Journal of Computational Physics **174,** 171–207 (2001)
doi:10.1006/jcph.2001.6888, available online at http://www.idealibrary.com on IDEAL®

Exactly Conservative Semi-Lagrangian Scheme for Multi-dimensional Hyperbolic Equations with Directional Splitting Technique

Takashi Nakamura,* Ryotaro Tanaka,† Takashi Yabe,‡ and Kenji Takizawa†

**Department of Electrical and Electronic Engineering, Utsunomiya University, 7-1-2 Yoto, Utsunomiya 321-8585, Japan; †Department of Energy Sciences, Tokyo Institute of Technology, 4259 Nagatsuta Midori-ku, Yokohama 226-8502, Japan; and ‡Department of Mechanical Engineering and Science, Tokyo Institute of Technology, 2-12-1 O-okayama Meguroku, Tokyo 152-8552, Japan*

E-mail: *ntakashi@cc.utsunomiya-u.ac.jp, †rtanaka@es.titech.ac.jp, ‡yabe@mech.titech.ac.jp

Received February 2, 2001; revised July 10, 2001

A new numerical method that guarantees exact mass conservation is proposed to solve multidimensional hyperbolic equations in semi-Lagrangian form. The method is based on the constrained interpolation profile (CIP) scheme and keeps the many good characteristics of the original CIP scheme. The CIP strategy is applied to the integral form of variables. Although the advection and nonadvection terms are separately treated, mass conservation is kept in the form of a spatial profile inside a grid cell. Therefore, it retains various advantages of the semi-Lagrangian solution with exact conservation, which has been beyond the capability of conventional semi-Lagrangian schemes. © 2001 Elsevier Science

Key Words: computational algorithm; CIP-CSL2; R-CIP-CSL2; semi-Lagrangian; mass conservation; multi-dimensions; cubic interpolation; monotone preserving.

1. INTRODUCTION

In recent years, with the computing environment being improved, a demand for high-precision, stable numerical methods is rapidly increasing in various fields of technology. The constrained interpolation profile (CIP) scheme, developed by Yabe and co-workers [1–4], for solving hyperbolic equations, also known as the cubic interpolated pseudoparticle/propagation scheme, has attracted a great deal of attention [5]. The CIP scheme is a low-diffusion and stable scheme and can solve hyperbolic equations with third-order accuracy in space [6]. This scheme has been successfully applied to various complex fluid flow problems, covering both compressible and incompressible flows, such as laser-induced evaporation, shock-wave generation, elastic-plastic flow, bubble collapse, and milk crown (for review see [7–9]). Furthermore the CIP scheme is essentially written as the semi-Lagrangian formulation. Therefore, it can be used for high Courant–Friedrichs–Lewy (CFL) conditions in explicit form and is stable for multiphase flow calculations.

Recently, a great deal of attention has been paid to semi-Lagrangian schemes and some excellent numerical schemes have been developed. These semi-Lagrangian schemes have been widely incorporated into numerical models for atmospheric flow (see Ref. [10] for one of the pioneering works). A review that serveys the semi-Lagrangian schemes to date is found in Ref. [11]. The semi-Lagrangian scheme is based on a Lagrangian invariant solution and the solution gives the time development of the value only at the spatial points. Therefore, the semi-Lagrangian schemes are naturally associated with non-conservative numerical formulations. In order to overcome the lack of conservation, many numerical approaches have been studied: schemes based on a higher order Hermite interpolations [12, 13], finite-volume approaches [14], and flux-form semi-Lagrangian schemes [15] based on the piecewise parabolic method [16]. Many applications require exact conservation of mass. For example, when we treat black-hole formation and plasma dynamics, small fractions of mass and charge generate a gravity wave and a large electric field, respectively, and therefore, the exact conservation of mass is necessary to the success of the numerical analysis. Therefore the development of the conservative semi-Lagrangian scheme should still be worthy further effort.

It is frequently demonstrated that the CIP method shows good conservation of mass, although the method is written in a nonconservative form. In a special case such as the solution of the Vlasov equation, it is possible to cast and improve the CIP method to exactly conserve mass [17]. However, it is not easy to apply this numerical technique to the solution of general hyperbolic equations. Therefore, the development of a conservative CIP method is earnestly desired. In such situations, authors have recently succeeded in developing new conservative semi-Lagrangian schemes called CIP–CSL4 [18] and CIP–CSL2 [19]. The schemes are based on the concept of the CIP scheme and preserve the excellent numerical features of the CIP scheme. In order to include these various families of schemes, we here extend the name CIP to mean constrained interpolation profile and CSL to mean conservative semi-Lagrangian scheme. CSL4 and CSL2 use fourth-order and quadratic polynomials, respectively. The schemes are written as semi-Lagrangian formulations and provide stable solutions under large CFL with exact mass conservation. In previous paper [18, 19], the scheme was applied to many problems in linear and nonlinear one-dimensional hyperbolic equations.

In this paper, we shall extend the CIP–CSL2 scheme to multi-dimensional equations and show some solutions by the schemes in two and three dimensions. The extension of CIP–CSL4 to multi-dimensions is found in Ref. [20]. In the following section, we present a brief introduction of the one-dimensional CIP and CIP–CSL2 schemes. Furthermore, we propose a rational CIP–CSL2 (R–CIP–CSL2) scheme, which has additional excellent numerical features such as monotone preserving and nonoscillatory features. In Section 3, the numerical procedure for the two-dimensional scheme is detailed and some examples of solutions in two dimensions are presented. Then, we describe the numerical procedure for the three-dimensional scheme and show some numerical solutions. Furthermore, because the fractional step technique is employed to extend the scheme to multi-dimensions although the scheme is essentially written in the semi-Lagrangian form, we try to apply the present multi-dimensional scheme to a semi-Lagrangian solution and examine the range of computational time intervals with which the scheme can provide reasonable numerical results. Finally, in Section 4, we summarize this paper briefly.

2. REVIEW OF CIP AND CIP–CSL2 METHODS

2.1. *CIP Method*

Although nature is continuous, digitization is unavoidable for implementation in numerical simulations. The primary goal of the numerical algorithm will be to retrieve the lost information inside the grid cell between these digitized points. Most of the numerical schemes proposed before, however, did not take care of real solutions inside the grid cell and resolution has been limited to the grid size. The CIP method proposed by one of the authors tries to construct a solution inside the grid cell close enough to this real solution of the given equation with some constraints. We here explain its strategy by using an advection equation,

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0. \tag{1}$$

When the velocity is constant, the solution of Eq. (1) gives a simple translational motion of field f with a velocity u. The initial profile (solid line of 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 of the profile inside the grid cell has been lost and it is hard to imagine the original profile and it is natural to imagine a profile like that shown by the solid line in Fig. 1c. Thus, numerical diffusion arises when we construct the profile by the linear interpolation even with the exact solution at grid points as 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, it suffers from overshooting. This process is the Lax–Wendroff scheme or Leith scheme.

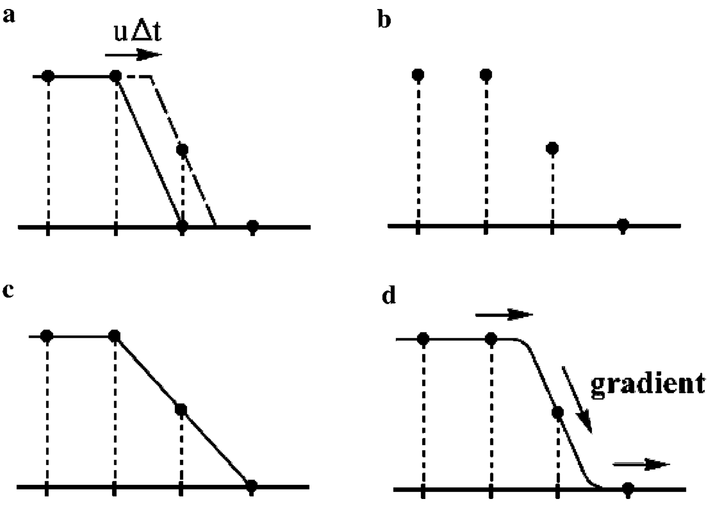

FIG. 1. The principle of the CIP method. (a) The solid line is the initial profile and the dashed line is an exact solution after advection, whose solution (b) is at discretized points. (c) When (b) is linearly interpolated, numerical diffusion appears. (d) In the CIP, the spatial derivative also propagates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

What made this solution worse? It is because we neglected the behavior of the solution inside the grid cell and merely followed the smoothness of the solution. From this experience, we understand that a method incorporating the real solution into the profile within a grid cell is quite an important subject. We propose to approximate the profile as shown below. Let us differentiate Eq. (1) with spatial variable x; then we get

$$\frac{\partial g}{\partial t} + u\frac{\partial g}{\partial x} = -\frac{\partial u}{\partial x}g, \tag{2}$$

where $g \equiv \partial f/\partial x$ stands for the spatial derivative of f. 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 a velocity u. By this equation, we can trace the time evolution of f and g on the basis of Eq. (1). If g could be predicted to propagate as shown by the arrows in Fig. 1d, it is easy to imagine that by this constraint, the solution would become 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 consistency has been demonstrated previously [9, 19].

If both the values of f and g are given at two grid points, the profile between these points can be interpolated by a cubic polynomial [3],

$$F_i^n(x) = a_i X^3 + b_i X^2 + g_i^n X + f_i^n, \tag{3}$$

$$\begin{aligned}
a_i &= \frac{g_i^n + g_{iup}^n}{\Delta x_i^2} + \frac{2\left(f_i^n - f_{iup}^n\right)}{\Delta x_i^3},\\
b_i &= \frac{3\left(f_{iup}^n - f_i^n\right)}{\Delta x_i^2} - \frac{2g_i^n + g_{iup}^n}{\Delta x_i},\\
\Delta x_i &= x_{iup} - x_i,\\
iup &= i - \operatorname{sgn}(u_i),\\
X &= (x - x_i),
\end{aligned} \tag{4}$$

where sgn(u) stands for the sign of u. Thus, the profile at the $(n+1)$th step is readily obtained by shifting the profile by $u_i\Delta t$, so that $f_i^{n+1} = F_i^n(x_i - u_i\Delta t)$ and $g_i^{n+1} = dF_i^n(x_i - u_i\Delta t)/dx$; then

$$\begin{aligned}
f_i^{n+1} &= a_i\xi_i^3 + b_i\xi_i^2 + g_i^n\xi_i + f_i^n,\\
g_i^{n+1} &= 3a_i\xi_i^2 + 2b_i\xi_i + g_i^n,
\end{aligned} \tag{5}$$

where we define $\xi_i = -u_i\Delta t$.

2.2 CIP–CSL2 Scheme

In this section, we shall describe a method to solve the one-dimensional conservative equation

$$\frac{\partial f}{\partial t} + \frac{\partial (uf)}{\partial x} = 0, \tag{6}$$

where u is a variable now. The CIP scheme given in the previous section uses the value f and its first-order spatial derivative $\partial_x f = \partial f/\partial x$ at the computational grid points as constraints for constructing a profile inside the grid cell. The CIP–CSL2 [19] and CSL4 [18] schemes require an additional constraint of the value integrated over neighboring two grid points,

$$\rho_i^n = \int_{x_i}^{x_{i+1}} f(x,t)\,dx, \tag{7}$$

where n indicates the time. Although the CIP method uses the time evolution of f and $g = \partial f/\partial x$ as constraints to define a cubic polynomial, it would be interesting to find a way to apply the CIP to the integrated value of f instead of to f itself. The motivation to employ this analogy stems from the following advection equation,

$$\frac{\partial D}{\partial t} + u\frac{\partial D}{\partial x} = 0. \tag{8}$$

Interestingly, if we take a spatial derivative of Eq. (8) and define $D' \equiv \partial D/\partial x$, we obtain a conservative-type equation,

$$\frac{\partial D'}{\partial t} + \frac{\partial (uD')}{\partial x} = 0. \tag{9}$$

Recalling that Eq. (9) is the same as Eq. (6), we come to the idea of using $D' = f$ in Eq. (9) and $D = \int f\,dx$ in Eq. (8). This procedure is exactly the same as that in Eq. (1), simply replacing f by $\int f\,dx$, together with Eq. (2), in which g is replaced by f. Thus the CIP procedure can be used for a pair of $\int f\,dx$ and f instead of f and $\partial f/\partial x$.

By this analogy, we shall introduce a function

$$D_i^n(x) = \int_{x_i}^{x} f(x',t)\,dx'. \tag{10}$$

$D_i^n(x)$ represents the accumulated mass from x_i to the upstream point x. We shall use a cubic polynomial to approximate this profile,

$$D_i^n(x) = \phi_i X^3 + \eta_i X^2 + f_i^n X, \tag{11}$$

where $X = x - x_i$. The role of spatial gradient g in the CIP method is now played by f, which is the spatial gradient of D in this scheme. By using the above relation, a profile of $f(x,t)$ between x_i and x_{i+1} is then obtained by taking the derivative of Eq. (11):

$$F_i^n(x) = \frac{\partial D_i^n(x)}{\partial x} = 3\phi_i X^2 + 2\eta_i X + f_i^n. \tag{12}$$

From the definition of D in Eq. (10), it is clear that

$$D_i^n(x_i) = 0, \quad D_i^n(x_{i+1}) = \rho_i^n. \tag{13}$$

Since $\partial D/\partial x$ gives a functional value f, it is also clear that

$$\frac{\partial D_i^n(x_i)}{\partial x} = f_i^n, \quad \frac{\partial D_i^n(x_{i+1})}{\partial x} = f_{i+1}^n. \tag{14}$$

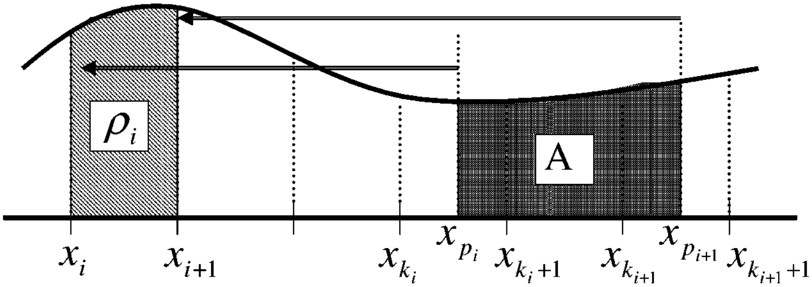

FIG. 2. The remapping procedure gives the time evolution of ρ. The region A can be found by tracing the trajectory.

Therefore, the coefficients ϕ_i and η_i are determined to satisfy the constraints of Eqs. (13) and (14). As a result of the above simultaneous equations, the coefficients are determined explicitly without any matrix solution as

$$\phi_i = \frac{f_i^n + f_{i+1}^n}{\Delta x_i^2} - \frac{2\rho_i^n}{\Delta x_i^3} \tag{15}$$

$$\eta_i = -\frac{2f_i^n + f_{i+1}^n}{\Delta x_i} + \frac{3\rho_i^n}{\Delta x_i^2}, \tag{16}$$

where $\Delta x_i \equiv x_{i+1} - x_i$. It is interesting to observe by comparing Eqs. (15) and (16) with Eq. (4) that the role of f, g in Eq. (4) is played by D, f, respectively, remembering that $D_i^n(x_i) - D_i^n(x_{i+1}) = -\rho_i^n$ from Eq. (13).

