

The Constrained Interpolation Profile Method for Multiphase Analysis

Takashi Yabe,* Feng Xiao,*[†] and Takayuki Utsumi[†][‡]

**Department of Mechanical Engineering and Science, Tokyo Institute of Technology, O-okayama, Meguro-ku, Tokyo 152-8552, Japan;* [†]*Advanced Photon Research Center, Kansai Research Establishment, Japan Atomic Energy Research Institute, 25-1 Miiminami-machi, Neyagawa, Osaka 572-0019, Japan;* and [‡]*Frontier Research System for Global Change, SEAVANS North, 1-2-1 Shibaura, Minato-ku, Tokyo 105-6791, Japan*

Received December 16, 1999; revised August 3, 2000

We present a review of the constrained interpolation profile (CIP) method that is known as a general numerical solver for solid, liquid, gas, and plasmas. This method is a kind of semi-Lagrangian scheme and has been extended to treat incompressible flow in the framework of compressible fluid. Since it uses primitive Euler representation, it is suitable for multiphase analysis. The recent version of this method guarantees the exact mass conservation even in the framework of a semi-Lagrangian scheme. We provide a comprehensive review of the strategy of the CIP method, which has a compact support and subcell resolution, including a front-capturing algorithm with functional transformation, a pressure-based algorithm, and other miscellaneous physics such as the elastic–plastic effect and surface tension. Some practical applications are also reviewed, such as milk crown or coronet, laser-induced melting, and turbulent mixing layer of liquid–gas interface. © 2001 Academic Press

Key Words: numerical algorithm; semi-Lagrangian schemes; interface capturing; fluid dynamics; mass conservation; multiphase flows; surface tension; rheological materials.

1. INTRODUCTION

Recent high technology requires new tools for combined analysis of materials in different phase states, e.g., solid, liquid, and gas. A universal treatment of all phases by one simple algorithm would be useful and we are at the point of attacking this goal. For these types of problems, for example, welding and cutting processes, we need to treat topology and phase changes of the structure simultaneously, where the grid system aligned to the solid or liquid surface has no meaning and sometimes the mesh is distorted and even broken up. To solve these problems with Lagrangian representation in finite-difference, finite-element, and boundary-element methods will be a challenging task.

Even without phase change, solving the problem of structure–fluid interaction is not an easy task. In most cases, the grid cannot always be adapted to the solid surface. Therefore, the description of moving solid surfaces of complicated shapes in the Cartesian grid system is a challenging subject [57].

To attack the problems mentioned above, we must first find a method to treat a sharp interface and to solve the interaction of compressible gas with incompressible liquid or solid. For compressible fluid, elaborate schemes like TVD (total variation diminishing) or ENO (essentially nonoscillating) proved to be effective in capturing shock waves. However, since these schemes employ a conservative form of fluid equations, divergence of velocity which becomes zero in the incompressible limit cannot be treated independent of the advection part. Furthermore, as Karni [15] pointed out, the conservative algorithm sometimes gives fictitious pressure undulation at the boundary of multiphase materials.

On the other hand, incompressible schemes like QUICK or higher-order upwind schemes can treat divergence-free fluid vorticity and turbulence. However, these schemes cannot always treat a shock wave as a sharp discontinuity.

We need a scheme for treating both compressible and incompressible fluids with large density ratios simultaneously in one program to simulate the interaction of gas with a liquid or solid. Fully implicit solvers can handle this procedure, but the convergence of iteration in a highly distorted state is still a problem. Toward this goal, we take a Eulerian approach based on the CIP (cubic-interpolated propagation) method [40, 41, 60–62] which does not need an adaptive grid system and therefore eliminates the problems of grid distortion caused by structural breakup and topology change. The material surface can almost be captured by one grid throughout the computation [64, 65]. Furthermore, the code can treat all the phases of matter from solid state through liquid and from two-phase state to gas without restriction on the time step from high-speed [63].

A pressure-based algorithm coupled with a semi-Lagrangian approach such as the CIP proved to be stable and robust in analyzing these subjects. One disadvantage of this method was the lack of conservative property. Recent versions of the CIP-CSL4 (conservative semi-Lagrangian) [43] can overcome this difficulty and provide exactly a conservative semi-Lagrangian scheme. Since these schemes do not use cubic polynomials but rather different orders of polynomials, we have renamed these CIP families “constrained interpolation profile” and kept the abbreviation CIP. This means that various constraints such as the time evolution of a spatial gradient, which is used in the original CIP method, or spatially integrated conservative quantities can be used to construct the profile. In this paper, we review the CIP method and related schemes and address these important subjects.

In Section 2, the CIP method is derived for advection calculation. After a description of the fundamental idea of the method with a 1D problem, some variants and practical extensions are discussed. They include, for example, multidimensional formulation, oscillation-suppressing interpolation, and sharpness enhancement. A formulation of the CIP method applied to general hydrodynamics is presented in Section 3. Section 4 discusses a unified procedure to compute both compressible and incompressible fluids with several examples. Some numerical formulations for interfacial flows, including surface tension and the elastic–plastic effect, are presented in Section 5. As a recent improvement to the CIP method, conservative variants of the method are briefly discussed with a few numerical examples. The paper ends with a short summary referencing some applications in other fields.

2. THE ADVECTION PROCESS

2.1. CIP Formulation in One Dimension

Although nature operates in a continuous world, a discretization process is unavoidable for implementing numerical simulations. The primary goal of any numerical algorithm will be to retrieve the information lost inside the grid cell between these digitized points. The CIP method proposed by one of the authors tries to construct a solution inside the grid cell that is close enough to the real solution of the given equation, with some constraints. We here explain the strategy of the CIP method by using an advection equation,

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0. \quad (1)$$

When the velocity is constant, the solution of Eq. (1) gives a simple translational motion of a wave with velocity u . The initial profile (the solid line in Fig. 1a) moves like a dashed line in a continuous representation. At this time, the solution at grid points is denoted by circles and is the same as the exact solution. However, if we eliminate the dashed line as in Fig. 1b, then the information about the profile inside the grid cell has been lost, and it is difficult to imagine the original profile and natural to imagine a profile such as that shown by the solid line in (c). Thus, numerical diffusion arises when we construct the profile by the linear interpolation, even with the exact solution shown in Fig. 1c. This process is called the first-order upwind scheme. On the other hand, if we use a quadratic polynomial for interpolation, the model suffers from overshooting. This process is called the Lax–Wendroff scheme or the Leith scheme [18].

What made this solution worse? This decline in accuracy is the reason we neglect the behavior of the solution inside the grid cell and merely follow the smoothness of the solution. From this consideration, we can see that it is important to develop a method incorporating the real solution into the profile within a grid cell. We propose an approximation of the profile as shown below. If we differentiate Eq. (1) with spatial variable x , we get

$$\frac{\partial g}{\partial t} + u \frac{\partial g}{\partial x} = -\frac{\partial u}{\partial x} g, \quad (2)$$

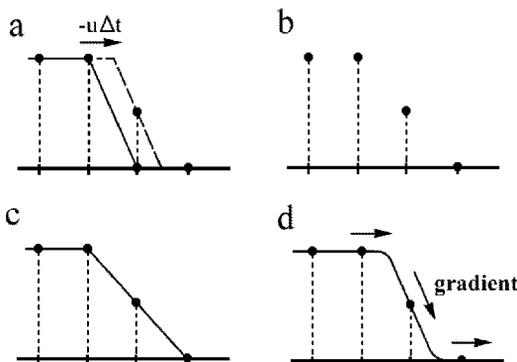


FIG. 1. The principle of the CIP method. (a) The solid line is the initial profile and the dashed line is an exact solution after advection, shown in (b) at discretized points. (c) When (b) is linearly interpolated, numerical diffusion appears. (d) In the CIP, the spatial derivative also propagates and the profile inside a grid cell is retrieved.

where g stands for the spatial derivative of f , $\partial f/\partial x$. In the simplest case where the velocity u is constant, Eq. (2) coincides with Eq. (1) and represents the propagation of a spatial derivative with velocity u . Using this equation, we can trace the time evolution of f and g on the basis of Eq. (1). If g is predicted after propagation as shown by the arrows in Fig. 1d, the profile after one step is limited to a specific profile. It is easy to imagine that by this constraint, the solution becomes much closer to the initial profile that is the real solution. Most importantly, the solution thus created gives a profile consistent with Eq. (1) even inside the grid cell. The importance of this constraint is demonstrated in the next section.

If two values of f and g are given at two grid points, the profile between these points can be interpolated by the cubic polynomial $F(x) = ax^3 + bx^2 + cx + d$. Thus, the profile at the $n + 1$ step can be obtained by shifting the profile by $u\Delta t$ so that

$$\begin{aligned} f^{n+1} &= F(x - u\Delta t), \\ g^{n+1} &= dF(x - u\Delta t)/dx. \end{aligned} \quad (3)$$

$$\begin{aligned} a_i &= \frac{g_i + g_{iup}}{D^2} + \frac{2(f_i - f_{iup})}{D^3}, \\ b_i &= \frac{3(f_{iup} - f_i)}{D^2} - \frac{2g_i + g_{iup}}{D}, \\ f_i^{n+1} &= a_i\xi^3 + b_i\xi^2 + g_i^n\xi + f_i^n, \\ g_i^{n+1} &= 3a_i\xi^2 + 2b_i\xi + g_i^n, \end{aligned} \quad (4)$$

where we define $\xi = -u\Delta t$. Here, $D = -\Delta x$, $iup = i - 1$ for $u \geq 0$ and $D = \Delta x$, $iup = i + 1$ for $u < 0$. Figure 2a shows a profile after 1000 steps with this CIP method for the propagation of a square wave.

The CIP advection scheme can be sorted out as a kind of semi-Lagrangian method in the sense that the CIP advection scheme employs a Lagrangian invariant solution. Semi-Lagrangian methods that trace back along the characteristics in time depend on an interpolation of the initial profile to determine the value at the upstream departure point (which may not coincide with the computational grid point).

Although there are various polynomial functions such as linear, quadratic Lagrange, cubic Lagrange, cubic spline, and quintic Lagrange [37], all of these schemes (except those using the linear interpolation function) need at least three points for constructing interpolation approximations in one dimension. A more compact scheme by which one

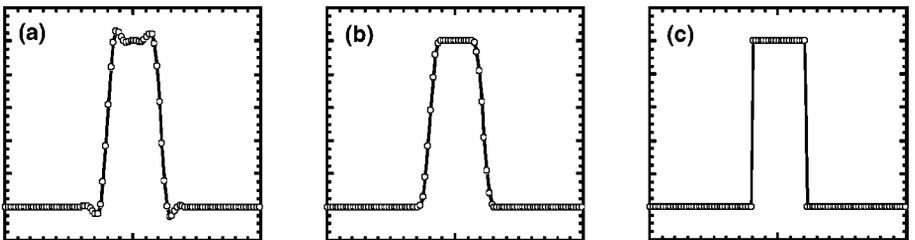


FIG. 2. (a) Initial condition, and the profile after one complete revolution with (b) CIP, and (c) rational CIP and tangent-transformed CIP.

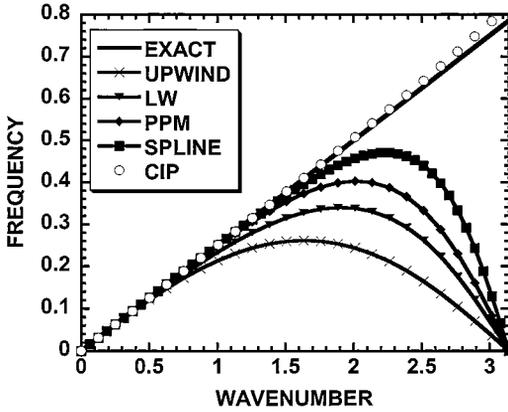


FIG. 3. Phase error of various schemes such as first-order upwind, Lax–Wendroff (LW), PPM, Spline, and CIP.

can construct interpolation functions of high accuracy with fewer computational stencils is desired in many situations, such as calculating discontinuities or large gradients. A scheme that is based on a compact support is able to localize dispersion errors to the regions where large local gradients appear. Moreover, in a model with a limited computational domain, different approximations for the derivatives must be used at the grid points close to boundaries; these approximations are usually of lower order than the approximations used deeper in the interior. Thus, a scheme that uses fewer stencils may be advantageous in treating computational boundaries since fewer boundary points need to be handled. Another attractive feature of reducing the number of stencils may be the reduction in data transfer in parallel implementations on distributed memory architectures. In this sense, the CIP seems to be attractive since it uses only one cell for computation even in three dimensions.

2.2. Mathematical Analysis of CIP

It is interesting to examine the phase error of various schemes using the method proposed by Purnell [34] and Utsumi *et al.* [46]. Figure 3 summarizes those results. As is well known, phase speeds in conventional schemes depart from the exact speed, shown by the solid line, which is around $k\Delta x = \pi/2$. Surprisingly, however, the CIP can reproduce the correct phase speed even up to $k\Delta x = \pi$. This is remarkable because $k\Delta x = \pi$ means that one wavelength is described by three mesh points. Let us consider the case shown in Fig. 4, where values of the three points are zero. Even in this case, one wave can exist as shown in Fig. 4. The CIP gives correct spatial gradients, which are non-zero at these points, and

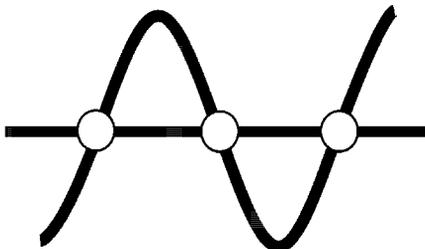


FIG. 4. The CIP can correctly recognize one wavelength with three grid points.

therefore it recognizes the existence of the wave inside the grid cell. Any code that only takes into account this value, which is zero here, cannot correctly recognize the wave even if a higher order polynomial is employed.

The importance of Eq. (2), which predicts the propagation of gradients, can be clearly demonstrated by comparison with the cubic spline [34]. Although the cubic spline uses the same cubic polynomial as the CIP, it cannot reproduce the result of the CIP, because the gradient of the spline is determined merely from the smoothness requirement. As is easily recognized, a constraint that is independent of the original equation will not help to retrieve the profile inside the grid cell.

A possible objection to the CIP method is that it uses both a function and its derivative and seems to be the same as other schemes that use twice as many mesh points. We address the following points against this objection.

(1) The cubic spline shown in Fig. 3 uses both a function and its derivative, as does the CIP, but fails to give a similar result.

(2) The calculation costs do not increase even if an additional variable such as a derivative is introduced. For example, the CPU time required for a simple advection problem in one dimension is $\text{Spline/CIP} = 1.68$ (the Thomas method is used to solve the matrix), $\text{RCIP (rational CIP)/CIP} = 1.77$, $\text{PPM (piecewise parabolic method) [6]/CIP} = 2.31$. RCIP will be defined in Section 2.4.

