions in various ionization stages [12] and so on. Furthermore, it is very hard to conserve both total momentum and energy in this method.

In this article, we choose a different way from the fully Lagrangian particle method and prefer a more Eulerian way such as Harlow's method. In our method, required memory capacity is much reduced because only the particle's mass and locations are stored in the particle's memory and we use finite-sized particles for reduction of noise. The difficulties of artificial relaxation coefficients are not met and hence the extension to more general hyperbolic system is straightforward. The following sections are dedicated to the introduction of our new scheme.

III. Fluid Macroparticle Code

In this section, we describe our new macroparticle method in detail, which is based on the original PIC method. The cells and particles are similar to it. Before we start, we briefly discuss the method to overcome the defects of the PIC method mentioned in the previous section.

The numerical noise caused by NGP (nearest-grid-point) procedure employed in the PIC method may be reduced by the area-weighting technique [13] but some modification is required before it is applied to the fluid PIC method. The full particle method [7–9], which means that all quantities are kept in the particle's memory, may use the area-weighting technique without special care but the problem of local anomalies is still a subject of considerable debate. The finite-sized particle also allows a large density variation because the density is determined by the fractional area of a particle belonging to a cell. It should be noted that the variation, however, is sometimes fictitious because of an incompressible nature of finite-sized particle. In uniformly spaced grids, the correctly represented range of density is decided by initial particle numbers per cell and is not less than that of one particle. If the grids are nonuniformly spaced, the minimum density is much lower than that.

Nonuniformly spaced grids also improve the resolution. Furthermore, it is possible to use moving grids for a local fine resolution because all physical quantities are temporarily stored within a finite-sized particle only when it moves.

Having these in mind, let us start to describe the basic procedure of our scheme. In the Eulerian phase, neglecting the advection terms, we solve Eqs. (1)–(3). Taking account of the conservations, the following difference forms are adopted:

$$\rho_{i,j}^{n} \frac{\tilde{u}_{i,j} - u_{i,j}^{n}}{\Delta t^{n}} = -\frac{1}{\Delta x_{i}^{n}} (p_{i+1/2,j}^{n} - p_{i-1/2,j}^{n}), \tag{9}$$

$$\rho_{i,j}^{n} \frac{\tilde{v}_{i,j} - v_{i,j}^{n}}{\Delta t^{n}} = -\frac{1}{\Delta y_{j}^{n}} (p_{i,j+1/2}^{n} - p_{i,j-1/2}^{n}), \tag{10}$$

$$\rho^n_{i,j}\frac{\tilde{E}_{i,j}-E^n_{i,j}}{\Delta t^n} = -\frac{(p^n\tilde{u})_{i+1/2,j}-(p^n\tilde{u})_{i-1/2,j}}{\Delta x^n_i} - \frac{(p^n\tilde{v})_{i,j+1/2}-(p^n\tilde{v})_{i,j-1/2}}{\Delta y^n_j}. \tag{11}$$

Quantities labeled with integer-plus-one-half subscript are obtained as an interpolation from the two adjacent cells. The interpolation of the products of pressure and velocity in Eq. (11) is defined to be

$$(p^n u)_{i+1/2,j} = p^n_{i+1/2,j} * u_{i+1/2,j},$$

for the correspondence to the kinetic equations, Eqs. (9) and (10). Equations (9) and (10) rigorously conserve the momentum and Eq. (11) conserves the total energy. In this phase, no other specific problems occur.

In the Lagrangian phase, some modifications are required for a reasonable accurate treatment. If the area-weighting method is applied to the scheme in a usual manner analogus to the NGP method, the momentum and the total energy of four adjacent cells are attached to the particles as shown in Fig. 1a. These quantities are redistributed uniformly within the particle. After it moves as shown in Fig. 1b, these are repartitioned to new four adjacent cells. This process is expressed as

$$\begin{gathered} U^* = A_1 U^{\text{OLD}}_{i,j} + A_2 U^{\text{OLD}}_{i+1,j} + A_3 U^{\text{OLD}}_{i,i,j+1} + A_4 U^{\text{OLD}}_{i+1,j+1}, \\ U^{\text{NEW}}_{i,j} = B_1 U^*, \qquad U^{\text{NEW}}_{i+1,j} = (B_2 + B_3) U^*, \\ U^{\text{NEW}}_{i,j+1} = (B_4 + B_7) U^*, \qquad U^{\text{NEW}}_{i+1,j+1} = (B_5 + B_6 + B_8 + B_9) U^*, \end{gathered} \tag{12'}$$

for the case depicted in the figure. Here, U represents a conservative quantity during the advection. It is easily understood that the process is mistaken and produces the zeroth-order diffusion. Even when the particle does not move, that is, $B_1 = A_1$,

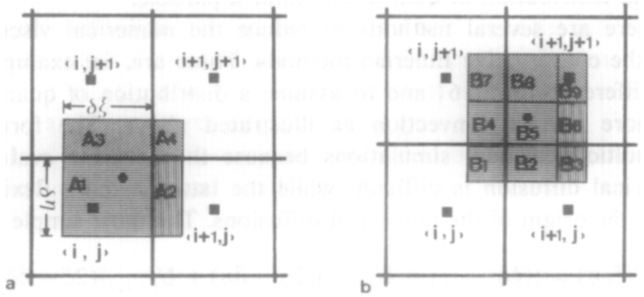

FIG. 1. Schematics of a finite-sized particle's location (●) and the cell center (■) (a) before and (b) after the particle moves. In the first-order scheme (Eq. (12)), the cell boundary lines of fractional areas (A1, A2, A3, A4) are memorized during the particle motion, that is, the area B7 + B8 in (b) corresponds to A3 in (a), for example.

$B_3 = A_2, B_7 = A_3, B_9 = A_4$ and $B_2 = B_4 = B_5 = B_6 = B_8 = 0$, $U^{\rm NEW}$ differs from $U^{\rm OLD}$ because of the averaging procedure mentioned above.