The time development of ρ is determined from the volume formed by two upstream departure points as shown in Fig. 2 and can be calculated from the classical conservative form of the conservation equation [14]

$$\rho_i^{n+1} = \int_{x_i}^{x_{i+1}} f(x, t + \Delta t)\, dx = \int_{x_{p_i}}^{x_{p_{i+1}}} f(x, t)\, dx, \tag{17}$$

where x_{p_i} is the particle position of the upstream departure point, calculated by

$$x_{p_i} = x_i + \int_{t+\Delta t}^{t} u\, dt. \tag{18}$$

This time integration is performed along the particle trajectory. Since the profile of the physical value f has already been interpolated by Eqs. (12), (15), and (16), the integration (17) can be estimated as

$$\rho_i^{n+1} = \int_{x_{p_i}}^{x_{k_i}+1} F_{k_i}^n(x)\, dx + \sum_{m=k_i+1}^{k_{i+1}-1} \int_{x_m}^{x_{m+1}} F_m^n(x)\, dx + \int_{x_{k_{i+1}}}^{x_{p_{i+1}}} F_l^n(x)\, dx, \tag{19}$$

where k_i indicates the cell which includes the departure points x_{p_i} and is determined by

$$x_{k_i} < x_{p_i} < x_{k_i+1}. \tag{20}$$

By using the relation $\int_{x_i}^{x} F_i^n(x)\,dx = \int_{x_i}^{x} (\partial D_i^n(x)/\partial x)\,dx = D_i^n(x)$, we can rewrite Eq. (19) as below and the time development of ρ is calculated:

$$\begin{aligned}\rho_i^{n+1} &= \rho_{k_i}^n - \int_{x_{k_i}}^{x_{p_i}} F_{k_i}^n(x)\,dx + \sum_{m=k_i+1}^{k_{i+1}-1} \int_{x_m}^{x_{m+1}} F_m^n(x)\,dx + \int_{x_{k_{i+1}}}^{x_{p_{i+1}}} F_l^n(x)\,dx \\ &= \left(D_{k_{i+1}}^n\left(x_{p_{i+1}}\right) - D_{k_i}^n\left(x_{p_i}\right)\right) + \sum_{m=k_i}^{k_{i+1}-1} \rho_m^n. \end{aligned} \tag{21}$$

In Eq. (21), it is assumed that $k_{i+1} - 1 \geq k_i$. In practice, the last term of the right-hand side of Eq. (21) vanishes when $k_{i+1} = k_i$.

Since Eq. (21) is written as the difference of $D_{k_i}^n(x_{p_i})$ for the time evolution of ρ, the sum of $D_{k_i}^n(x_{p_i})$ over the entire computational domain will vanish exactly; thus we can get an exact conservation relation between ρ_i^n and ρ_i^{n+1} by taking a summation of (21):

$$\sum_i \rho_i^{n+1} = \sum_i \rho_i^n. \tag{22}$$

Equation (22) guarantees the conservation defined in terms of the sum of the integrals over grid points $\sum_i \rho_i$ defined by Eq. (7) instead of a sum over values at grid points $\sum_i f_i$.

Next, let us turn to the time evolution of the value f. We calculate the value f in the same way as the original CIP scheme. The conservation Eq. (6) is rewritten as

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = G, \tag{23}$$

where $G \equiv -f\,\partial u/\partial x$. On the basis of the time-splitting algorithm of the CIP scheme [3], we split the solution of Eq. (23) into two steps:

advection phase

$$\partial f/\partial t + u\partial f/\partial x = 0, \tag{24}$$

non-advection phase

$$\partial f/\partial t = G. \tag{25}$$

After the advection phase is solved, the non-advection phase is calculated on the basis of values resulting from the advection phase.

In the advection phase, we make use of the local analytic solution of Eq. (24), which is well known as the Lagrangian invariant solution:

$$f(x_i, t + \Delta t) = f(x_{p_i}, t). \tag{26}$$

Since the profile of $f(x,t)$ between x_{k_i} and x_{k_i+1} is given by Eq. (12), the solution of the advection phase f^* is calculated as

$$f_i^* = F_{k_i}^n\left(x_{p_i}\right) = 3\phi_{k_i}\langle\xi\rangle^2 + 2\eta_{k_i}\langle\xi\rangle + f_{k_i}^n, \tag{27}$$

where $\langle\xi\rangle$ is the distance between these two points:

$$\langle\xi\rangle = x_{p_i} - x_{k_i}. \tag{28}$$

It should be noticed that $\langle\xi\rangle$ is neither $-u_i\Delta t$ nor $\int_{t+\Delta t}^{t} u\,dt$. In Eq. (27), ϕ_{k_i}, η_{k_i} are given by simply replacing i by k_i in Eqs. (15) and (16).

After the advection phase is calculated using Eq. (27), the result of the advection phase f^* is advanced to the value of the next time step f^{n+1} in the non-advection phase. The non-advection phase can be solved by conventional forward, finite difference method as

$$f_i^{n+1} = f_i^* + G\Delta t, \tag{29}$$

where $G = -f_i^*(\partial u(x_i, t)/\partial x)$ and the spatial derivative of the velocity $\partial u/\partial x$ is approximated by the simple centered finite difference. It should be noticed that although f is calculated separately by Eqs. (24) and (25) in non-conservative form, the mass conservation is recovered in constructing the spatial profile of f so as to satisfy Eq. (7).

Figure 3a shows the results of the linear propagation of the square wave after 1000 time steps. We see from Fig. 3, the CIP–CSL2 scheme provides a result quite similar to the original CIP scheme shown in Fig. 3b. As shown previously, this scheme can correctly

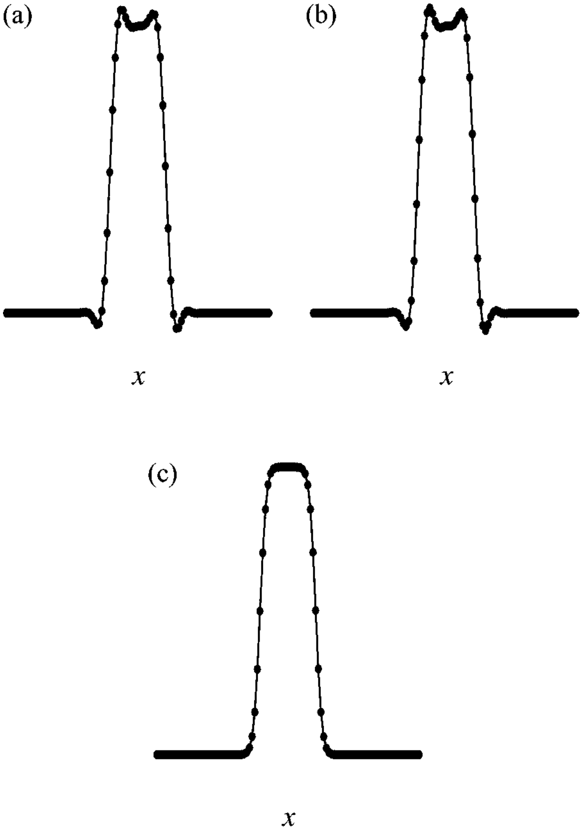

FIG. 3. Linear wave propagation with (a) the CIP–CSL2, (b) the CIP, and (c) the R–CIP–CSL2 after 1000 time steps with $u\Delta t/\Delta x = 0.2$.

calculate the propagation speed of the shock wave for the Burgers equation even without numerical viscosity [18, 19].

2.3 Rational CIP–CSL2 (R–CIP–CSL2) Method

Here, in order to show the possibility of extension of the present scheme, we point out that we can use other interpolation functions instead of the cubic polynomial represented by Eq. (11). For example, we can use a rational interpolation function,

$$D^n_{rational}(x) = \frac{(\kappa_i X^3 + \chi_i X^2 + \phi_i X)}{(1 + \alpha\beta_i X)}, \tag{30}$$

which had been proposed as a rational CIP scheme by Xiao *et al.* [21]. In the interpolation function (30), α is a switching parameter. The coefficients χ_i, ϕ_i, and β_i are temporally determined so that the interpolation function (30) satisfies the conditions (13) and (14) by setting $\kappa_i = 0$ and $\alpha = 1$ at first. By keeping only this β_i, and then κ_i, χ_i and ϕ_i are finally determined by the conditions (13) and (14) with $\alpha = 0$. Therefore, κ_i vanishes when $\alpha = 1$ and the interpolation function (30) coincides with the cubic polynomial function (11) when $\alpha = 0$ (for details see [21]):

$$\phi_i = f^n_i + \rho^n_{i+1}\alpha\beta_i \tag{31}$$

$$\chi_i = S_i\alpha\beta_i + \left(S_i - f^n_i\right)\big/\Delta x_i - \kappa_i\Delta x_i \tag{32}$$

$$\kappa_i = \left[f^n_i - S_i + \left(f^n_{i+1} - S_i\right)(1 + \alpha\beta_i\Delta x_i)\right]\big/\Delta x_i \tag{33}$$

$$\beta_i = \left[\left|\left(S_i - f^n_i\right)\big/\left(f^n_{i+1} - S_i\right)\right| - 1\right]\big/\Delta x_i \tag{34}$$

$$S_i = \rho^n_{i+1}\big/\Delta x_i \tag{35}$$

$$\alpha = \begin{cases} 1 & \text{for } \left(S_i - f^n_i\right)\big/\left(f^n_{i+1} - S_i\right) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{36}$$

Even for this interpolation function, the time development of ρ can be calculated according to Eq. (21) and the solution of the advection phase for f is given by

$$f^*_i = \frac{\partial D^n_{rational}(x_i + \xi)}{\partial x} = \frac{(3\kappa_i\xi^2 + 2\chi_i\xi + \phi_i)}{(1 + \alpha\beta_i\xi)} - \alpha\beta_i\frac{(\kappa_i\xi^3 + \chi_i\xi^2 + \phi_i\xi)}{(1 + \alpha\beta_i\xi)^2}. \tag{37}$$

It had been proved that the profile interpolated by this rational function is monotone [21]. Thus, we can construct the CIP–CSL2 scheme, which has excellent numerical features such as being monotone preserving and nonoscillatory. We show one typical result by this rational CIP–CSL2 scheme in Fig. 3c.

3. EXTENSION TO TWO DIMENSIONS

3.1. Formulation in Two Dimensions

In this and the following sections, we shall extend the one-dimensional CIP–CSL2 scheme to higher dimensions by the fractional step technique. By using the fractional step technique, the one-dimensional scheme can be easily extended to a multi-dimensional scheme and we

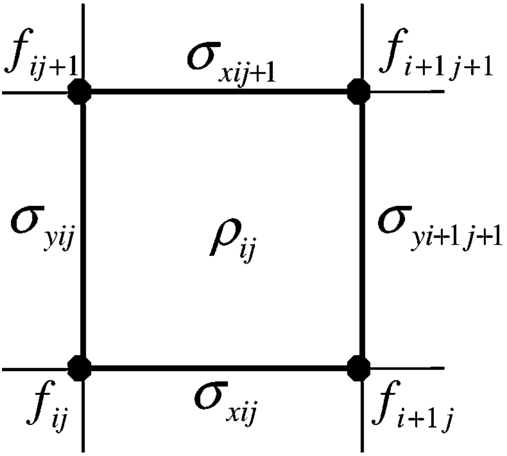

FIG. 4. The variables in the two-dimensional CIP–CSL2 scheme.

can avoid tedious efforts to implement the codes because the solution is performed by reciprocal use of one-dimensional solutions.

First, in order to simplify the discussion in the following, we shall introduce the 1D algorithm

$$\mathit{CIPCSL1D}(U, P, PN, R, RN, m, \lambda). \tag{38}$$

An explicit procedure for the algorithm *CIPCSL1D* is described in Appendix A. Thus, the above one-dimensional CIP–CSL2 scheme is given by *CIPCSL1D*$(u, f^n, f^{n+1}, \rho^n, \rho^{n+1}, i, x)$.

In the two-dimensional scheme, as shown in Fig. 4, an integrated value of f within a cell surrounded by four computational grid points (i, j), $(i+1, j)$, $(i, j+1)$, and $(i+1, j+1)$ is introduced,

$$\rho_{ij}^n = \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} f(x, y, t)\, dx\, dy, \tag{39}$$

and the scheme conserves the sum of ρ_{ij} exactly during computation.

Let us consider a two-dimensional conservative equation in the Cartesian coordinates x, y,

$$\frac{\partial f}{\partial t} + \frac{\partial (uf)}{\partial x} + \frac{\partial (vf)}{\partial y} = 0, \tag{40}$$

where u and v correspond to the velocities in the x- and y-directions, respectively.

For simplicity, in this paper, we assume that f is a scalar and describe the implementation in the Cartesian coordinates. However, it has already been demonstrated that the CIP scheme can handle non-scalars and be extended to non-Cartesian coordinates such as a curvilinear coordinate system [22]. As mentioned in the previous section, the present CIP–CSL2 and R–CIP–CSL2 schemes take over the same concept and numerical procedures as the CIP scheme, and therefore can be easily extended to the solution of non-scalars and a non-Cartesian coordinate system. The implementation of this scheme in the non-Cartesian coordinate and its application to the Navier–Stokes equation are now in progress.

It is well known that a one-dimensional solver is easily extended to a multidimensional solver by the fractional step technique. In the fractional step technique, the solution of the two-dimensional conservative Eq. (40) is split into two sequential solutions:

$$\text{Step 1}\quad \partial f/\partial t + \partial(uf)/\partial x = 0, \tag{41}$$

$$\text{Step 2}\quad \partial f/\partial t + \partial(vf)/\partial y = 0. \tag{42}$$

The solution of the original two-dimensional conservative equation (40) is given by sequential solution of Eqs. (41) and (42). As easily understood in view of Fig. 4, the solution of Step 1 is given by

$$\mathit{CIPCSL1D}\big(u,\ f^n,\ f^{step1}, \sigma_x^n, \sigma_x^{step1}, i, x\big). \tag{43}$$

We should note that the line density σ_x defined by

$$\sigma_{xij}^n = \int_{x_i}^{x_{i+1}} f(x, y_j, t)\, dx \tag{44}$$

plays the same role as ρ in Eq. (7) in the case of the one-dimensional algorithm in the previous section, as clearly seen from Fig. 5a. In a similar way, using the line density

$$\sigma_{yij}^n = \int_{y_j}^{y_{j+1}} f(x_i, y, t)\, dy, \tag{45}$$

the solution of Step 2 is given by

$$\mathit{CIPCSL1D}\big(u,\ f^{step1},\ f^{n+1}, \sigma_y^{step1}, \sigma_y^{n+1}, j, y\big). \tag{46}$$

However, we must note that σ_y^{step1} is not yet obtained after Step 1. Therefore we need a method to approximate it. This situation is quite similar to the fractional step solution of the CIP method [2, 17].

Next, we shall discuss how to estimate the evolution of σ_y in Step 1. In deriving an equation for σ_y, we assume that the advection velocity $u(x, y, t)$ is uniform from (i, j) to $(i, j+1)$ along the y-direction within one cell and set to $\overline{u}_j(x, t) = (u(x, y_j, t) + u(x, y_{j+1}, t))/2$ for $y_j \le y \le y_{j+1}$. Thus, the integration of Eq. (41) in the y-direction,

$$\int_{y_j}^{y_{j+1}} \left\{ \frac{\partial f}{\partial t} + \frac{\partial(\overline{u}_j f)}{\partial x} \right\} dy = 0, \tag{47}$$

leads to the following advection equation for the governing equations of a function of the line density $\sigma_{yj}(x, t) = \int_{y_j}^{y_{j+1}} f(x, y, t)\, dx$,

$$\frac{\partial \sigma_{yj}}{\partial t} + \frac{\partial(\overline{u}_j \sigma_{yj})}{\partial x} = 0. \tag{48}$$

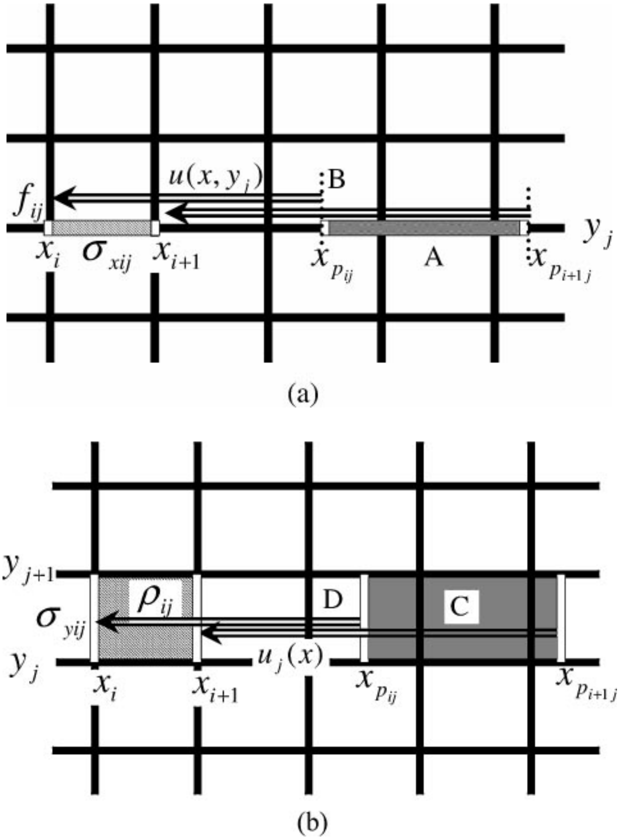

FIG. 5. Schematic view of the numerical procedure in two dimensions. (a) Time revolution of σ_x is given by a remapping procedure. The region A can be found by the tracing the trajectory in the x-direction with the velocity $u(x, y_j)$. The integration along the line A gives the solution of σ_x. Time evolution of f is given by interpolating the value at a point B. (b) Time revolution of ρ is given by a remapping procedure. The region C can be found by the tracing the trajectory in the x-direction with the velocity $\bar{u}_j(x)$. The volume within a region C gives the solution of ρ and the time evolution of σ_y is given by line density along a line D.

