Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 194 (2004) 57–77
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Higher-order schemes with CIP method and adaptive Soroban grid towards mesh-free scheme

Takashi Yabe [a,*], Hiroki Mizoe [a,b], Kenji Takizawa [a], Hiroshi Moriki [a], Hyo-Nam Im [a], Youichi Ogata [a]

[a] *Department of Mechanical Engineering and Science, Tokyo Institute of Technology, O-okayama, Meguro-ku, Tokyo 152-8552, Japan*
[b] *Tohoku Electric Power Company, Inc., 7-2-1 Nakayama, Aoba-ku, Sendai 981-0952, Japan*

Received 8 April 2003; received in revised form 16 July 2003; accepted 28 August 2003

Abstract

A new class of body-fitted grid system that can keep the third-order accuracy in time and space is proposed with the help of the CIP (constrained interpolation profile/cubic interpolated propagation) method. The grid system consists of the straight lines and grid points moving along these lines like abacus – Soroban in Japanese. The length of each line and the number of grid points in each line can be different. The CIP scheme is suitable to this mesh system and the calculation of large CFL (>10) at locally refined mesh is easily performed. Mesh generation and searching of upstream departure point are very simple and almost mesh-free treatment is possible. Adaptive grid movement and local mesh refinement are demonstrated.

Keywords: Third-order scheme; Numerical algorithm; Curvilinear coordinate; Mesh free; Semi-Lagrangian schemes

1. Introduction

The use of curvilinear coordinate system is important to represent the detailed flow structure-like boundary layer around complex bodies [1]. In constructing such coordinate, however, little attention to the accuracy has been made. This is the key issue because the degradation of accuracy by introducing curvilinear coordinate may cancel the advantage of curvilinear coordinate.

Besides this, there exist several interface capturing schemes at present. These are VOF (volume of fluid) [2,3], level set [4], particle tracking [5] and the CIP (constrained interpolation profile/cubic interpolated propagation) method [6–12]. These methods can accurately described severely distorted interfaces even with fixed Cartesian grid. The merit of these schemes are that they can treat the deformation of the surface and even break up, but the description of the boundary layer still needs locally refined mesh.

* Corresponding author. Tel.: +81-3-57342165; fax: +81-3-573422165.
E-mail address: yabe@mech.titech.ac.jp (T. Yabe).

0021-9991/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.jcp.2003.08.019

Although we applied the CIP scheme to curvilinear system [13], we did not examine the accuracy. In this paper, we shall propose a specific mesh system that can preserve the third-order accuracy in time and space even in non-orthogonal coordinate system. For illustrative purpose, one-dimensional system is examined in Section 2. In Section 3, a new mesh system is proposed to preserve the accuracy. Section 4 is devoted to the discussion and demonstration of moving adaptive grid and almost mesh-free computation.

2. One-dimensional system

It would be easier to understand the intrinsic problem caused by curvilinear system in one-dimensional case. Let us consider the linear wave propagation in non-uniform mesh system with an advection equation,

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

There exist many higher-order algorithms to solve this equation. The primary concern of this paper is to examine the accuracy of the scheme in non-uniform grid. If we use the non-uniform grid and the grid points are numbered in uniform ξ space as $x(\xi)$, then Eq. (1) can be put into a form

$$\frac{\partial f}{\partial t} + u\left(\frac{\partial x}{\partial \xi}\right)^{-1}\frac{\partial f}{\partial \xi} = 0. \tag{2}$$

Although each term like $\partial x/\partial\xi$ and $\partial f/\partial\xi$ can be put into higher-order finite difference form, it will be difficult to generate meshes keeping the higher-order accuracy for $\partial x/\partial\xi$. For example, the CIP with $\partial x/\partial\xi$ described by the fourth-order scheme becomes second-order if the ratio of neighboring mesh sizes becomes more than 1.01 as shown later. This is because the term "fourth-order" is merely in the sense of Taylor expansion while the higher-order errors are no more small for abrupt mesh change and become dominant for the mesh system whose spacing changes stepwise.

In one dimension, it may be possible to construct the higher-order scheme by directly applying it to Eq. (1) in non-uniform grid. However, it would be a hard task to extend it to multi-dimensionally deformed grid system. In addition, we must introduce the time difference. Even in uniform mesh system, the accuracy is reduced to the first-order when only the forward finite difference is used in time step even if the third-order accurate spatial scheme is adopted. Therefore, both the time and space must be designed to be the same order. It is also a subject of major concern to find an algorithm that can cancel the error caused by time step difference in curvilinear system.

In this sense, the CIP scheme is one of the interesting schemes for curvilinear coordinate as well as for the Cartesian coordinate because it has a compact support and the third-order accuracy both in time and space as shown in the following section.

2.1. Advection with constant velocity in uniform mesh

Let us review the CIP method briefly. Although the nature operates in a continuous world, a discretization process is unavoidable in order to implement numerical simulations. The primary goal of any numerical algorithm will be to retrieve the lost information inside the grid cell between these digitized points. The CIP method 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 Eq. (1).

When the velocity is constant, the solution of Eq. (1) gives a simple translational motion of wave with velocity u. The initial profile (solid line of Fig. 1(a)) moves like a dashed line in a continuous representation.

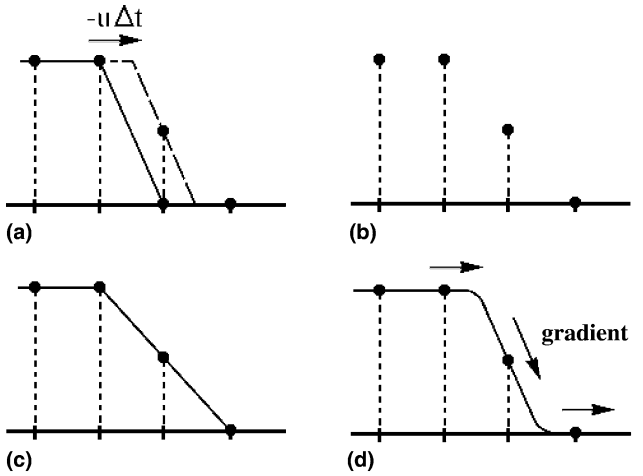

Fig. 1. The principle of the CIP method: (a) solid line is initial profile and dashed line is an exact solution after advection, whose solution (b) at discretized points. (c) When (b) is linearly interpolated, numerical diffusion appears. (d) In the CIP, spatial derivative also propagates and the profile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

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. 1(b), 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 solid line in (c). Thus, numerical diffusion arises when we construct the profile by the linear interpolation even with the exact solution as shown in Fig. 1(c). This process is called the first-order upwind scheme. On the other hand, if we use quadratic polynomial for interpolation, it suffers from overshooting. This process is the Lax–Wendroff scheme or Leith scheme [14].

What made this solution worse? This is why we neglect the behavior of the solution inside grid cell and merely follow after the smoothness of the solution. From this consideration, we understand that a method incorporating the real solution into the profile within a grid cell is an important subject. We proposed to approximate the profile as shown below. Let us differentiate Eq. (1) with spatial variable x, then we get

$$\frac{\partial g}{\partial t} + u\frac{\partial g}{\partial x} = -\frac{\partial u}{\partial x}g, \tag{3}$$

where g stands for the spatial derivative of f, $\partial f/\partial x$. In the simplest case where the velocity u is constant, Eq. (3) coincides with Eq. (1) and represents the propagation of 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 is predicted after propagation as shown by the arrows in Fig. 1(d), the profile after one step is limited to a specific profile. It is easy to imagine that by this constraint, the solution becomes very closer to the initial profile that is real solution. Most importantly, the solution thus created gives a profile consistent with Eq. (1) even inside the grid cell and thus a wave having wavelength of two-grid sizes can be accurately treated [12].