(3) The memory increase owing to an additional variable is less significant for higher dimensions. Let L_t be the number of time-dependent variables; then memory increases as $(\alpha + 1)L_t$ in the CIP and as $n^{(\alpha+1)}L_t$ in conventional schemes if $\Delta t/\Delta x$ is fixed, where α is the dimension and n is the mesh refinement. These rates will be $4L_t$ and $16L_t$ in three dimensions for the CIP and for twice-refined ($n = 2$) conventional schemes, respectively.

(4) Adding to (3), in most practical applications, we use additional variables such as thermal conductivity, viscosity, and temporal variables for matrix solution. If L_0 is the number of these variables, the memory requirement is then $(\alpha + 1)L_t + L_0$ and $n^\alpha L_t + L_0$. Since $L_0 \gg L_t$ usually, the memory requirement of the CIP is similar to that of other schemes even for unrefined mesh $n = 1$.

2.3. CIP Formulation in Multidimensions

In the CIP method, the first-order spatial derivatives of the interpolation function are treated as dependent variables as shown in the previous sections. The governing equations for those derivatives in multidimensions are derived by differentiating the advection equation with respect to the spatial coordinates. This scheme is different from the conventional semi-Lagrangian methods for computation of derivatives. In the latter methods, as mentioned above, the gradient is calculated based on the function values at neighboring grid points by either assuming the continuity of the quantity, or of the first- and sometimes the second-order derivatives of the quantity at the mesh boundaries [34], or by using approximations based on local grid points [51]. By special treatment of the first derivatives of the interpolation function, the CIP method achieves a compact form that uses only one mesh cell to construct the interpolation profile and provides subcell resolution.

The model equation to be solved is an advection equation,

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} + \mathbf{u} \cdot \nabla f(\mathbf{x}, t) = 0, \quad (5)$$

where \mathbf{x} means (x, y, z) , $\mathbf{u} = (u(x, y, z, t), v(x, y, z, t), w(x, y, z, t))$ is the characteristic speed, and $f(x, y, z, t)$ is the dependent variable. The gradient operator is defined as $\nabla = (\partial_x, \partial_y, \partial_z)$.

Semi-Lagrangian methods usually make use of the solution as a Lagrangian invariant,

$$f(\mathbf{x}, t) = f(\mathbf{x} - \hat{\mathbf{x}}, t - \Delta t), \quad (6)$$

where $\hat{\mathbf{x}}$ is the distance a particle travels in time increment Δt given by $d\mathbf{x}/dt = \mathbf{u}$. If it is possible to trace the trajectory across several grid points in Δt , Eq. (6) can be used even for a large CFL number $\equiv u\Delta t/\Delta x$. One of the main efforts in semi-Lagrangian methods is constructing the interpolation function based on grid values to determine the field value at a departure point not coinciding with a grid point.

A CIP-type method evaluates the first-order derivatives of the profile by treating the derivatives as additional dependent variables and adding an equation, derived from Eq. (5) as

$$\frac{\partial(\partial_\chi f)}{\partial t} + \mathbf{u} \cdot \nabla(\partial_\chi f) = -\frac{\partial \mathbf{u}}{\partial \chi} \cdot \nabla f, \quad (7)$$

where χ denotes the space coordinate x, y , or z , and $\partial_\chi f$ represents $\partial f/\partial \chi$.

One then gets an additional advection equation in each dimension with a forcing term for the first-order derivative of that respective dimension. The set of governing equations for the three-dimensional advection problem is consequently written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{U} = \mathbf{G}, \quad (8)$$

where

$$\mathbf{U} = \begin{pmatrix} f \\ \partial_x f \\ \partial_y f \\ \partial_z f \end{pmatrix}$$

and

$$\mathbf{G} = \begin{pmatrix} 0 \\ -\partial_x u \partial_x f - \partial_x v \partial_y f - \partial_x w \partial_z f \\ -\partial_y u \partial_x f - \partial_y v \partial_y f - \partial_y w \partial_z f \\ -\partial_z u \partial_x f - \partial_z v \partial_y f - \partial_z w \partial_z f \end{pmatrix}.$$

A CIP-type scheme is equivalent to the two-step semi-Lagrangian procedure

$$\tilde{\mathbf{U}}(\mathbf{x}, t) = \mathbf{U}_h(\mathbf{x} - \hat{\mathbf{x}}, t - \Delta t) \quad (9)$$

and

$$\mathbf{U}(\mathbf{x}, t) = \tilde{\mathbf{U}}(\mathbf{x}, t) + \int_\tau \mathbf{G}(\tilde{\mathbf{U}}) d\tau, \quad (10)$$

where \mathbf{U}_h represents the interpolation approximation to \mathbf{U} , and τ denotes the trajectory that connects $(\hat{\mathbf{x}}, t - \Delta t)$ and (\mathbf{x}, t) .

Once the interpolation function is determined from the continuity condition imposed on the dependent variable and its first-order derivatives at the grid points, we immediately get the solution to the advection equation from Eqs. (9) and (10).

Several forms of the multidimensional cubic polynomial have been proposed [2, 62]. Among various families of polynomial, the simplest one is written as [62]

$$F_{i,j}(x, y) = C_{3,0}X^3 + C_{2,0}X^2 + f_{xi,j}X + f_{i,j} + C_{0,3}Y^3 + C_{0,2}Y^2 + f_{yi,j}Y + C_{2,1}X^2Y + C_{1,1}XY + C_{1,2}XY^2, \quad (11)$$

which has a form consistent with one-dimensional CIP in one direction. Here we define $X = x - x_i$, $Y = y - y_j$. Thus 10 unknowns are determined from the continuity of f ; $\partial f/\partial x$; $\partial f/\partial y$ at (i, j) , $(i + 1, j)$, $(i, j + 1)$; and the continuity of f at $(i + 1, j + 1)$.

Although Eq. (11) is not the only candidate for the interpolation in two dimensions (a variant is in fact presented in [2]), it is sufficient for a small time step $u\Delta t/\Delta x < 0.5$ as one can imagine from the definition: $\partial f/\partial x$, $\partial f/\partial y$ at $(i + 1, j + 1)$ are not continuous and are not accurate. In Fig. 5, we show the result of a two-dimensional solid-body revolution [74]. Figure 5a shows an initial profile and Fig. 5b the profile after one revolution when Eq. (11) is applied. Although the scheme is simple, it can produce a result comparable to existing higher order schemes. If we apply tangent transformation (which will be introduced in Section 2.5) to this problem, the result is much improved, as shown in Fig. 5e.

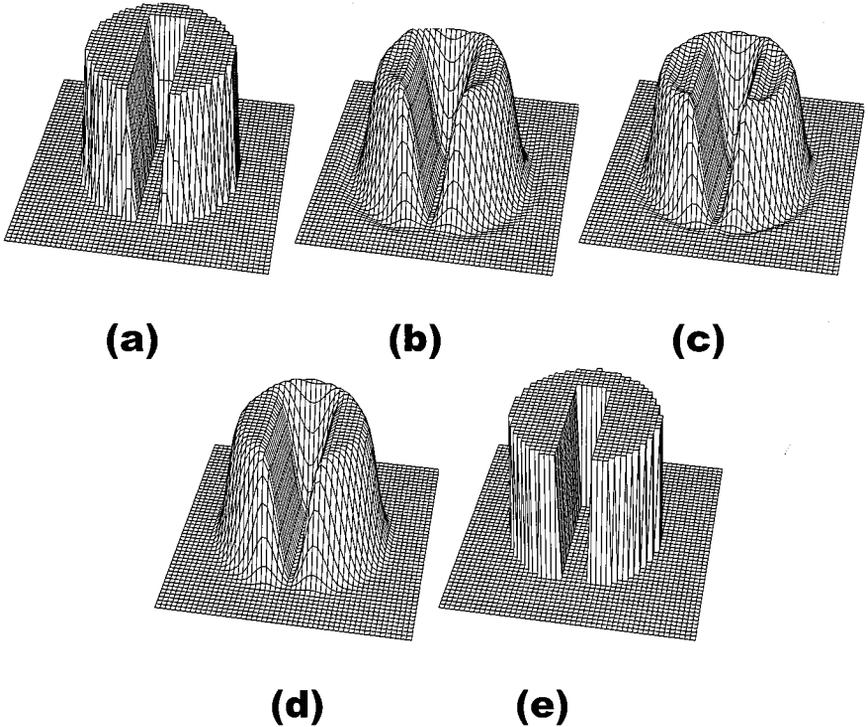


FIG. 5. (a) Initial condition and the profile after one complete revolution with (b) CIP, (c) splitting CIP, (d) rational CIP, and (e) tangent-transformed CIP.

In these cases, the multidimensional profile is shifted in the direction of velocity vector \mathbf{u} . In addition, a directional splitting technique can be used to perform sequential one-dimensional advection in each direction. The latter method is very promising because it would be difficult to construct a cubic polynomial in six-dimensional hyperspace such as the Vlasov equation, and a conservative scheme can be readily constructed using this splitting scheme [21]. Figure 5, which includes the result of this scheme, shows that the symmetry of the profile is preserved well.

2.4. The RCIP Scheme for the Advection Equation

In practical implementation, an attractive advection scheme should be both low in diffusion and free of oscillation. In modern Eulerian high-resolution schemes, manipulations such as numerical viscosity are usually done to degrade the scheme to a lower order in the presence of discontinuities in order to eliminate spurious oscillation. Some of these schemes are reviewed by Toro [44]. Preserving the shape of the advected field is also one of the primary goals for semi-Lagrangian schemes, since a scheme with an interpolator higher than second order will produce spurious oscillations near large gradients or discontinuities. Williamson and Rasch [51] discuss several shape-preserving interpolation schemes. The monotonicity of a scheme is improved by imposing derivative constraints on a Hermite cubic or a rational-cubic interpolation function. Bermejo and Staniforth [4] also reported their work on overcoming the numerical oscillation for semi-Lagrangian schemes, where a minimum/maximum limiter is imposed on the calculated results from any conventional semi-Lagrangian scheme, based on the argument that no new extremum should be created by an advection scheme.

The original CIP method, which uses a cubic polynomial interpolant, produces numerical oscillations near the area where the dependent variable has a degree of smoothness no more than 1, which is only C^0 for a step or a triangle distribution. A numerical scheme to get a shape-preserving scheme for a CIP-type method was developed from a rational interpolation function over one mesh cell [54, 55]. This scheme, which we call the RCIP (rational-cubic interpolation propagation) scheme, shows good properties in keeping the topological nature of data, such as preserving convexity or concavity.

The 1D scheme is based on the rational function

$$R_i^{\text{1D}}(x) = \sum_{0 \leq l_i \leq 3} \left(\sum_{0 \leq p \leq 1} \alpha_p \beta_p X^p \right)^{-1} C_{l_i} X^{l_i}, \quad (12)$$

where $X = x - x_i$ and

$$\begin{cases} C_0 = f_i \\ C_1 = g_i + f_i \alpha_1 \beta_1 \\ C_2 = S_i \alpha_1 \beta_1 + (S_i - d_i) \Delta_i^{-1} - C_3 \Delta_i \\ C_3 = [g_i - S_i + (g_{\text{iup}} - S_i)(1 + \alpha_1 \beta_1 \Delta_i)] \Delta_i^{-2} \\ \alpha_0 = 1.0 \\ \beta_0 = 1.0 \\ \beta_1 = [|(S_i - g_i)/(g_{\text{iup}} - S_i)| - 1] \Delta_i^{-1}; \end{cases}$$

here

$$\begin{cases} \Delta_i = x_{iup} - x_i \\ S_i = (f_{iup} - f_i)\Delta_i^{-1} \\ g_i = (df/dx)_i. \end{cases}$$

In deriving the above coefficients, we first determined β_1 by setting $C_3 = 0$, which means a second-order polynomial is the numerator of Eq. (12) because β_1 should be effective at the discontinuity where the scheme becomes lower order. Therefore four constraints from f and g at neighboring points are sufficient to determine these coefficients. After determining β_1 , we use it as a fixed value and then recover C_3 and uniquely determine the cubic polynomial by the same four constraints [55].

Then a solution to the advection equation can be written as

$$f_i^{n+1} = R_i^{1D}(x_i - u\Delta t) = \frac{C_0 + C_1\xi + C_2\xi^2 + C_3\xi^3}{1 + \alpha_1\beta_1\xi}, \quad (13)$$

and the first-order derivative of the dependent variable is calculated by

$$g_i^{n+1} = \partial_x f_i^{n+1} = \partial_x R_i^{1D}(x_i - u\Delta t) = (C_1 + 2C_2\xi + 3C_3\xi^2)(1 + \alpha_1\beta_1\xi)^{-1} - \alpha_1\beta_1(C_0 + C_1\xi + C_2\xi^2 + C_3\xi^3)(1 + \alpha_1\beta_1\xi)^{-2} - \left(\frac{\partial u}{\partial x} \frac{\partial f}{\partial x}\right)_i^n, \quad (14)$$

where $\xi = -u\Delta t$. All the coefficients in the above expressions can be computed from the quantities of f and g at step n . It is interesting to see that the second term on the right-hand side of Eq. (14) is proportional to $-\alpha_1\beta_1$ and hence plays a role in reducing the contribution from the highest order polynomial, the $\alpha_1\beta_1 C_3\xi^3$ term, to g at the discontinuity.

The parameter $\alpha_1 \in [0, 1]$ provides flexibility to choose between a rational function and a cubic function for interpolation. In practical computations, we recommend the following switching formulation to put the interpolant “cubic” in the smooth region and the “rational” near discontinuity:

$$\alpha_1 = \begin{cases} 1, & g_i \cdot g_{iup} < 0; \\ 0, & g_i \cdot g_{iup} \geq 0. \end{cases} \quad (15)$$

The subscript *iup* indicates an upwind grid point. The smoothness is detected by seeing whether the first-order derivative, which is computed as a dependent variable in the CIP method, has the same sign at neighboring grid points. When the switching parameter α_1 is set to zero, algorithm (13), (14) is identical to the original CIP method. Let $\alpha_1 = 1$; under the CFL condition, scheme (13), (14) is convex–concave preserving and monotone preserving if the given data are nonconcave or nonconvex, as proved in Ref. [54].

Numerical experiments, shown in Fig. 2b demonstrate that the scheme of Eqs. (13) and (14) is capable of suppressing the spurious oscillation near discontinuities.

Fully two- and three-dimensional schemes can also be constructed by making use of the following interpolation functions (see [55] for details).

$$R_{i,j}^{2D}(x, y) = \left(\sum_{0 \leq p+q \leq 1} \alpha_{p,q} \beta_{p,q} X^p Y^q \right)^{-1} \sum_{0 \leq l_x + l_y \leq 3} C_{l_x, l_y} X^{l_x} Y^{l_y}, \quad (16)$$

and

$$R_{i,j,k}^{3D}(x, y, z) = \left(\sum_{0 \leq p+q+r \leq 1} \alpha_{p,q,r} \beta_{p,q,r} X^p Y^q Z^r \right)^{-1} \sum_{0 \leq l_x+l_y+l_z \leq 3} C_{l_x,l_y,l_z} X^{l_x} Y^{l_y} Z^{l_z}. \quad (17)$$

Figure 5 includes the test run with this RCIP scheme.