The above advection equation is the same as Eq. (41) except that the values f and u are replaced by σ_{yj} and $\bar{u}_j$. This means that σ_{yj} is advected in the same manner as f as clearly seen from Fig. 5b. Furthermore, we should remember that in the solution of Step 1, as a natural extension of the one-dimensional case, the time evolution of ρ_{ij} is calculated using a classical conservative form,

$$\rho_{ij}^{step1} = \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} f(x, y, t + \Delta t)\, dy\, dx = \int_{x_{p_{ij}}}^{x_{p_{i+1j}}} \int_{y_j}^{y_{j+1}} f(x, y, t)\, dy\, dx, \tag{49}$$

where x_{pij} is an upstream departure point for x_i and is calculated using the equation

$$x_{pij} = x_i + \int_{t+\Delta t}^{t} \bar{u}_j(x, t)\, dt. \tag{50}$$

In the case where the time interval Δt is small enough, we can assume that the velocity is constant in time and the upstream departure points x_{pij} are estimated by $x_{pij} = x_i - u(x_i, y_{j+1/2}, t)\Delta t$. It is well known that when the fractional step technique is used in each direction of the two-dimensional advection equations, the accuracy of the trajectory in multi-dimensional space will deteriorate for large CFL numbers. However, there are many merits to using the fractional steps technique, such as the simple numerical procedure and easy extension to higher dimensions. Furthermore, as will be shown later, the numerical results are reasonably good except where the CFL number is too large.

By using a function of the line density $\sigma_{yj}(x,t) = \int_{y_j}^{y_{j+1}} f(x, y, t)\, dx$, Eq. (49) is rewritten as

$$\rho_{ij}^{step1} = \int_{x_i}^{x_{i+1}} \sigma_{yj}(x, t + \Delta t)\, dx = \int_{x_{p_i}}^{x_{p_{i+1}}} \sigma_{yj}(x, t)\, dx. \tag{51}$$

Because this is analogous to Eq. (17) but σ_{yj} is used instead of f, it is obvious that the time development of ρ can be estimated by applying the one-dimensional CIP–CSL2 scheme to a pair of ρ and σ_y instead of ρ and f. Therefore, in order to calculate σ_y^{step1} and ρ^{step1}, we can directly apply the one-dimensional algorithm

$$CIPCSL1D\big(\overline{u}, \sigma_y^n, \sigma_y^{step1}, \rho^n, \rho^{step1}, i, x\big). \tag{52}$$

In the same way, σ_x^{step1} is advanced to σ_x^{n+1} in Step 2 by

$$CIPCSL1D\big(\overline{v}, \sigma_x^{step1}, \sigma_x^{n+1}, \rho^{step1}, \rho^{n+1}, j, y\big), \tag{53}$$

where $\overline{v}_i(y,t) = (v(x_i, y, t) + v(x_{i+1}, y, t))/2$.

Here, it should be noticed that the conservation of ρ is guaranteed exactly while σ is not conserved like f in one dimension. We summarize explicitly all calculation procedures for the two-dimensional scheme in Appendix A.

In the above procedure, ρ does not seem to be directly connected to f but only connected through σ_x and σ_y, which are temporally introduced. Furthermore, in this two-dimensional CIP–CSL2, only the one-dimensional profile of f is interpolated. Hence it is not clear whether $f(x, y, t)$ thus obtained can really satisfy Eq. (39). Thus, a question arises if whether exact conservation is really attained by this whole procedure, or not. In the following, we shall prove that the profile of f thus obtained is consistent with the integrated value ρ.

3.1.1. *Proof of Conservation in 2D*

In Fig. 6, we illustrate the process of interpolation of the value $f(x, y, t)$ inside the grid cell of $x_i \le x \le x_{i+1}$ and $y_j \le y \le y_{j+1}$. First, f at the point $A(x, y_j)$ between (x_i, y_j) and (x_{i+1}, y_j) is interpolated in the x-direction. If we use the quadratic polynomial similar to Eqs. (12), (15), and (16), then we obtain

$$f_{(A)}^n = 3\phi_{xij} X^2 + 2\eta_{xij} X + f_{ij}^n, \tag{54}$$

$$\phi_{xij} = \frac{f_{ij}^n + f_{i+1j}^n}{\Delta x^2} - \frac{2\sigma_{xij}^n}{\Delta x^3}, \tag{55}$$

$$\eta_{xij} = -\frac{2 f_{ij}^n + f_{i+1j}^n}{\Delta x} + \frac{3\sigma_{xij}^n}{\Delta x^2}, \tag{56}$$

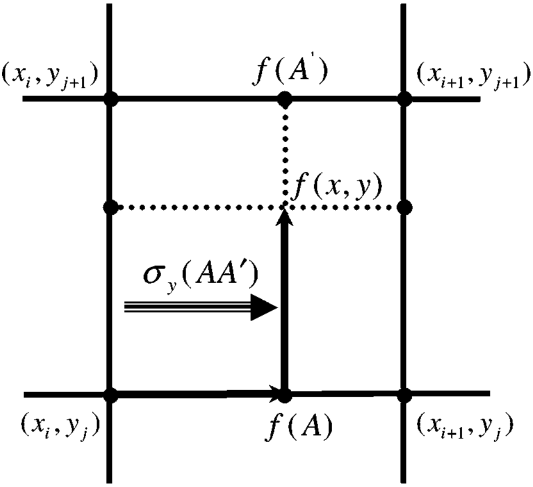

FIG. 6. The process of construction of the two-dimensional profile within a computational cell.

where $X = x - x_i$ and $\Delta x = x_{i+1} - x_i$. Similarly, we can interpolate f at the point $A'(x, y_{j+1})$ between (x_i, y_{j+1}) and (x_{i+1}, y_{j+1}) as follows:

$$f^n_{(A')} = 3\phi_{xij} X^2 + 2\eta_{xij} X + f^n_{ij+1}, \tag{57}$$

$$\phi_{xij} = \frac{f^n_{ij+1} + f^n_{i+1j+1}}{\Delta x^2} - \frac{2\sigma^n_{xij+1}}{\Delta x^3}, \tag{58}$$

$$\eta_{xij} = -\frac{2 f^n_{ij+1} + f^n_{i+1j+1}}{\Delta x} + \frac{3\sigma^n_{xij+1}}{\Delta x^2}. \tag{59}$$

Next, the value of $f(x, y, t)$ is interpolated between A and A' in the y-direction, and we have

$$F^n_{ij}(x, y) = 3\phi_{yij} Y^2 + 2\eta_{yij} Y + f^n_{(A)}, \tag{60}$$

$$\phi_{yij} = \frac{f^n_{(A)} + f^n_{(A')}}{\Delta y^2} - \frac{2\sigma^n_{y(AA')}}{\Delta y^3}, \tag{61}$$

$$\eta_{yij} = -\frac{2 f^n_{(A)} + f^n_{(A')}}{\Delta y} + \frac{3\sigma^n_{y(AA')}}{\Delta y^2}, \tag{62}$$

where $Y = y - y_j$, $\Delta y = y_{j+1} - y_j$, and $\sigma^n_{y(AA')}$ is the line density along the line $A - A'$ and is given by the interpolation of σ^n_y in the x-direction:

$$\sigma^n_{y(AA')} = 3\phi_{ij} X^2 + 2\eta_{ij} X + \sigma^n_{yij}, \tag{63}$$

$$\phi_{xij} = \frac{\sigma^n_{yij} + \sigma^n_{yi+1j}}{\Delta x^2} - \frac{2\rho^n_{ij}}{\Delta x^3}, \tag{64}$$

$$\eta_{xij} = -\frac{2\sigma^n_{yij} + \sigma^n_{yi+1j}}{\Delta x} + \frac{3\rho^n_{ij}}{\Delta x^2}. \tag{65}$$

The above interpolation gives us the two-dimensional profile of f within a computational cell and Eqs. (54)–(59) correspond to the procedure (43), Eqs. (60)–(62) to Eq. (46), and

Eqs. (63)–(65) to Eq. (52). If we write down the above two-dimensional interpolation, we obtain the explicit expression

$$F_{ij}^n(x, y) = \sum_{l=0}^{2} \sum_{m=0}^{2} C_{lm} X^l Y^m, \tag{66}$$

where the coefficients C_{lm} are given in Appendix B. It is verified easily from Eqs. (54)–(65) that the interpolated profile (66) leads to the nine relations

$$\begin{gathered}
\int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} F_{ij}^n(x, y)\, dx\, dy = \rho_{ij}^n, \\
\int_{x_i}^{x_{i+1}} F_{ij}^n(x, y_j)\, dx = \sigma_{xij}^n, \quad \int_{x_i}^{x_{i+1}} F_{ij}^n(x, y_{j+1})\, dx = \sigma_{xij+1}^n, \\
\int_{y_j}^{y_{j+1}} F_{ij}^n(x_i, y)\, dy = \sigma_{yij}^n, \quad \int_{y_j}^{y_{j+1}} F_{ij}^n(x_{i+1}, y)\, dy = \sigma_{yi+1j}^n, \\
F_{ij}^n(x_i, y_j) = f_{ij}^n, \quad F_{ij}^n(x_{i+1}, y_j) = f_{i+1j}^n, \\
F_{ij}^n(x_i, y_{j+1}) = f_{ij+1}^n, \quad F_{ij}^n(x_{i+1}, y_{j+1}) = f_{i+1j+1}^n,
\end{gathered} \tag{67}$$

and the profile $F_{ij}^n(x, y)$ is consistent with the definition of σ_{xij}^n, σ_{yij}^n, ρ_{ij}^n, and f_{ij}^n. In particular, Eq. (67) proves that the spatial profile of $F_{ij}^n(x, y)$ gives the correct value of ρ when it is integrated within a cell, thus guaranteeing mass conservation. Furthermore, in many applications, the above two-dimensional profile of f within a computational cell is quite useful for the calculation of surface tension, for example.

3.1.2. *Numerical Solution in Two Dimensions*

To demonstrate the accuracy and efficiency of the present method, some numerical solutions in two dimensions will be given in this section.

First, in order to examine the basic numerical features of the two-dimensional CIP–CSL2 scheme using the fractional step technique, we shall describe solutions in which the time interval is small enough, such as CFL < 1.

We use a rectangular grid with uniform spacing $\Delta x = \Delta y = 1$ and employ 100×100 grid points. The time interval is determined in order to satisfy CFL ≤ 0.4 in all grid points. At the beginning of the computation, we set the value of f and then the initial value of ρ is calculated by $\rho_{ij} = (f_{i,j} + f_{i+1,j} + f_{ij+1} + f_{i+1,j+1})\Delta x \Delta y/4$.

At first, we apply the present method to the two-dimensional solid-body rotation problem known as Zalesak's solid-body problem [23]. Figure 7a shows the schematic view of this test problem. The value of f inside the cut-off cylinder is 1.0, while outside the cylinder $f = 0.0$, and the solid-body rotation is defined with velocity component

$$u = -2\pi y, \quad v = 2\pi x. \tag{68}$$

The lateral boundary condition was made to be a free-slip boundary. Figures 7b and 7c show profiles of ρ at the initial time and after one complete revolution. In Fig. 7, the line contour of ρ is plotted from $\rho = 0.0$ to $\rho = 1.2$ with increments of 0.1. The maximum and minimum values are $f = +1.06$ and -9.24×10^{-2}, respectively. The present method restores well

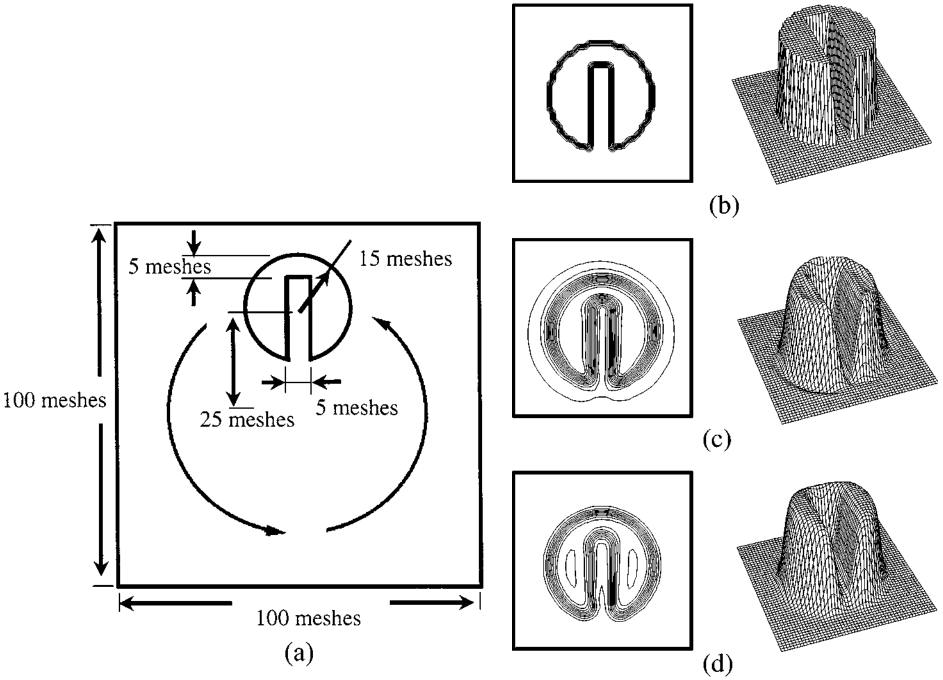

FIG. 7. (a) Schematic view of Zalesak's solid-body rotation problem. The value of f inside a cut-out cylinder is 1.0, while outside $f = 0.0$. The velocities are given by $v_x = -\omega y$ and $v_y = \omega x$, where ω is an angular velocity and $\omega = 2\pi$. Contour plots and three-dimensional views of the profile of ρ for Zalesak's problem: (b) initial state, (c) the computational result after one complete evolution, and (d) the computational result with the rational interpolation function.

the shape of the initial profiles and gives a stable, weakly diffusive but nonmonotone result. Furthermore, Fig. 7d shows the result with the R–CIP–CSL2, where it is seen that the monotone and nonoscillatory properties are maintained, and the maximum and minimum values are $f = +1.00$ and -3.98×10^{-4}, respectively. The conservation errors of total mass of the CIP–CSL2 and R–CIP–CSL2 schemes are less than 2.40×10^{-7} and 1.51×10^{-8}, respectively, until the computation ends. These errors are caused by mass loss at the free boundaries. The root mean square error and the numerical scores are represented in Table I.