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

$$a_i = \frac{g_i + g_{iup}}{D^2} + \frac{2(f_i - f_{iup})}{D^3},$$

$$b_i = \frac{3(f_{iup} - f_i)}{D^2} - \frac{2g_i + g_{iup}}{D}, \tag{4}$$

$$f_i^{n+1} = a_i X^3 + b_i X^2 + g_i^n X + f_i^n,$$

$$g_i^{n+1} = 3a_i X^2 + 2b_i X + g_i^n, \tag{5}$$

where we define $X = -u\Delta t$. Here, $D = -\Delta x$, $iup = i - 1$ for $u \geqslant 0$ and $D = \Delta x, iup = i + 1$ for $u < 0$.

For constant velocity, the scheme becomes exactly conservative. Even in variable velocity field, we can obtain the exactly conservative scheme [15–17] and the same procedure given in this paper can also be used in such class of schemes. Interestingly, the CIP is proven to be the third-order in time and space easily by Taylor expanding Eqs. (4) and (5) [18]. Numerically, this can be proven by the following test problem with a propagation of sine wave.

The initial profile is

$$f(x, t = 0) = 2.0 + \sin(2\pi x),$$

$$\partial_x f(x, t = 0) = 2\pi \cos(2\pi x) \quad (0 \leqslant x \leqslant 1) \tag{6}$$

and the system is periodic in space.

Changing the mesh spacing, the error is estimated by the following root mean square:

$$\epsilon \equiv \frac{\sqrt{\sum_i^{NX} (f_{\mathrm{Num}} - f_{\mathrm{exact}})^2}}{\sum_i^{NX} f_{\mathrm{exact}}}, \tag{7}$$

where f_{Num} is the numerical result and f_{exact} is the exact solution, where NX is the number of meshes. The time step Δt is defined so that $\mathrm{CFL} \equiv u\Delta t/\Delta x$ is kept to be 0.2 and $u = 1$. Fig. 2 shows the numerical error at $t = 4.0$.

For comparison, we depict the results in Fig. 2 with the first-order upwind and the Lax–Wendroff scheme, which are the first- and the second-order, respectively, in time as well as in space. The numerical test shown in Fig. 2 clearly shows this dependence.

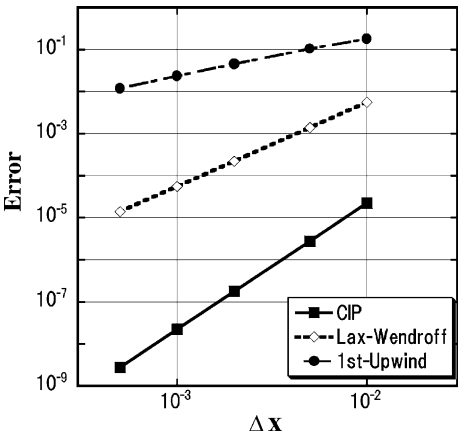

Fig. 2. The numerical error of the CIP, Lax–Wendroff and first-order upwind scheme at $t = 4.0$.

It should be noticed that the scheme of the first-order in time becomes the first-order even if the third-order scheme is used in space because the advection proceeds equally in time and space and CFL $= u\Delta t/\Delta x$ is fixed.

2.2. Advection with variable velocity

Next we apply the method to a conservation law as

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = -u\frac{\partial u}{\partial x} \equiv H. \tag{8}$$

We also need an equation for the derivative

$$\frac{\partial g}{\partial t} + u\frac{\partial g}{\partial x} = H' - g\frac{\partial u}{\partial x}. \tag{9}$$

Thus we separate all the equations into advection phase

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0,$$

$$\frac{\partial g}{\partial t} + u\frac{\partial g}{\partial x} = 0, \tag{10}$$

and non-advection phase:

$$\frac{\partial f}{\partial t} = -u\frac{\partial u}{\partial x},$$

$$\frac{\partial g}{\partial t} = -2g\frac{\partial u}{\partial x} - f\frac{\partial^2 u}{\partial x^2}. \tag{11}$$

The advection phase is calculated by Eqs. (4) and (5), but we must remind that the advection velocity changes during the propagation. As for the advection velocity, we have several choices but we discuss only three cases. (1) u_i at the grid point. (2) u^* at the upstream departure point. (3) These average $u_{\text{ave}} \equiv (u_i + u^*)/2$.

As for non-advection phase, we used a finite difference in previous papers. For the application towards the mesh-less form as will be given later in this paper, we choose to use f and g directly. If u is given, these equations can be solved in closed form for f and g without finite difference. Thus, Eq. (11) is put into a numerical solution as

$$f_i^{n+1} = f_i^* \exp[-\partial u/\partial x \Delta t],$$

$$g_i^{n+1} = g_i^* \exp[-\partial u/\partial x \Delta t] - f^* \partial^2 u/\partial x^2 \Delta t, \tag{12}$$

where f^*, g^* are the results after advection phase.

This scheme is tested by the following analytical solution. If the advection velocity is

$$u(x) = \frac{1}{1+ax}, \tag{13}$$

then the analytical solution is easily obtained as

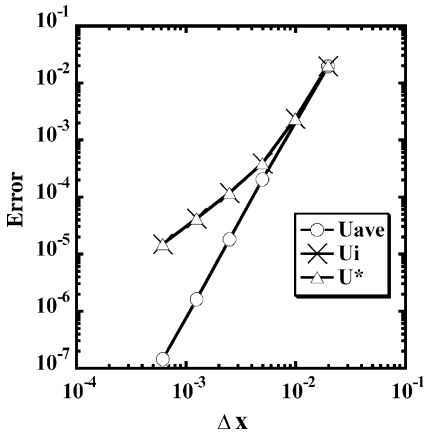

Fig. 3. Error in conservative equation. Three advection velocities are tested.

$$f(x_i,t)=\frac{f(x_0,0)u(x_0)}{u(x_i)}, \tag{14}$$

$$x_0=\frac{\sqrt{1+2a(x_i+x_i^2/2-t)}-1}{a}.$$

The initial condition is $f=\exp(-(x-0.3)^2/0.05^2)$ and the spatial system size is 1, $\Delta t=0.2\Delta x$ and $a=1$. The boundary value is fixed to 0 because no perturbation reaches the boundary during calculation. All the calculation in Section 3 also use the same boundary condition. The accuracy of the solution is estimated by Eq. (7) at $t=0.4$ and is given in Fig. 3. It is very interesting to observe that the solution becomes the third-order if we use the averaged advection velocity both for advection and non-advection terms. With other choices, the scheme becomes the first-order. It should be noticed that we have used time-splitting technique and separately solved advection and non-advection terms. Usually, the time-splitting has been believed to deteriorate the accuracy. If the advection velocity is correctly estimated, the present scheme has been proved to retain the third-order in time and space since the solution given in Eq. (14) develops in a similar manner both in time and space.

2.3. CIP in non-uniform mesh

In the previous section, we confirmed that the CIP method has the third-order accuracy in time and space in uniform mesh. In this section, we show how the accuracy of the CIP method will be preserved in non-uniform mesh.

When the CIP method is applied to non-uniform mesh system, two different approaches are possible. One is to directly use the CIP algorithm in non-uniform mesh and the other is to use the CIP in imaginary space of curvilinear coordinate as proposed in the previous paper [13]. In the former method, the mesh size at the grid point i is assumed to be Δx_i, thus the CIP method in non-uniform mesh can be described by just replacing Δx with Δx_i in Eqs. (4) and (5).