2.5. Interface Tracking: A Sharpness-Preserving Method

Treatment of the interface that lies between materials of different properties remains a formidable challenge to the computation of multiphase fluid dynamics. Eulerian methods have proven robust in simulating flows with interfaces of complex topology. Generally, Eulerian methods use color function to distinguish among regions containing different materials. To accurately reproduce the physical processes that occur across the interface transition region, maintaining the compact thickness of the interface is very important. The finite-difference schemes constructed on a Eulerian grid, however, intrinsically produce numerical diffusions to the solution of the advection equation by which the interface is predicted temporally. Thus, the direct implementation of finite-difference schemes (even those of high order) cannot maintain the compactness of the interface.

Various kinds of methods have been developed to achieve a compact and correctly defined interface by introducing extra programming. Among the most common algorithms are the level-set methods and the VOF (volume-of-fluid) methods for front capturing, and others for front tracking [45]. The level set method that was first proposed by Osher *et al.* [24, 29, 38] gets around the computation of interfacial discontinuity by evaluating the field in higher dimensions. The interface of interest is then recovered by taking a subset of the field. Practically, the interface is defined as the zero-level set of a distance function from the interface.

In VOF methods, on the other hand, the interface needs to be reconstructed based on the volume fraction of fluid. VOF methods are mainly classified as SLIC (simple line interface calculation) algorithms and PLIC (piecewise linear interface calculation) algorithms according to the interpolation function used to represent the interface. The SLIC [11] algorithm makes use of piecewise constant reconstruction, and the interfaces are approximated by lines aligned with mesh coordinates. A significant improvement in the VOF method was made by Youngs with the PLIC algorithm [71]. Since then, some improvements in the reconstruction of the VOF interface have been reported [16, 32, 33]. The PLIC algorithm estimates the interface with a truly piecewise linear approximation that greatly improves the geometrical faithfulness of the method. A comparison of various methods for tracking interfaces can be found in [35].

In [64] and [65], we devised an interface tracking technique that appears to be efficient, geometrically faithful, and diffusionless. The method is a combination of the CIP advection solver and a tangent function transformation.

Consider K kinds of impermeable materials occupying closed areas $\{\Omega_k(t), k = 1, 2, \dots, K\}$ in computational domain $D \in \mathbf{R}^3(x, y, z)$. We identify them with color functions or density functions $\{\phi_k(x, y, z, t), k = 1, 2, \dots, K\}$ according to the following definition.

$$\phi_k(x, y, z, t) = \begin{cases} 1, & (x, y, z) \in \Omega_k(t), \\ 0, & \text{otherwise.} \end{cases}$$

Suppose these materials move at the local speed; then the color functions evolve according to the advection equation

$$\frac{\partial \phi_k}{\partial t} + \mathbf{u} \cdot \nabla \phi_k = 0, \quad k = 1, 2, \dots, K, \quad (18)$$

where \mathbf{u} is the local velocity.

It is known that solving the above equation by finite-difference schemes in a Eulerian representation will produce numerical diffusion and tend to smear the initial sharpness of the interfaces. In our method, rather than the original variable ϕ_k itself, its transformation, say $F(\phi_k)$, is calculated by the CIP method. We specify $F(\phi_k)$ to be a function of ϕ_k only, which means that the new function $F(\phi_k)$ is also governed by the same equation as (18). Hence, we have

$$\frac{\partial F(\phi_k)}{\partial t} + \mathbf{u} \cdot \nabla F(\phi_k) = 0, \quad (19)$$

and all the algorithms proposed for ϕ_k (schemes for advection equations) can be used for $F(\phi_k)$. We hope the considerable simplicity of this kind of technique will make it very attractive for practical implementation. Here we use a transformation of a tangent function for $F(\phi_k)$; that is,

$$F(\phi_k) = \tan[(1 - \epsilon)\pi(\phi_k - 1/2)], \quad (20)$$

$$\phi_k = \tan^{-1} F(\phi_k) / [(1 - \epsilon)\pi] + 1/2, \quad (21)$$

where ϵ is a small positive constant. The factor $(1 - \epsilon)$ enables us to get around $-\infty$ for $\phi_k = 0$ and ∞ for $\phi_k = 1$ and to tune for desired steepness of the transition layer.

Remarks:

- Although ϕ_k undergoes a rapid change from 0 to 1 at the interface, $F(\phi_k)$ shows regular behavior. Because most of the values of $F(\phi_k)$ are concentrated near $\phi_k = 0$ and 1, the function transformation improves locally the spatial resolution near the large gradients. Thus, the sharp discontinuity can be described easily.

- A transformation of this kind is effective only for the case where the value of ϕ_k is limited to a definite range throughout the calculation, as is the color function defined above.

- An analysis (done by Brackbill in [5]) shows that transforming ϕ_k to $F(\phi_k)$ results in a modification in the advection speed. The effective velocity guarantees the correctness of the advection speed along the $\phi_k = 0.5$ surface of the interface transition layer and tends to produce a solution that counters the smearing across the transition layer with intrinsic anti-diffusion.

- The tangent function transformation performs well with other third-order schemes such as the PPM method [6], but it is not encouraging when incorporated with low-order schemes or dispersion schemes, such as the first-order upwind scheme or the Lax–Wendroff scheme. This can be explained by the fact that $\phi = 0.5$ is not the middle point of the transition layer for an advection scheme with significant errors in dispersion or dissipation.

- This method does not involve any interface construction procedure and is economical in computational complexity. One of the interesting examples is the shock-wave interaction with a liquid drop, in which the deformable shape of the drop has been successfully captured [64]. Note that the presented method is more attractive in 3D computation since the extension of the scheme to three dimensions is straightforward.

Figure 2c shows a 1D square wave propagation computed by the CIP method together with the tangent transformation. The initial sharpness is well preserved and the discontinuities are advected with a correct speed. A 3D rotating notched brick, which was also used in [16, 33] to evaluate the performance of the VOF method, was calculated with the tangent transformation as well. As displayed in Fig. 6, the geometry is satisfactorily preserved.

3. A SEMI-LAGRANGIAN APPROACH TO HYDRODYNAMIC EQUATIONS

3.1. Basic Equations

Before presenting a method to solve all the phases of materials, we must first construct a unified equation to describe all the phases. For this purpose we use the following set of hydrodynamic equations:

$$\frac{\partial \mathbf{f}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{f} = \mathbf{S}. \quad (22)$$

Here, $\mathbf{f} = (\rho, \mathbf{u}, T)$, $\mathbf{S} = (-\rho \nabla \cdot \mathbf{u} + Q_m, -\nabla p / \rho + Q_u, -P_{\text{TH}} \nabla \cdot \mathbf{u} / \rho C_v + Q_E)$, where ρ is the density, \mathbf{u} the velocity, p the pressure, and T the temperature, Q_m represents the mass source term, Q_u represents viscosity, elastic stress tensor, surface tension, etc., and Q_E represents viscous heating, thermal conduction, and heat source. C_v is the specific heat for constant volume and we define $P_{\text{TH}} = T(\partial p / \partial T)_\rho$, which is derived from the first principle of thermodynamics and the Helmholtz free energy. Here, P_{TH} is not merely the pressure. In the special case of ideal fluid, however, P_{TH} is exactly the pressure p because the pressure linearly depends on temperature. The next simpler example is the two-phase flow described by the Clausius–Clapeyron relation

$$p = p_0 \exp\left(-\frac{L}{RT}\right), \quad P_{\text{TH}} = T\left(\frac{\partial p}{\partial T}\right) \propto L, \quad (23)$$

where R is the gas constant. In this case, P_{TH} becomes proportional to the latent heat L . Therefore, P_{TH} describes the heat loss due to latent heat when the ratio of gas increases in two-phase flow. A more general form of C_v and P_{TH} is given by semi-analytical formula or tabulated data.

3.2. The Fractional Step Approach

The underlying physics included in the above equations for continuum dynamics is complex and may include processes that have different time scales of variation. It is expedient to separate the solution procedure into several fractional steps. For example, for the advection phase,

$$\frac{\partial \mathbf{f}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{f} = 0, \quad (24)$$

and for the non-advection phase,

$$\frac{\partial \mathbf{f}}{\partial t} = \mathbf{S}. \quad (25)$$

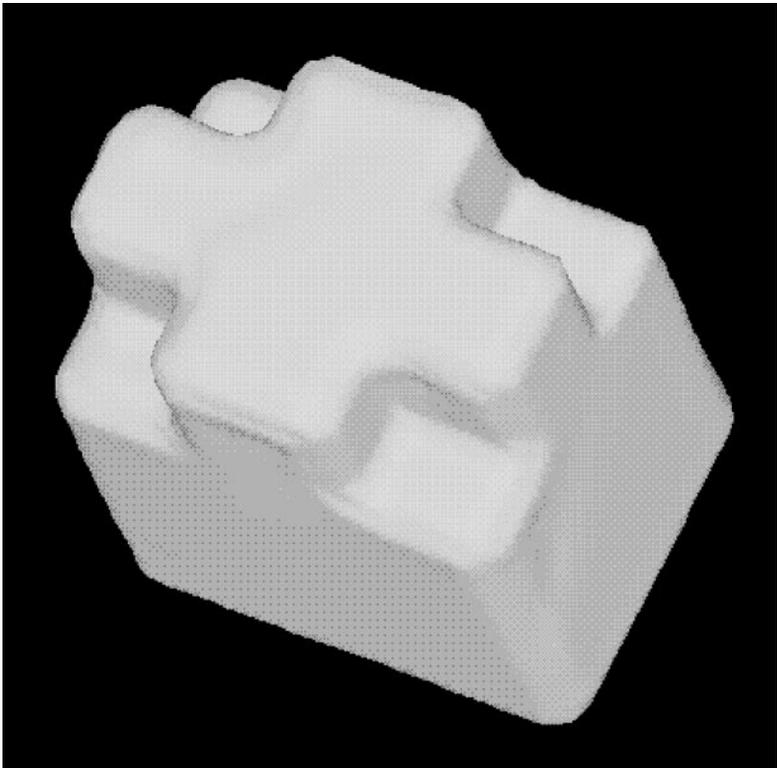
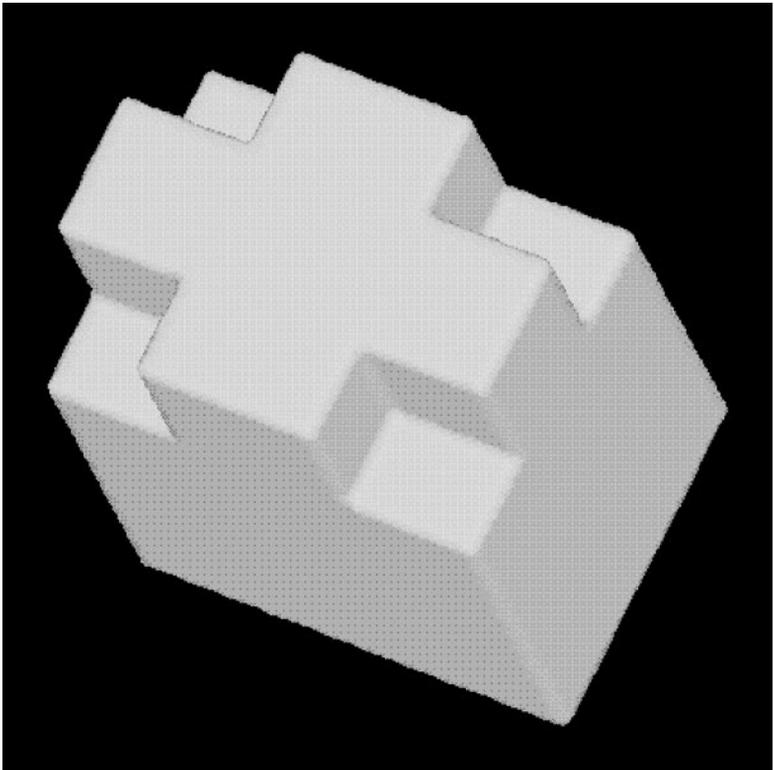


FIG. 6. The 0.5 isosurface of the notched brick. Displayed are the initial shape (top) and the computed result after one revolution computed by the interface-tracking method (bottom).

For the advection phase, the procedure given in Section 2 is used. The time evolution of the spatial gradient in the non-advection phase is again calculated according to Eq. (25) and thus we have

$$\frac{\partial(\partial_\chi \mathbf{f})}{\partial t} = \partial_\chi \mathbf{S}. \quad (26)$$

Usually it is not easy to get a finite-difference form of $\partial_\chi \mathbf{S}$. One method for estimating this term is [61]

$$\frac{(\partial_\chi \mathbf{f})^{n+1} - (\partial_\chi \mathbf{f})^*}{\Delta t} = \frac{(\delta_\chi \mathbf{f})^{n+1} - (\delta_\chi \mathbf{f})^*}{\Delta t}, \quad (27)$$

where $\delta_\chi \mathbf{f}$ represents a centered finite-difference form of $\partial_\chi \mathbf{f}$ in the χ direction. Thus the time evolution of $\partial_\chi \mathbf{f}$ is estimated by the time evolution of \mathbf{f} already given by Eq. (25).

3.3. Application to Shock Waves

As is well known, in a primitive Euler scheme, as in Eq. (22), the equation of internal energy is decoupled from kinetic energy. Therefore, in order to correctly transform dissipated kinetic energy to internal energy, some real physical mechanism must be introduced. In ordinary hydrodynamics, viscosity plays this role. In most cases of practical interest, the shock width is smaller than the mesh size. However, since the dissipation scale is limited to mesh size in computation, viscosity with this scale will be much smaller and hence will not give the correct dissipation energy. Therefore von Neumann and Richtmyer [22] proposed the use of an artificially large viscosity coefficient to achieve a sufficient viscosity effect even with a shock width equal to the mesh size. The correct form of this artificial viscosity is given by the Rankine–Hugoniot relation as shown by Wilkins [49], and its improved form free from directional dependence in 3D was proposed by Ogata and Yabe [23].

The CIP method with this numerical viscosity has been tested by an example of two interacting blast waves given by Woodward and Colella [52]. In this case, initial pressure $p = 1000$ for $x < 0.1$, $p = 0.01$ for $0.1 < x < 0.9$, and $p = 100$ for $0.9 < x < 1$. Although Woodward and Colella used a minimum grid size of $\Delta x = 1/9600$, we have succeeded in reproducing the result even with a uniformly spaced grid of $\Delta x = 1/400$ (Fig. 7b) and it should be noted again that the present scheme used the primitive Euler representation to capture these shock waves.