As the first candidate to overcome this diffusion, the particles' inner boundary lines are temporarily memorized during the particle movement. In the example depicted in Fig. 1b, the procedure can be expressed to be

$$\begin{aligned} U^{\rm NEW}_{i,j} &= B_1 U^{\rm OLD}_{i,j}, \\ U^{\rm NEW}_{i,j+1} &= B_4 U^{\rm OLD}_{i,j} + B_7 U^{\rm OLD}_{i,j+1}, \\ U^{\rm NEW}_{i+1,j} &= B_2 U^{\rm OLD}_{i,j} + B_3 U^{\rm OLD}_{i+1,j}, \\ U^{\rm NEW}_{i+1,j+1} &= B_5 U^{\rm OLD}_{i,j} + B_6 U^{\rm OLD}_{i+1,j} + B_8 U^{\rm OLD}_{i,j+1} + B_9 U^{\rm OLD}_{i+1,j+1}. \end{aligned} \tag{12}$$

Although this method can conserve the total momentum, it only has the first-order accuracy in the space step and is equivalent to the NGP method using a large number of small point particles, which are uniformly distributed within the area of a macroparticle. This form can conserve the momentum and total energy. This area-weighting procedure removes the zeroth-order diffusion, but the first-order diffusion, which is the so-called numerical viscosity or numerical heat conduction, still remains. The effect is usually not crucial, except for the case where fluid velocity relative to Euler grid is greater than local sound speed and velocity gradient is relatively large. This phenomena may be understood by the following explanation, which, although it is not mathematically correct, is illustrative. Since the particle transports the momentum $m_{\rm p} u_{\rm p}$ and total energy $Q_{\rm p}$, the internal energy S is calculated by $S = \sum Q_{\rm p} - (\sum m_{\rm p} u_{\rm p})^2/(2 \sum m_{\rm p})$, where $\sum$ means the sum of the particles in a cell. The inequality $(\sum m_{\rm p} u_{\rm p})^2/\sum m_{\rm p} < \sum m_{\rm p} u_{\rm p}^2 \equiv$ (real kinetic energy) tells us that the internal energy calculated by the above formula is always larger than the real internal energy $S_{\rm r} \equiv \sum Q_{\rm p} - \sum m_{\rm p} u_{\rm p}^2/2$. We may call this process as "numerical viscosity" since the kinetic energy is converted to the internal energy by an averaging process. The difference $|Q - Q_{\rm r}|$ may depend on the square of the difference of $v_{\rm p}$ among particles, because $(a^2 + b^2)/2 - [(a+b)/2]^2 = [(a-b)/2]^2$. This idea forces us to put some spatial distribution of quantities within a particle.

Although there are several methods to reduce the numerical viscosity, here we consider only the conservative Eulerian methods. These are, for example, to subtract it in a finite difference form [6] and to assume a distribution of quantities within a particle for more correct convection as illustrated above. The former is hardly extended to multidimensional simulations because the accurate evaluations of the effective numerical diffusion is difficult, while the latter is more flexible because it tries to remove the origin of the numerical diffusions. The most simple distribution is

$$\begin{aligned} U(\xi, \eta) = {} & \{(U_{i+1,j+1} - U_{i+1,j})(2\eta - \delta\eta) + U_{i+1,j}\}(2\xi - \delta\xi) \\ & - \{(U_{i,j+1} - U_{i,j})(2\eta - \delta\eta) + U_{i,j}\}(2\xi - \delta\xi - 1), \end{aligned} \tag{13}$$

where the particle size is set to unity and the origin of the coordinate (ξ, η) is at the bottom left-hand corner of the particle in Fig. 1a. The physical quantities of four parts after convection are easily obtained by integrating $U(\xi, \eta)$ over each new area in Fig. 1b. Equation (13) preserves total amounts of physical quantities, such as momentum and energy, and if the particle does not move, no diffusion occurs. This is verified by integrating the above equation over each domain. For example, when the particle does not move, we integrate the equation over the area A_1 and obtain

$$\int_0^{\delta\eta} \int_0^{\delta\xi} U(\xi, \eta)\, d\xi\, d\eta = U_{i,j}.$$

This method removes the first-order truncation error in the space step Δx and rigorously conserves the total integration of the quantities U such as momentum and energy over an above space.

All these procedures are straightforwardly extended to a nonuniformly spaced grid system. If the arbitrarily spaced grid system is employed, the locations of particles in cells are only found with the comparison of the particles' positions and all the cell boundaries' positions. These operations are hard to accept. In order to simplify the operations, we employ a uniform subgrid system, whose spacing is δx and δy as shown in Fig. 2. The subgrid, to which a particle belongs, can be easily found by a usual method. Namely, if a particle has the position (x_p, y_p), the number (IS, JS) of the subgrid is calculated with

$$IS = \mathrm{INT}(x_p/\delta x) + 1,$$
$$JS = \mathrm{INT}(y_p/\delta y) + 1.$$

The cell, to which each subgrid belongs, is beforehand defined, that is, (IS, JS) is initially related to the cell number (i, j) for a fixed grid system or at each calculation step for a moving grid system. Consequently, the cell center coordinate (x_i, y_j) is given by

$$x_i = \sum_{k=1}^{i-1} \Delta x_k + \frac{\Delta x_i}{2}, \tag{14}$$

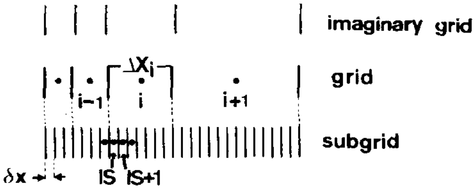

FIG. 2. For a nonuniformly spaced grid system, the grid spacings are taken to be an integral times of the subgrid, which is uniformly spaced and is taken to be as small as possible for a smooth grid movement. The grid to which a particle belongs can be easily found through such a double procedure. The relation of the grid i to the subgrid IS can be expressed by some function IGRID[IS], for example, IGRID[IS] = i, IGRID[IS + 1] = i.

$$y_j = \sum_{k=1}^{j-1} \Delta y_k + \frac{\Delta y_j}{2}. \tag{15}$$

The grid widths Δx_k and Δy_k are selected to the integral times of a small size δx and δy, respectively, for the convenience of the calculation of particle's location in the cell. In this method, grids move in a discrete manner, and hence δx and δy must be chosen as small as possible for a smoothly moving grid system. The grid motion is achieved by the following method. Initially we define "imaginary" grids whose arrangement is the same as that of real grids at initial time. The boundaries of the "imaginary" grids move with some velocities, which are assigned from outside, at every time step. If we specify the location of the boundary of the "imaginary" grid of grid i as X_i^{IB} at some time, the number IS_i^{IB} defined by

$$IS_i^{\mathrm{IB}} = \mathrm{INT}(X_i^{\mathrm{IB}}/\delta x + 0.5)$$

gives us the information of the grid location on the subgrid system. This relation of the grid to the subgrid IS can be expressed by some function IGRID $[IS]$. In the example depicted in Fig. 2, this function gives us $\mathrm{IGRID}[IS] = i$, $\mathrm{IGRID}[IS + 1] = i, \ldots$, etc. The grid width Δx_i can be calculated by

$$\Delta x_i = \{IS_{i+1}^{\mathrm{IB}} - IS_i^{\mathrm{IB}}\}\, \delta X.$$

Furthermore, real grid-boundary X_i^{RB} is given by

$$X_i^{\mathrm{RB}} = IS_i^{\mathrm{IB}}\, \delta x.$$

The same definition is used in another direction y. This means that the real grid-boundary moves in a discrete manner, whose minimum increments are δx and δy, and hence they should be as small as possible for a smooth grid movement. It should be noted that the "imaginary" grids move smoothly and do not depend on δx and δy. In this paper, grids move in a specific way, as given by the examples in the following sections, but the optimization problem of grid velocity is not discussed here.