In the latter case, the basic equations are transformed into

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial \xi}=0, \tag{15}$$

$$\frac{\partial(\partial_\xi f)}{\partial t} + U\frac{\partial(\partial_\xi f)}{\partial \xi} - u\left[\frac{1}{\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2}\frac{\mathrm{d}^2x}{\mathrm{d}\xi^2}\right]\partial_\xi f = 0, \tag{16}$$

where $\partial_\xi f$ means the derivative of f in ξ coordinate since we are using the CIP for Eq. (15) in ξ coordinate instead of x. $U = u/(\mathrm{d}x/\mathrm{d}\xi)$ is the contravariant velocity. Although there are several method to represent $\mathrm{d}x/\mathrm{d}\xi$ and $\mathrm{d}^2x/\mathrm{d}\xi^2$ in finite difference form, we here choose the fourth-order scheme:

$$\frac{\mathrm{d}x}{\mathrm{d}\xi} = \frac{-x_{i+2} + 8(x_{i+1} - x_{i-1}) + x_{i-2}}{12\Delta\xi}, \tag{17}$$

$$\frac{\mathrm{d}^2x}{\mathrm{d}\xi^2} = \frac{-x_{i+2} + 16x_{i+1} - 30x_i + 16x_{i-1} - x_{i-2}}{12\Delta\xi^2}. \tag{18}$$

Case 1. Both schemes are tested by employing two kinds of mesh size for the same initial condition of f and $\partial_x f$ as in the uniform case (Eq. (6)). In the first case, the mesh size is abruptly changed stepwise only at two points but is uniform elsewhere as

$$r(i) = \begin{cases} 1.0, & IL \leqslant i \leqslant IR, \\ \alpha, & \text{otherwise}, \end{cases} \tag{19}$$

where $\alpha = (0.5, 1.0, 1.01, 1.05, 1.2, 1.5)$. Therefore, mesh size $\Delta x_i = \Delta x \cdot r(i)$ in which $\Delta x \equiv 1.0/\sum_{i=0}^{NX} r(i)$.

The number of employed meshes is changed as $NX = (100, 200, 500, 1000, 2000)$ for each α, and the grid numbers $IL = NX/4$, $IR = IL + 20 \times NM - 1$, $NM = NX/100$. Fig. 4 shows the numerical error ϵ for each α.

The remarkable point of this result is that the direct CIP method keeps the third-order accuracy and the error for $\alpha = 0.5$ is almost the same as uniform mesh CIP. The CIP with coordinate transformation, however, becomes the second-order even when the ratio of mesh size is only 1.01. This is very significant conclusion because it would be hard to generate the mesh that changes only by 1.01 in all the neighbouring meshes.

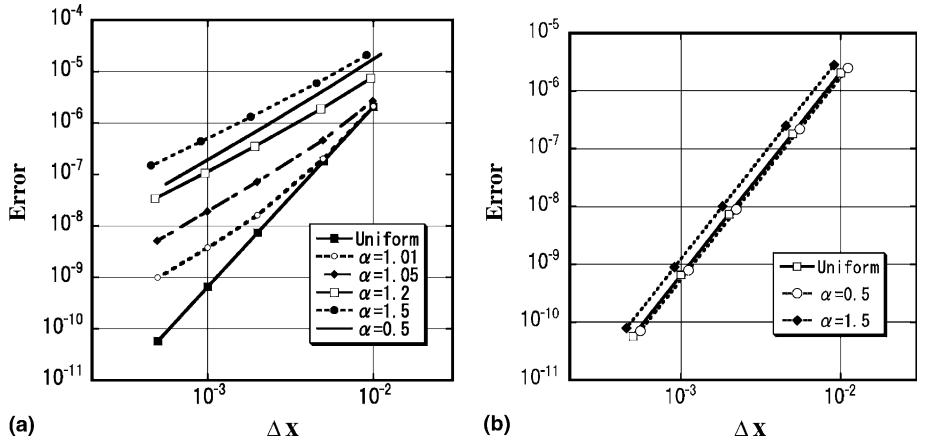

Fig. 4. The numerical error ϵ for each α in case 1 with (a) the coordinate transformed CIP and (b) the direct CIP.

Case 2. In the second case, the mesh is gradually changed as

$$r(i) = \begin{cases} 1.0 + \beta \sin\left(2\pi(i - IL)/(IR - IL)\right), & IL \leqslant i \leqslant IR, \\ 1.0 & \text{otherwise}, \end{cases} \tag{20}$$

where $\beta = (0.0, 0.05, 0.2, 0.35, 0.5)$. The mesh size $\Delta x_i = \Delta x \cdot r(i)$ in which $\Delta x \equiv 1.0/\sum_{i=0}^{NX} r(i)$.

Fig. 5 shows the nonuniform mesh size for $\beta = 0.5$ in Eq. (20) and initial profile. The CFL condition $u\Delta t/\Delta x$ is 0.2 in both cases. The number of employed meshes is changed as $NX = (100, 200, 500, 1000, 2000)$ for each β, and the grid numbers $IL = NX/4$, $IR = IL + 60 \times NM - 1$, $NM = NX/100$. Fig. 6 shows the numerical error ϵ for each β. Even for such smooth change of mesh, the error is still significant.

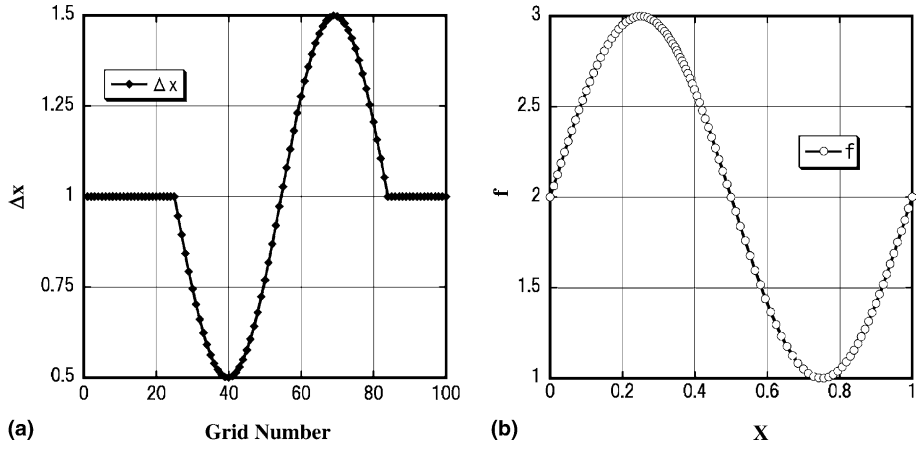

Fig. 5. (a) The mesh size Δx_i and (b) initial condition of f for $NX = 100$ and $\alpha = 0.5$.

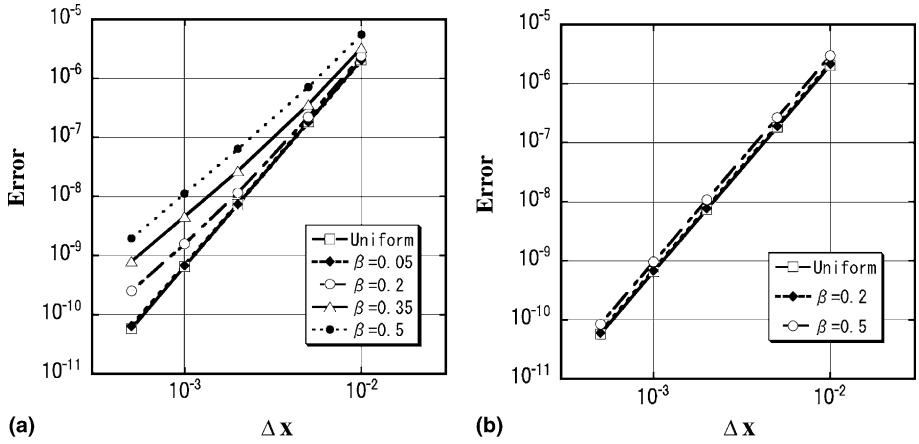

Fig. 6. The numerical error ϵ for each α in case 2 with (a) the coordinate transformed CIP and (b) the direct CIP.

3. Two-dimensional system and Soroban grid

3.1. Accuracy of splitting scheme

As shown in the previous section, direct application of the CIP to non-uniform grid system can achieve the third-order accuracy in time and space. However, the application of the method to multi-dimensions is not possible in general. This is the reason why we used the coordinate-transformation in the previous paper [13]. It would be useful to consider a special mesh system that keeps the higher-order accuracy. In adopting the special mesh system, we take advantage of the characteristics of the method.