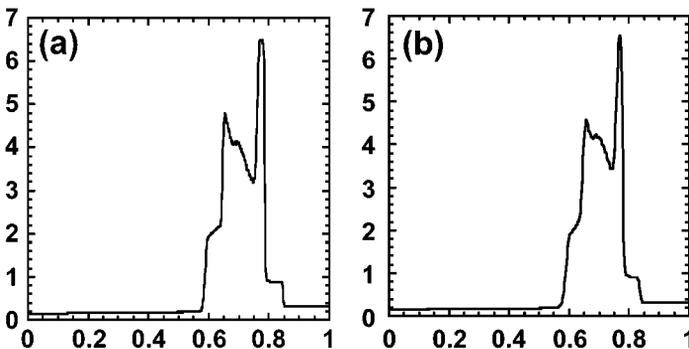


FIG. 7. Interacting shock waves. Density profile at $t = 0.038$ with (a) 800 and (b) 400 equally spaced grids.

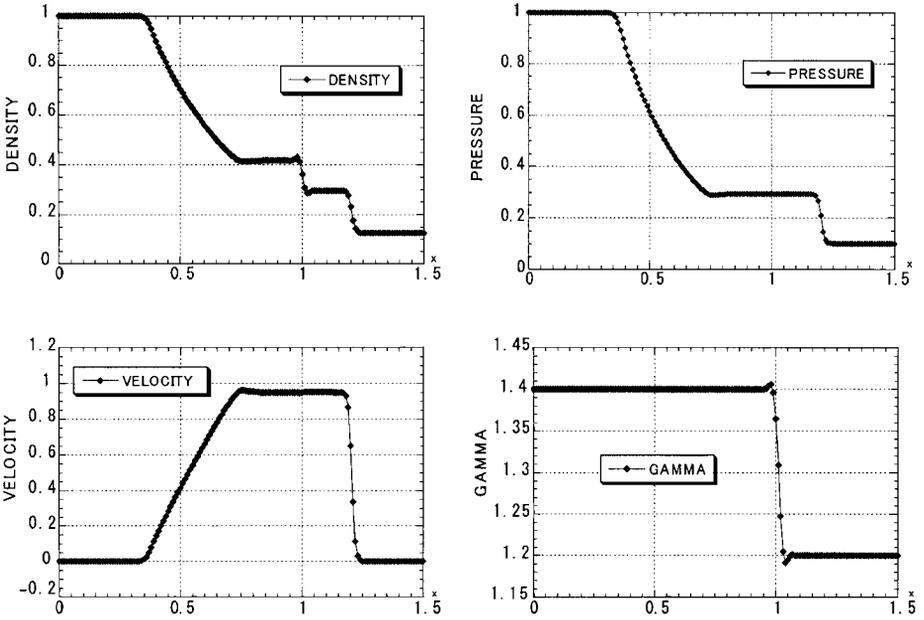


FIG. 8. Shock propagation in multiphase media.

The merit of primitive Euler representation of the CIP method can be demonstrated by the example of shock propagation through multiphase media. Figure 8 shows the result of a typical example, as discussed by Karni [15]. The specific heat ratio varies and is 1.4 on the left and 1.2 on the right. The specific heat ratio propagates according to Eq. (18) together with the contact discontinuity. At this point, fictitious pressure undulation appears when conservative schemes are used. However, the CIP can correctly treat the problem without any pressure undulation, as shown in Fig. 8. Note that in all the calculations shown in this section, we used neither RCIP nor tangent transformation to demonstrate the performance of the simple CIP. Small undulations of density at the contact discontinuity can be eliminated by RCIP and the profile of γ can be sharpened by tangent transformation. In addition, for the pressure (temperature) solution, Eq. (22) is explicitly solved and the implicit technique given in the following section is not used.

4. A PRESSURE-BASED ALGORITHM IN A PRIMITIVE EULER SCHEME

4.1. Pressure-Based Algorithm

The CIP method uses the primitive Euler method to solve Eq. (22); thus the formulation into a simultaneous solution of incompressible and compressible fluid is readily obtained. In order to understand this strategy, we first examine why it has been difficult to solve these equations together. In ordinary compressible fluid, the density ρ is solved by the mass conservation equation and then the temperature T is obtained by the energy equation. After that, from the equation of state (EOS), schematically shown in Fig. 9, the pressure $p = p(\rho, T)$ is calculated. At the low-density side, $p \propto \rho T$ like ideal fluid and dependence is relatively weak, but at solid or liquid density p steeply rises as the density rises. This means that extremely high pressure is needed to compress solid or liquid even slightly. In

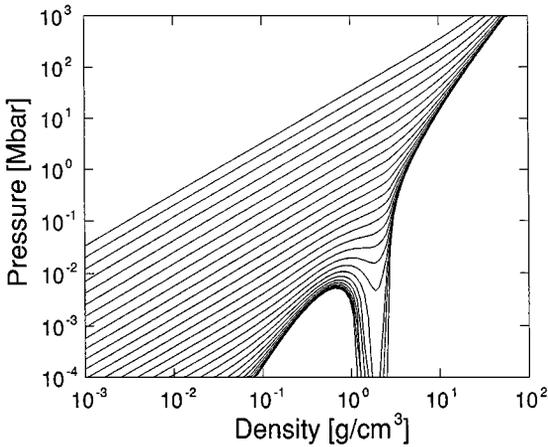


FIG. 9. An example of an equation of state. Each line represents isotherms.

other words, for solid or liquid, the sound speed $C_s = (\partial p / \partial \rho)^{1/2}$ is very large. Therefore, if we look at the process in which density is first calculated, we find that only a small amount of density error—10%, for example—causes a large pressure pulse of 3–4 orders of magnitude.

In such a situation, incompressible approximation is normally adopted; that is, the pressure equation to ensure $\nabla \cdot \mathbf{u} = 0$ is derived from the equation of motion and mass conservation. This scheme is called a pressure-based scheme and MAC [10], SMAC [1], SIMPLE [28], and SIMPLER [27] are typical examples.

In order to extend this idea to compressible fluid, we need to modify the EOS shown in Fig. 9. If we rotate the figure by 90 degrees, then the steep pressure curve becomes a flat density curve. This means that if we could first solve the pressure and then estimate the density in terms of $\rho(p, T)$ by using the EOS shown in Fig. 9, the problem at liquid density would be eliminated. In addition, since the EOS in lower density gas depends linearly on other quantities, this reverse procedure creates no problems in that case.

Then how do we realize this reverse procedure? For this purpose, we should predict how the pressure reacts to changes in density and temperature. Such a unified procedure to incorporate compressible fluid with incompressible fluid has been initiated by Harlow and Amsden as the ICE (implicit continuous Eulerian) [9]. The ICE has been improved by the PISO [12] (pressure implicit with splitting of operators). In both cases, however, conservative equations are used as a starting point. The main difference between the ICE and the PISO is in the treatment of the convection term.

On the other hand, the CCUP (CIP-combined and unified procedure) [63] uses primitive Euler equations and splits the advection term from the other terms, related sound waves. This simplification also simplifies the pressure equation and greatly improves our ability to attack multiphase flow. One year after this proposal, Zienkiewicz [73] proposed a similar method but applied it to the finite-element method. Unfortunately, however, with their scheme it is not as easy to remove the difficulty stemming from a large-density ratio at the boundary between liquid and gas, as discussed below.

We now present a condensed and generalized description of the ICE. In both the ICE and the PISO, conservation equations of mass and momentum are used in a finite-difference

form,

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = -\frac{\partial(\rho u)'}{\partial x} \quad (28)$$

$$\frac{(\rho u)' - (\rho u)^n}{\Delta t} = -\frac{\partial p^{n+1}}{\partial x} + H(u) \quad (29)$$

$$H(u) \equiv -\frac{\partial(\rho u^2)}{\partial x}. \quad (30)$$

Substituting Eq. (29) into Eq. (28), we get

$$\frac{\partial^2 p^{n+1}}{\partial x^2} = \frac{\rho^{n+1} - \rho^n}{\Delta t^2} + \frac{1}{\Delta t} \left(\frac{\partial \rho u}{\partial x} \right)^n + \frac{\partial H}{\partial x}. \quad (31)$$

Next, if we assume that density changes in proportion to pressure change,

$$\Delta p = \left(\frac{\partial p}{\partial \rho} \right)_T \Delta \rho = C_s^2 \Delta \rho, \quad (32)$$

then density change on the right-hand side of Eq. (31) is replaced by pressure change,

$$\frac{\partial^2 p^{n+1}}{\partial x^2} = \frac{p^{n+1} - p^n}{C_s^2 \Delta t^2} + \frac{1}{\Delta t} \left(\frac{\partial \rho u}{\partial x} \right)^n + \frac{\partial H}{\partial x}. \quad (33)$$

In the ICE, the term H is estimated at the step n , while in the PISO H is predicted by an equation of motion,

$$\frac{\rho^n u^p - (\rho u)^n}{\Delta t} = -\frac{\partial p^n}{\partial x} + H(u^p), \quad (34)$$

and finally gives

$$\frac{\partial^2 (p^{n+1} - p^n)}{\partial x^2} = \frac{p^{n+1} - p^n}{C_s^2 \Delta t^2} + \frac{1}{\Delta t} \left(\frac{\partial \rho^n u^p}{\partial x} \right)^n. \quad (35)$$

The original PISO is more complicated because it repeats this predictor–corrector algorithm a few times and some complication appears to diagonalize the H term to solve Eq. (34) in terms of u^p .

4.2. CCUP Method

Yabe and Wang [63] adopted the primitive Euler form instead of the conservative form to construct the pressure equation. Furthermore, the advection part is separated from the other terms, since the advection term can be processed free from the CFL condition in a semi-Lagrangian procedure. Fortunately, this splitting led to an unexpected advantage to the solution in multiphase flow, as shown below.

The original CCUP method [63] was proposed only for a special equation of state such as Eq. (32), but here we rebuild it with a more general EOS [56]. That is, for small changes of density and temperature, the pressure change can be linearly proportional to them as

$$\Delta p = \left(\frac{\partial p}{\partial \rho} \right)_T \Delta \rho + \left(\frac{\partial p}{\partial T} \right)_\rho \Delta T, \quad (36)$$

where Δp means the pressure change $p^{n+1} - p^*$ during one time step and $*$ is the profile

after advection. This also applies to ρ , T . From this relation, once $\Delta\rho$ and ΔT are predicted, Δp is predicted based on Eq. (36); $\partial p/\partial\rho$, $\partial p/\partial T$ are given by the EOS.

Since the CIP separates the non-advection terms from the advection terms, we can concentrate on the non-advection terms related to sound waves, which are the primary cause of the difficulty posed by the large sound speed of liquid, and hence ρ , T are simply given by

$$\Delta\rho = -\rho^* \nabla \cdot \mathbf{u}^{n+1} \Delta t \quad \rho^* C_v \Delta T = -P_{\text{TH}} \nabla \cdot \mathbf{u}^{n+1} \Delta t, \quad (37)$$

where C_v is the specific heat ratio at constant volume. \mathbf{u}^{n+1} in this equation is given by an equation of motion as

$$\Delta\mathbf{u} = -\frac{\nabla p^{n+1}}{\rho^*} \Delta t. \quad (38)$$

Since $\Delta\mathbf{u} = \mathbf{u}^{n+1} - \mathbf{u}^*$, Eqs. (36)–(38) lead to a pressure equation [56, 63]

$$\nabla \left(\frac{1}{\rho^*} \nabla p^{n+1} \right) = \frac{p^{n+1} - p^*}{\Delta t^2 \left(\rho^* C_s^2 + \frac{P_{\text{TH}}^2}{\rho C_v T} \right)} + \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}. \quad (39)$$

Then substituting the given p^{n+1} into Eq. (38), we obtain the velocity \mathbf{u}^{n+1} and then the density ρ^{n+1} from Eq. (37). From this procedure, density can be solved in terms of pressure, which is analogous to rotating Fig. 9 by 90 degrees. Equation (39) has many important features. This equation shows that, at sharp discontinuities, $\mathbf{n} \cdot (\nabla p/\rho)$ is continuous. Since $\nabla p/\rho$ is the acceleration, it is essential that this term be continuous since the density changes by several orders of magnitude at the boundary between liquid and gas. In this case, the denominator of $\nabla p/\rho$ changes by several orders, and the pressure gradient must be calculated accurately enough to ensure continuous change of acceleration. Equations (33) and (35) derived by the ICE and the PISO seem to be similar to Eq. (39) but the continuity of $\nabla p/\rho$ in the ICE and the PISO is not guaranteed. However, Eq. (39) works robustly even with a density ratio larger than 1000 and enables us to treat both compressible and incompressible fluids. Computationally, the solution of Eq. (39) provides a pressure distribution that can be used to project the velocity field for variable density flow; i.e., the resulting pressure field is weighted by the inverse density. Other projection methods with variable density for incompressible fluid can be found in [3] and [32].

It is interesting to examine the meaning of this pressure equation. If the $\nabla \cdot \mathbf{u}$ term is absent, this equation is merely the diffusion equation. The origin of this term is as follows. During time step Δt , the sound wave propagates for a distance $C_s \Delta t$. In the next step, the signal also propagates backward and forward since the sound wave should isotropically propagate. Then statistically, 50% propagates backward and another 50% forward. This process is similar to the random walk. The diffusion coefficient of the random walk is given by the quivering distance $\Delta x = C_s \Delta t$ as $D = \Delta x^2/\Delta t$. This leads to the diffusion equation for pressure. From this consideration, we can see how the effect of sound waves is implemented.

4.3. Two-Dimensional Driven Cavity Flow

As a model problem for verifying the numerical scheme, we first examine two-dimensional incompressible flow in a square cavity with a top wall moving at a constant velocity. This problem has been studied by many researchers as a benchmark problem. In order to simulate incompressible flow with compressible-type equations, the sound velocity C_s is set

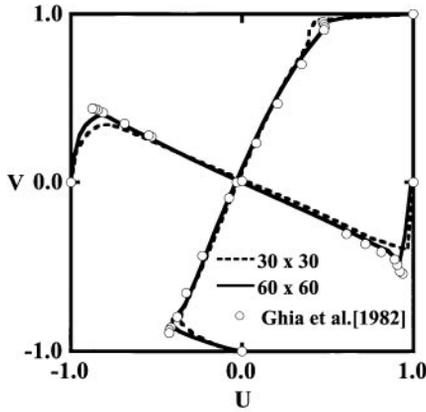


FIG. 10. The velocity profiles in a two-dimensional driven cavity flow with 30×30 and 60×60 Cartesian grids. For comparison, the result by Ghia *et al.* [7] using 256×256 multigrids is shown by circles.

to 1.0×10^{10} instead of to ∞ . In this calculation, we need to include the viscosity term in Eq. (38) but it is possible to treat it separately by a splitting procedure [63] or by a predictor–corrector scheme [70]. The calculations are performed on a uniform grid structure for the Reynolds number 10,000. The velocity profiles [70] in the x and y directions are shown in Fig. 10. For comparison, the result of Ghia *et al.* [7] from using 256×256 multigrids is shown by circles. As shown in the figure, the solution with 60×60 meshes is accurate enough to reproduce Ghia’s result near the wall as well as in the central region, although the one with 30×30 meshes failed near the wall.