In a nonuniform grid system, a particle size may be set to the minimum size of four adjacent cells. As the size of a particle changes in conjunction with the grid spacing, sometimes the fluid suffers a false compression or expansion if the grid spacings are poorly set up. There may be two possible methods to avoid the phenomena. One is to optimize the grid motion. The other is the particle generation and extinction. One of the procedures in the latter case is that the size of the particle is not determined from the grid spacings but from the fluid dynamics. The size of each particle is decided from the adjacent particles or is stored in the particle's memory. If the particle size spreads over three or more grid widths, the particle is divided into two or more to avoid the spreading of the particle over many cells, otherwise the distribution method is hard to employ. Although the particle generation technique can be easily introduced into the code and some test runs produced favorable results, it is not employed in the test runs shown in the following sections, in which our new scheme without particle generation is enough for the purpose. A new powerful scheme which includes the automated optimization of grid velocity and the particle generation will be discussed in another paper.

IV. Test Runs

Now we shall demonstrate the abilities of our scheme to the fluid dynamics. Some examples were shown in a previous note [14]. Here we make a more detailed discussion about the nature of our code. In this section, the discussion is limited to a planar configuration and the following problems are investigated; the 1-D adiabatic expansion process, the 1-D shock wave propagation and the Rayleigh–Taylor instability. The next section is devoted to the problems in a polar coordinate system with a brief comment on the extension of our scheme. All calculations have been performed on the FACOM M-200 computer at the Institute of Plasma Physics, Nagoya University.

A. *Adiabatic Expansion*

The PIC method is usually not good in the expansion process, because the particle density rarefies in that case and numerical noise grows intolerably. One of the countermeasures for it is the cloud-in-cell technique [13]. But it does not work well below the density, which is the minimum one determined by one particle. Usually the value is not sufficient for many calculations. Our scheme makes it possible by using the area-weighting procedure in a nonuniformly spaced moving grid and, if necessary, by the particle generation. In the PIC method, the numerical viscosity exists even for an expansion phase. The numerical viscosity is effective in the region where the local fluid velocity u_{R} relative to the grids is higher than the local sound speed, because the implicit numerical viscosity has a form of $\frac{1}{2}\rho\,|u_{\mathrm{R}}|\,\partial u_{\mathrm{R}}/\partial x\,\Delta x$. If the magnitude of u_{R} is greater than the local sound speed, viscosity pressure is comparable to the static pressure of the fluid. Our second-order scheme, that is, the velocity distribution technique, can successfully reduce the effect as shown below.

Let us show three example calculations for the adiabatic expansion: (1) the uniformly spaced fixed grid, (2) nonuniformly spaced fixed grid and (3) moving grids. In the first example, initially 20 particles per cell are loaded in uniform 20 grids and the same spaced grids are placed in a vacuum region. In the second example, initially 5 particles per cell are loaded in uniform 20 grids and the nonuniformly spaced grids, whose spacings are stretched adjusting the expansion profile, are placed in a vacuum region. In the moving grid method, only two particles per cell are loaded and two grids are supplemented in a vacuum region. These results are shown in Fig. 3 with the analytical solution denoted by the solid line. In such a range of density variation, the three methods produce almost the same result. The rarefaction wave front is smeared because of the inaccuracy. So the propagation speed of the rarefaction wave is slightly higher than the sound speed around the wave front. In the other region, where the higher components of the Fourier series are much

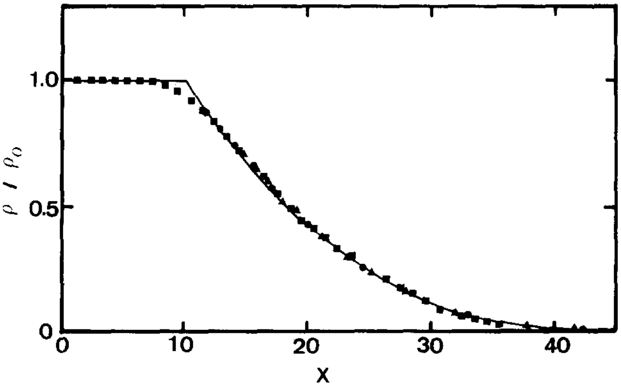

FIG. 3. Spatial profile of density in the adiabatic expansion process for $\gamma = 5/3$. Here, ■, ▲ and ● are the results obtained by the first-order scheme with uniformly spaced grid, the second-order scheme with nonuniformly spaced grid and the moving grid scheme, respectively.

smaller than the lower, a good agreement is obtained between the calculations and the analytic solution. The sonic point is located at $\rho = 0.422$ and hence the numerical viscosity is not remarkable for the density range shown in the figure. The viscous effect is clearly seen in the examples shown below.

In Fig. 4, the $p - \rho$ relations of the following three methods are compared. The filled rectangles are obtained making use of the nonuniformly spaced fixed grids and first-order repartition given by Eq. (12). A large density variation is realized but the $p - \rho$ relation is intolerably detached from the adiabatic relation at a lower density. The triangles are obtained making use of the same grids and second-order repartition given by Eq. (13). The numerical viscosity is reduced and the $p - \rho$ relation is tolerably improved. With the moving grid method, the fluid velocity relative to the grid can be suppressed much smaller than the sound speed even in much lower density. Then the adiabatic relation is maintained both for the first- and second-order

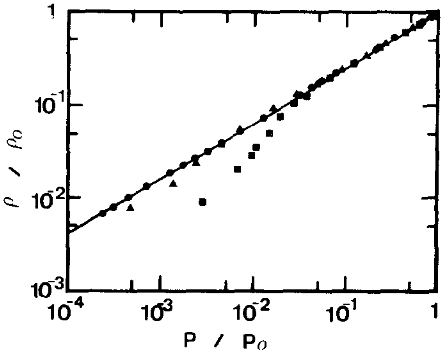

FIG. 4. Comparison among the theoretical adiabatic relation (the solid line) and simulation results, where ■, ▲ and ● are the results given by the first-order scheme in nonuniform fixed grid, the second-order scheme in nonuniform fixed grid and the first-order scheme in the moving grid, respectively.

repartition methods as shown by the circles in Fig. 4. In this case the moving velocity of the "imaginary" grid is 80% of the local fluid velocity and the subgrid spacing is one-twentieth of the initial grid spacing. These examples show us that the velocities of grid motion can be arbitrarily chosen to a certain degree.