As the next example, Fig. 8 shows the propagation of a square wave at an angle of 45° to the computational grid orientation with the velocity $u = v = 1.0$. A periodic boundary

TABLE I

Results of the Solid-Body Rotation Problem with the CIP–CSL2 Scheme and R–CIP–CSL2 Scheme: Root Mean Square Error (RMS), Relative Error of the Total Mass Conservation (mass error), and Minimum (min) and Maximum (max) Value of *f*

	RMS	Mass error	Max	Min
CSL2	1.61×10^{-3}	2.40×10^{-3}	$+1.09$	-9.24×10^{-2}
Rational CSL2	2.10×10^{-3}	1.51×10^{-8}	$+1.00$	-3.98×10^{-4}

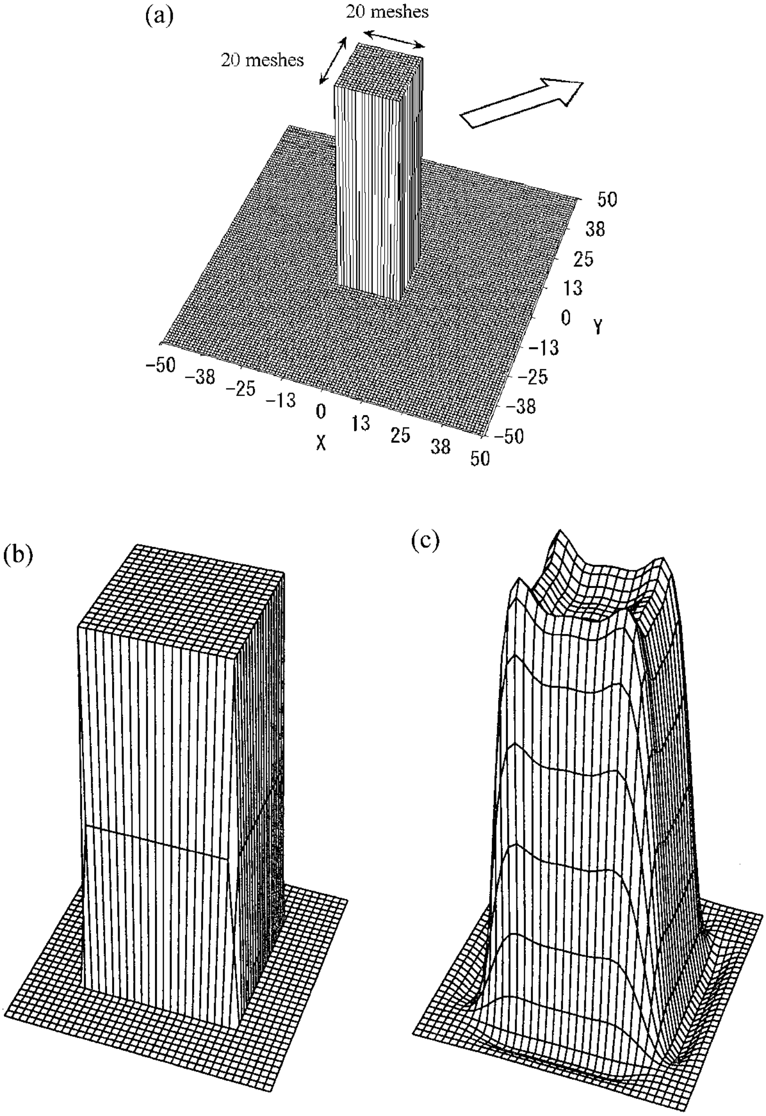

FIG. 8. (a) Schematic view of the advection of a square wave to the oblique direction. The value of f inside the rectangle is 1, while outside $f = 0$. The computational domain is periodic in x and y. (b) A profile of ρ at an initial time. (c) A profile of ρ after 1000 time steps with the constant velocity $u = v = 1.0$.

condition is used. The initial condition is

$$f = \begin{cases} 1 & (|x|, |y| < 10) \\ 0 & \text{otherwise.} \end{cases} \tag{69}$$

Figures 8b and 8c show the profile of ρ at the initial step and after 1000 steps, respectively. The maximum and minimum values at the end of computation are $f = +1.10$ and $f = -5.34 \times 10^{-2}$, respectively, and the overshoot and undershoot of the profile of f is suppressed well.

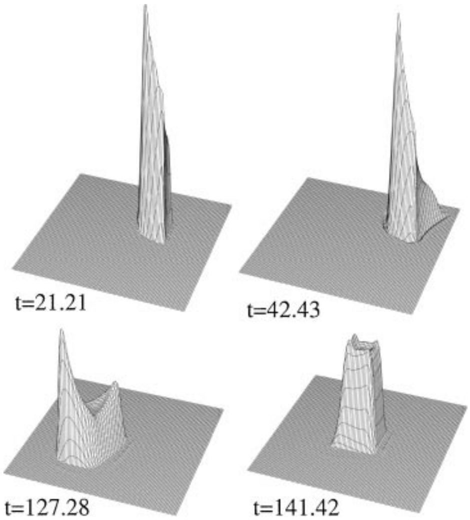

FIG. 9. Three-dimensional views of the time development of ρ in the fluctuating velocity field given by Eq. (70). ρ is advected to the oblique direction and compressed and expanded severely by a large fluctuation of the velocity.

Next, the velocity is changed to a fluctuating oblique velocity field given by

$$u = v = \frac{1}{\sqrt{2}} \frac{1}{1 + 0.5 \sin\left[\frac{2\pi}{100}(x + y)\right]}. \tag{70}$$

Figure 9 shows the time development of ρ. In spite of a large velocity fluctuation, the calculation was stably executed without severe numerical oscillation. Compression and expansion of the rectangle are correctly calculated. The comparison is made in Fig. 10 for

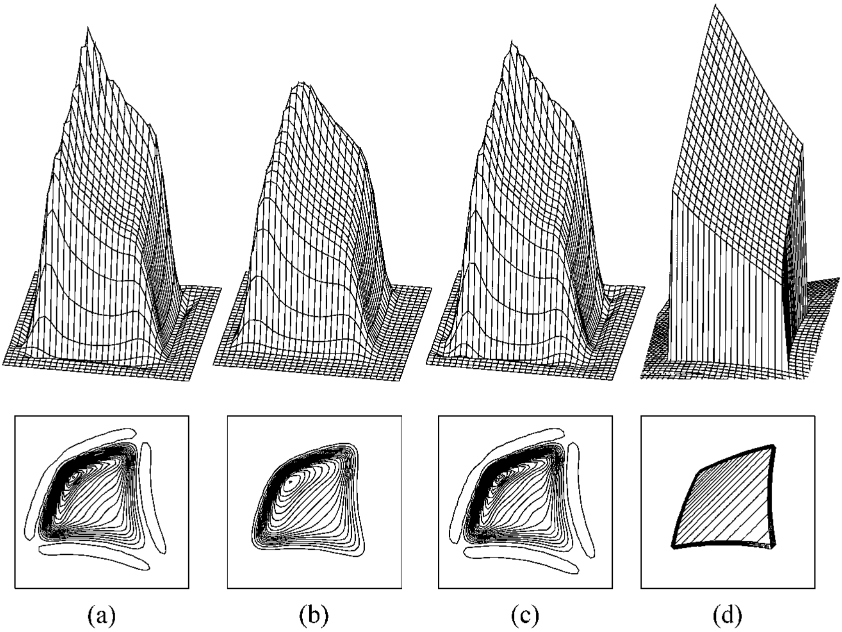

FIG. 10. Overview profile and line contour of f at $t = 98.99$ using (a) the CIP–CSL2, (b) the R–CIP–CSL2, (c) original CIP scheme using the two-dimensional interpolation procedure called type A, and (d) analytical result.

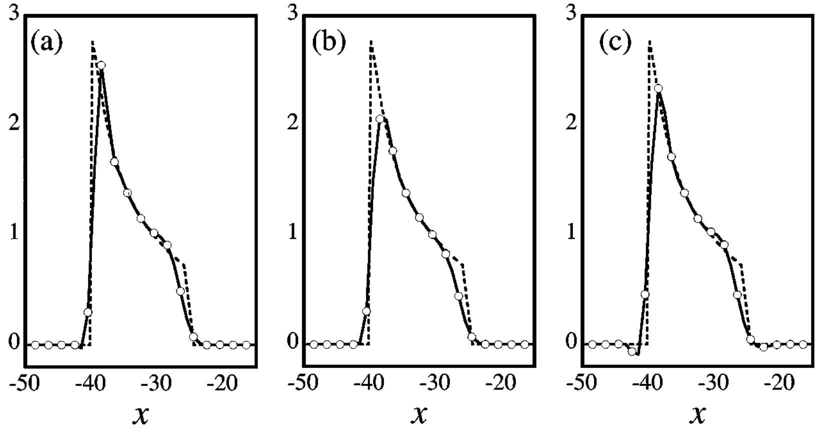

FIG. 11. Cross section of the profile of f along a line $x = y$ at $t = 98.99$. The dashed line represents the analytic solution and the solid lines represent the results of (a) present scheme, (b) present scheme using a rational interpolation function, and (c) original CIP.

the profile of f at $t = 98.99$ calculated by (a) the CIP–CSL2, (b) the R–CIP–CSL2, (c) the original CIP scheme, and (d) the analytical solution. The line contour of f is plotted from $f = 0.0$ to $f = 3.0$ with increments of 0.1. Figure 11 shows the profiles of f on the line $x = y$. Although it is well known that the CIP scheme excels in capturing discontinuities, the present scheme can also model discontinuities as well as the CIP scheme. Furthermore, as seen from the figures, the present scheme using a rational function provides the nonoscillatory and monotone profile. Figure 12 shows the relative error of mass conservation. It is verified that the present scheme conserves the total mass exactly within the computational round-off error. The numerical scores of each scheme are presented in Table II. In Table II, the negative mass ratio means that the proportion of a volume of the negative mass to the total volume and is calculated by $\sum_{\rho<0} \rho_{ij} / \sum_{All} \rho_{ij}$.

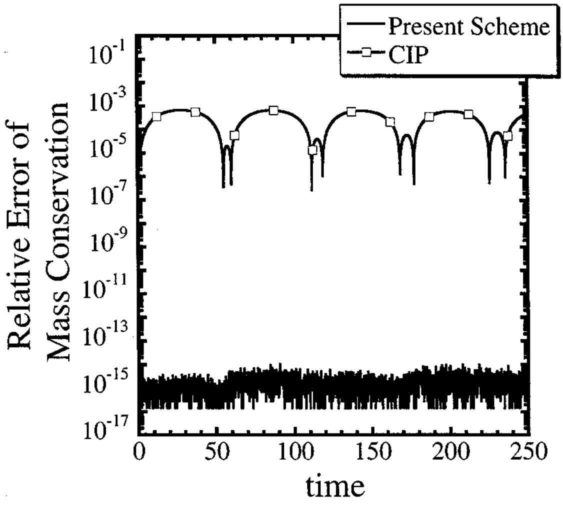

FIG. 12. Time evolution of the relative error of the mass conservation.

TABLE II

Results of the Wave Propagation Problem in the Fluctuating Oblique Velocity Field with the CIP–CSL2 Scheme and R–CIP–CSL2 Scheme after t = 98.99

	Mass error	Max	Min	Negative mass ratio
CSL2	9.09×10^{-15}	$+2.46$	-8.76×10^{-2}	2.01×10^{-2}
Rational CSL2	2.70×10^{-15}	$+2.00$	-7.10×10^{-25}	2.70×10^{-25}
CIP	3.06×10^{-4}	$+2.46$	-1.04×10^{-1}	2.38×10^{-2}
Theory	—	$+2.80$	0.00	—

Note. The proportion of the volume of the negative mass to the total volume is the negative mass ratio.

As the next example, the velocity field is set to

$$u = -\frac{x}{\sqrt{x^2+y^2}}, \quad v = -\frac{y}{\sqrt{x^2+y^2}}, \tag{71}$$

corresponding to a uniform convergence of the mass over the entire computational domain into one point. Such a velocity field often appears in the solution of problems that require exact conservation of mass, such as in the star formation. The boundary condition was made to be a free-slip boundary. The value of f within a cylinder of 40-grid radius located at the origin is 1.0, while outside the cylinder $f = 0.0$ at $t = 0$. The time development of ρ by using the CSL2 scheme is shown in Fig. 13a. While the theoretical minimum value of f is always 0.0, the computed minimum value of f at $t = 40.0$ is $f = -0.600$, which

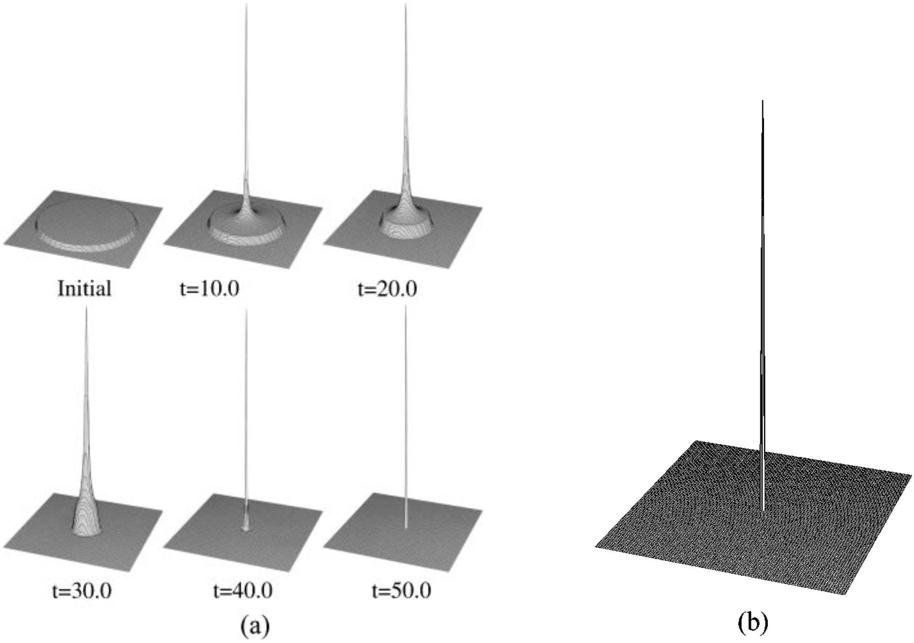

FIG. 13. (a) Three-dimensional views of the time development of ρ in the velocity field given by Eq. (71) with the CIP–CSL2 scheme. All the mass is finally compressed into the origin and condensed within one computational cell. (b) Three-dimensional views of ρ at t = 40.0 with the R–CIP–CSL2 scheme.

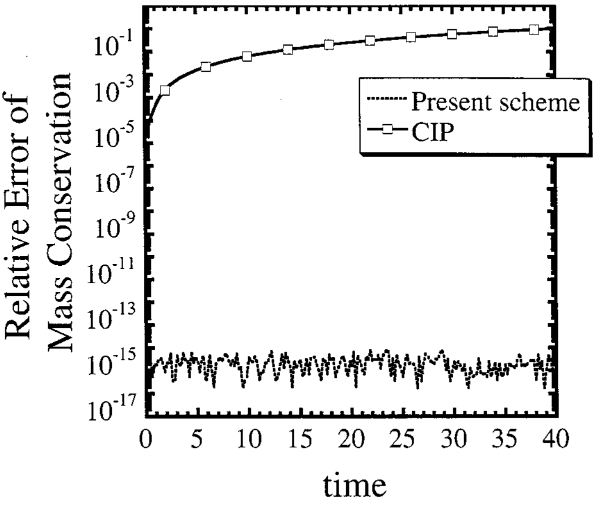

FIG. 14. Time evolution of the relative error of the mass conservation. The dotted and solid curves represent the results of the present scheme and CIP scheme, respectively.

corresponds to 1.2×10^{-2}% of the maximum value of f. As shown in Fig. 13b, however, this undershooting vanishes when the R–CIP–CSL2 scheme is used. The mass converges toward the origin as time proceeds, and at $t = 40$ all the mass is concentrated into the central cell as predicted theoretically from the initial condition and velocity field. Until the computation ends after $t = 40$, all the mass is stably retained at the origin. The relative error of the mass conservation is shown in Fig. 14 as a function of time. The original CIP scheme loses the mass for the most part around the origin where the direction of the velocity changes, while the CIP–CSL2 scheme conserves the total mass very well until the computation ends.