4.4. Milk Crown Formation

The next example demonstrates the capturing of the liquid–gas interface. Simulation of the coronet or “milk crown” has long been a dream in computational physics [8]. The authors successfully simulated this problem [66, 67] and discussed the effect of surface tension comparing expected effects with experimental data [72]. The coronet is not only a consequence of a free-surface problem; we must solve the problem of surrounding gas as well.

Figure 11 shows an iso-surface contour of density in which 100×100 (horizontal) $\times 34$ (vertical) fixed equally spaced grids are used with $D/16$ grid spacing, where D is the diameter of the water drop. A thin water film with a thickness of $D/4$ is placed on a solid plate and a water drop impacts from the top at a speed of U ; $\text{Re} = UD/\nu = 8000$. We solved for air as well as for water and the density ratio at the interface is almost 1000.

Surprisingly, as this three-dimensional simulation shows, the wavelength of the irregularity or finger along the rim depends on the density of ambient gas and increases with the density. Even for the highest ambient density, the density ratio of liquid and gas is still 33 and such a low-density gas has been believed not to cause such a significant change because the growth rate of the Rayleigh–Taylor (R–T) instability is proportional to $(\rho_{\text{Liq}} - \rho_{\text{gas}})^{1/2}$.

We proposed a stabilization mechanism by ambient gas [72]. Since the wind of ambient gas blows toward the rim radially in the moving frame of the rim, it can act to stabilize the R–T instability like a wall. If this wind velocity is U_0 , then the effect of the rim pressure should be added in the perturbed form of $\partial \rho_{\text{gas}} U_0^2 / \partial x$, which is roughly $\rho_{\text{gas}} U_0 \partial(\xi/\Delta t) / \partial x \sim \rho_{\text{gas}} U_0 k \xi \gamma_{\text{RT}}$, where ξ is the amplitude of the perturbed surface, $\Delta t \sim 1/\gamma_{\text{RT}}$, and x is in

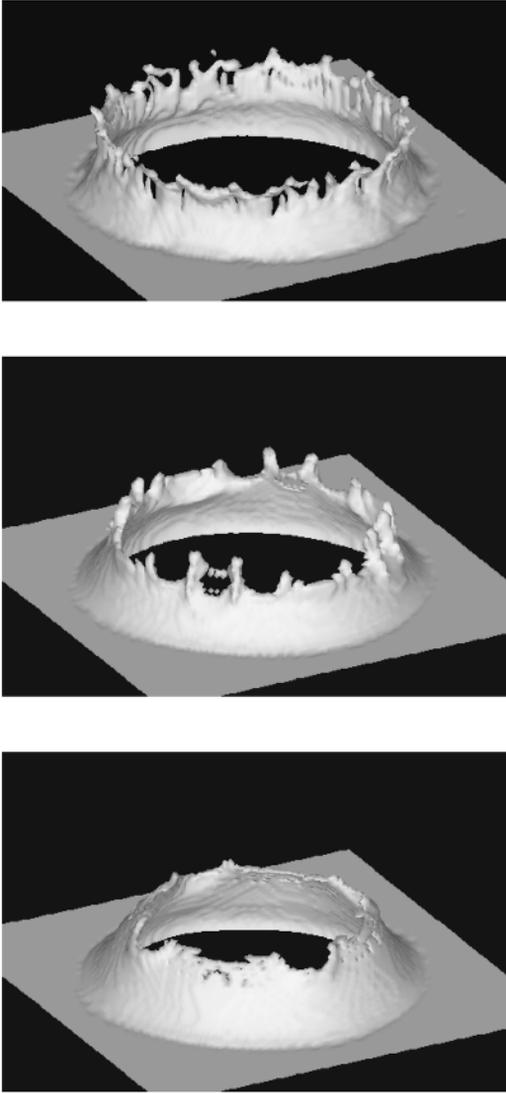


FIG. 11. A water surface plot in the coronet-formation process. A $100 \times 100 \times 34$ Cartesian grid is used. The figure shows the plots at $t = 8.2D/U$. Ambient gas density used in the simulation is $\rho_{\text{gas}}/\rho_{\text{Liq}} = 0.002$ (top), 0.02 (middle), and 0.03 (bottom). The irregular structure disappears as the gas density increases.

the radial direction. Thus we obtain a model equation [72]:

$$\rho_{\text{Liq}} \frac{\partial^2 \xi}{\partial t^2} = \rho_{\text{Liq}} \gamma_{\text{RT}}^2 \xi - \rho_{\text{gas}} U_0 k \xi \gamma_{\text{RT}}. \quad (40)$$

Therefore the criteria for the growth of perturbations will be

$$\gamma_{\text{RT}} > \frac{\rho_{\text{gas}}}{\rho_{\text{Liq}}} k U_0. \quad (41)$$

Thus, the maximum wavenumber for instability is inversely proportional to ρ_{gas}^2 , since γ_{RT} is proportional to $k^{1/2}$.

In the simulation, we have included kinematic viscosity for water and air as $UD/\nu = 8400$ and 560 , respectively. Since the viscosity effect appears in a form of kinematic viscosity in the R–T instability, the effect of the ambient density does not explicitly appear in the result. The effect of surface tension has also been discussed in Ref. [72] and has been compared to experimental results.

Since the wavelength of the fingers along the rim strongly depends on the density of ambient gas, the finger formation observed here is not an artifact of finite grid size, although it might have provided seeds of the instability. As is clear from this example, simulations can provide important information about coronet-formation physics by modeling situations that would be hard to create experimentally. The phenomena observed here also give valuable information about GDI (gasoline direct injection) engines, in which the impact of splashing fuel on the combustion chamber is the key issue for efficient combustion [47] because of increasing surface area. Since ambient pressure in the chamber should be high, the generation of droplets might be greatly reduced as indicated here.

4.5. Laser-Induced Evaporation

We now add further complication to this interaction, i.e., phase change. Figure 12 shows melting and evaporation of aluminum under the illumination of laser light, where the density

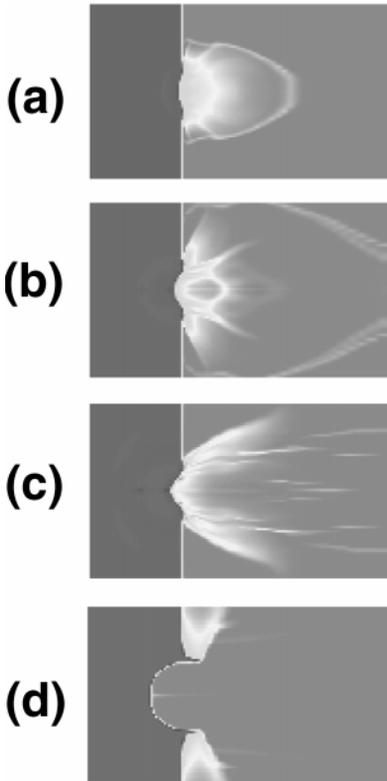


FIG. 12. The density contour of aluminum (on the left) illuminated by laser light. The filamentary structure explains the experimental results. The time sequence is (a) 40 ns, (b) 90 ns, and (c) 290 ns with an elastic–plastic effect. The crater does not grow after this. However, if the elastic–plastic effect is switched off, it continues to grow like the contour at (d) 500 ns.

changes from 2.7 to 10^{-4} g/cm³. Aluminum solid is initially treated as an elastic material, whose treatment is given in Section 5, and then as a liquid and vapor during phase transition. This change is simple as realized by the equation of state shown in Fig. 9. This example shows that the code has a strong ability to describe a sharp interface and is robust enough to treat both compressible and incompressible fluid simultaneously. In this calculation, we used two-dimensional cylindrical coordinates and a grid of 90 (axial) \times 45 (radial). Laser light is normally incident and is simply treated as a straight line depositing its energy at the solid–gas interface with an absorption rate of 30%.

The experiment was performed at the Institute of Laser Engineering, Osaka University [68]. A YAG laser of 650 mJ and an 8-ns pulse width are used to illuminate an aluminum-slab target. Final crater depth and shape agree well with the simulation. It is interesting to note that the crater is not formed during the laser pulse but develops gradually over a time scale of several hundred nanoseconds well after the laser pulse has ended, as shown in Fig. 12. The very high temperature plasma of more than a few tens of electronvolts are produced by the 8-ns laser pulse, and most of them expand from the target. However, some stay near the target for a long time after the laser pulse because of the recoil force from expanded plasmas and act as a heat source to melt the aluminum over a time scale of several hundred nanoseconds.

When the plasma temperature becomes less than the melting temperature—around 290 ns (the time is measured from the laser peak)—aluminum with a strength of 0.248 Mbar and a yield strength of 2.2976 kbar starts being affected from the stress and no distortion occurs after that time. This yield stress plays an important role in determining the final crater size. Without yield strength, the crater develops further even after 490 ns and becomes several times larger, as shown in Fig. 12d, although less difference is seen at the beginning around 90 ns.

The plasma-heated crater formation leads to other interesting phenomena. Since the plasma acts not only as a heat source but also as a pressure source, the dynamic expansion of evaporated material at a later time is strongly modified. Since a high-pressure region is just in front of the evaporation surface, the vapor is forced to bypass this region, flowing through a narrow channel between the metal surface and this pressure source. Therefore, the vapor preferentially flows toward a circumference with a large angle to the target normal. This effect is the exactly the same as that obtained in the experiment [68].

The simulation predicts additional interesting behavior. The expansion at $t < 40$ ns is uniform because its temperature is high—a few tens of electronvolts. The experiment supports this result and the debris around 0 degrees is very fine and indistinguishable with an optical microscope. On the other hand, the simulation result at $t = 290$ ns shows some filamentary streams flowing from the surface. The experiment also supports this result and the debris at 75 degrees consists of particles of 1 to 20 μm .

5. TREATMENT OF EMBEDDED BOUNDARIES IN CARTESIAN GRIDS

5.1. *Fluid–Structure Interaction*

Directly computing the interactions between solid bodies and suspended fluid is necessary for suspension flows in which the suspended bodies interact with the surrounding fluid and substantially influence the flow motion. An efficient computational model for flows containing distortionless rigid bodies can be constructed by using the numerical procedures introduced in Section 2.5.

Using color functions, we can easily treat solid bodies or objects that have any complex shape or heterogeneous density distributions.

The advection equation for the color function ϕ_l ,

$$\frac{\partial \phi_l}{\partial t} + \mathbf{u}_{b(l)} \cdot \nabla \phi_l = 0, \quad l = 1, 2, \dots, L, \quad (42)$$

is solved by the method in Section 2.5 to recognize all the solid bodies, where $\mathbf{u}_{b(l)}$ is the velocity field for object l . The motion of the object l is determined by

$$\mathbf{u}_{b(l)} = \bar{\mathbf{u}}_l + \mathbf{r} \times \bar{\boldsymbol{\Omega}}_l, \quad (43)$$

where $\bar{\mathbf{u}}_l$ is the translational speed of the mass center of object l , $\bar{\boldsymbol{\Omega}}_l$ is the angular speed, and \mathbf{r} is the distance to the mass center. $\mathbf{r} = \mathbf{x} - \mathbf{X}$, $\mathbf{X} = \int \mathbf{x} \rho \phi_l dV / M$, and $M = \int \rho \phi_l dV$. These quantities are predicted by Newton's laws of motion

$$\frac{d\bar{\mathbf{u}}_l}{dt} = \frac{1}{M} \int \frac{d\mathbf{u}}{dt} \rho \phi_l dV, \quad (44)$$

and

$$\frac{d}{dt} (\bar{\boldsymbol{\Pi}}_{(l)} \bar{\boldsymbol{\Omega}}_l) = \bar{\boldsymbol{\Gamma}}_l \equiv \int \mathbf{r} \times \frac{d\mathbf{u}}{dt} \rho \phi_l dV, \quad (45)$$

where $\bar{\boldsymbol{\Gamma}}_l$ is the torque for object l . $\bar{\boldsymbol{\Pi}}_{(l)}$ is the tensor of inertia moment $\equiv \int \mathbf{r} \mathbf{r} \rho \phi_l dV$.

All the forces (represented by $\rho d\mathbf{u}/dt$, including both the body force and the fluid stress) are calculated at all grids in a volume force form. It is convenient to compute the net force on the mass center of object l by summing up all the forces over the entire domain, but only the region recognized by $\phi_l = 1$ contributes to the integration in Eqs. (44) and (45). Different from the so-called ‘‘surface force’’ formulation, the volume force-based schemes, as we used here, do not need information about the body surface, such as orientations and surface areas, which appears in other difficult problems in computation.

A non-slip condition is used to impose solid motion onto the velocity field of the fluid, which in turn is driven through the velocity coupling as

$$\frac{d\mathbf{u}}{dt} = -\nu \sum_l \phi_l (\mathbf{u} - \bar{\mathbf{u}}_{b(l)}). \quad (46)$$

This slip boundary condition can be treated similarly but is slightly more complicated [69].

We computed a solid object undergoing steady translation at a low speed \mathbf{v}_s in Stokes flows. Some similar examples can also be found in the works of Pan and Banerjee [25, 26]. The Reynolds number $\text{Re} = 2a\mathbf{v}_s/\nu$ is 0.006, where a is the radius being described by two grids in Cartesian coordinates. Figure 13 shows the velocity component perpendicular to the velocity of the particle and the analytical results at different levels apart from the particle, and here we find agreement between the numerical results and the analytical solutions.

As an example of treating complex geometry, we simulated a rotating spherical cage rising from fluid under the force of floating. The cage is a hollow sphere with six holes on the surface, and the thickness of the cage is described by seven grid sizes. This geometry is

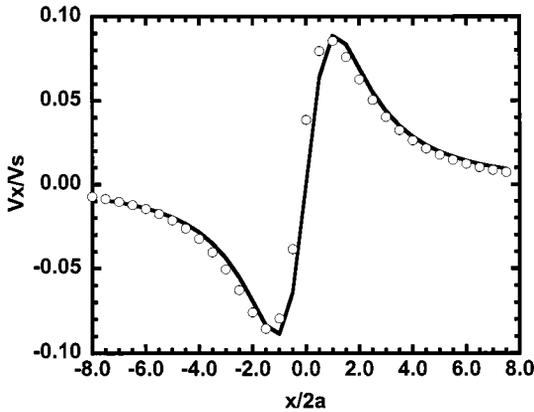


FIG. 13. A solid sphere moving at a constant speed v_s in a viscous fluid produces steady flows in the surrounding fluid. Displayed is the velocity distribution of the velocity component normal to the moving speed of the particle, v_s , at distances of $(z_p - z)/(2a) = 1.5$ below the particle. z_p is the particle location and a is the radius of the particle. The solid line represents the analytical solution, and the circles denote our computational results.

too complex for many numerical methods to handle. The cage has a density 10% that of the fluid and rotates initially along the gravity direction. Figure 14 shows the time development of the process. The cage rose from under water and drove out the surrounding fluid. Part of the fluid was carried upward by the cage and then leaked out through the holes. The cage then approached its equilibrium state and stayed on the fluid's surface.