B. *Reflected Shock Wave*

The second example is the production and propagation of a steady-state shock. The PIC method has treated many shock problems fairly well. Because our scheme reduces the intrinsic numerical viscosity, artificial viscosity is required to stabilize the calculation of shock wave. Even in the PIC method, the artificial viscosity of a form such as

$$\begin{aligned} q_A &= -a\rho C_s\, \partial u/\partial x\, \Delta x \qquad && \text{if} \quad \partial u/\partial x < 0, \\ &= 0 && \text{otherwise,} \end{aligned} \tag{16}$$

is required for the reflected shock wave discussed below because the intrinsic numerical viscosity is proportional to the local fluid velocity relative to the grid and is small in the stagnation region.

In our scheme, the artificial viscosity of a form

$$q = q_A + q_B \tag{17}$$

is employed [15], where q_B is the second-order or the von Neumann-type viscosity [16];

$$\begin{aligned} q_B &= b\rho(\partial u/\partial x)^2\, \Delta x^2 \qquad && \text{if} \quad \partial u/\partial x < 0, \\ &= 0 && \text{otherwise.} \end{aligned} \tag{18}$$

In the full particle method, different techniques are used to stabilize the overshooting. In the GAP method, the artificial relaxation constant is set to unity around the shock front and the scheme becomes more diffusive there, such as the scheme given by Eq. (12′). On the other hand, Leboeuf *et al.* [9] have used the averaging of pressure over a few grids at the shock front, which also makes the scheme more diffusive. We can use similar techniques in our scheme but prefer a more explicit way.

Let us describe the simulation results. Initially two particles per cell are loaded on a 200-cell system to produce a shock wave as shown in Fig. 5. The fluid initially moves with the velocity v_0 towards the left boundary, which is the rigid wall. The figure shows the time development of the density profile. Except for a few boundary cells, good agreement between numerical and analytic calculation is observed. The maximum error of the density is about 1% and pressure is 0.2% in this case. The shock front is smeared as usual and the width is about four grid spacings in case of a small overshooting as shown in the figure. In this example, we adopt $a = 1.2$ and $b = 1.6$ for the artificial viscosity given by Eq. (17). It should be noted that such a smooth profile in Fig. 5 can be obtained even with a few particles per cell, only two particles in this case.

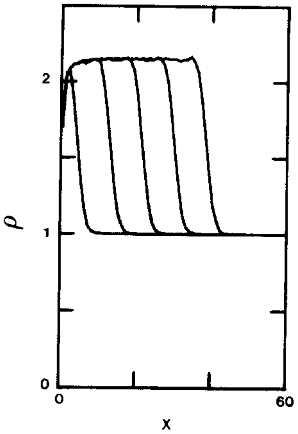

FIG. 5. Time evolution of the spatial density profile in the reflected shock wave. The fluid flows in from the right-hand side and the left boundary is a rigid wall.

Let us briefly discuss the selection of the artificial viscosity. To suppress the long-wavelength overshooting and numerical oscillation and minimize the width of the shock front, it is preferable to use the viscosity of lower order such as Eq. (16). But in the strong shock, the first-order viscosity such as Eq. (16) is small at the shock front. If only the viscosity of Eq. (16) is used, the coefficient should be widely changed adjusting to the strength of the shock wave for the best calculation. On the contrary, if only the second-order viscosity such as Eq. (18) is used for the suppression of the long-wavelength oscillations, the width of the shock front becomes very large, about 7–10 grid spacings. The accuracy of the Rankine–Hugoniot relation weakly depends on the species of the viscosity in our calculations. From the above reason, we employ a linear combination of the quadratic (q_B) and linear (q_A) viscosity.

C. *Rayleigh–Taylor Instability*

The third example is the Rayleigh–Taylor instability [17, 18]. A heavy fluid is superposed over a light fluid in a gravitational field. The initial perturbation is of the type $\nabla \cdot \vec{v} = 0$;

$$u = u_0 \sin(kx) * (2H(y) - 1) * \exp(-k\,|y|),$$
$$v = u_0 \cos(kx) * \exp(-k\,|y|),$$

where $\vec{v}$ is the fluid velocity, k the wavenumber, the initial interface coincides with the origin of the vertical axis ($y = 0$) and

$$H(y) = 1, \qquad y > 0,$$
$$= 0, \qquad y < 0.$$

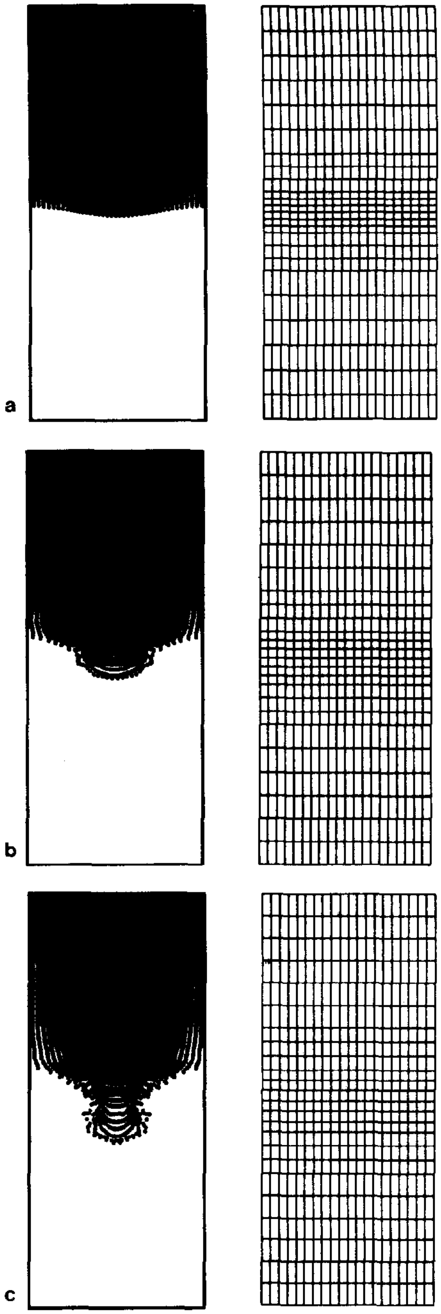

FIG. 6. Particle plots and grid arrangement at $\gamma_T t =$ (a) 1.92, (b) 3.21 and (c) 4.01. The second-order scheme is employed, the initial velocity is $u_0/C_s = 8.2 \times 10^{-3}$, and the density ratio of two superposed fluid is 10.