Let us start with a conservation law in two dimensions:

$$\frac{\partial f}{\partial t}+\frac{\partial uf}{\partial x}+\frac{\partial vf}{\partial y}=0. \tag{21}$$

As in one-dimensional scheme, spatial derivative of this equation leads to

$$\frac{\partial g_x}{\partial t}+u\frac{\partial g_x}{\partial x}+v\frac{\partial g_x}{\partial y}=R_x,$$

$$\frac{\partial g_y}{\partial t}+u\frac{\partial g_y}{\partial x}+v\frac{\partial g_y}{\partial y}=R_y, \tag{22}$$

where $g_x\equiv\partial f/\partial x, g_y\equiv\partial f/\partial y$. We have defined the non-advection terms R_x, R_y as

$$R_x=-g_x\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)-g_y\frac{\partial v}{\partial x}-f\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right),$$

$$R_y=-g_x\frac{\partial u}{\partial y}-g_y\left(\frac{\partial u}{\partial x}+2\frac{\partial v}{\partial y}\right)-f\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right). \tag{23}$$

Eqs. (21) and (22) are split into advection and non-advection phases as in one-dimensional case.

Advection phase:

$$\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}+v\frac{\partial f}{\partial y}=0,$$

$$\frac{\partial g_x}{\partial t}+u\frac{\partial g_x}{\partial x}+v\frac{\partial g_x}{\partial y}=0,$$

$$\frac{\partial g_y}{\partial t}+u\frac{\partial g_y}{\partial x}+v\frac{\partial g_y}{\partial y}=0. \tag{24}$$

Non-advection phase:

$$\frac{\partial f}{\partial t}=-f\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right),$$

$$\frac{\partial g_x}{\partial t} = R_x,$$

$$\frac{\partial g_y}{\partial t} = R_y. \tag{25}$$

The time evolution of f *in advection phase* at the grid point (x_i, y_i) is realized by transferring f at the upstream departure point given by $(x_i - u\Delta t, y_i - v\Delta t)$. Retrieving the information at this departure point between four neighbouring grid points is merely the interpolation procedure inside the grid cell. The direct CIP in multi-dimensions introduces the cubic polynomial in multi-dimensional space. There exists another choice to construct the polynomial by splitting scheme that uses the one-dimensional scheme sequentially in all directions [7]. In this paper, we use the latter method.

Therefore, Eq. (24) is separated as

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial y} = 0, \tag{26}$$

$$\frac{\partial g_y}{\partial t} + v\frac{\partial g_y}{\partial y} = 0, \tag{27}$$

$$\frac{\partial g_x}{\partial t} + v\frac{\partial g_x}{\partial y} = 0, \tag{28}$$

and then

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0, \tag{29}$$

$$\frac{\partial g_x}{\partial t} + u\frac{\partial g_x}{\partial x} = 0, \tag{30}$$

$$\frac{\partial g_y}{\partial t} + u\frac{\partial g_y}{\partial x} = 0. \tag{31}$$

It is interesting to examine the accuracy of the proposed scheme in two dimensions because the directional splitting usually deteriorates the accuracy. For this purpose, we use the same analytical solution Eq. (14) but apply it to 45° direction and adopt the advection velocity $u = v = 1/[1 + a(x + y)]$. Non-advection phase Eq. (25) is solved only by algebraic manipulation as in Eq. (12) once the analytical form is used for u, v. We tested this procedure for the propagation of a profile

$$f = \exp\left(-\frac{(x-0.3)^2 + (y-0.3)^2}{0.05^2}\right) \tag{32}$$

and the system size is $x = 1$, $y = 1$, and $\Delta t = 0.2\Delta x$, $a = 1$.

Since we have already employed two splitting procedures, it seems to be hard to expect the accurate solution with the conventional schemes. Surprisingly, however, two splitting schemes, the Type-M and Type-C schemes that will be described later on, have the third-order accuracy in time and space as shown in

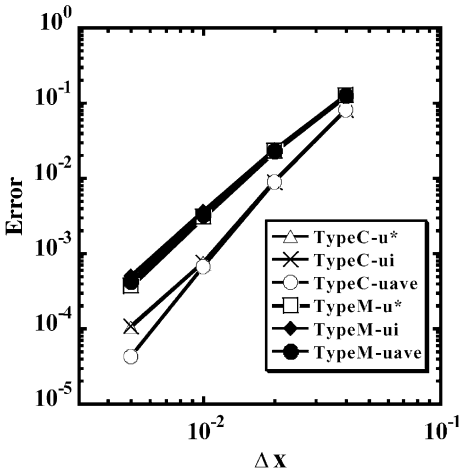

Fig. 7. Error in two-dimensional conservative equation. The region of $\Delta x > 0.005$ coincides with Fig. 3 for Type-C, while Type-M is not sensitive to advection speed.

Fig. 7 where the error estimated at $t = 0.4$ in the same manner as before. The result with Type-C agrees with that in Fig. 3 for the same range of Δx. Since Type-M is already less accurate owing to linear interpolation, it is insensitive to the choice of advection speed.

3.2. Type-M scheme

Let us consider a special mesh system shown in Fig. 8, where the vertical mesh (y-direction) is the straight line, while the grid points move along each line. In this system, we obtain the solution as follows. If the upstream departure point is given as $(\xi, \eta) = (x_i - u\Delta t, y_i - v\Delta t)$, at first *one pair of lines* satisfying $x_{i1} < \xi < x_{i1+1}$ is searched. Then two pairs of points satisfying $y_{j1} < \eta < y_{j1+1}$ and $y_{j2} < \eta < y_{j2+1}$ are searched along two lines at $x = x_{i1}$ and $x = x_{i1+1}$, respectively. Finally, the interpolation is performed with four points $(i1, j1), (i1, j1 + 1), (i1 + 1, j2), (i1 + 1, j2 + 1)$.

At first, the interpolation along the vertical straight line gives $A_{i1,\eta}$ and $A_{i1+1,\eta}$ by using the one-dimensional scheme. This means that f and g_y are transported as shown in Eq. (5) along y-direction.

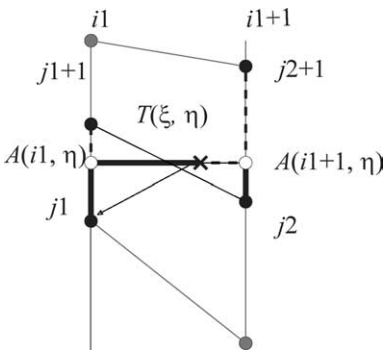

Fig. 8. Schematics of the Soroban grid. All the grid points move along the vertical lines. The number of grid points in each line can be different.

This corresponds to Eqs. (26) and (27). Therefore, A and $\partial_y A$ are readily obtained, where $\partial_y A$ represents $\partial A/\partial y$.

In the second step, T is given by interpolation along the straight line connecting $A_{i1,\eta}$ and $A_{i1+1,\eta}$ by using the one-dimensional scheme in the x-direction (see Fig. 8). In this step, A and $\partial_x A$ must be transported according to Eqs. (29) and (30). However, in the first step, only $\partial_y A$ is obtained and we need some method to estimate $\partial_x A$ from g_x, which is the procedure to solve Eq. (28).

In the previous paper [7], we proposed to use the first-order scheme for Eqs. (28) and (31) because the derivative in the direction perpendicular to the propagating direction is not sensitive and hence can be estimated only roughly. Such a splitting scheme was at first proposed in [7] which we call the "Type-M" scheme, while the direct multi-dimensional scheme commonly used is called the "Type-A" [10].