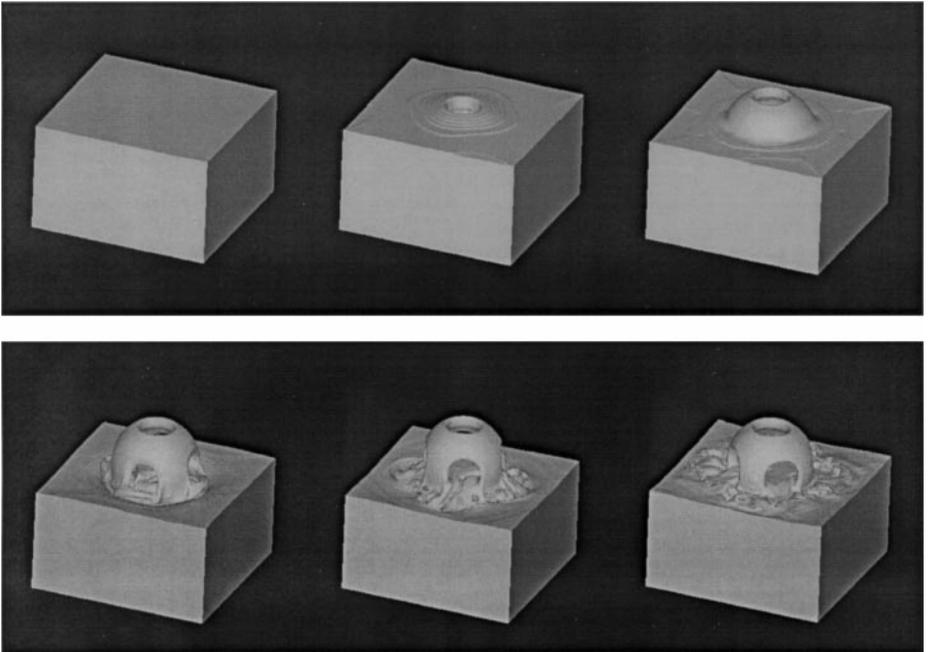


FIG. 14. A spherical cage floating up from under water with an initial rotation along the gravitational direction. A $150 \times 150 \times 150$ Cartesian grid is used. Time increases from left to right and from top to bottom. This solid cage is described by a color function with tangent transformation.

5.2. Numerical Formulation for Surface Tension

The CSF (continuum surface force) formulation [5] is widely used in the CFD (computational fluid dynamic) society. In the CSF computational model, the surface tension is converted into a form of volume force and the resulting force is proportional to the product of the interface gradient and the surface curvature. A similar expression can be easily obtained using the solution of the variational problem for the total energy conservation [13]. The effects of surface tension are consequently included in the computational model through an external forcing term added to the momentum equation. Then, in the *external-forcing-and resource-terms-related phase*, we must solve the equation

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\sigma \kappa(\phi) \nabla \phi, \quad (47)$$

where ϕ is the color function defined above. The local mean curvature is calculated as

$$\kappa(\phi) = \nabla \frac{\nabla \phi}{|\nabla \phi|}. \quad (48)$$

Computing the curvature requires the gradient field of ϕ . The gradients can be directly obtained from the distance function in a level set method in which the quantity $\nabla \phi$ always indicates a slope of 1 but needs to be artificially smoothed in a VOF-type method in which an interface is presented via a discontinuity in the VOF function.

Using a CIP-type scheme to calculate the color function ϕ , one can simultaneously get the solutions to ϕ and its first-order derivatives. This fact motivated us to make use of the readily known values of $\partial_x \phi$, $\partial_y \phi$, and $\partial_z \phi$ from the interface-tracking calculation. In evaluating the curvature $\kappa(\phi)$, we can directly manipulate the gradients of the color function as dependent variables.

A 2D example of an equilibrium rod in [5] was used to validate the computational model. The 2D RCIP scheme was used in the calculation of the color function.

When the viscosity effect is neglected and a constant surface-tension coefficient is assumed, from Laplace's formula we can obtain the theoretical prediction of the pressure-jump inside an infinite cylinder as $\Delta p_{\text{theor.}} = \sigma/R$, where R is the cylinder radius. We computed the equilibrium rod problem with the model introduced above. A Cartesian grid with uniform mesh spacing ($\Delta_x = \Delta_y$) was used and the background pressure is 1.0. Calculations were carried out with drops with various radii and surface-tension coefficients. The numerical error is measured with the L_2 norm defined by

$$L_2 = \sqrt{\frac{\sum_{i,j} [(\Delta p_{\text{numer.},i,j} - \Delta p_{\text{theor.}})^2 \phi_{i,j}]}{\Delta p_{\text{theor.}}^2 \sum_{i,j} \phi_{i,j}}}.$$

In all the cases, equilibrium pressure jumps were built up and underwent no noticeable change after 500 calculation steps.

Results after 1500 time steps are displayed in Fig. 15 and Table I. Figure 15 exhibits an oscillation-less plateau of a pressure field and a sharp transition region that covers less than two grid points. As we evaluate the surface tension from a color function which has a compact transition thickness, we see that the force is only imposed on a narrow region along the interface. This looks different from the CSF method, which gives a smoothed pressure jump across the interface due to the use of a smoothed color function. Table I

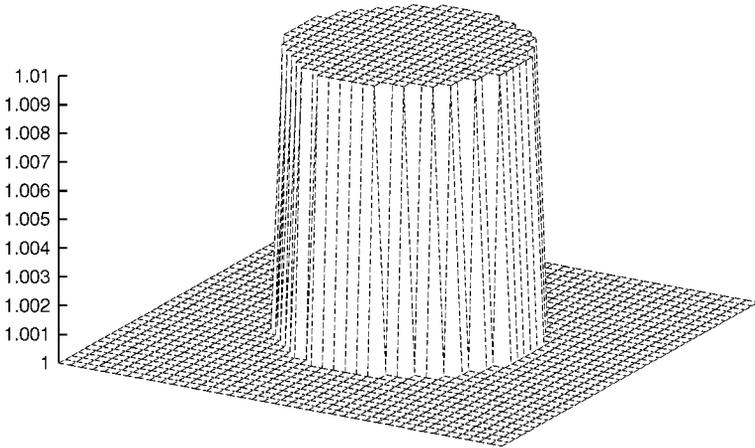


FIG. 15. Pressure field of the 2D equilibrium rod ($R/\Delta = 10$, $\sigma = 0.1$).

shows agreement between the numerical and theoretical results. Compared with the case of $R/\Delta_x = 20$, that of $R/\Delta_x = 5$ also gives a competitive computational accuracy even with such a low spatial resolution.

5.3. Computation of the Constitution Equation

Modeling of fluids which contain flexible bodies has been done so far with numerical methods based on Eulerian representations [19, 30, 31]. In these methods, the rheological effects are included by using either massless Lagrangian particles to evaluate the strain or finite-element techniques on subdomains to treat the flexible bodies. McMaster [19] discusses three computer codes to model fluid–structure interaction, where Eulerian finite differencing is used to compute the fluid and a finite-element method is used to solve the structure. The key strategy underlying those numerical models is to couple the fluid and the structure by an iteration procedure that results in a continuous velocity boundary condition on the interface between two material components. The advantage of the CIP method is that it can be applied to all the materials including flexible bodies using a single algorithm with no iterative procedure.

TABLE I

Comparison of the Computed Mean Drop Pressure Jump $\widehat{\Delta p}_{\text{numer.}} = \sum_{i,j} \Delta p_{\text{numer.},i,j} \phi_{i,j} / \sum_{i,j} \phi_{i,j}$ and the Theoretical Pressure, and the L_2 Errors for the Equilibrium Rod Problem

$R/\Delta x$	σ	$\widehat{\Delta p}_{\text{numer.}} / \Delta p_{\text{theor.}}$	L_2
5	0.1	0.982	5.8×10^{-2}
5	0.2	0.951	6.2×10^{-2}
10	0.1	0.972	3.5×10^{-2}
10	0.2	0.965	3.9×10^{-2}
20	0.1	0.994	3.8×10^{-2}
20	0.2	0.992	3.1×10^{-2}

Based on the fractional-step solution procedure mentioned in the previous sections, we can construct a numerical treatment for the constitutive equation–related phase [58] separately from other processes. The constitutive relation, which has a differential form and relates the velocity to the stress or stress rate, is solved by the finite-difference method. The resulting velocity is then used as a provisional value in the remaining computations.

Constitutive equations, which give prescribed relations between stress and strain, are usually expressed in terms of stress or stress rate and strain or strain rate. Since velocity is one of the dependent variables in the present computational model, a constitutive equation in terms of strain rate is more convenient to formulate. Thus an equation such as

$$\frac{ds_{ij}}{dt} = 2G \left(\frac{1}{2} \frac{d\epsilon_{ij}}{dt} - \frac{1}{3} \frac{d\epsilon_{kk}}{dt} \delta_{ij} \right) \quad \frac{d\epsilon_{ij}}{dt} \equiv \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}$$

is used to obtain a stress tensor s_{ij} from velocity fields with Young modulus G . By using a staggered grid, the conventional central differencing results in a compact computational formulation. The time integration of the “velocity–stress” relations can be computed by a multistep explicit scheme based on the Taylor expansion [58] or by a semi-implicit scheme using the ADI (alternative directional implicit) technique [53].

Figure 16 shows the sequence of two kinds of viscoelastic bodies sedimenting in fluid under the force of gravity. The object is initially put in a direction oblique to gravity which points downward. Once released, the object moves down and is imposed on by forces from the surrounding fluid. As expected from the theoretical analysis [59], the net external force along the body will bend the long axis into a curve. The bent object will then produce a torque that tends to push it toward a horizontal position. Thus, the object vacillates while it is sedimenting in the fluid. A constitutive equation for a Hook elastic solid is used for the pure elastic body. As can be seen from Fig. 16 (top), the elastic force resists the forcing from the fluid and prevents further deformation of the body. For the Maxwell body, a constitutive equation for a Maxwell solid body is used, which includes both elastic and viscous effects, and a relaxation on elastic stresses takes place. In the present calculation (Fig. 16, bottom), the Deborah number ($N_D = \tau/t_e$ with t_e being the total integration time) is about $2/3$. The stress is released significantly during the computation. The body is bent to a larger extent and behaves more like fluid.

Figure 17 shows another example. An elastic ball collides with a plate which is fixed on both sides. Both ball and plate move through a 100×100 fixed Cartesian grid. Here only 10 grids are used to describe the thickness of the plate at its initial location. The black part in the figure is filled with air and is also taken into account in the calculations. After the collision, the ball again detached from the plate. Such behavior is difficult to simulate with adapted grid methods.

6. CONSERVATIVE SEMI-LAGRANGIAN SCHEMES

It is useful to look for conservative semi-Lagrangian schemes because the semi-Lagrangian method can be effective on parallel computers, is suitable for multiphase flow, and enables the advection calculation to be made with a large time step free from the CFL condition. Although the semi-Lagrangian scheme has been successfully used in short-term atmospheric problems, the loss of exact conservation makes the scheme inappropriate for long-term problems and oceanic problems.

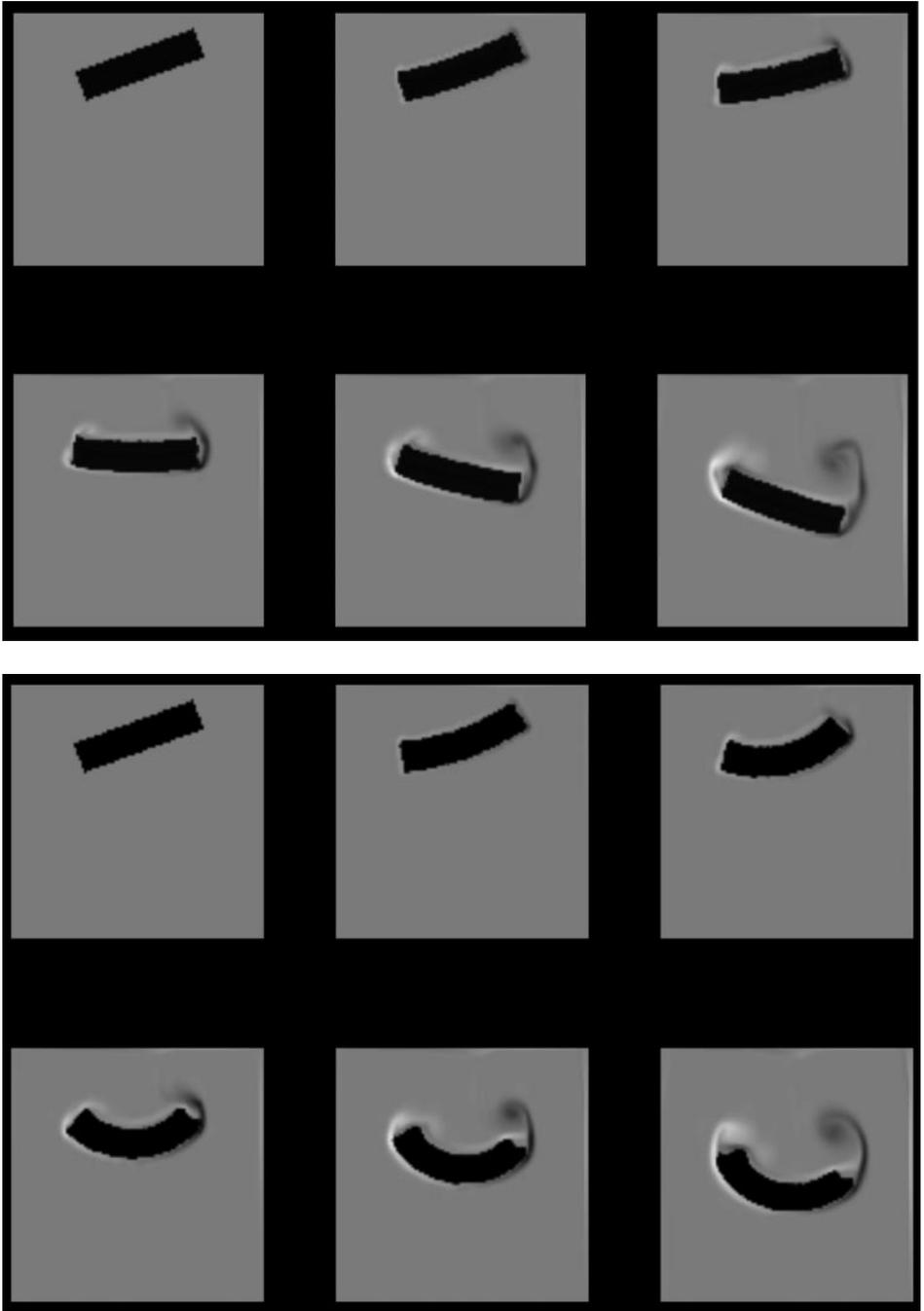


FIG. 16. Sedimentation of a pure elastic body (top) and a Maxwell body (bottom) in fluid. Time increases from left to right and from top to bottom. The vorticity of the environmental flow is also plotted.