Then, the amplitude ξ of the surface wave evolves according to

$$\xi = u_0/\gamma_T \sinh(\gamma_T t)\cos(kx). \tag{19}$$

Here, the linear growth rate γ_T of R–T instability of the invicid incompressible fluid is $\sqrt{agk}$ and a is the Atwood number. At an early state ($\gamma_T t < 1$), Eq. (19) reduces to

$$\xi = u_0 t \cos(kx)$$

and the amplitude grows only with an initial velocity. Let us call this stage the "ballistic phase." When $\gamma_T t > 1$, Eq. (19) leads us to a so-called "linear phase";

$$\xi = u_0/(2\gamma_T)\, e^{\gamma_T t}\cos(kx).$$

When the amplitude grows large enough and $k\xi \sim 1$ is reached, Eq. (19) is no longer valid and the "nonlinear phase" sets in. Because a fine grid spacing is required for the description of initial growth, partially nonuniformly spaced grids are preferred. The initial behavior sometimes affects the nonlinear growth as shown later on. Some calculations [18] of the R–T instability employ a relatively large initial velocity and hence the nonlinear phase sets in around $\gamma_T t \sim 1$, that is, the nonlinear phase directly begins from the ballistic phase and the linear phase is not clearly observed.

Now let us discuss simulation results. Figure 6 shows the time evolution of instability, where the density ratio of two superposed fluids is 10 and the magnitude of the initial perturbation u_0/C_{s0} is set to 8.2×10^{-3}, where C_{s0} is the sound speed at

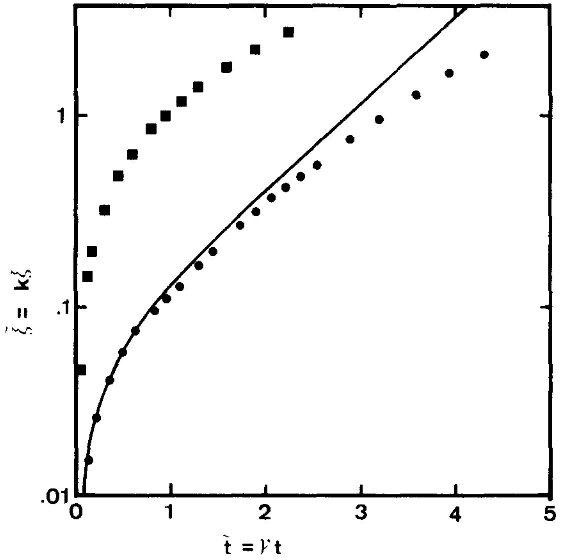

FIG. 7. Time evolution of the surface perturbation amplitude ξ for $u_0/C_s = 8.2 \times 10^{-3}$ (●) and $u_0/C_s = 3.3 \times 10^{-1}$ (■). The solid line shows the theoretical prediction in the former.

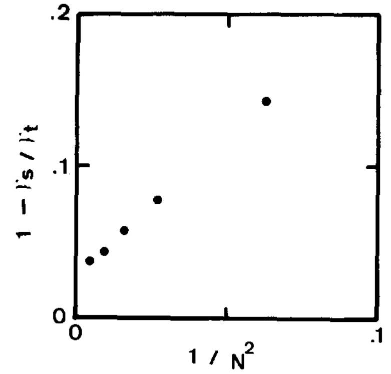

FIG. 8. Comparison of the growth rate between the theory and simulation for various grid spacings in the horizontal direction. The grid numbers in a half wavelength of perturbation are denoted by N. The data seem to be scaled as N^{-2} in the second-order scheme.

the top of the lower light fluid. In this case, the second-order scheme is employed with the moving 24 grids in the vertical direction and uniformly spaced fixed grids in the horizontal direction. The grids are forced to move adjusting to the amplitude of the surface instability. The time evolution of the amplitude is depicted in Fig. 7 and compared with the linear theory Eq. (19) denoted by the solid line. The ballistic, linear and nonlinear phases are clearly demonstrated. Because the grid moves in the vertical direction and its spacing is fine enough for the initial stage, the discrepancy between the theory and simulation in the figure mainly comes from the coarseness of the grid spacing in the horizontal direction. The linear growth rate γ_s in the simulation is compared with γ_T for various grid spacings in the horizontal direction and is depicted in Fig. 8. The result seems to be scaled as N^{-2}, where N is the grid

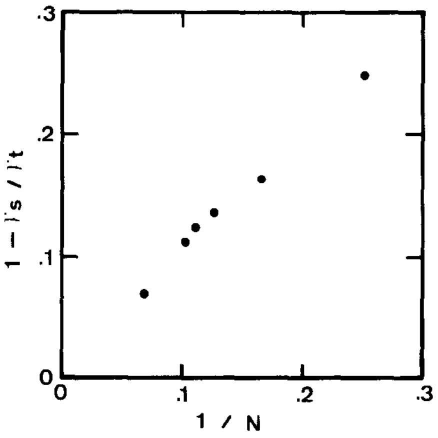

FIG. 9. Calculation of the same kind as in Fig. 8 for the first-order scheme, where the data seem to be scaled as N^{-1}.

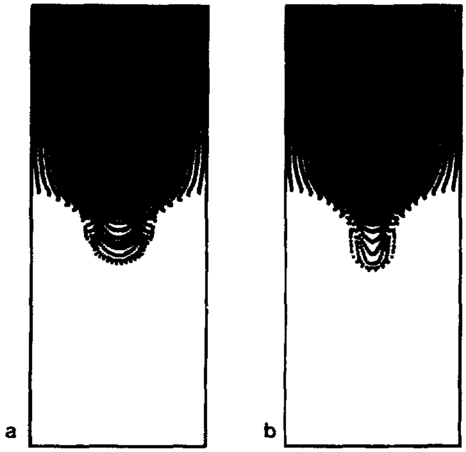

FIG. 10. Comparison of the profile in nonlinear phase. (a) The first-order scheme, $u_0/C_s = 8.2 \times 10^{-3}$, $\gamma_T t = 4.01$. (b) The second-order scheme, $u_0/C_s = 3.3 \times 10^{-1}$, $\gamma_T t = 0.48$. These figures are in the same phase as in Fig. 6c, which is obtained by the second-order scheme and $u_0/C_s = 8.2 \times 10^{-3}$. The numerical viscosity stabilizes the K–H instability (Fig. 10a). For a large initial perturbation, there is no time for the K–H instability to grow (Fig. 10b).

numbers in a half wavelength of perturbation, but γ_s does not tend to γ_T for $N^{-2} \to 0$. This may come from the inaccuracy of higher-order or finite grid spacing in the vertical direction and so forth.

For a comparison, the same test runs are repeated for the first-order scheme given by Eq. (12). In contrast to the second-order scheme, the result may be scaled as N^{-1} as shown in Fig. 9. This is a stimulating result. As it is shown, the second-order scheme can be used with less grids than the first-order scheme.