Here we represent the cubic polynomial of the procedure Eq. (5) by

$$f_i^{n+1} = \mathrm{CIP\_1D}(f_i, g_i, f_{iup}, g_{iup}, x = X),$$

$$g_i^{n+1} = \frac{\partial}{\partial x}\mathrm{CIP\_1D}(f_i, g_i, f_{iup}, g_{iup}, x = X). \tag{33}$$

This means the polynomial is given in terms of x connecting the points i and iup, and then the substitution of $x = X$ gives the updated value f_i^{n+1}. The spatial derivative of this polynomial gives the updated value g_i^{n+1}. In addition, if the linear interpolation is written as $\mathrm{FDM\_1D}(f_i, f_{iup}, x = X)$, then all the procedures are given by

Step 1.

$$A_{i1,\eta} = \mathrm{CIP\_1D}(f_{i1,j1}, \partial_y f_{i1,j1}, f_{i1,j1+1}, \partial_y f_{i1,j1+1}, y = \eta) \to \text{Eq. (26)},$$

$$\partial_y A_{i1,\eta} = \frac{\partial}{\partial y}\mathrm{CIP\_1D}(f_{i1,j1}, \partial_y f_{i1,j1}, f_{i1,j1+1}, \partial_y f_{i1,j1+1}, y = \eta) \to \text{Eq. (27)},$$

$$\partial_x A_{i1,\eta} = \mathrm{FDM\_1D}(\partial_x f_{i1,j1}, \partial_x f_{i1,j1+1}, y = \eta) \to \text{Eq. (28)}. \tag{34}$$

The same procedure is repeated also for $i1 + 1$ and $j2$.

Step 2.

$$T_{\xi,\eta} = \mathrm{CIP\_1D}(A_{i1,\eta}, \partial_x A_{i1,\eta}, A_{i1+1,\eta}, \partial_x A_{i1+1,\eta}, x = \xi) \to \text{Eq. (29)},$$

$$\partial_x T_{\xi,\eta} = \frac{\partial}{\partial x}\mathrm{CIP\_1D}(A_{i1,\eta}, \partial_x A_{i1,\eta}, A_{i1+1,\eta}, \partial_x A_{i1+1,\eta}, x = \xi) \to \text{Eq. (30)},$$

$$\partial_y T_{\xi,\eta} = \mathrm{FDM\_1D}(\partial_y A_{i1,\eta}, \partial_y A_{i1+1,\eta}, x = \xi) \to \text{Eq. (31)}. \tag{35}$$

Thus the value and the derivatives at $(i1, j1)$ at the next time step $n + 1$ are given as $f_{i1,j1}^{n+1} = T_{\xi,\eta}$, $\partial_x f_{i1,j1}^{n+1} = \partial_x T_{\xi,\eta}$, $\partial_y f_{i1,j1}^{n+1} = \partial_y T_{\xi,\eta}$.

For clear understanding of this procedure, we here show a program for Type-M scheme. Last two blocks named "interpolation along line" and "interpolation horizontally" correspond to the above procedure. Other lines are finding upstream points and neighbouring grid points as will be described in Section 4.2. $f^n = F,\ \partial_x f^n = GX,\ \partial_y f = GY,\ f^{n+1} = FN,\ \partial_x f^{n+1} = GXN,\ \partial_y f^{n+1} = GYN$.

```
c    ********************************************************************
c -- at n step F,GX,GY    at n+1 step FN,GXN,GYN
c -- these values are defined at grid No.=igrid, along the line No.=iline
c -- upstream departure point is (xnew, ynew)
c    current position (XG,YG),  advection velocity (U,V)
c    ********************************************************************
       xnew=XG(iline)-U(igrid)*DT      !  XG :x coordinate of line
       ynew=YG(igrid)-V(igrid)*DT      !  YG :y coordinate of grid
c    ********************************************************************
c -- find the line number in which xnew is included (see section 4.2)
c    ********************************************************************
       ix=xnew/DX
       iline0=IS(ix)
c    ********************************************************************
c -- find the nearest two grids along each line (iline0, iline0+1)
c -- dividing all the points along a line into two groups, find to which
c -- groups the ynew belongs.
c -- upstream points are between two grids (j1) and (j1+1) along a line
c   ********************************************************************
      DO 200 n=1,2
       iline1=iline0+n-1
       jstt=IGRIDST(iline1)   !  IGRIDST(iline) : start grid No. of iline
       jend=IGRIDED(iline1)   !  IGRIDED(iline) : end grid No. of iline
       jmdl=(jend-jstt)/2
 1000  continue
        if((YG(jstt)-ynew)*(YG(jmdl)-ynew).le.0.0) then
           jend=jmdl
           jmdl=(jmdl-jstt)/2
        else
           jstt=jmdl
           jmdl=jmdl+(jend-jmdl)/2
        endif
      if((jend-jstt).gt.1) go to 1000
        j1=jstt
c    ************************
c -- interpolation along line
c    ************************
          dd=YG(j1)-YG(j1+1)
          yy=ynew-YG(j1)
          a01 = ( (GY(j1)+GY(j1+1) )*dd+2.0*( F(j1)-F(j1+1) ) )/dd**3
          a11 = (3.0*(F(j1+1)-F(j1))-(GY(j1+1)+2.*GY(j1))*dd)/(dd*dd)
        A (n) = ((a01*yy+a11)*yy+GY(j1))*yy+F(j1)
        AY(n) = (3.*a01*yy+2.*a11)*yy+GY(j1)
        AX(n) = GX(j1)+yy/dd*(GX(j1+1)-GX(j1))  ! Linear interpolation
  200  continue
c    ************************
c -- interpolation horizontally
c    ************************
          dd=XG(iline+1)-XG(iline)
          xx=xnew-XG(iline)
          a01 = ( (AX(1)+AX(2) )*dd+2.0*( A(1)-A(2) ) )/dd**3
          a11 = (3.0*(A(2)-A(1))-(AX(2)+2.*AX(1))*dd)/(dd*dd)
        FN(igrid) = ((a01*xx+a11)*xx+AX(1))*xx+A(1)
        GXN(igrid) = (3.*a01*xx+2.*a11)*xx+AX(1)
        GYN(igrid) = AY(1)+xx/dd*(AY(2)-AY(1))   ! Linear interpolation
```

3.3. Type-C scheme

Although the Type-M scheme is sufficient for many applications, a little more accurate scheme is possible at the price of memory requirement. This scheme is proposed by Aoki [19] and we call it "Type-C scheme" hereafter. In this scheme, independent variables are now $f, \partial_x f = g_x, \partial_y f = g_y, \partial_{xy} f = \partial_x g_y = \partial_y g_x$ in two dimensions. Instead of using linear interpolation in the last line of step 1, the one-dimensional CIP scheme is applied to the advection of $(\partial_x f)$ and $\partial_y(\partial_x f)$. The latter corresponds to the spatial derivative of Eq. (28)

$$\frac{\partial(\partial_y g_x)}{\partial t} + v\frac{\partial(\partial_y g_x)}{\partial y} = 0. \tag{36}$$

Thus the CIP is applied to two pairs (f, g_y) in Eqs. (26) and (27) and $(g_x, \partial_y g_x)$ in Eqs. (28) and (36). Then all the procedures are given as

Step 1.

$$A_{i1,\eta} = \mathrm{CIP\_1D}(f_{i1,j1}, \partial_y f_{i1,j1}, f_{i1,j1+1}, \partial_y f_{i1,j1+1}, y = \eta),$$

$$\partial_y A_{i1,\eta} = \frac{\partial}{\partial y}\mathrm{CIP\_1D}(f_{i1,j1}, \partial_y f_{i1,j1}, f_{i1,j1+1}, \partial_y f_{i1,j1+1}, y = \eta),$$

$$\partial_x A_{i1,\eta} = \mathrm{CIP\_1D}(\partial_x f_{i1,j1}, \partial_{xy} f_{i1,j1}, \partial_x f_{i1,j1+1}, \partial_{xy} f_{i1,j1+1}, y = \eta),$$

$$\partial_y(\partial_x A_{i1,\eta}) = \frac{\partial}{\partial y}\mathrm{CIP\_1D}(\partial_x f_{i1,j1}, \partial_{xy} f_{i1,j1}, \partial_x f_{i1,j1+1}, \partial_{xy} f_{i1,j1+1}, y = \eta). \tag{37}$$

By use of all these values, four independent variables at (ξ, η) in Fig. 8 are calculated as follows.