As a final example, we apply the scheme to idealized kinematic frontogenesis in meteorology [24] in order to demonstrate that the scheme provides good results even if the velocity is strongly fluctuating and more complex. The details of the problem are given by Doswell [25] and the analytical solution is known. In the problem, a circular vortex is assumed and the velocity is given by

$$v_x = -v_T \frac{y}{r} \tag{72}$$

$$v_y = v_T \frac{x}{r} \tag{73}$$

$$v_T(r) = \operatorname{sech}^2(r)\tanh(r)/v_{T0}, \tag{74}$$

where $r = \sqrt{x^2 + y^2}$ and v_{T0} is a constant which is determined so that maximum value of v_T is equal to unity. The initial condition of f is distributed in the y-direction and given by

$$f = -\tanh(y/2). \tag{75}$$

As with other computational conditions such as the mesh condition or width of the domain, except the time interval Δt, the same conditions as those of Ranĉić [24] are employed. We use a computational domain $R = \{(x, y) \mid -4 < x < 4, -4 < y < 4\}$ and employ 60×60

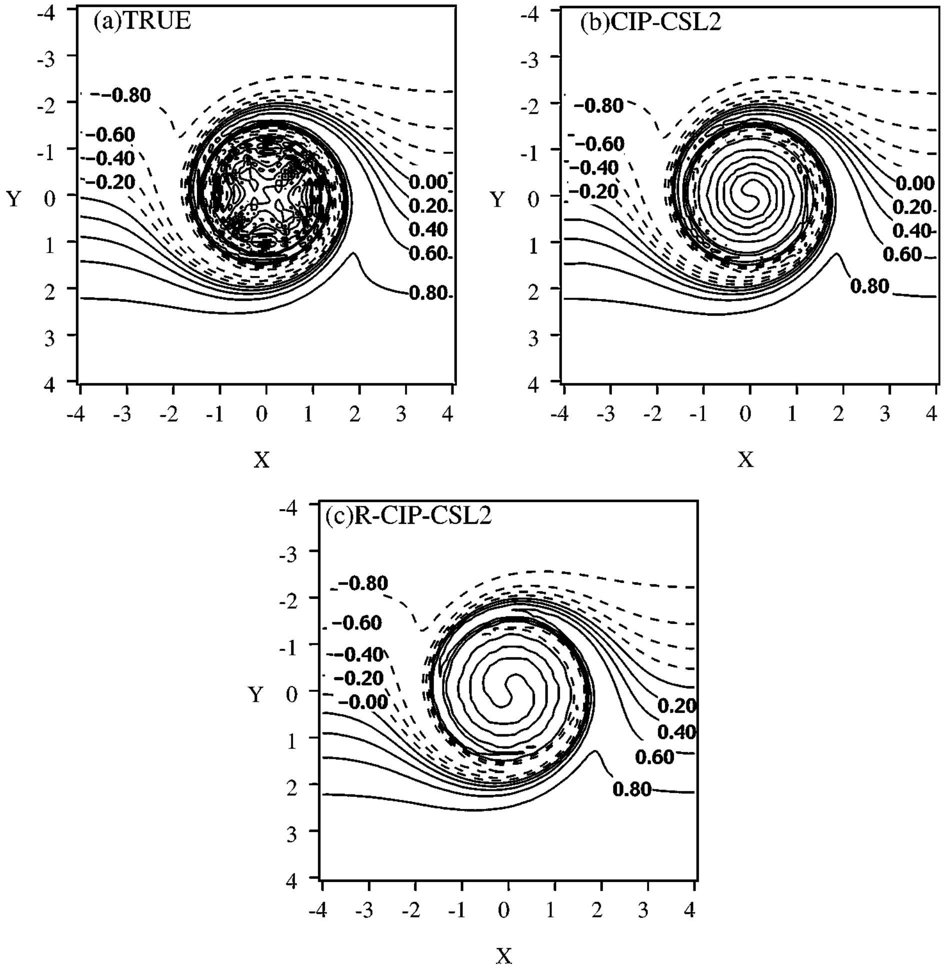

FIG. 15. Numerical results of the non-semi-Lagrangian solution of Doswell's frontogenesis experiment with the CIP–CSL2 scheme and the R–CIP–CSL2 scheme at $t = 16$. Courant number CFL $= 0.4$ is used. (a) Theoretical solution, (b) results with CIP–CSL2, and (c) results with R–CIP–CSL2.

grid points. The computations are executed with the CFL $= 0.4$. The line contour of the numerical result of the CIP–CSL2 scheme and the analytical result are shown in Figs. 15b and 15a, respectively. Furthermore, Fig. 15c shows the numerical result of R–CIP–CSL2. In each panel of Fig. 15, the line contour is plotted from $f = -0.8$ to $f = 0.8$ at increments of 0.2. In comparison with the theoretical solution presented in Fig. 15a, the numerical result agrees very well with the theoretical one. This is much more evident in Fig. 16, which shows three-dimensional views of the profile of f at $t = 16$. As shown in these figures, the scheme can stably calculate the fine structure around the origin. Especially the R–CIP–CSL2 scheme provides the monotone profile of f in addition to the accuracy. The scores of the numerical results are given in Table III; RMS indicates the root mean square. Edisp, Ediss, and Etotal are dispersion error, dissipation error, and the summation of Edisp and Ediss, respectively. Each error is estimated in the same way as that described in Ref. [24].

TABLE III

Results of Doswell's Frontogenesis Problem at *t* = 16 with CFL = 0.40 (Non-Semi-Lagrangian Solution) Dissipation (Ediss), Dispersion (Edisp), and Total (Etotal) Errors

	RMS	Edisp	Ediss	Etotal
CSL2	7.92×10^{-2}	6.49×10^{-3}	6.54×10^{-4}	6.56×10^{-3}
Rational CSL2	9.30×10^{-2}	8.79×10^{-3}	1.27×10^{-4}	8.92×10^{-3}

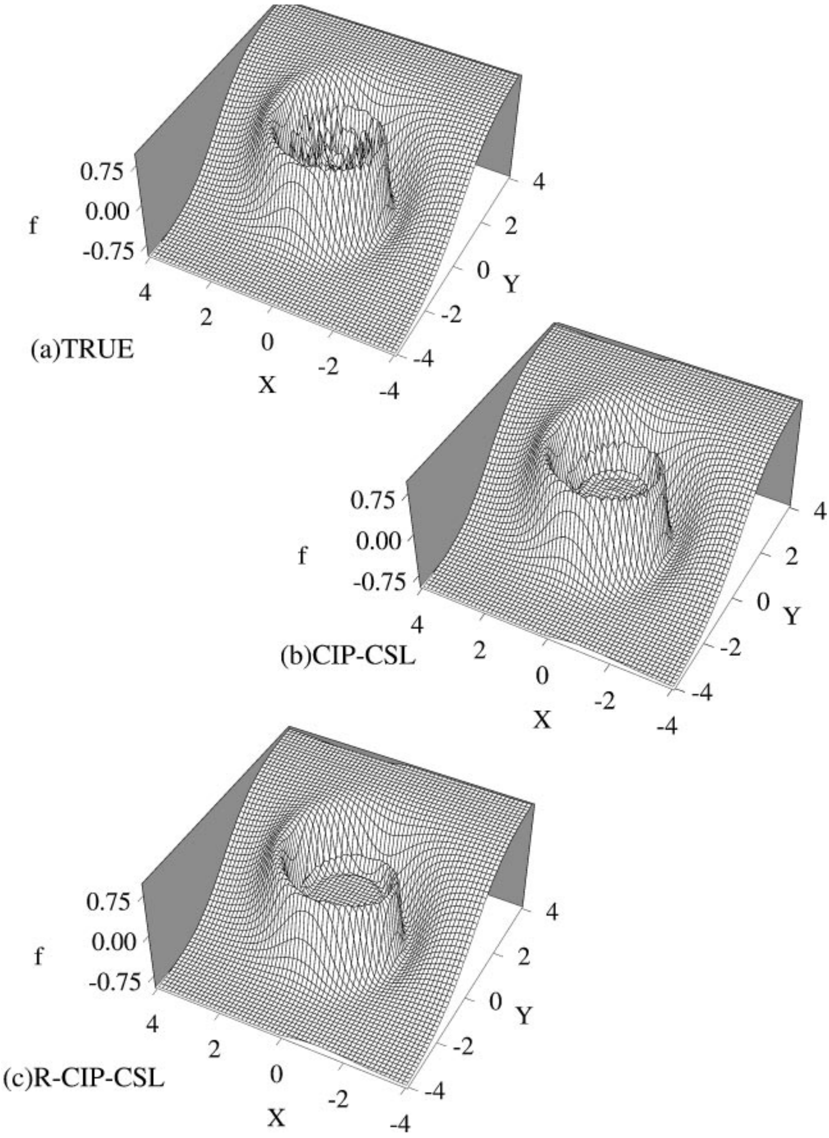

FIG. 16. Three-dimensional views of the numerical results of Doswell's frontogenesis experiment with the CIP–CSL2 scheme and the R–CIP–CSL2 scheme at $t = 16$. (a) Theoretical solution, (b) results with CIP–CSL2, and (c) results with R–CIP–CSL2.

3.2. Extension to Three Dimensions

3.2.1. *Formulation in Three Dimensions*

Once the two-dimensional scheme is established, it is straightforward to extend it to three dimensions. As a natural extension of the two-dimensional version, an integrated value of f within a cell surrounded by eight computational grid points (x_i, y_j, z_k), (x_{i+1}, y_j, z_k), (x_i, y_{j+1}, z_k), (x_i, y_j, z_{k+1}), (x_{i+1}, y_{j+1}, z_k), (x_i, y_{j+1}, z_{k+1}), (x_{i+1}, y_j, z_{k+1}), and $(x_{i+1}, y_{j+1}, z_{k+1})$,

$$\rho^n_{ijk} = \int_{z_k}^{z_{k+1}} \int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} f(x, y, z, t)\, dx\, dy\, dz \tag{76}$$

is introduced and $\sum_{ijk} \rho_{ijk}$ is conserved exactly by using the present three-dimensional scheme. We make use of the fractional step technique; thus, the solution of the three-dimensional conservation equation

$$\partial f/\partial t + \partial(v_x f)/\partial x + \partial(v_y f)/\partial y + \partial(v_z f)/\partial z = 0 \tag{77}$$

is split into three sequential steps:

$$\text{Step 1} \quad \partial f/\partial t + \partial(uf)/\partial x = 0, \tag{78}$$

$$\text{Step 2} \quad \partial f/\partial t + \partial(vf)/\partial y = 0, \tag{79}$$

$$\text{Step 3} \quad \partial f/\partial t + \partial(wf)/\partial z = 0. \tag{80}$$

In the three-dimensional case, the procedure of each step is almost the same as that in two-dimensional case, and the solution is given simply by repeating the one-dimensional scheme. For example, in Step 1, the value f_{ijk} is advected by the one-dimensional CIP–CSL2 algorithm

$$\mathit{CIPCSL1D}\big(u,\ f^n,\ f^{step1}, \sigma^n_x, \sigma^{step1}_x, i, x\big), \tag{81}$$

where σ_x is the line density shown in Fig. 17 and is given by integrating f in the x-direction

$$\sigma^n_{xijk} = \int_{x_i}^{x_{i+1}} f(x, y_j, z_k, t)\, dx. \tag{82}$$

The line densities in the other directions, $\sigma^n_{yijk} = \int_{y_j}^{y_{j+1}} f(x_i, y, z_k, t)\, dy$ and $\sigma^n_{zijk} = \int_{z_k}^{z_{k+1}} f(x_i, y_j, z, t)\, dz$, are advected by using the surface densities S instead of ρ as in the two-dimensional solution,

$$\mathit{CIPCSL1D}\big(u', \sigma^n_y, \sigma^{step1}_y, S^n_{xy}, S^{step1}_{xy}, i, x\big) \tag{83}$$

$$\mathit{CIPCSL1D}\big(u'', \sigma^n_z, \sigma^{step1}_z, S^n_{zx}, S^{step1}_{zx}, i, x\big), \tag{84}$$

where $u'(x, t) = (u(x, y_j, z_k, t) + u(x, y_{j+1}, z_k, t))/2$, $u''(x, t) = (u(x, y_j, z_k, t) + u(x, y_j, z_{k+1}, t))/2$, and the surface densities S_{xy}, S_{yz}, and S_{zx} are defined by

$$S^n_{xyijk} = \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} f(x, y, z_k, t)\, dy\, dx, \tag{85}$$

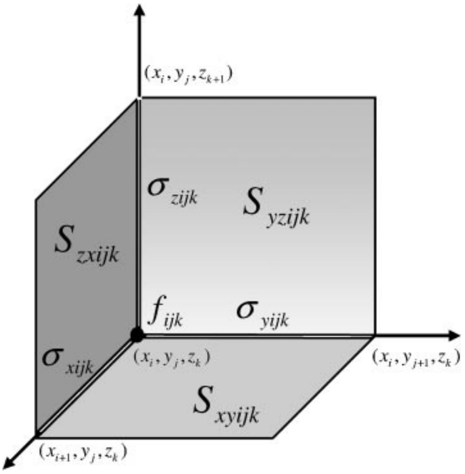

FIG. 17. A location of the variable f_{ijk}, the line densities σ_{xijk}, σ_{yijk}, σ_{zijk}, and the surface densities S_{xyijk}, S_{yzijk}, S_{zxijk} in the three-dimensional CIP–CSL2 scheme.

$$S^n_{yzijk} = \int_{z_k}^{z_{k+1}} \int_{y_j}^{y_{j+1}} f(x_i, y, z, t)\, dy\, dz, \tag{86}$$

$$S^n_{zxijk} = \int_{x_i}^{x_{i+1}} \int_{z_k}^{z_{k+1}} f(x, y_j, z, t)\, dz\, dx. \tag{87}$$

Then, the task left for us is the estimation of S^{step1}_{yz}, which is needed to advect the line densities in the Step 2 and Step 3 solutions. Similarly to the advection equation for the line density (48) and with the assumption that the velocity is constant for $y_j \le y \le y_{j+1}$ and $z_k \le z \le z_{k+1}$,

$$\begin{aligned} u(x, y, z, t) &= u'''(x, t) \\ &= (u(x, y_j, z_k, t) + u(x, y_{j+1}, z_k, t) + u(x, y_j, z_{k+1}, t) \\ &\quad + u(x, y_{j+1}, z_{k+1}, t))/4, \end{aligned} \tag{88}$$

we obtain the following advection equation for the surface density defined by $S_{yz}(x, t) = \int_{y_j}^{y_{j+1}} \int_{z_k}^{z_{k+1}} f(x, y, z, t)\, dz\, dy$:

$$\frac{\partial S_{yz}}{\partial t} + \frac{\partial (u''' S_{yz})}{\partial x} = 0. \tag{89}$$

The above advection equation is the same as Eq. (41) except that the values of f and u are replaced by S_{yz} and u'''. This means that S is advected in the same manner as f in the one-dimensional solution. Furthermore, we should remember that ρ is just the integrated value of S_{yz} in x-direction. Thus, a pair of values (ρ, S_{yz}) corresponds to a pair (ρ, f) in the one-dimensional solution. Therefore, the one-dimensional CIP–CSL2 solution is applied to the time development of (ρ, S_{yz}), and the solutions $(\rho^{step1}, S^{step1}_{yz})$ are given by

$$\mathit{CIPCSL1D}\big(u''', S^n_{yz}, S^{step1}_{yz}, \rho^n, \rho^{step1}, i, x\big). \tag{90}$$

By the above procedure, the set of values $\{\rho^{step1}, S^{step1}_{xy}, S^{step1}_{yz}, S^{step1}_{zx}, \sigma^{step1}_x, \sigma^{step1}_y, \sigma^{step1}_z, f^{step1}\}$ that are the solution of Step 1 is calculated. We can solve Step 2 and Step 3 similarly, and we summarize explicitly all the calculation procedures in three dimensions in Appendix A.

The three-dimensional profile of f within a computational cell is described by the form

$$f(x, y, z, t) = \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{2} C_{lmn} X^l Y^m Z^n, \tag{91}$$

where $Z = z - z_k$ and the 27 coefficients C_{lmn} are determined so that the interpolated profile (91) is consistent with the values of f, σ, S, and ρ in the same way as in two dimensions. We summarize all the coefficients of this three-dimensional profile in Appendix B.

3.2.2. *Cost of Computation*

It is natural to pose the objection that the CIP–CSL2 uses the integrated values in addition to the physical value and the schemes seem to be the same as other schemes that use twice as many mesh points. We address the following points concerning such an ojection.