The difference between both schemes can be clearly seen in the nonlinear phase. In Fig 6, a mushroom-like structure is produced due to the Kelvin–Helmholz (K–H) instability [19]. On the contrary, it is not observed for the first-order scheme as shown in Fig. 10a. This means that the K–H instability is stabilized due to the numerical viscosity in the first-order scheme.

Before closing this section, we should comment on the initial velocity u_0 and its effects. Because the mushroom-like structure comes from the K–H instability, it needs a time to grow. If the initial velocity is large, the nonlinear phases directly begin from the ballistic phases as shown in Fig. 7 and the time to reach some amplitude becomes too short for the K–H instability to grow. This is clearly seen by comparing the results given by Fig. 10b, where a relatively large initial velocity is used, with the structure given by Fig. 6c. This structure in Fig. 10b is similar to that given by Daly *et al.* [18], who employed a relatively large initial velocity. From the result, we should note that the initial behavior often affects the nonlinear phase.

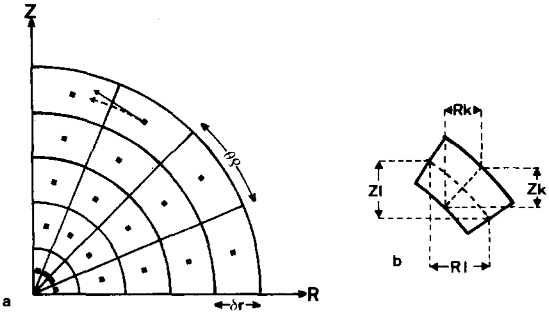

FIG. 11. (a) Schematics of the grid arrangement in polar coordinate system. The figure is rotationally symmetric around Z-axis. In a finite time step, the particle moves straightforwardly in Cartesian coordinate (solid arrow) and moves curvedly in polar coordinate (dashed arrow). This is an origin of artificial centrifugal force. (b) Projected cell width in $R-Z$ plane for the pressure work calculation.

V. EXTENSION TO POLAR COORDINATE SYSTEM

There exist many situations where the polar coordinate system is useful for a description of the fluid dynamics. The main problems, when the particle code is extended to the polar coordinate system, are the fictitious accelerations, for example, centrifugal or Coriolis acceleration, which originate from the curvilinear coordinate. This is schematically shown in Fig. 11, which is rotationally symmetric around the vertical axis. When a particle has a velocity V_θ only in the θ-direction and the particle location is advanced according to $d\theta/dt = V_\theta/r$, the location differs from that calculated by using $d\vec{x}/dt=\vec{u}$ in the Cartesian coordinate because of the finite time step Δt. A simple way to reduce the effect is to push the particle in the Cartesian coordinate $(R-Z)$. For this purpose, the velocity components (u, v) in $R-Z$ coordinate had better be calculated rather than $(V_r,\ V_\theta)$ even in the Eulerian phase. Projecting the momentum equations to the $R-Z$ plane, we obtain the following finite difference equations:

$$M^n_{i,j}\frac{\tilde{u}_{i,j}-u^n_{i,j}}{\Delta t^n} = -R^n_{i,j}Z^n_{1\ i,j}(p^n_{i+1/2,j}-p^n_{i-1/2,j}) + R^n_{i,j}Z^n_{k\ i,j}(p^n_{i,j+1/2}-p^n_{i,j-1/2}), \tag{20}$$

$$M^n_{i,j}\frac{\tilde{v}_{i,j}-v^n_{i,j}}{\Delta t^n} = -R^n_{i,j}R^n_{1\ i,j}(p^n_{i+1/2,j}-p^n_{i-1/2,j}) - R^n_{i,j}R^n_{k\ i,j}(p^n_{i,j+1/2}-p^n_{i,j-1/2}), \tag{21}$$

where Z_k, Z_1, R_k and R_1 are the effective length as shown in Fig. 11 and M is the total mass within a cell. This form can maintain the spherical symmetry. Associated with the energy equation, the integration over the volume of each cell gives us

$$\frac{\tilde{Q}_{i,j} - Q^n_{i,j}}{\Delta t^n} = -[S^n_{i+1/2,j}(p^n \bar{V}_r)_{i+1/2,j} - S^n_{i-1/2,j}(p^n \bar{V}_r)_{i-1/2,j}]$$

$$- [S^n_{i,j+1/2}(p^n \bar{V}_\theta)_{i,j+1/2} - S^n_{i,j-1/2}(p^n \bar{V}_\theta)_{i,j-1/2}], \qquad (22)$$

$$V_r = u \sin\theta + v \cos\theta, \qquad V_\theta = -u \cos\theta + v \sin\theta,$$

where Q is the total energy of the cell and S is the surface area of each cell. This form can rigorously conserve the total energy. The basic calculating sequences are the same as in the case of the rectangular cell.

In the Lagrangian phase, the fluid particles are transported in the $R - Z$ plane. The new particle position $(R^{n+1}_{\mathrm{p}}, Z^{n+1}_{\mathrm{p}})$ is calculated by

$$R^{n+1}_{\mathrm{p}} = R^n_{\mathrm{p}} + u_{\mathrm{p}} \Delta t^n, \qquad (23)$$

$$Z^{n+1}_{\mathrm{p}} = R^n_{\mathrm{p}} + v_{\mathrm{p}} \Delta t^n, \qquad (24)$$

where $(R^n_{\mathrm{p}}, Z^n_{\mathrm{p}})$ are the old particle position and u_{p} and v_{p} are the R, Z components of the particle velocity. Although particles move in the $R - Z$ plane, the area-weighting procedure should be done in the $r - \theta$ plane. For this purpose, we must transform the position $(R^{n+1}_{\mathrm{p}}, Z^{n+1}_{\mathrm{p}})$ into $(r^{n+1}_{\mathrm{p}}, \theta^{n+1}_{\mathrm{p}})$ using the inverse trigonometrical function. The inverse trigonometrical function requires much computation time which corresponds to a half of that of other parts of the particle transport. In our situation, this extra computation time is not crucial because other parts of the code, multigroup hot electron and radiation transport, heat conduction affected by magnetic field and so on [12], require much more computation time than does particle transport. The area-weighting should be carried out in the polar coordinate to make the transformations of the physical quantities between particle and cell easier, and to avoid fictitious fluxes. It should be emphasized that the mass distribution within a particle is uniform, namely, the cloud-in-cell procedure in the polar coordinate is not volume-weighting but cross-sectional-area-weighting of mass in order for the mass distribution in a particle not to change through a calculation. The fraction of mass partitioned from particle to cell and the contrary is proportional to each cross-sectional area normalized by r and θ, that is, the cross-sectional area of an imaginary "rectangular" cell in the $r - \theta$ plane. Accordingly, our area-weighting procedure in the polar coordinate is the same as that of the rectangular cells. The reason for this procedure is obvious. For simplicity, let us imagine that the fluid converges to the center with uniform velocity and a particle located at a cell center moves by a distance, which is half the length of cell in the r-direction. In our area-weighting, the change of mass distribution is half of its mass because the two parts of the shaded region in Fig. 12 have equal masses. On the contrary, in volume-weighting or real area-weighting, the two parts have different masses, because the volume and the real area change proportional to r^2 and r, respectively. So the mass ratios of two