Step 2.

$$T_{\xi,\eta} = \mathrm{CIP\_1D}(A_{i1,\eta}, \partial_x A_{i1,\eta}, A_{i1+1,\eta}, \partial_x A_{i1+1,\eta}, x = \xi),$$

$$\partial_x T_{\xi,\eta} = \frac{\partial}{\partial x}\mathrm{CIP\_1D}(A_{i1,\eta}, \partial_x A_{i1,\eta}, A_{i1+1,\eta}, \partial_x A_{i1+1,\eta}, x = \xi),$$

$$\partial_y T_{\xi,\eta} = \mathrm{CIP\_1D}(\partial_y A_{i1,\eta}, \partial_{xy} A_{i1,\eta}, \partial_y A_{i1+1,\eta}, \partial_{xy} A_{i1+1,\eta}, x = \xi),$$

$$\partial_x(\partial_y T_{\xi,\eta}) = \frac{\partial}{\partial x}\mathrm{CIP\_1D}(\partial_y A_{i1,\eta}, \partial_{xy} A_{i1,\eta}, \partial_y A_{i1+1,\eta}, \partial_{xy} A_{i1+1,\eta}, x = \xi). \tag{38}$$

Thus the value and the derivatives at $(i1, j1)$ at the next time step $n+1$ are given as $f^{n+1}_{i1,j1} = T_{\xi,\eta}$, $\partial_x f^{n+1}_{i1,j1} = \partial_x T_{\xi,\eta}$, $\partial_y f^{n+1}_{i1,j1} = \partial_y T_{\xi,\eta}$ and $\partial_{xy} f^{n+1}_{i1,j1} = \partial_{xy} T_{\xi,\eta}$.

3.4. Soroban grid and accuracy

It is very important to note that the number of grid points along each j line can be different. Corresponding four points $j1,\ j1+1, j2,\ j2+1$ including (ξ, η) point can be numerically searched. Most efficient method for finding the grid points in nonuniform mesh was proposed in [20]. By this method, the grid belonging to the upstream departure point is readily found with only one indexing procedure. Since this mesh system looks like an abacus, we call this system "Soroban" which is the Japanese name of abacus.

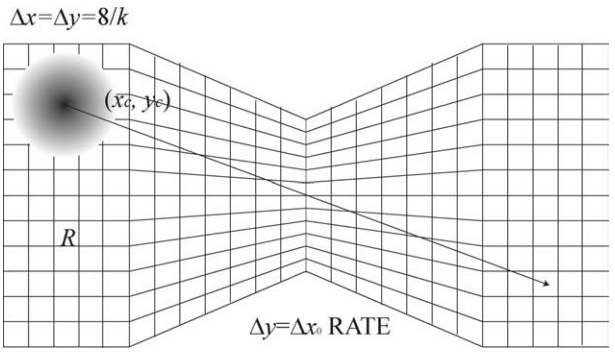

Fig. 9. Grid arrangement and initial profile for benchmark test.

For the benchmark test, we adopted a mesh system shown in Fig. 9. This mesh is symmetrical at the center in x-direction. The leftmost region is rectangular and it size is 40×96 and $\Delta x = \Delta y = \Delta x_0$. The length of the second trapezoidal region is 56 in x-direction and the mesh size Δx in x-direction is the same as that in the first region. The mesh size in y-direction changes linearly reaching $\Delta y = \mathrm{RATE} \times \Delta x_0$ at the center. The benchmark test was done for three cases $\mathrm{RATE} = 0.5, 1.0, 1.5$ by changing the mesh size as $\Delta x_0 = 8, 4, 2, 1$ for measuring accuracy. $\mathrm{RATE} = 1.0$ corresponds to the uniform region and $\mathrm{RATE} = 0.5$ gives a shape like Fig. 9.

Let us summarize the mesh arrangement denoting the mesh size as $\Delta x_0 = 8/k$ ($k = 1, 2, 4, 8$ is used to change the mesh size) and setting the grid point be (i, j), where $i = 0, 1, \ldots, 24k$, $j = 0, 1, \ldots, 12k$, then

$$\begin{aligned}
&\Delta y = \Delta x_0, \quad i = 0, \ldots, 5k,\\
&\Delta y = \Delta x_0 \times (1 + (\mathrm{RATE} - 1) \times (i - 5k)/7k), \quad i = 5k, \ldots, 12k,\\
&\Delta y = \Delta x_0 \times (1 + (\mathrm{RATE} - 1) \times (19k - i)/7k)), \quad i = 12k, \ldots, 19k,\\
&\Delta y = \Delta x_0, \quad i = 19k, \ldots, 24k,
\end{aligned} \tag{39}$$

where $\Delta x = \Delta x_0$ is fixed constant and Δy is constant along the line, $(i, j) = (0, 0)$ is the origin and $(i, j) = (96, 48)$ is the center.

We calculate the propagation of a profile

$$f(x, y) = \begin{cases} \dfrac{1+\cos\left(\pi\sqrt{(x-x_c)^2+(y-y_c)^2}/R\right)}{2}, & (x - x_c)^2 + (y - y_c)^2 < R^2, \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

and $R = 15, (x_c, y_c) = (16, 80), u = 0.1, v = -0.04, \Delta t/\Delta x_0 = 2$ are used for the test run. The initial value for the derivatives are estimated by analytically differentiating Eq. (40). At $t = 1600$, the initial profile moves to the opposite side of Fig. 9 and the numerical error is estimated by Eq. (7) and is shown in Fig. 10. Both Type-M and Type-C have approximately the third-order accuracy. The third-order accuracy of the Type-C is as is expected because it uses the CIP procedure all through the process. Although the Type-M uses the first-order scheme (linear interpolation) in estimating the derivative in perpendicular direction, it gives the accuracy better than second-order.

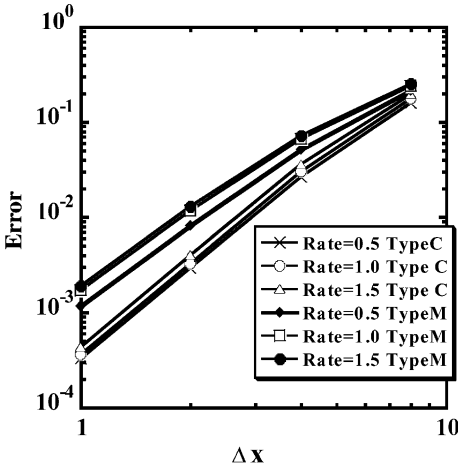

Fig. 10. The numerical error ϵ for different grid shape with Type-M and Type-C.

4. Moving adaptive Soroban grid

4.1. Adaptive grid

Here we shall discuss the possibility to use the Soroban grid as for the adaptive grid to the moving body. We must remind that all the discussion should straightforwardly apply to three dimensions. For the simplest choice of the monitoring function to the variation, we use the following quantity:

$$M(x,t) \equiv \left(1+\alpha\left(\frac{\partial f}{\partial x}\right)^2\right)^{1/2} + \beta\left(\frac{\partial^2 f}{\partial x^2}\right)^2. \tag{41}$$

Therefore monitoring function M becomes large for larger gradient region. Since the Soroban grid is straight in one-direction, it is much easier to generate the adaptive grid points along the line.

The reorganization of the grid point is easily performed by accumulated monitoring function as shown in Fig. 11. If we divide the accumulated function into equal pieces and get the x coordinate to get such integration

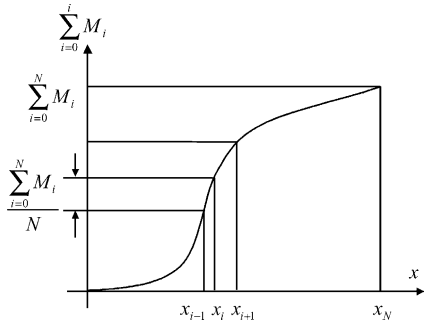

Fig. 11. Accumulated monitor function is divided into equal pieces. The x boundary of each piece gives the grid point.