(a) As shown previously [19], other interpolation schemes such as the cubic spline use cubic interpolation functions like CIP–CSL2 fail to give the correct result at high wave number.

(b) The cubic spline scheme uses the same cubic polynomial function as given by Eq. (3). While the cubic spline requires matrix solutions to determine the coefficients of a polynomial in the general cases, the coefficients in the CIP–CSL2 scheme are explicitly calculated according to Eqs. (15) and (16). Therefore, generally in one dimension, the computational cost of the CIP–CSL2 scheme for interpolation is expected to be much less than that of the cubic spline and the calculation costs do not increase even if an additional variable is introduced. Actually, the CPU time required for a simple advection problem in one dimension is cubic Lagrange/CIP = 1.0, spline/CIP = 1.68 (the Thomas method is used to solve the matrix), CIP–CSL2/CIP = 1.20, R–CIP–CSL2/CIP = 1.60, PPM [16]/CIP = 2.31.

(c) In the present scheme, the solution is given by the reciprocal use of the one-dimensional scheme and we can reuse the same subroutine to calculate all kinds of values in each fractional step. Therefore, the total computational load is proportional to the frequency of calling the subroutine. As shown in Appendix A, the number of required subroutine calls is proportional to the dimensions and given by $\alpha \times 2^{\alpha-1}$, where α is the dimension. However, due to the fact mentioned in (a), reducing the grid points by β can give $\beta^{\alpha+1}$ reduction of computational load for a fixed CFL. Therefore, the actual required computational cost is only proportional to $L_t = L_{one}(\alpha 2^{\alpha-1}/\beta^{\alpha+1})$, where L_{one} is the computation cost for the calculation in one dimension. The conventional numerical results and their comparison to the cubic spline scheme suggest that the β is larger than 2 [17, 19].

(d) In addition to (c), in most practical applications, we use variables such as thermal conductivity, viscosity and temporal variables for matrix solution. Let L_0 be the cost for the solution of a number of these variables; the required computational cost in then $L_t + L_0$. Since usually $L_0 \gg L_t$, the required computational cost of the CIP–CSL2 is similar to that in other schemes even for $\beta = 1$.

Furthermore, we are developing another formulation, which does not use the directional splitting technique, and more reduction of the computational cost is expected.

3.2.3. *Numerical Solution in Three Dimensions*

In order to evaluate the present three-dimensional solver, the problem of three-dimensional solid-body rotation is calculated. We use a 100 × 100 × 100 rectangular mesh system with uniform spacing $\Delta x = \Delta y = \Delta z = 1$, and the computational domain is $\{(x, y, z) \mid -50 < x < 50, -50 < y < 50, -50 < z < 50\}$. The velocity $\vec{v} = (u, v, w)$ is set as

$$\vec{v} = \vec{\omega} \times \vec{r}, \tag{92}$$

where $\vec{r} = (x, y, z)$ and $\vec{\omega} = 2\pi/\sqrt{2}(0, -1, 1)$. In order to satisfy CFL ≤ 0.2 in all grid points, the time step interval of $\Delta t = 4.547 \times 10^{-4}$ is employed. In Fig. 18a a schematic picture of this problem is shown. Initially, the center of the solid body is located at

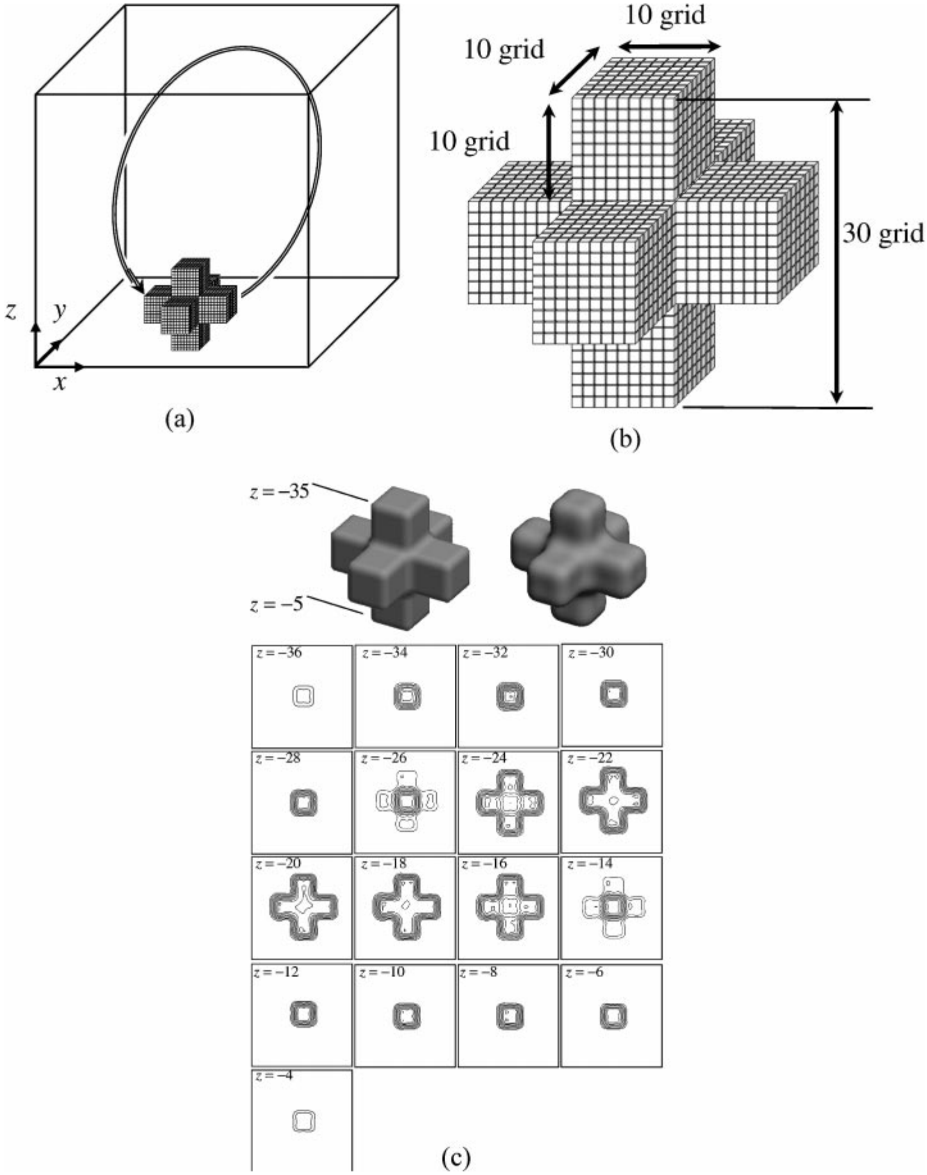

FIG. 18. (a) Schematic view of the three-dimensional solid-body, rotation problem. (b) Initial profile of the solid body. At the boundary and the inside, $f = 1.0$; elsewhere $f = 0.0$. (c) Computational result after one complete rotation. The pictures on the top row show the isosurface of $\rho = 0.5$ of the initial condition and the computational result. The line contours of the profile of ρ on the $z = k$ plane are shown below.

$(x, y, z) = (0, -20, -20)$, and the detailed profile of the initial state is drawn in Fig. 18b. At the boundary and inside the solid body, the value of f is set to 1.0, while $f = 0.0$ outside. The initial value of ρ is given by $\rho_{ijk} = \frac{1}{8}(f_{ijk} + f_{i+1jk} + f_{ij+1k} + f_{ijk+1} + f_{i+1j+1k} + f_{ij+1k+1} + f_{i+1jk+1} + f_{i+1j+1k+1})\Delta x \Delta y \Delta z$. Ideally, the profile after one complete evolution should be the same as the initial one. Figure 18c shows an isosurface of $\rho = 0.5$ for the numerical result and the profiles of ρ on the $z = k$ plane at $t = 1.0$ within $-25 \leq x \leq 25$ and $-45 \leq y \leq 5$. The line contours are drawn from 0.1 to 1.2 with increments of 0.1. The scheme restores the initial profile well.

3.3. Semi-Lagrangian Solution

In this section, we show a result of the semi-Lagrangian solutions. Although the CIP–CSL scheme is a semi-Lagrangian scheme as well as the CIP and the present two- and three-dimensional numerical procedures are written in the semi-Lagrangian formulation, too, we may lose the accuracy of the trajectory in exchange for the many merits of using the fractional step technique such as the simple numerical procedure and easy extension to higher dimensions. This is because the trajectory is estimated individually along each direction in fractional steps. However, it is valuable to examine the efficiency of the scheme with a large time interval.

For an example of a semi-Lagrangian solution, we apply the scheme to the idealized kinematic frontogenesis that has already been mentioned in the two-dimensional case. The same computational conditions except for time interval Δt are employed. In the case of CFL $\gg 1$, the trajectory to the upstream departure points has to be estimated accurately. For the computation of the trajectory, we can use well-known and highly precise schemes for ordinary differential equations such as the Runge–Kutta and the iterative method [14]. We employed the fourth-order Runge–Kutta scheme to solve Eq. (50) (for details see [26]). In order to examine the dependency on Δt, the computations are performed with CFL = 1.061, 2.121, 4.243. The numerical results of f at $t = 16$ with CFL $= 1.061, 2.121, 4.243$ are depicted in Figs. 19b–19d. The scores of the numerical results are given in Table IV.

In comparison with the theoretical solution presented in Fig. 19a, the numerical result with the large time interval CFL $= 4.243$ is slightly different from the theoretical solution and the accuracy is degraded, while the numerical results with CFL $= 1.061, 2.121$ agree well with the theoretical one.

Up to now, some excellent semi-Lagrangian schemes have been proposed, such as the piecewise parabolic method proposed by Woodward and co-workers [16] or the piecewise biparabolic scheme (PBM) proposed by Rančić [24]. Especially, the PBM scheme has been widely noticed recently as a small numerical diffusion method. The errors of the CIP–CSL2 for CFL $= 1.061, 2.121$ are almost equal or less than the results of the PBM scheme reported by Rančić. Actually, in the solution with CFL $= 1.061$, dispersion, dissipation, and total errors are about 7% smaller than for the results of the PBM scheme. On the other hand, however, the numerical score for CFL $= 4.243$ which is presented in Table IV is worse than the result of PBM scheme (see Table III in [24]). This degradation of accuracy for a large CFL number is reasonable because we adopt the fractional step technique to extend the scheme to multi-dimensions and the trajectory is estimated individually along each direction in fractional steps. However, if we take account of the simplicity of the procedure of the present scheme, we think that the error of the computational results still remains reasonably small.

TABLE IV

Result of Semi-Lagrangian Solutions of Doswell's Frontogenesis Problem with CIP–CSL2 Scheme at *t* = 16

CFL	RMS	Edisp	Ediss	Etotal	CPU
1.061	8.44×10^{-2}	7.32×10^{-3}	7.61×10^{-5}	7.40×10^{-3}	2895
2.121	1.09×10^{-1}	1.21×10^{-2}	9.29×10^{-5}	1.22×10^{-2}	1460
4.243	1.44×10^{-1}	2.10×10^{-2}	1.50×10^{-4}	2.11×10^{-2}	724

Notes. Line of CPU presents the total computation time measured in central processing unit time on a 333-MHz Pentium II personal computer.

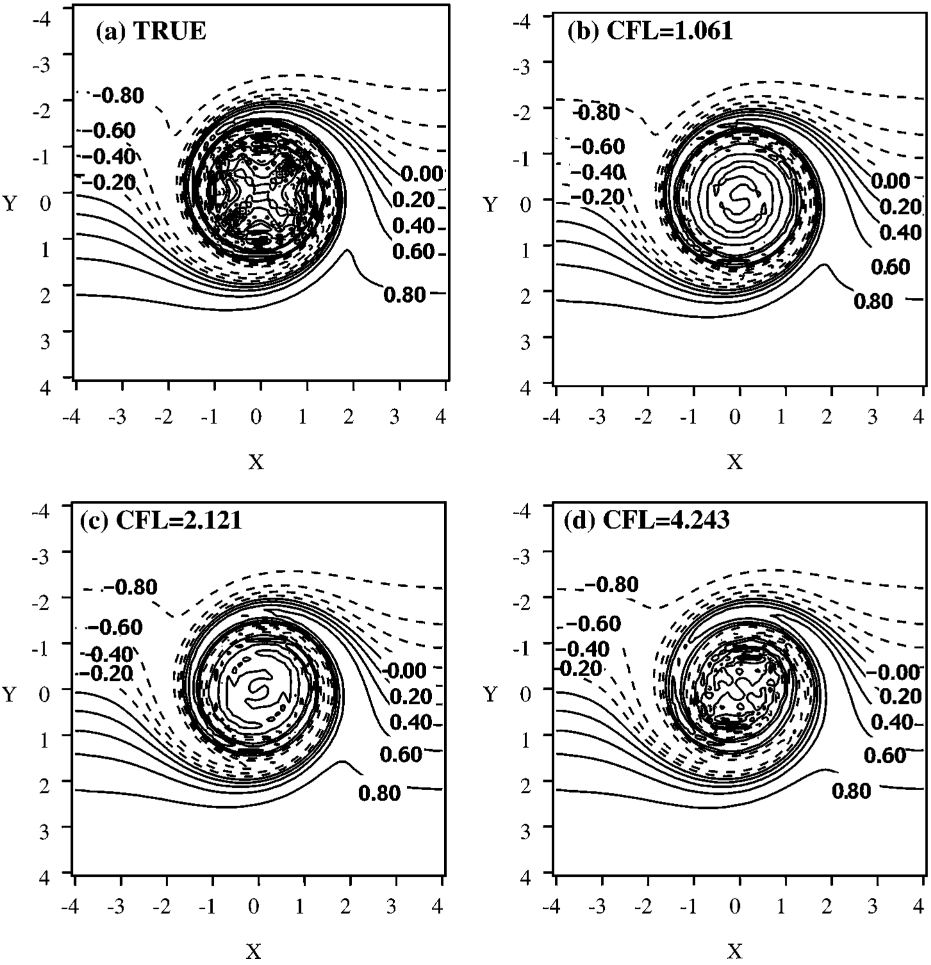

FIG. 19. Numerical results of the semi-Lagrangian solution of Doswell's frontogenesis experiment with the CIP–CSL2 scheme at $t = 16$. (a) The theoretical solution and the numerical results with (b) CFL = 1.061, (c) 2.121, and (d) 4.243 are represented.

4. SUMMARY

We have extended the CIP–CSL2 scheme to the multi-dimensional conservative equation. The line density or the surface density is introduced on each boundary of a computational cell. The cubic polynomial function is used to estimate the mass flow and the advection. The present scheme successfully solves the two- and three-dimensional problems with exact mass conservation and the numerical diffusion is quite small. Furthermore, the conservative, monotone preserving, and non-oscillatory scheme has been constructed by using the rational interpolation function instead of the cubic polynomial.

The present scheme has many excellent numerical features and provides highly accurate, low diffusive solutions with guaranteeing exact mass conservation. Although the scheme adopts the fractional step technique in the extension to multi-dimensions, it provides reasonably accurate results even for a semi-Lagrangian solution. When a much larger CFL number is employed, degradation of the accuracy is unavoidable because of the fractional step procedure. However, if we take account of the many merits of the fractional step technique, such as the simple numerical procedure and easy extension to higher dimensions, it is concluded that the solution by the present scheme is accurate enough even in the case of semi-Lagrangian solutions.