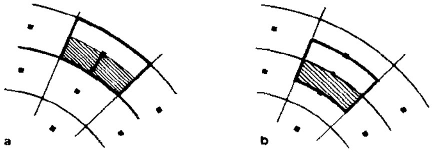

FIG. 12. Mass distribution of particle in the polar coordinate (a) before and (b) after transport. The particle moves in the r-direction with uniform velocity in this case. The oblique line cuts the length of the particle in half in the r-direction.

parts, the shaded and not shaded parts, are different in the two phases Fig. 12a and b. The ratio in Fig. 12a is less than that in Fig. 12b. This produces a fictitious mass flux toward the r-direction. In our area-weighting procedure, it is removed.

Associated with the distribution of the physical quantities in our second-order scheme, the same procedure as in Eq. (13) can be used with (ξ, η) being the particle coordinates in an imaginary "rectangular" $r-\theta$ plane as discussed above.

In the rest of this section, we show two example calculations; the imploding shock wave and the converging shock instability.

A. *Imploding Shock Wave*

A spherically symmetric stong shock wave travels towards the center of symmetry through a fluid having uniform initial density ρ_0. When the shock wave is strong

$$R = R_0(1 - t/t_0)^{\alpha}, \tag{25}$$

where R_0 is the initial radius, t_0 the collapse time and α the similarity exponent. In our calculation, the shock wave is generated by an outer high-pressure fluid, which encircles the lower-pressure fluid of uniform density ρ_0. All the fluid is initially at rest. The ratio of the two pressures in 5.3. Uniform fixed 90 or moving 30 grids in the radial direction and uniform fixed 5 grids in the azimuthal direction are used. Initially two particles per cell are loaded. The results are shown in Fig. 13. Here, the circles show the results obtained by moving grids or the second-order scheme (Eq. (13)) with fixed grids. The squares show the results obained by the first-order scheme (Eq. (12)) with fixed grids. In the case of moving grids, imaginary radial grids move with the local fluid velocity and the interval of the subgrids is 1/10 of the initial interval of the grid. The solid line denotes the analytical solution given by Eq. (25). The error bars in the figure imply the uncertainty of the collapse time. As in the previous section, the moving grid method or the second-order scheme gives good results. But in the first-order scheme, the numerical viscosity causes the lag of the shock propagation and hence the shock wave collapses late by 8% of the collapse time of other methods.

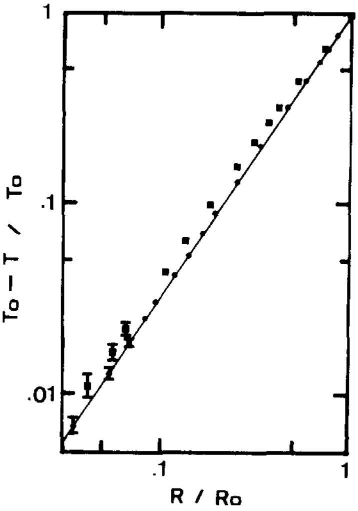

FIG. 13. Time evolution of shock locus in the converging shock wave. The second-order scheme with uniformly spaced fixed grid 90 × 5 ($r \times \theta$) and the first-order scheme with moving grid 30 × 5 produce good results (●). On the other hand, the first-order scheme with uniformly spaced fixed grid 90 × 5 shows the lag of the shock propagation (■).

B. *Converging Shock Instability*

As pointed out by Butler [21], the converging shock wave is unstable. If the shock front radius is perturbed from the average radius R by an amount ζ, the front vibrates and the relative amplitude ζ/R grows according to [22, 23]

$$\frac{\zeta}{R} \sim R^{-(\lambda+2)/(2\lambda) \pm i(\lambda+2)p/\lambda},$$

$$p = \frac{[4\lambda l(l+1) - (\lambda+2)^2]^{1/2}}{2(\lambda+2)}, \tag{26}$$

$$\lambda = \frac{\gamma+2}{\gamma} + \left(\frac{2\gamma}{\gamma-1}\right)^{1/2},$$

where γ is the specific heat ratio, l the mode number of the perturbation. In the test calculation the shock wave is produced by the same method as in Section V.A, but the interface of the two fluids is initially perturbed such as $R = R_0 + \zeta_0 P_8(\cos\theta)$, where $P_8(\cos\theta)$ is the Legendre function of mode 8. The time evolution of the shock front is depicted in Fig. 14. The amplitude oscillation and growth are clearly seen in Fig. 15, which shows the evolution of the relative amplitude ζ/R, where the average radius R of the shock front is used instead of time. In this case, $\gamma = 5/3$, $l = 8$ and hence $\lambda = 4.44$ and Eq. (26) reduces to

$$\zeta/R \sim R^{-0.725} \cos[3.96 \ln(R/R_0)]. \tag{27}$$

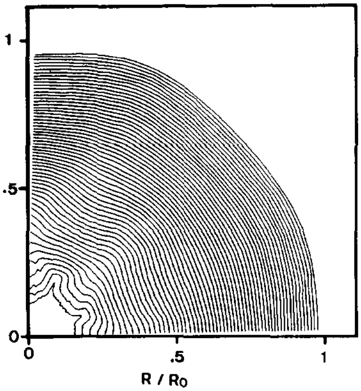

FIG. 14. Time evolution of the 2-D shock locus. The second-order scheme with uniformly spaced 90 × 40 grid is employed.

Therefore, the perturbation amplitude tends to zero at $R/R_0 = 0.673$, 0.3045, ---. The computed results show a good agreement with the linear analysis as for the frequency of the oscillation as shown in Fig. 15. But the growth rate does not coincide with the linear analysis. This may be due to the nonlinearity because ζ/R is more than 10% at the second peak in this case.