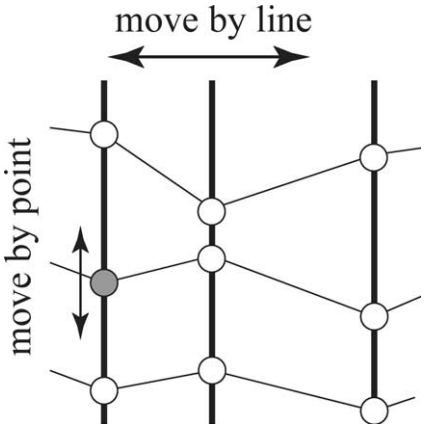

Fig. 12. The straight lines can move in horizontal direction and grid points moves along the straight lines.

$$M_i = \int_{x_i}^{x_{i+1}} M(x,t)\,\mathrm{d}x = \frac{1}{N}\int_{x_1}^{x_N} M(x,t)\,\mathrm{d}x \quad (i = 1,\ldots,N), \tag{42}$$

then these points can be used as grid points.

In the two-dimensional mesh shown in Fig. 12, mesh moving is performed as follows.

- Calculate $M(y,t)$ along each line.
- Generate the points along each line.
- Calculate the average $M(x,t)$ from all the points along each line.
- Move the lines.

This moving grid scheme is applied to the solid body revolution proposed by Zalesak [21]. The mesh of 101×101 is used and the rotation center is located at $(x_c, y_c) = (50, 50)$. The initial profile is

$$f(x,y) = \begin{cases} 1, & R \leqslant 17 \text{ and } (|x-26| > 3 \text{ or } y > 60), \\ 0, & \text{otherwise}, \end{cases} \tag{43}$$

where $R = \sqrt{(x-26)^2 + (y-51)^2}$. The mesh size is initially $\Delta x = \Delta y = 1.0$. The time step is fixed to $\Delta t = 1$, and $\alpha = 1$ and $\beta = 0.3$. Revolution speed is set so that the revolution is completed after 800 steps:

$$\begin{cases} u = -2\pi(y - y_c)/800, \\ v = 2\pi(x - x_c)/800. \end{cases} \tag{44}$$

Fig. 13 shows the contour where increment is 0.1 and the corresponding grid points are depicted. It is important to note that the time step is fixed to $\Delta t = 1$, thus as the mesh is moving as shown in the figure, the mesh can be very small and the CFL can easily exceed 1. In this simulation, we do not need the information of the connectivity between two points in different lines. The upstream departure point is searched along the line and can be separated by several mesh size from the terminal point. Actually, the maximum CFL was 22 at the locally refined mesh.

4.2. Local mesh refinement

It would be useful to notice that the present mesh system is very flexible and can describe quite a large class of surface as shown in Fig. 14. In the left figure, the length of each line and the number of grid points on each line can be different. It is important to note that the connectivity of the points along neighbouring

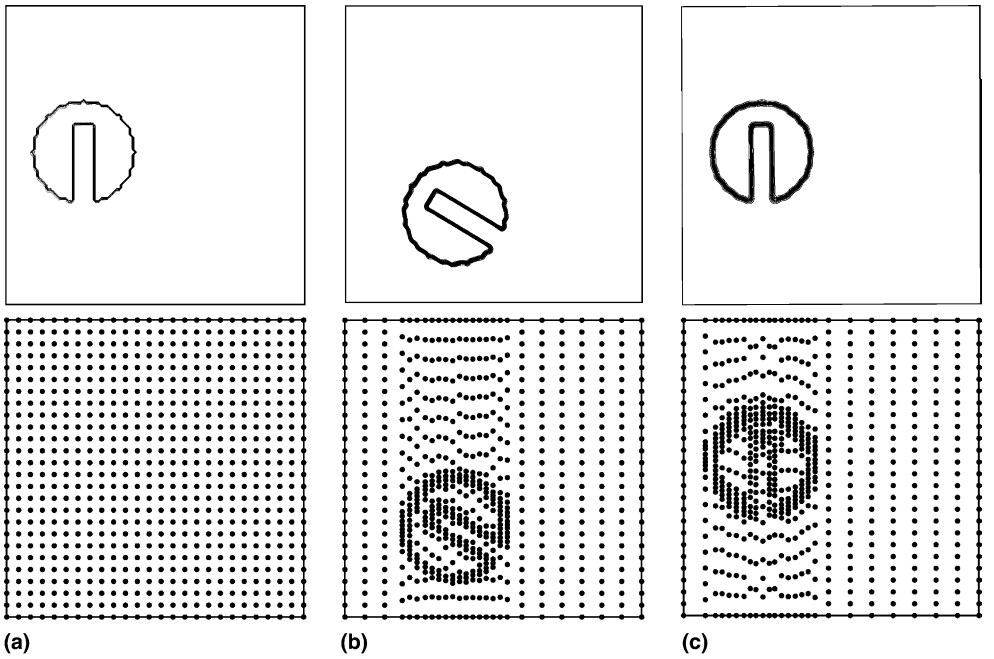

Fig. 13. Solid body revolution. Density contour and grid points. (a) $t = 0$, (b) $t = 133$, (c) $t = 800$.

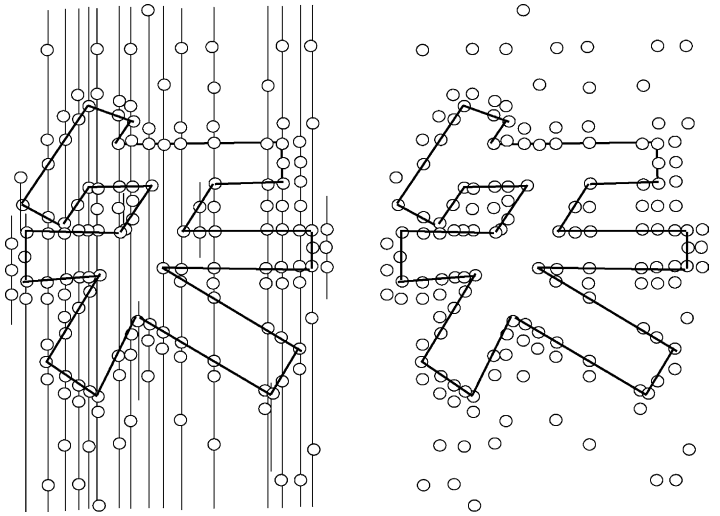

Fig. 14. Flexible adaptive mesh is possible by the Soroban grid.

two lines is not required and only the grid points shown in the right figure of Fig. 14 are used for local calculation.

The Soroban grid can easily incorporate the local mesh refinement [22]. Let us prepare two Soroban grids and calculate the same problem as in Fig. 13. Child grid system moves loosely coupled with the rigid body. We can use the adaptive scheme that was introduced in the previous section again within this child

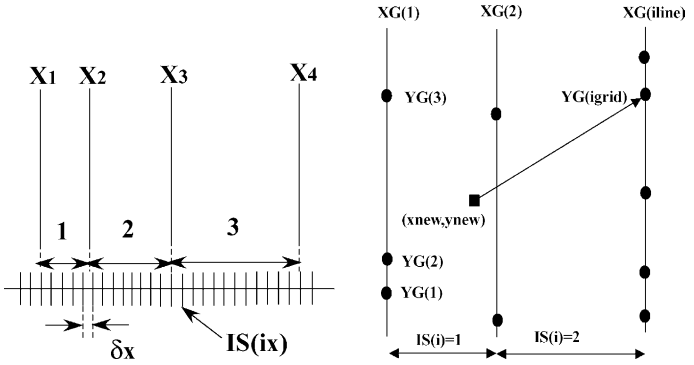

Fig. 15. (Left) A method to find a line that includes upstream point. (Right) $(x_{\text{new}}, y_{\text{new}})$ is the upstream point. YG(igrid) is the current position.

grid. For simplicity, we will not treat this problem here. As already clarified, only procedure we need is to find upstream departure points.