Furthermore, some subjects require exact conservation of mass. A typical example is black-hole formation and emission of gravity waves [36]. In this case, a small fraction of mass is converted into gravity waves and strict mass conservation is essential. Another example is plasma simulation in which a Vlasov equation in six-dimensional phase space

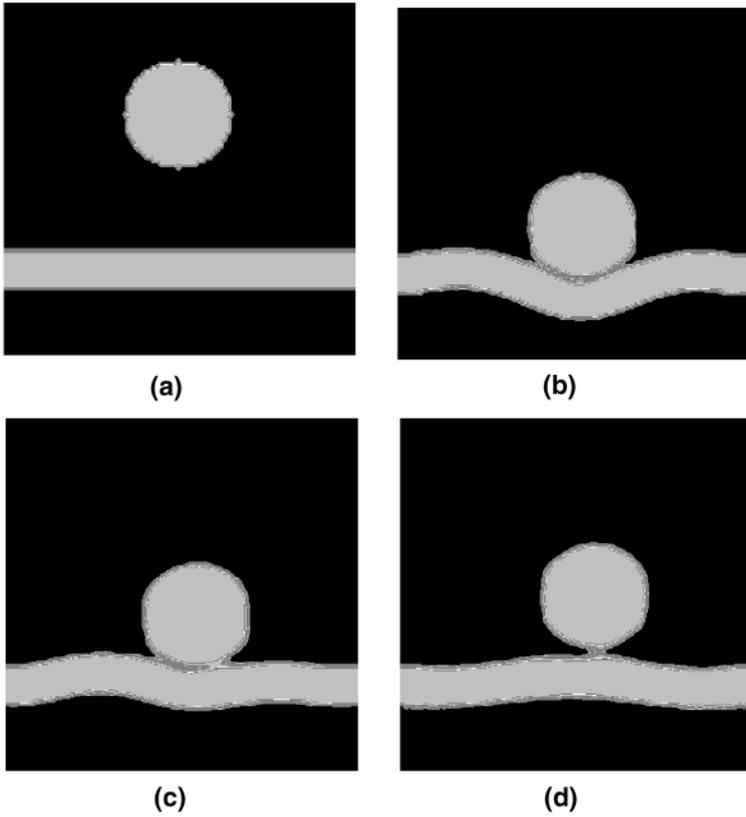


FIG. 17. Example of elastic–plastic flow. The system is described by a 100×100 Cartesian fixed grid, and only 10 grids are used for the width of the plate. The plate and ball are moving through a fixed-grid system. The black part is filled with air and is also taken into account in the calculations.

must be solved and total particle density must be conserved or else a large electric field appears. The CIP method can be constructed to exactly conserve the total mass in a Vlasov system of equally spaced grids in phase space [21]. However, the use of non-equally spaced grids or other general grids can save on computational cost and is worthy of investigation.

In this sense, establishing exact conservation in a semi-Lagrangian form would be a challenging task. In shall section, we propose a non-conservative or semi-Lagrangian scheme that guarantees exact mass conservation.

6.1. CIP-CSLA [43]

We first discuss the conservation law, which is computed as

$$\frac{\partial f}{\partial t} + \frac{\partial(uf)}{\partial x} = 0. \quad (49)$$

As already seen, the CIP adopts an additional constraint, i.e., the spatial gradient, to represent the profile inside the grid cell. In order to ensure the conservative property, we here add another constraint,

$$\rho_{i+1/2}^n = \int_{x_i}^{x_{i+1}} f^n dx. \quad (50)$$

Therefore the spatial profile must be constructed to satisfy this additional constraint as well. If this is possible, f is advanced in the non-conservative form, but conservation is realized by ρ which is advanced by remapping of the Lagrangian volume, as described below.

The i th function piece $F_i(x)$ is determined so as to satisfy the continuity condition

$$\begin{cases} F_i(x_{i-1}) = f(x_{i-1}), & F_i(x_i) = f(x_i) \\ \partial F_i(x_{i-1})/\partial x = g(x_{i-1}), & \partial F_i(x_i)/\partial x = g(x_i) \\ \int_{x_{i-1}}^{x_i} F_i(x) dx = \rho_{i-1/2}. \end{cases} \quad (51)$$

In order to meet the above condition, a fourth-order polynomial can be chosen as an interpolation function $F_i(x)$; the time development of f and g is calculated simply by shifting the interpolation function $F_i(x)$ by $u\Delta t$ in the same way as Eq. (4) of the CIP method as

$$f_i^* = F_i(x_i - u_i^n \Delta t) = a_i^n \xi_i^4 + b_i^n \xi_i^3 + c_i^n \xi_i^2 + g_i^n \xi_i + f_i^n, \quad (52)$$

$$g_i^* = \partial F_i(x_i - u_i^n \Delta t)/\partial x = 4a_i^n \xi_i^3 + 3b_i^n \xi_i^2 + 2c_i^n \xi_i + g_i^n, \quad (53)$$

where $\xi_i = -u_i^n \Delta t$ and

$$\begin{aligned} a_i^n &= \frac{-5(6(f_{iup} + f_i)\Delta x_i - (g_{iup} - g_i)\Delta x_i^2 + 12\text{sgn}(u_i^n)\rho_{i-\text{sgn}(u_i^n)/2})}{2\Delta x_i^5}, \\ b_i^n &= \frac{4(7(f_{iup} + 8f_i)\Delta x_i - (g_{iup} - (3/2)g_i)\Delta x_i^2 + 15\text{sgn}(u_i^n)\rho_{i-\text{sgn}(u_i^n)/2})}{\Delta x_i^4}, \\ c_i^n &= \frac{-3(4(2f_{iup} + 3f_i)\Delta x_i - (g_{iup} - 3g_i)\Delta x_i^2 + 20\text{sgn}(u_i^n)\rho_{i-\text{sgn}(u_i^n)/2})}{2\Delta x_i^3}, \end{aligned} \quad (54)$$

$$\Delta x_i = x_{iup} - x_i,$$

$$iup = \begin{cases} i - 1 & \text{if } u_i^n \geq 0, \\ i + 1 & \text{if } u_i^n < 0, \end{cases}$$

$$\text{sgn}(u_i^n) = \begin{cases} +1 & \text{if } u_i^n \geq 0, \\ -1 & \text{if } u_i^n < 0. \end{cases}$$

The problem left for us is to calculate the time development of ρ . If we define the upstream departure point of x_i as

$$x_i^p = x_i - \int_t^{t+\Delta t} u_i dt, \quad (55)$$

the density contained between $[x_{i-1}^p, x_i^p]$ is simply remapped into $[x_{i-1}, x_i]$ at $t + \Delta t$. Therefore in the simplest case of $u\Delta t/\Delta x < 1$, the time development of ρ is calculated by the following equation.

$$\rho_{i-1/2}^{n+1} = \rho_{i-1/2}^n + \Delta\rho_{i-1}^n - \Delta\rho_i^n. \quad (56)$$

With the aid of Fig. 18, it is clear that $\Delta\rho_i^n$ is defined by the equation

$$\Delta\rho_i^n = \int_{x_i - u_i^n \Delta t}^{x_i} F_i^n(x) dx = -\left(\frac{a_i^n}{5}\xi_i^4 + \frac{b_i^n}{4}\xi_i^3 + \frac{c_i^n}{3}\xi_i^2 + \frac{g_i^n}{2}\xi_i + f_i^n\right)\xi_i, \quad (57)$$

where $\xi_i = -u_i^n \Delta t$, and each coefficient is equivalent to Eq. (54) at time step n .

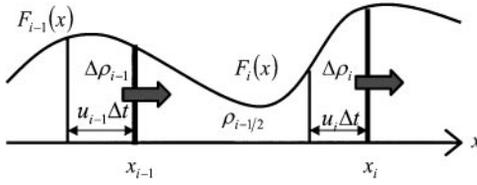


FIG. 18. The inflow and outflow of flux during Δt .

Therefore, the solution of Eq. (49) is given by Eqs. (52)–(57). Extension to more general equations is similar to the CIP method.

6.2. Numerical Tests

We present some sample calculations to test the procedure given in the previous section. The first example is an advection with variable velocity and Eq. (49) is solved under a given velocity field,

$$u = 1 + 0.5 \sin(2\pi x / 100),$$

with the initial condition

$$f(0, x) = \begin{cases} 1 & \text{if } 40 \leq x \leq 60 \\ 0 & \text{otherwise,} \end{cases}$$

where equally spaced grid points of $\Delta x = 100/(N - 1)$ and a time-step size of $\Delta t = 10/(N - 1)$ are used, N being the number of grid points.

We repeated the calculation by changing the total number of grid points to 101, 301, and 1001 to test grid dependence of the present scheme, the CIP scheme, the upwind scheme, and TVD with a superbee limiter [39], as shown in Fig. 19. The result of the first-order upwind scheme with $N = 10,001$ is also shown in Fig. 19. All profiles shown are after $10(N - 1)$ time steps, which corresponds to $t = 100$. Since no analytical solution is available, the result of the present scheme with $N = 1001$ is shown by the solid line in the figure.

We confirm that the accuracy has been improved at the discontinuity by the present scheme. Note that the first-order upwind scheme needs 10,001 grid points to obtain the same result as the present scheme or the CIP method using 101 grid points, which converges to one solution regardless of grid size. Furthermore, it has already been shown that most of the modern schemes such as TVD and ENO fail to reproduce the result with 101 grid points [42], as shown in Fig. 19.

The following one-dimensional Burger's equation is an interesting example of application to a non-linear equation.

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \lambda \frac{\partial^2 u}{\partial x^2} \equiv S. \quad (58)$$

The viscosity term is calculated with a finite-difference approach.

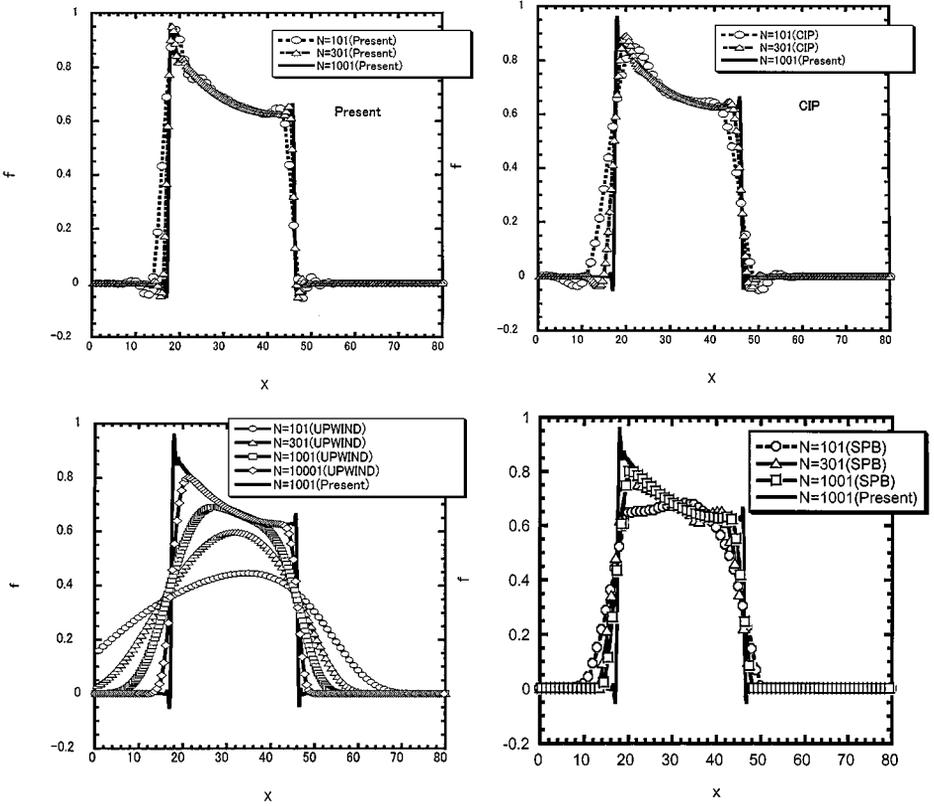


FIG. 19. Propagation of a square wave with a given velocity field by the present scheme, the CIP method, the upwind scheme, and the TVD scheme with a superbee limiter (SPB) at $t = 100$. The number of grids is 101 (circles), 301 (triangles), 1,001 (squares), and 10,001 (diamonds). For comparison, the result of the present scheme with 1,001 grids is shown by the solid line.

In order to calculate the spatial derivative of u , we differentiate Eq. (58); then

$$\frac{\partial u'}{\partial t} + u \frac{\partial u'}{\partial x} = -u'^2 + S'. \quad (59)$$

This equation is split into two phases: $\partial u'/\partial t + u \partial u'/\partial x = 0$ (advection) and $\partial u'/\partial t = -u'^2 + S'$ (non-advection). We calculate the advection phase with Eq. (4), and the non-advection phase with the following equation, applying a finite-difference method.

$$u_i^{n+1} = u_i^* - (u_i^*)^2 \Delta t.$$

The S and S' terms are treated by the method given in Section 3.2. In fact, first-order time integration is not necessarily used here. A higher order integration along the trajectory can be easily implemented.

For the calculation of $\rho (\equiv \int u dx)$, we transform Eq. (58) into the conservation form as follows.

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2/2)}{\partial x} = S. \quad (60)$$

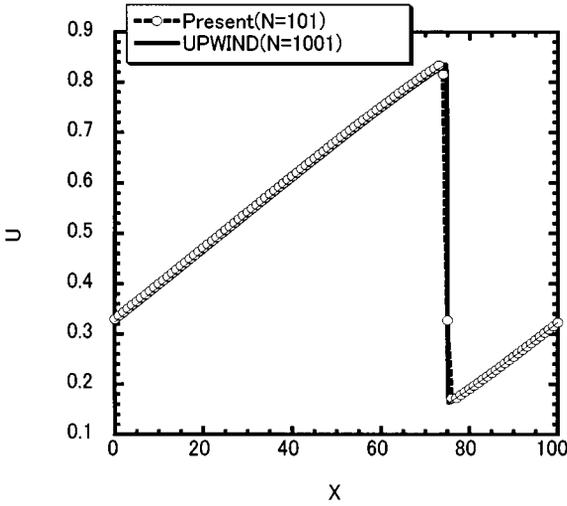


FIG. 20. The result of Burger's equation without viscosity with the present scheme with 101 grid points at $t = 100$. For comparison, the result of the first-order upwind scheme with 1001 grid points at $t = 100$ is shown by the solid line.

Figure 20 shows the calculation result at $t = 100$ with $\lambda = 0$ and the initial condition

$$u(0, x) = 0.5 + 0.4 \cos(2\pi x/100), \quad (61)$$

and equally spaced grid points of $\Delta x = 1.0$, a time-step size of $\Delta t = 0.1$, and mesh number $N = 101$ are used.

In order to check the exact speed of a shock wave, we show the result of the calculation with the conservative form of Burger's equation using the first-order upwind scheme with $N = 1001$.