Let us examine the effects of the numerical viscosity and the grid spacing in the azimuthal direction. In the radial direction, the uniformly spaced fixed 90 grids are

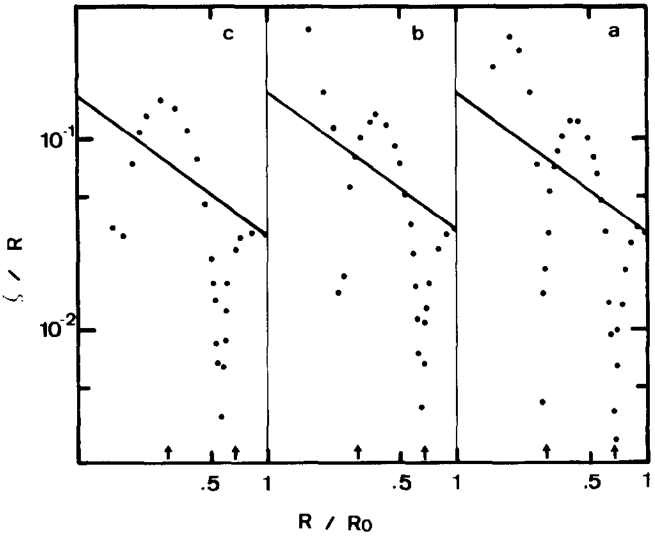

FIG. 15. Evolution of shock front perturbation in the first- and second-order schemes with uniformly spaced fixed grid (a) 90 × 40, (b) 90 × 30 and (c) 90 × 20. The period of amplitude oscillation depends on the grid coarseness. The solid lines denote the theoretical prediction of amplitude growth.

used in all calculations described below. As for the azimuthal direction, the test runs of the same kind are repeated with the uniformly spaced fixed 20, 30 and 40 grids both for the first- and the second-order schemes. Initially four particles per cell are loaded. The first nodal point, which is predicted to be $\zeta/R = 0.673$ by the linear theory, is compared among the computed results. The fewer the grids, the longer the oscillation period is.

Lastly, we should note that the oscillation characteristics have no difference regardless of the scheme, such as the first-order, the second-order or the moving grid methods, employed. The nodal points appear at almost the same radius in any method. The points mainly depend on the grid coarseness in the azimuthal direction. It should be noted, however, that the time evolution of the average radius R has different characteristics as pointed out in Section V.A.

VI. Summary

A new fluid particle code (SOAP) is developed. The numerical noise is suppressed, and the required particle number for calculations and the calculation time are substantially reduced. Due to the second-order accuracy in the advection, the numerical viscosity in the first order is removed. The scheme is different from a Lagrangian full particle method and tries to improve the accuracy of the advective term in a more Eulerian way. The variable spaced grids and moving grids improve the range of the variations of the physical quantities, the local fine resolution and the accuracy of the solutions.

An extension to the polar coordinate system is successfully done without the numerical centrifugal force, although the treatment of the pole at the center may be a subject of considerable debate.

There are many problems remaining for further improvement: the automated optimization of grid movement, the demonstration of the particle generation and extinction, the treatment of the reflection point, such as the center of the polar coordinate, the automatic operation of a troublesome initial setup of the particle and optimal grid spacings, especially in a nonuniform grid system, and so forth.

References

1. J. P. Boris and D. L. Book, *J. Comput. Phys.* **11** (1973), 38.
2. W. D. Schultz, *in* "Method in Computational Physics," Vol. 3, Academic Press, New York, 1964.
3. W. P. Crowley, Flag, a free-Lagrange method for numerically simulating hydrodynamics flow in two dimensions, *in* "Proceedings, Second International Conference on Numerical Methods in Fluid Dynamics," Springer–Verlag, Berlin/New York, 1971.
4. C. W. Hirt, A. A. Amsden, and J. L. Cook, *J. Comput. Phys.* **14** (1974), 227.
5. R. J. Gelinas and S. K. Doss, *J. Comput. Phys.* **40** (1981), 202.
6. M. W. Evans and F. H. Harlow, "The Particle-in-Cell Method for Hydrodynamic Calculations," Los Alamos Scientific Laboratory Report LA-2139, 1957; F. H. Harlow, *in* "Method in

Computational Physics," Vol. 3, Academic Press, New York, 1964; A. A. AMSDEN, "The Particle-in-Cell Method for the Calculation of the Dynamics of Compressible Fluids," Los Alamos Scientific Laboratory Report LA-3466, 1966.

7. B. M. MARDER, *Math. Comp.* **29** (1975), 434.
8. R. L. MCCRORY, R. L. MORSE, AND K. A. TAGGART, *Nuclear Sci. Engrg.* **64** (1977), 163.
9. J. N. LEBOEUF, T. TAJIMA, AND J. M. DAWSON, *J. Comput. Phys.* **31** (1979), 379.
10. A. NISHIGUCHI, T. YABE, N. UEDA, H. DOI, N. ASAMI, AND C. YAMANAKA, *Tech. Rep. Osaka Univ.* **31** (1981), 23.
11. S. I. BRAGINSKII, "Review of Plasma Physics," Vol. 1, Consultants Bureau, New York.
12. G. B. ZIMMERMAN, Lawrence Livermore Laboratory Report, UCLA-74811, 1973.
13. C. K. BIRDSALL AND D. FUSS, *J. Comput. Phys.* **3** (1969), 494.
14. A. NISHIGUCHI AND T. YABE, *J. Comput. Phys.* **47** (1982), 297.
15. R. LANDSHOFF, "A Numerical Method for Treating Fluid Flow in the Presence of Shocks," Los Alamos Scientific Laboratory Report LA-1930, 1955; M. L. WILKINS, *J. Comput. Phys.* **36** (1980), 281.
16. J. VON NEUMANN AND R. D. RICHTMYER, *J. Appl. Phys.* **21** (1950), 232.
17. L. RAYLEIGH, "Theory of Sound," Vol. 2, 2nd ed., Dover, New York, 1894; G. I. TAYLOR, *Proc. Roy. Soc. London Ser. A* **201** (1950), 192; F. H. HARLOW AND J. W. WELCH, *Phys. Fluids* **9** (1966), 842.
18. B. J. DALY, *Phys. Fluids* **10** (1967), 297; J. R. FREEMAN, M. J. CLAUSER, AND S. L. THOMPSON, *Nuclear Fusion* **17** (2) (1977), 223.
19. S. CHANDRASEKHAR, "Hydrodynamic and Hydromagnetic Stability," Oxford Univ. Press, London/New York, 1961.
20. K. P. STANYUKOVICH, "Unsteady Motion of Continuous Media," Gostekhizdat, Moscow, 1955. [English transl. (M. Holt, Ed.), Academic Press, New York, 1960]; G. GUDERLEY, *Luftfahrtforschung* **19** (1942), 302–312.
21. D. S. BUTLER, A.R.D.E. Report No. 18 (1956), 56.
22. W. CHESTER, *Philos. Mag.* **45** (1954) 1293; R. F. CHISNESS, *Proc. Roy. Soc. London Ser. A* **232** (1955), 350; G. B. WHITHAM, *J. Fluid Mech.* **2** (1057), 146.
23. J. H. GARDNER, D. L. BOOK, AND I. B. BERNSTEIN, *J. Fluid Mech.* **114** (1982), 41.