APPENDIX A

Here we summarize numerical procedures in the CIP–CSL2 method in two and three dimensions. In order to clarify the solution procedure in the multiple dimensions, we shall define the 1D algorithm of the present scheme by *CIPCSL1D* $(U, F, FN, R, RN, m, \lambda)$. The explicit expression of the 1D algorithm of the present scheme is given as follow:

CIPCSL1D$(U, F, FN, R, RN, m, \lambda)$

/∗ Departure point ∗/

$$\lambda_{p_m} = \lambda_m + \int_{t+\Delta t}^{t} U(\lambda, t)\, dt. \tag{A1}$$

/∗ Grid point k_m satisfying $\lambda_{k_m} \le \lambda_{p_m} < \lambda_{k_m+1}$ ∗/

$$\text{if}\big(\lambda_j \le \lambda_{p_m} < \lambda_{j+1}\big) \text{ then } k_m = j \tag{A2}$$

$$\langle \lambda_m \rangle = \lambda_{p_m} - \lambda_{k_m} \tag{A3}$$

$$\Delta\lambda_{k_m} = \lambda_{k_m+1} - \lambda_{k_m}. \tag{A4}$$

/∗ Interpolation between λ_{k_m} and λ_{k_m+1} ∗/

$$\phi_{k_m} = \frac{F_{k_m} + F_{k_m+1}}{\Delta\lambda_{k_m}^2} - \frac{2R_{k_m}}{\Delta\lambda_{k_m}^3} \tag{A5}$$

$$\eta_{k_m} = -\frac{2F_{k_m} + F_{k_m+1}}{\Delta\lambda_{k_m}} + \frac{3R_{k_m}}{\Delta\lambda_{k_m}^2}. \tag{A6}$$

/∗ Time development of R ∗/

$$D_m = \phi_{k_m} \langle \lambda_m \rangle^3 + \eta_{k_m} \langle \lambda_m \rangle^2 + F_m \langle \lambda_m \rangle \tag{A7}$$

$$RN_m = \sum_{j=k_m}^{k_{m+1}-1} R_j + (D_{m+1} - D_m). \tag{A8}$$

/∗ Time development of F ∗/

$$F_m^* = 3\phi_{k_m} \langle\lambda_m\rangle^2 + 2\eta_{k_m} \langle\lambda_m\rangle + F_m \tag{A9}$$

$$FN_m = \left(1 - \frac{\partial U(\lambda_m, t)}{\partial\lambda}\right) \times F_m^*. \tag{A10}$$

Two-Dimensional Solver

The two-dimensional solution is given only by applying the one-dimensional solution repeatedly. In the Step 1 solution, the sets of values $(f^n,\ \sigma_x^n)$ and $(\rho^n,\ \sigma_y^n)$ are advanced to $(f^{step1},\ \sigma_x^{step1})$ and $(\rho^{step1},\ \sigma_y^{step1})$ by applying the one-dimensional solution to each set. In Step 2, each set of values $(f^{step1},\ \sigma_y^{step1})$ and $(\rho^{step1},\ \sigma_x^{step1})$ is advanced:

$$\text{Step 1:}\quad \bar{u}_{ij} = (u_{ij} + u_{ij+1})/2 \tag{A11}$$

$$CIPCSL1D\big(\bar{u}, \sigma_y^n, \sigma_y^{step1}, \rho^n, \rho^{step1}, i, x\big) \tag{A12}$$

$$CIPCSL1D\big(u,\ f^n,\ f^{step1}, \sigma_x^n, \sigma_x^{step1}, i, x\big). \tag{A13}$$

$$\text{Step 2:}\quad \bar{v}_{ij} = (v_{ij} + v_{i+1j})/2 \tag{A14}$$

$$CIPCSL1D\big(\bar{v}, \sigma_x^{step1}, \sigma_x^{n+1}, \rho^{step1}, \rho^{n+1}, j, y\big) \tag{A15}$$

$$CIPCSL1D\big(v,\ f^{step1},\ f^{n+1}, \sigma_y^{step1}, \sigma_y^{n+1}, j, y\big). \tag{A16}$$

Three-Dimensional Solver

The three-dimensional solution is given only by applying the one-dimensional solution repeatedly as follows.

$$\text{Step 1:}\quad CIPCSL1D\big(u,\ f^n,\ f^{step1}, \sigma_x^n, \sigma_x^{step1}, i, x\big) \tag{A17}$$

$$u'_{ijk} = (u_{ijk} + u_{ij+1k})/2 \tag{A18}$$

$$CIPCSL1D\big(u', \sigma_y^n, \sigma_y^{step1}, S_{xy}^n, S_{xy}^{step1}, i, x\big) \tag{A19}$$

$$u''_{ijk} = (u_{ijk} + u_{ijk+1})/2 \tag{A20}$$

$$CIPCSL1D\big(u'', \sigma_z^n, \sigma_z^{step1}, S_{zx}^n, S_{zx}^{step1}, i, x\big) \tag{A21}$$

$$u'''_{ijk} = \big(u_{ijk} + u_{ijk+1} + u_{ij+1k} + u_{ij+1k+1}\big)/4 \tag{A22}$$

$$CIPCSL1D\big(u''', S_{yz}^n, S_{yz}^{step1}, \rho^n, \rho^{step1}, i, x\big). \tag{A23}$$

$$\tag{A24}$$

$$\text{Step 2:}\quad CIPCSL1D\big(v,\ f^{step1},\ f^{step2}, \sigma_y^{step1}, \sigma_y^{step2}, j, y\big) \tag{A25}$$

$$v'_{ijk} = (v_{ijk} + v_{i+1jk})/2 \tag{A26}$$

$$CIPCSL1D\big(v', \sigma_x^{step1}, \sigma_x^{step2}, S_{xy}^{step1}, S_{xy}^{step2}, j, y\big) \tag{A27}$$

$$v''_{ijk} = (v_{ijk} + v_{ijk+1})/2 \tag{A28}$$

$$CIPCSL1D\left(v'', \sigma_z^{step1}, \sigma_z^{step2}, S_{yz}^{step1}, S_{yz}^{step2}, j, y\right) \tag{A29}$$

$$v'''_{ijk} = (v_{ijk} + v_{ijk+1} + v_{i+1jk} + u_{i+1jk+1})/4 \tag{A30}$$

$$CIPCSL1D\left(v''', S_{zx}^{step1}, S_{zx}^{step2}, \rho^{step1}, \rho^{step2}, j, y\right). \tag{A31}$$

$$\text{Step 3:}\quad CIPCSL1D\left(w,\ f^{step2},\ f^{n+1}, \sigma_z^{step2}, \sigma_z^{n+1}, k, z\right) \tag{A32}$$

$$w'_{ijk} = (w_{ijk} + w_{i+1jk})/2 \tag{A33}$$

$$CIPCSL1D\left(w', \sigma_x^{step2}, \sigma_x^{n+1}, S_{zx}^{step2}, S_{zx}^{n+1}, k, z\right) \tag{A34}$$

$$w''_{ijk} = (w_{ijk} + w_{ij+1k})/2 \tag{A35}$$

$$CIPCSL1D\left(w'', \sigma_y^{step2}, \sigma_y^{n+1}, S_{yz}^{step2}, S_{yz}^{n+1}, k, z\right) \tag{A36}$$

$$w'''_{ijk} = (w_{ijk} + w_{ij+1k} + w_{i+1jk} + w_{i+1j+1k})/4 \tag{A37}$$

$$CIPCSL1D\left(w''', S_{xy}^{step2}, S_{xy}^{n+1}, \rho^{step2}, \rho^{n+1}, k, z\right). \tag{A38}$$

APPENDIX B

Here we summarize the coefficients of the two-dimensional or three-dimensional profile within a computational cell.

Two-Dimensional Profile

$$F_{ij}^n(x,y) = \sum_{l=0}^{2}\sum_{m=0}^{2} C_{lm}(x-x_i)^l(y-y_j)^m \quad \text{for } x_i \le x \le x_{i+1} \quad \text{and} \quad y_j \le y \le y_{j+1}.$$

$$C_{00} = f_{ij}^n$$

$$C_{10} = \frac{1}{\Delta x}\left[-\left(2 f_{ij}^n + f_{i+1j}^n\right) + \frac{3\sigma_{xij}^n}{\Delta x}\right]$$

$$C_{01} = \frac{1}{\Delta y}\left[-\left(2 f_{ij}^n + f_{ij+1}^n\right) + \frac{3\sigma_{yij}^n}{\Delta y}\right]$$

$$C_{20} = \frac{1}{\Delta x^2}\left[\left(f_{ij}^n + f_{i+1j}^n\right) - \frac{2\sigma_{xij}^n}{\Delta x}\right]$$

$$C_{02} = \frac{1}{\Delta y^2}\left[\left(f_{ij}^n + f_{ij+1}^n\right) - \frac{2\sigma_{yij}^n}{\Delta y}\right]$$

$$C_{11} = \frac{4}{\Delta x\,\Delta y}\left[\left(2C - f_{i+1j+1}^n + 2 f_{ij}^n\right) + 3\frac{2\sigma_{xij}^n + \sigma_{xij+1}^n}{\Delta x} + 3\frac{2\sigma_{yij}^n + \sigma_{yi+1j}^n}{\Delta y} + \frac{9\rho_{ij}^n}{\Delta x\,\Delta y}\right]$$

$$C_{21} = \frac{6}{\Delta x^2\Delta y}\left[-\left(C + f_{i+1j}^n + f_{ij}^n\right) + 2\frac{2\sigma_{xij}^n + \sigma_{xij+1}^n}{\Delta x} + 3\frac{\sigma_{yij}^n + \sigma_{yi+1j}^n}{\Delta y} - \frac{6\rho_{ij}^n}{\Delta x\,\Delta y}\right]$$

$$C_{12} = \frac{6}{\Delta x\,\Delta y^2}\left[-\left(C + f_{ij+1}^n + f_{ij}^n\right) + 3\frac{\sigma_{xij}^n + \sigma_{xij+1}^n}{\Delta x} + 2\frac{2\sigma_{yij}^n + \sigma_{yi+1j}^n}{\Delta y} - \frac{6\rho_{ij}^n}{\Delta x\,\Delta y}\right]$$

$$C_{22} = \frac{9}{\Delta x^2\Delta y^2}\left[C - 2\frac{\sigma_{xij}^n + \sigma_{xij+1}^n}{\Delta x} - 2\frac{\sigma_{yij}^n + \sigma_{yi+1j}^n}{\Delta y} + \frac{4\rho_{ij}^n}{\Delta x\,\Delta y}\right]$$

$$C = \left(f_{i+1j+1}^n + f_{i+1j}^n + f_{ij+1}^n + f_{ij}^n\right).$$

Three-Dimensional Profile

$$F^n_{ijk}(x, y, z) = \sum_{l=0}^{2}\sum_{m=0}^{2}\sum_{n=0}^{2} C_{lmn}(x - x_i)^l (y - y_j)^m (y - y_j)^n$$

$$\text{for } x_i \le x \le x_{i+1},\ y_j \le y \le y_{j+1} \quad \text{and} \quad z_k \le z \le z_{k+1}.$$

$$C_{000} = f^n_{ijk}$$

$$C_{100} = \frac{1}{\Delta x}\left[-\left(2 f^n_{ijk} + f^n_{i+1jk}\right) + \frac{3\sigma^n_{xijk}}{\Delta x}\right]$$

$$C_{010} = \frac{1}{\Delta y}\left[-\left(2 f^n_{ijk} + f^n_{ij+1k}\right) + \frac{3\sigma^n_{yijk}}{\Delta y}\right]$$

$$C_{001} = \frac{1}{\Delta z}\left[-\left(2 f^n_{ijk} + f^n_{ijk+1}\right) + \frac{3\sigma^n_{zijk}}{\Delta z}\right]$$

$$C_{200} = \frac{1}{\Delta x^2}\left[\left(f^n_{ijk} + f^n_{i+1jk}\right) - \frac{2\sigma^n_{xijk}}{\Delta x}\right]$$

$$C_{020} = \frac{1}{\Delta y^2}\left[\left(f^n_{ijk} + f^n_{ij+1k}\right) - \frac{2\sigma^n_{yijk}}{\Delta y}\right]$$

$$C_{002} = \frac{1}{\Delta z^2}\left[\left(f^n_{ijk} + f^n_{ijk+1}\right) - \frac{2\sigma^n_{zijk}}{\Delta z}\right]$$

$$C_{110} = \frac{4}{\Delta x\,\Delta y}\left[\left(2C_{xy} - f^n_{i+1j+1k} + 2 f^n_{ijk}\right) + 3\frac{2\sigma^n_{xijk} + \sigma^n_{xij+1k}}{\Delta x}\right.$$

$$\left. + 3\frac{2\sigma^n_{yijk} + \sigma^n_{yi+1jk}}{\Delta y} + \frac{9\rho^n_{ijk}}{\Delta x\,\Delta y}\right]$$

$$C_{101} = \frac{4}{\Delta x\,\Delta z}\left[\left(2C_{zx} - f^n_{i+1jk+1} + 2 f^n_{ijk}\right) + 3\frac{2\sigma^n_{xijk} + \sigma^n_{xijk+1}}{\Delta x}\right.$$

$$\left. + 3\frac{2\sigma^n_{zijk} + \sigma^n_{zi+1jk}}{\Delta z} + \frac{9\rho^n_{ijk}}{\Delta x\,\Delta z}\right]$$

$$C_{011} = \frac{4}{\Delta z\,\Delta y}\left[\left(2C_{yz} - f^n_{ij+1k+1} + 2 f^n_{ijk}\right) + 3\frac{2\sigma^n_{zijk} + \sigma^n_{zij+1k}}{\Delta z}\right.$$

$$\left. + 3\frac{2\sigma^n_{yijk} + \sigma^n_{yijk+1}}{\Delta y} + \frac{9\rho^n_{ijk}}{\Delta z\,\Delta y}\right]$$

$$C_{210} = \frac{6}{\Delta x^2\,\Delta y}\left[-\left(C_{xy} + f^n_{i+1jk} + f^n_{ijk}\right) + 2\frac{2\sigma^n_{xijk} + \sigma^n_{xij+1k}}{\Delta x}\right.$$

$$\left. + 3\frac{\sigma^n_{yijk} + \sigma^n_{yi+1jk}}{\Delta y} - \frac{6\rho^n_{ijk}}{\Delta x\,\Delta y}\right]$$

$$C_{120} = \frac{6}{\Delta x\,\Delta y^2}\left[-\left(C_{xy} + f^n_{ij+1k} + f^n_{ijk}\right) + 3\frac{\sigma^n_{xijk} + \sigma^n_{xij+1k}}{\Delta x}\right.$$

$$\left. + 2\frac{2\sigma^n_{yijk} + \sigma^n_{yi+1jk}}{\Delta y} - \frac{6\rho^n_{ijk}}{\Delta x\,\Delta y}\right]$$

$$\begin{aligned} C_{201} = {} & \frac{6}{\Delta x^2 \Delta z}\left[-\left(C_{zx} + f^n_{i+1jk} + f^n_{ijk}\right) + 2\frac{2\sigma^n_{xijk} + \sigma^n_{xijk+1}}{\Delta x}\right. \\ & \left. + 3\frac{\sigma^n_{zijk} + \sigma^n_{zi+1jk}}{\Delta z} - \frac{6\rho^n_{ijk}}{\Delta x \Delta z}\right] \end{aligned}$$

$$\begin{aligned} C_{102} = {} & \frac{6}{\Delta x \Delta z^2}\left[-\left(C_{zx} + f^n_{ijk+1} + f^n_{ijk}\right) + 3\frac{\sigma^n_{xijk} + \sigma^n_{xijk+1}}{\Delta x}\right. \\ & \left. + 2\frac{2\sigma^n_{zijk} + \sigma^n_{zi+1jk}}{\Delta z} - \frac{6\rho^n_{ijk}}{\Delta x \Delta z}\right] \end{aligned}$$

$$\begin{aligned} C_{012} = {} & \frac{6}{\Delta z^2 \Delta y}\left[-\left(C_{yz} + f^n_{ijk+1} + f^n_{ijk}\right) + 2\frac{2\sigma^n_{zijk} + \sigma^n_{zij+1k}}{\Delta z}\right. \\ & \left. + 3\frac{\sigma^n_{yijk} + \sigma^n_{yijk+1}}{\Delta y} - \frac{6\rho^n_{ijk}}{\Delta z \Delta y}\right] \end{aligned}$$

$$\begin{aligned} C_{021} = {} & \frac{6}{\Delta z \Delta y^2}\left[-\left(C_{yz} + f^n_{ij+1k} + f^n_{ijk}\right) + 3\frac{\sigma^n_{zijk} + \sigma^n_{zij+1k}}{\Delta z}\right. \\ & \left. + 2\frac{2\sigma^n_{yijk} + \sigma^n_{yijk+1}}{\Delta y} - \frac{6\rho^n_{ijk}}{\Delta z \Delta y}\right] \end{aligned}$$