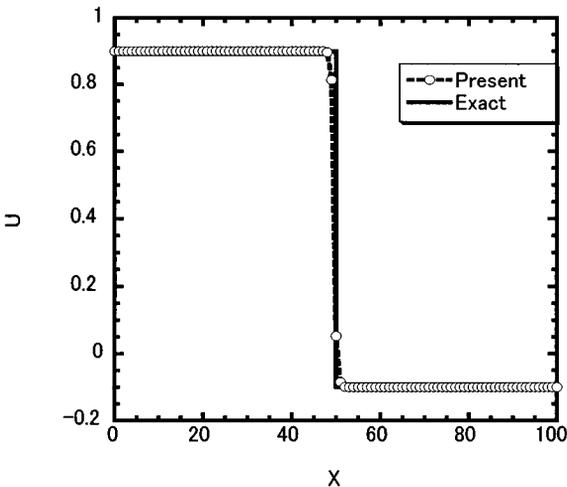


FIG. 21. The result of Burger's equation with viscosity with the present scheme at $t = 100$. The exact solution is shown by the solid line.

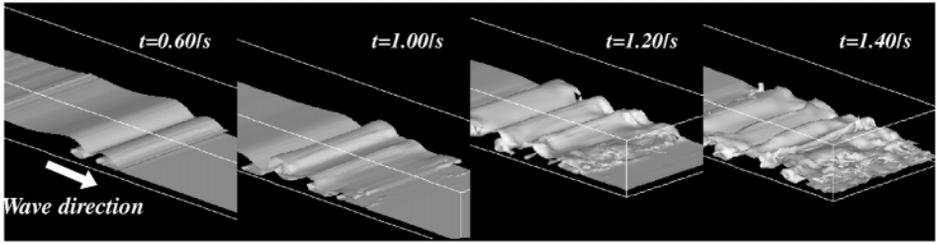


FIG. 22. Bird's-eye view of a gas-liquid turbulence mixing layer.

In this calculation, although the viscosity term is not considered, the speed of a shock wave can be exactly reproduced by the present scheme.

Next we solve Eq. (58) with a viscosity term and condition as

$$u(0, x) = \begin{cases} 0.9 & \text{if } x \leq 10 \\ -0.1 & \text{if } x > 10, \end{cases}$$

where equally spaced grid points of $\Delta x = 1$, and a time-step size of $\Delta t = 0.1$ are used. The coefficient of the viscosity term is set to $\lambda = 0.15$.

Figure 21 shows the profile after 1000 time steps that corresponds to $t = 10$, and the propagation speed agrees well with the exact solution.

7. SUMMARY

We have reviewed various families of the CIP method to address the problem of multi-phase flow. Significant improvements are still being made but we cannot discuss all of those efforts, such as, application to the finite-element method, extension to higher order polynomials, application to other equations such as diffusion or the Poisson equation. Application of the CIP method to various fields such as volcanic eruption, astrophysical jets [17], the GDI engine, and combustion is also expanding. Some of these fields are discussed in the special issue of the CFD Journal on the CIP method (vol. 8, No. 1, 1999). Before closing, we introduce an interesting application done by Mutsuda and Yasuda [20]. Figure 22 shows a bird's-eye view of a gas-liquid turbulence mixing layer driven by jet ejection, in which a three-dimensional domain of a solitary wave leads to very forceful wave-breaking on a double reef with two steps, and the surface profiles and entrained air bubble motions develop quickly into the turbulent mixing layer with strong three-dimensionality. This simulation has been done with one CPU on an Intel-based personal computer (with a 300-MHz Pentium II processor).

REFERENCES

1. A. A. Amsden and F. H. Harlow, A simplified MAC technique for incompressible fluid flow calculations, *J. Comput. Phys.* **6**, 322 (1970).
2. T. Aoki, Multi-dimensional advection of CIP (cubic-interpolated propagation) scheme, *CFD J.* **4**, 279 (1995).
3. J. B. Bell and D. L. Marcus, A second-order projection method for variable-density flows, *J. Comput. Phys.* **101**, 334 (1992).

4. R. Bermejo and A. Staniforth, The conversion of semi-Lagrangian advection schemes to quasi-monotone schemes, *Mon. Weather Rev.* **120**, 2622 (1992).
5. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
6. P. Colella and P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *J. Comput. Phys.* **54**, 174 (1984).
7. U. Ghia, K. N. Ghia, and C. T. Shin, High-resolution for incompressible flow using the Navier-Stokes equations and a multi-grid method, *J. Comput. Phys.* **48**, 387 (1982).
8. F. H. Harlow and J. P. Shannon, The splash of a liquid drop, *J. Appl. Phys.* **38**, 3855 (1967).
9. F. H. Harlow and A. A. Amsden, Numerical simulation of almost incompressible flow, *J. Comput. Phys.* **3**, 80 (1968).
10. F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow with free surface, *Phys. Fluids* **8**, 2182 (1965).
11. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
12. R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, *J. Comput. Phys.* **62**, 40 (1985).
13. D. Jacqmin, A variational approach to deriving smeared interface surface tension models, in *Barriers and Challenges in CFD*, edited by V. Venkatakrishnan *et al.* (Kluwer Academic, Dordrecht/Norwell, MA, 1998), p. 231.
14. K. C. Karki and S. V. Patankar, Pressure-based calculation procedure for viscous flows at all speeds in arbitrary configurations, *AIAA J.* **27**, 1167 (1989).
15. S. Karni, Multicomponent flow calculations by a consistent primitive algorithm, *J. Comput. Phys.* **112**, 31 (1994).
16. D. B. Kothe, W. J. Rider, S. J. Mosso, and J. S. Brock, *Volume tracking of interfaces having surface tension in two and three dimensions*, AIAA Paper 96-0859, 1996.
17. T. Kudoh, R. Matsumoto, and K. Shibata, Magnetically driven jets from accretion disks. III. 2.5-dimensional nonsteady simulations for thick disk case, *Astrophys. J.* **508**, 186 (1998).
18. C. E. Leith, Numerical simulation of the earth's atmosphere, in *Methods in Computational Physics, Vol. 4*, edited by B. Alder *et al.* (Academic Press, New York, 1965), p. 1.
19. W. H. McMaster, *1984 Computer codes for fluid-structure interactions*, Lawrence Livermore Laboratory Report UCRL-89724, 1984.
20. H. Mutsuda and T. Yasuda, Physical mechanism based on numerical simulations of strongly turbulent air-water mixing layer within surf-zone, *Proceedings 27th Int. Conf. on Coastal Eng., Sydney, 1999*.
21. T. Nakamura and T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space, *Comput. Phys. Commun.* **120**, 122 (1999).
22. J. Von Neumann and R. D. Richtmyer, A method for the numerical calculation of the hydrodynamic shocks, *J. Appl. Phys.* **21**, 232 (1950).
23. Y. Ogata and T. Yabe, Shock capturing with improved numerical viscosity in primitive Euler representation, *Comput. Phys. Commun.* **119**, 179 (1999).
24. S. Osher and J. A. Sethian, Front propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
25. Y. Pan and S. Banerjee, Numerical simulation of particle interactions with wall turbulence, *Phys. Fluids* **8**, 2733 (1996).
26. Y. Pan and S. Banerjee, Numerical investigation of the effects of large particles on wall-turbulence, *Phys. Fluids* **9**, 3786 (1997).
27. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (McGraw-Hill, New York, 1980).
28. S. V. Patankar and D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *Int. J. Heat Mass Transfer* **15**, 1787 (1972).
29. D. Peng, B. Merriman, S. Osher, H. K. Zhao, and M. Kang, A PDE-based fast local level-set method, *J. Comput. Phys.* **155**, 410 (1999).

30. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
31. C. S. Peskin and D. M. McQueen, A three-dimensional computational model for blood flow in the heart. I. Immersed elastic fibers in viscous incompressible fluid, *J. Comput. Phys.* **81**, 372 (1989).
32. E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* **130**, 269 (1997).
33. E. G. Puckett and J. S. Saltzman, A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics, *Physica D* **60**, 84 (1992).
34. D. K. Purnell, Solution of the advective equation by upstream interpolation with a cubic spline, *Mon. Weather Rev.* **104**, 42 (1975).
35. W. J. Rider and D. B. Kothe, *Stretching and Tearing Interface Tracking Methods*, AIAA-95-1717, 33rd Aerospace Sciences Meeting and Exhibit, Reno, 1995.
36. S. L. Shapiro and S. A. Teukolsky, *Black Holes, White Dwarfs, and Neutron Stars* (Wiley, New York, 1983).
37. A. Staniforth and J. Côté, Semi-Lagrangian integration scheme for atmospheric model—A review, *Mon. Weather Rev.* **119**, 2206 (1991).
38. M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).
39. P. K. Sweby, High-resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. Numer. Anal.* **21**, 995 (1984).
40. H. Takewaki, A. Nishiguchi, and T. Yabe, The cubic-interpolated pseudo-particle (CIP) method for solving hyperbolic-type equations, *J. Comput. Phys.* **61**, 261 (1985).
41. H. Takewaki and T. Yabe, Cubic-interpolated pseudo particle (CIP) method—application to nonlinear or multi-dimensional problems, *J. Comput. Phys.* **70**, 355 (1987).
42. R. Tanaka, T. Nakamura, T. Yabe, and H. Wu, A class of conservative formulation of the CIP method, *CFD J.* **8**, 1 (1999).
43. R. Tanaka, T. Nakamura, and T. Yabe, Constructing an exactly conservative scheme in a non-conservative form, *Comput. Phys. Commun.* **126**, 232 (2000).
44. E. T. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer-Verlag, Berlin, 1997).
45. S. O. Unverdi and G. A. Tryggvasson, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).
46. T. Utsumi, T. Kunugi, and T. Aoki, Stability and accuracy of the cubic interpolated propagation scheme, *Comput. Phys. Commun.* **101**, 9 (1996).
47. T. Wakisaka, S. Takeuchi, and J. H. Chung, A numerical study on the mixture formation process in fuel injection engines by means of the CIP method, *CFD J.* **8**, 82 (1999).
48. P. Y. Wang, T. Yabe, and T. Aoki, A general hyperbolic solver—the CIP method—applied to curvilinear coordinates, *J. Phys. Soc. Japan* **62**, 1865 (1993).
49. M. L. Wilkins, Use of artificial viscosity in multidimensional fluid dynamic calculations, *J. Comput. Phys.* **36**, 281 (1980).
50. M. L. Wilkins, *Computer Simulation of Dynamic Phenomena* (Springer-Verlag, New York 1999).
51. D. L. Williamson and P. J. Rasch, Two-dimensional semi-Lagrangian transport with shape-preserving interpolation, *Mon. Weather Rev.* **117**, 102 (1989).
52. P. R. Woodward and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* **54**, 115 (1984).
53. F. Xiao, *Numerical Scheme for Advection Equation and Multi-Layered Fluid Dynamics*, Ph.D thesis (Tokyo Institute of Technology, 1996).
54. F. Xiao, T. Yabe, and T. Ito, Constructing an oscillation-preventing scheme for the advection equation by a rational function, *Comput. Phys. Commun.* **93**, 1 (1996).
55. F. Xiao, T. Yabe, G. Nizam, and T. Ito, Constructing a multi-dimensional oscillation-preventing scheme for the advection equation by a rational function, *Comput. Phys. Commun.* **94**, 103 (1996).
56. F. Xiao *et al.*, An Efficient model for driven flow and application to a gas circuit breaker, *Comput. Model. Sim. Eng.* **1**, 235 (1996).

57. F. Xiao, T. Yabe, T. Ito, and M. Tajima, An algorithm for simulating solid objects suspended in stratified flow, *Comput. Phys. Commun.* **102**, 147 (1997).
58. F. Xiao and T. Yabe, Computation of complex flows containing rheological bodies, *CFD J.* **8**, 43 (1999).
59. X. Xu and A. Nadim, Deformation and orientation of an elastic slender body sedimenting in viscous liquid, *Phys. Fluids* **6**, 2889 (1994).
60. T. Yabe and E. Takei, A new higher-order Godunov method for general hyperbolic equations, *J. Phys. Soc. Japan* **57**, 2598 (1988).
61. T. Yabe and T. Aoki, A universal solver for hyperbolic-equations by cubic-polynomial interpolation. I. One-dimensional solver, *Comput. Phys. Commun.* **66**, 219 (1991).
62. T. Yabe, T. Ishikawa, P. Y. Wang, T. Aoki, Y. Kadota, and F. Ikeda, A universal solver for hyperbolic-equations by cubic-polynomial interpolation. II. Two- and three-dimensional solvers, *Comput. Phys. Commun.* **66**, 233 (1991).
63. T. Yabe and P. Y. Wang, Unified numerical procedure for compressible and incompressible fluid, *J. Phys. Soc. Japan* **60**, 2105 (1991).
64. T. Yabe and F. Xiao, Description of complex and sharp interface during shock wave interaction with liquid drop, *J. Phys. Soc. Japan* **62**, 2537 (1993).
65. T. Yabe and F. Xiao, Description of complex and sharp interface with fixed grids in incompressible and compressible fluid, *Computer Math. Applic.* **29**, 15 (1995).
66. T. Yabe, Interface capturing and universal solution of solid, liquid and gas by CIP method, *Proceedings Conf. High-Performance Computing on Multi-phase Flow*, Tokyo, 1997.
67. T. Yabe, Y. Zhang, and F. Xiao, A numerical procedure—CIP—to solve all phases of matter together, *Lecture Note in Physics*, Sixteenth Int. Conf. on Numerical Methods in Fluid Dynamics, (Springer-Verlag, New York, 1998), p. 439.
68. T. Yabe, Simulation of laser-induced melting and evaporation dynamics by the unified solver CIP for solid, liquid and gas, in *Mathematical Modeling of Weld Phenomena 4*, edited by H. Cerjak (Cambridge Univ. Press, Cambridge, U.K., 1998), p. 26.
69. T. Yabe, F. Xiao, and Y. Zhang, *Strategy for unified solution of solid, liquid, gas and plasmas*, AIAA Paper No. 99-3509, 30th AIAA Fluid Dynamics Conference, Norfolk, 1999.
70. S. Y. Yoon and T. Yabe, The unified simulation for incompressible and compressible flow by the predictor-corrector scheme based on the CIP method, *Comput. Phys. Commun.* **119**, 149 (1999).
71. D. L. Youngs, Time-dependent multi-material flow with large fluid distortion, in *Numerical Methods for Fluid Dynamics*, edited by K. W. Morton and M. J. Baines (Academic Press, New York, 1982), p. 273.
72. Y. Zhang and T. Yabe, Effect of ambient gas on three-dimensional breakup in coronet formation, *CFD J.* **8**, 378 (1999).
73. O. C. Zienkiewicz and J. Wu, A general explicit or semi-explicit algorithm for compressible and incompressible flows, *Int. J. Num. Meth. Eng.* **35**, 457 (1992).
74. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithm for fluids, *J. Comput. Phys.* **31**, 335 (1979).