$$\begin{aligned} C_{111} = {} & \frac{8}{\Delta x \Delta y \Delta z}\left[-\left(2C_{xyz} + 6f^n_{ijk} + 2f^n_{i+1jk} + 2f^n_{ij+1k} + 2f^n_{ijk+1} - f^n_{i+1j+1k+1}\right)\right. \\ & + 3\frac{2C_{\sigma_x} + 2\sigma^n_{xijk} - \sigma^n_{xij+1k+1}}{\Delta x} + 3\frac{2C_{\sigma_y} + 2\sigma^n_{yijk} - \sigma^n_{yi+1jk+1}}{\Delta y} \\ & + 3\frac{2C_{\sigma_z} + 2\sigma^n_{zijk} - \sigma^n_{zi+1j+1k}}{\Delta z} + 9\frac{2S^n_{xyijk} + S^n_{xyijk+1}}{\Delta x \Delta y} + 9\frac{2S^n_{yzijk} + S^n_{yzi+1jk}}{\Delta y \Delta z} \\ & \left. + 9\frac{2S^n_{zxijk} + S^n_{zxij+1k}}{\Delta x \Delta z} + \frac{27\rho^n_{ijk}}{\Delta x \Delta y \Delta z}\right] \end{aligned}$$

$$C_{220} = \frac{9}{\Delta x^2 \Delta y^2}\left[C_{xy} - 2\frac{\sigma^n_{xijk} + \sigma^n_{xij+1k}}{\Delta x} - 2\frac{\sigma^n_{yijk} + \sigma^n_{yi+1jk}}{\Delta y} + \frac{4\rho^n_{ijk}}{\Delta x \Delta y}\right]$$

$$C_{202} = \frac{9}{\Delta x^2 \Delta z^2}\left[C_{zx} - 2\frac{\sigma^n_{xijk} + \sigma^n_{xijk+1}}{\Delta x} - 2\frac{\sigma^n_{zijk} + \sigma^n_{zi+1jk}}{\Delta z} + \frac{4\rho^n_{ijk}}{\Delta x \Delta z}\right]$$

$$C_{022} = \frac{9}{\Delta z^2 \Delta y^2}\left[C_{yz} - 2\frac{\sigma^n_{zijk} + \sigma^n_{zij+1k}}{\Delta z} - 2\frac{\sigma^n_{yijk} + \sigma^n_{yijk+1}}{\Delta y} + \frac{4\rho^n_{ijk}}{\Delta z \Delta y}\right]$$

$$\begin{aligned} C_{112} = {} & \frac{12}{\Delta x \Delta y \Delta z^2}\left[\left(2C_{xyz} - f^n_{i+1j+1k} + 2f^n_{ijk} - f^n_{i+1j+1k+1} + 2f^n_{ijk+1}\right)\right. \\ & + 3\frac{C_{\sigma_x} + \sigma^n_{xijk} + \sigma^n_{xijk+1}}{\Delta x} + 3\frac{C_{\sigma_y} + \sigma^n_{yijk} + \sigma^n_{yijk+1}}{\Delta y} - 2\frac{2C_{\sigma_z} + 2\sigma^n_{zijk} - \sigma^n_{zi+1j+1k}}{\Delta z} \\ & \left. + 9\frac{C_{S_{xy}}}{\Delta x \Delta y} - 6\frac{2S^n_{yzijk} + S^n_{yzi+1jk}}{\Delta y \Delta z} - 6\frac{2S^n_{zxijk} + S^n_{zxij+1k}}{\Delta x \Delta z} - \frac{18\rho^n_{ijk}}{\Delta x \Delta y \Delta z}\right] \end{aligned}$$

$$
\begin{aligned}
C_{211} = \frac{12}{\Delta x^2 \Delta y \Delta z} \Bigg[& \left(2C_{xyz} - f^n_{ij+1k+1} + 2 f^n_{ijk} - f^n_{i+1j+1k+1} + 2 f^n_{i+1jk}\right) \\
& + 3\frac{C_{\sigma_z} + \sigma^n_{zijk} + \sigma^n_{zi+1jk}}{\Delta z} + 3\frac{C_{\sigma_y} + \sigma^n_{yijk} + \sigma^n_{yi+1jk}}{\Delta y} - 2\frac{2C_{\sigma_x} + 2\sigma^n_{xijk} - \sigma^n_{xij+1k+1}}{\Delta x} \\
& + \frac{9C_{S_{yz}}}{\Delta y \Delta z} - 6\frac{2S^n_{xyijk} + S^n_{xyijk+1}}{\Delta x \Delta y} - 6\frac{2S^n_{zxijk} + S^n_{zxij+1k}}{\Delta x \Delta z} - \frac{18\rho^n_{ijk}}{\Delta x \Delta y \Delta z} \Bigg] \\
C_{121} = \frac{12}{\Delta x \Delta y^2 \Delta z} \Bigg[& \left(2C_{xyz} - f^n_{i+1jk+1} + 2 f^n_{ijk} - f^n_{i+1j+1k+1} + 2 f^n_{ij+1k}\right) \\
& + 3\frac{C_{\sigma_x} + \sigma^n_{xijk} + \sigma^n_{xij+1k}}{\Delta x} + 3\frac{C_{\sigma_z} + \sigma^n_{zijk} + \sigma^n_{zij+1k}}{\Delta z} - 2\frac{2C_{\sigma_y} + 2\sigma^n_{yijk} - \sigma^n_{yi+1jk+1}}{\Delta y} \\
& + \frac{9C_{S_{zx}}}{\Delta x \Delta z} - 6\frac{2S^n_{yzijk} + S^n_{yzi+1jk}}{\Delta y \Delta z} - 6\frac{2S^n_{xyijk} + S^n_{xyijk+1}}{\Delta x \Delta y} - \frac{18\rho^n_{ijk}}{\Delta x \Delta y \Delta z} \Bigg] \\
C_{221} = \frac{18}{\Delta x^2 \Delta y^2 \Delta z} \Bigg[& -\left(2C_{xyz} - f^n_{i+1j+1k+1} - f^n_{i+1jk+1} - f^n_{ij+1k+1} - f^n_{ijk+1}\right) \\
& + 2\frac{2C_{\sigma_x} - \sigma^n_{xijk+1} - \sigma^n_{xij+1k+1}}{\Delta x} + 2\frac{2C_{\sigma_y} - \sigma^n_{yijk+1} - \sigma^n_{yi+1jk+1}}{\Delta y} + \frac{3C_{\sigma_z}}{\Delta z} \\
& - 4\frac{2S^n_{xyijk} + S^n_{xyijk+1}}{\Delta x \Delta y} - \frac{6C_{S_{yz}}}{\Delta y \Delta z} - \frac{6C_{S_{zx}}}{\Delta x \Delta z} + \frac{12\rho^n_{ijk}}{\Delta x \Delta y \Delta z} \Bigg] \\
C_{122} = \frac{18}{\Delta x \Delta y^2 \Delta z^2} \Bigg[& -\left(2C_{xyz} - f^n_{i+1j+1k+1} - f^n_{i+1jk+1} - f^n_{i+1j+1k} - f^n_{i+1jk}\right) \\
& + 2\frac{2C_{\sigma_z} - \sigma^n_{zi+1jk} - \sigma^n_{zi+1j+1k}}{\Delta z} + 2\frac{2C_{\sigma_y} - \sigma^n_{yi+1jk} - \sigma^n_{yi+1jk+1}}{\Delta y} + \frac{3C_{\sigma_x}}{\Delta x} \\
& - 4\frac{2S^n_{yzijk} + S^n_{yzi+1jk}}{\Delta y \Delta z} - \frac{6C_{S_{xy}}}{\Delta x \Delta y} - \frac{6C_{S_{zx}}}{\Delta x \Delta z} + \frac{12\rho^n_{ijk}}{\Delta x \Delta y \Delta z} \Bigg] \\
C_{212} = \frac{18}{\Delta x^2 \Delta y \Delta z^2} \Bigg[& -\left(2C_{xyz} - f^n_{i+1j+1k+1} - f^n_{i+1j+1k} - f^n_{ij+1k+1} - f^n_{ij+1k}\right) \\
& + 2\frac{2C_{\sigma_x} - \sigma^n_{xij+1k} - \sigma^n_{xij+1k+1}}{\Delta x} + 2\frac{2C_{\sigma_z} - \sigma^n_{zij+1k} - \sigma^n_{zi+1j+1k}}{\Delta z} + \frac{3C_{\sigma_y}}{\Delta y} \\
& - 4\frac{2S^n_{zxijk} + S^n_{zxij+1k}}{\Delta x \Delta z} - \frac{6C_{S_{yz}}}{\Delta y \Delta z} - \frac{6C_{S_{xy}}}{\Delta x \Delta y} + \frac{12\rho^n_{ijk}}{\Delta x \Delta y \Delta z} \Bigg] \\
C_{222} = \frac{27}{\Delta x^2 \Delta y^2 \Delta z^2} \Bigg[& C_{xyz} - 2\frac{C_{\sigma_x}}{\Delta x} - 2\frac{C_{\sigma_y}}{\Delta y} - 2\frac{C_{\sigma_z}}{\Delta z} + \frac{4C_{S_{xy}}}{\Delta x \Delta y} + \frac{4C_{S_{yz}}}{\Delta y \Delta z} \\
& + \frac{4C_{S_{zx}}}{\Delta x \Delta z} + \frac{8\rho^n_{ijk}}{\Delta x \Delta y \Delta z} \Bigg] \\
C_{S_{xy}} = & \left(S^n_{xyijk+1} + S^n_{xyijk}\right) \\
C_{S_{yz}} = & \left(S^n_{yzi+1jk} + S^n_{yzijk}\right) \\
C_{S_{zx}} = & \left(S^n_{zxij+1k} + S^n_{zxijk}\right) \\
C_{\sigma_x} = & \left(\sigma^n_{xij+1k+1} + \sigma^n_{xij+1k} + \sigma^n_{xijk+1} + \sigma^n_{xijk}\right)
\end{aligned}
$$

$$
\begin{aligned}
C_{\sigma_y} &= \left(\sigma^n_{yi+1jk+1} + \sigma^n_{yi+1jk} + \sigma^n_{yijk+1} + \sigma^n_{yijk}\right)\\
C_{\sigma_z} &= \left(\sigma^n_{zi+1j+1k} + \sigma^n_{zij+1k} + \sigma^n_{zi+1jk} + \sigma^n_{zijk}\right)\\
C_{xy} &= \left(f^n_{i+1j+1k} + f^n_{i+1jk} + f^n_{ij+1k} + f^n_{ijk}\right)\\
C_{yz} &= \left(f^n_{ij+1k+1} + f^n_{ijk+1} + f^n_{ij+1k} + f^n_{ijk}\right)\\
C_{zx} &= \left(f^n_{i+1jk+1} + f^n_{i+1jk} + f^n_{ijk+1} + f^n_{ijk}\right)\\
C_{xyz} &= \left(f^n_{i+1j+1k+1} + f^n_{i+1j+1k} + f^n_{ij+1k+1} + f^n_{i+1jk+1} + f^n_{i+1jk} + f^n_{ij+1k} + f^n_{ijk+1} + f^n_{ijk}\right).
\end{aligned}
$$

ACKNOWLEDGMENTS

We thank Dr. F. Xiao for valuable discussions and the three anonymous referees for their constructive suggestions and kind advice, which improved the original manuscript. We also thank the editors for their kindness.

REFERENCES

1. H. Takewaki, A. Nishiguchi, and T. Yabe, Cubic interpolated pseudoparticle method (CIP) for solving hyperbolic type equations, *J. Comput. Phys.* **61**, 261 (1985).
2. H. Takewaki and T. Yabe, The cubic-interpolated pseudo particle (CIP) method: Application to nonlinear and multi-dimensional hyperbolic equations, *J. Comput. Phys.* **70**, 355 (1987).
3. 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).
4. 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 solver, *Comput. Phys. Commun.* **66**, 233 (1991).
5. The special issue of the CIP method, *CFD J.* **8** (1999).
6. T. Utsumi, T. Kunugi, and T. Aoki, Stability and accuracy of the cubic interpolated propagation scheme, *Comput. Phys. Commun.* **101**, 9 (1997).
7. T. Yabe, Unified solver CIP for solid, liquid and gas, in *Computational Fluid Dynamics Review 1998*, edited by M. M. Hafez and K. Oshima (World Scientific, Singapore, 1998), p. 1011.
8. T. Yabe, Y. Zhang, and F. Xiao, A numerical procedure—CIP—to solve all phases of matter together, in *Sixteenth International Conference on Numerical Methods in Fluid Dynamics,* Lecture Notes in Physics (Springer-Verlag, Berlin/New York, 1998), p. 439.
9. T. Yabe, F. Xiao, and T. Utsumi, The constrained interpolation profile method for multiphase analysis, *J. Comput. Phys.* **169**, 556 (2001), doi:10.1006/jcph. 2000.6625.
10. A. Robert, A semi-implicit and semi-Lagrangian numerical integration scheme for the primitive meteorological equations, *J. Meteorol. Soc. Jpn.* **60**, 319 (1982).
11. A. Staniforth and J. Côté, Semi-Lagrangian integration scheme for atmospheric model—A review, *Mon. Weather Rev.* **119**, 2206 (1991).
12. D. L. Williamson and P. J. Rasch, Two-dimensional semi-Lagrangian transport with shape-preserving interpolation, *Mon. Weather Rev.* **117**, 102 (1989).
13. R. Bermejo and A. Staniforth, The conversion of semi-Lagrangian advection schemes to quasi-monotone schemes, *Mon. Weather Rev.* **120**, 2622 (1992).
14. R. Laprise and A. Plante, SLIC: A semi-Lagrangian integrated-by-cell mass-conserving numerical transport scheme, *Mon. Weather Rev.* **123**, No. 2, 553 (1995).
15. S. Lin and R. B. Rood, Multidimensional flux-form semi-Lagrangian transportation schemes, *Mon. Weather Rev.* **124**, 2046 (1996).
16. R. L. Jr. Carpenter, K. K. Droegemeier, P. R. Woodward, and C. E. Hane, Application of the piecewise parabolic method (PPM) to meteorological modeling, *Mon. Wheather Rev.* **118**, 586 (1990).

17. T. Nakamura and T. Yabe, Cubic interpolation scheme for solving the hyperdimensional Vlasov–Poisson equation in phase space. *Comput. Phys. Commun.* **120**, 122 (1999).

18. R. Tanaka, T. Nakamura, and T. Yabe, Constructing exactly conservative scheme in non-conservative form, *Comput. Phys. Commun.* **126**, 232 (2000).

19. T. Yabe, R. Tanaka, T. Nakamura, and F. Xiao, An exactly conservative semi-Lagrangian scheme (CIP-CSL) in one dimension, *Mon. Weather Rev.* **129**, 332 (2001).

20. T. Nakamura, R. Tanaka, and T. Yabe, Multi-dimensional conservative scheme in non-conservative form, *CFD J.* **9**, 437 (2001).

21. F. Xiao, T. Yabe, and T. Ito, Oscillation preventing scheme for advection equation by rational function, *Comput. Phys. Commun.* **93**, 1 (1999).

22. P. Y. Wang, T. Yabe, and T. Aoki, A general hyperbolic solver—the CIP method—applied to curvilinear coordinate, *J. Phys. Soc. Jpn.* **62**, 1865 (1993).

23. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithm for fluids, *J. Comput. Phys.* **31**, 335 (1979).

24. M. Rančić, Semi-Lagrangian piecewise biparabolic scheme for two-dimensional horizontal advection of a passive scalar, *Mon. Weather Rev.* **120**, 1394 (1991).

25. C. A. Doswell II, A kinematic analysis of frontgenesis associated with a nondivergent vortex, *J. Atmos. Sci.* **41**, 1242 (1984).

26. K. Sakurai, T. Aoki, and T. Yabe, Semi-Lagrangian cubic-interpolated propagation (CIP) scheme for long parcel trajectories, *CFD J.* **10**, 76 (2001).