If the mesh-free grid system is employed, nearest neighbouring grid points for a specific upstream point can be found with the comparison of these positions. These operations are waste of computation time. In order to simplify the operations, we employ a uniform one-dimensional subgrid system [20], whose spacing is δx as shown in Fig. 15. The subgrid, to which a position belongs, can be easily found by a conventional method frequently used in the particle codes. Namely, if a position is located at $(x_{\text{new}}, y_{\text{new}})$, the number ix of the subgrid is calculated by

$$ix = \text{INT}(x_{\text{new}}/\delta x), \tag{45}$$

where INT means integer (see program in Section 3.2).

The cell, to which each subgrid belongs, is beforehand defined, that is, ix is initially related to the cell number $\text{IS}(ix)$, where $\text{IS} = 1$ if the subgrid lays between x_1 and x_2 and so on, as shown in Fig. 15.

The distance between two lines are selected to the integral times of a small size δx for the convenience of the calculation of points location. In this method, the lines move in a discrete manner, and hence δx must be chosen as small as possible for a smoothly moving grid system. Since $\text{IS}(ix)$ is one-dimensional array, a huge number of subgrid can be used. Once the line is specified, it is much easier to find nearest grid points along the line as shown in the program of Section 3.2.

Even if the parent grid points and child grid points are overlapped in the same place or become very closer with each other, we only need to find nearest four points neighbour to upstream point. Since CFL can be large as shown already, no difficulty occurs from such overlapping.

Fig. 16 shows the calculation results for the same problem as in Fig. 13. Parent grid is 30×30 and child grid is 60×60 covering the square area of 45×45 in size.

5. Conclusion

We have proposed Soroban grid which can be used to describe complex body. We confirmed that the CIP can realize the third-order accuracy both in time and space in such mesh system. The merit of the Soroban grid is

- It is easy to generate.
- Searching the upstream departure point is easy and can be done by one step by referring to the index.

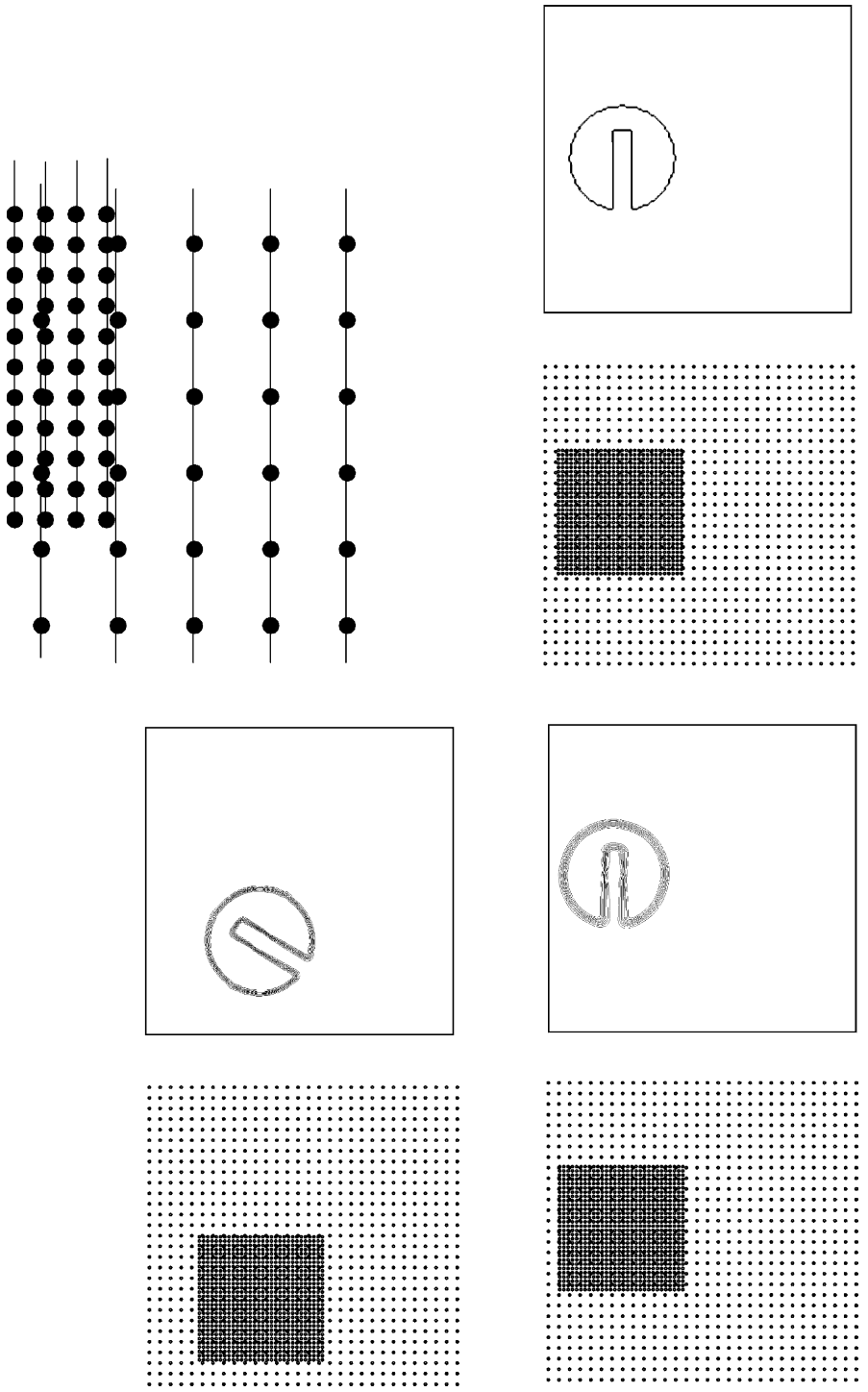

Fig. 16. Solid body revolution. Parent and child grids can be very close as in the top-left figure. Density contour and grid points at $t = 0$, $t = 133$ and $t = 800$.

- High accuracy.
- CFL can be large.

Since the spatial derivatives are also obtained by the CIP algorithm, the hyperbolic-type equation with advection and non-advection terms can be treated in the present algorithm [23] as in the case of Sections 2.2 and 3.1. The present scheme can be straightforwardly applied to the plasma simulation [24] and the propagation of electromagnetic wave. Application to the fluid dynamic equations will be discussed in future papers as well as three dimensional applications.

References

[1] T.E. Tezduyar, Finite element methods for flow problems with moving boundaries and interfaces, Arch. Comput. Methods Eng. 8 (2001) 83.
[2] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201.
[3] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluids Dynamics, 1982, p. 273.
[4] S. Osher, J.A. Sethian, Front propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12.
[5] S.O. Unverdi, G.A. Tryggvasson, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25.
[6] H. Takewaki, A. Nishiguchi, T. Yabe, The cubic-interpolated pseudo-particle (CIP) method for solving hyperbolic-type equations, J. Comput. Phys 61 (1985) 261.
[7] H. Takewaki, T. Yabe, Cubic-interpolated pseudo particle (CIP) method – application to nonlinear or multi-dimensional problems, J. Comput. Phys. 70 (1987) 355.
[8] T. Yabe, E. Takei, A new higher-order Godunov method for general hyperbolic equations, J. Phys. Soc. Jpn. 57 (1988) 2598.
[9] T. Yabe, T. Aoki, A universal solver for hyperbolic-equations by cubic-polynomial interpolation. I. One-dimensional solver, Comput. Phys. Commun. 66 (1991) 219.
[10] T. Yabe, T. Ishikawa, P.Y. Wang, T. Aoki, Y. Kadota, F. Ikeda, A universal solver for hyperbolic-equations by cubic-polynomial interpolation. II. Two- and three-dimensional solvers, Comput. Phys. Commun. 66 (1991) 233.
[11] T. Yabe, P.Y. Wang, Unified numerical procedure for compressible and incompressible fluid, J. Phys. Soc. Jpn. 60 (1991) 2105.
[12] T. Yabe, F. Xiao, T. Utsumi, Constrained interpolation profile method for multiphase analysis, J. Comput. Phys. 169 (2001) 556–593.
[13] P.Y. Wang, T. Yabe, T. Aoki, A general hyperbolic solver – the CIP method – applied to curvilinear coordinate, J. Phys. Soc. Jpn. 62 (1993) 1865.
[14] C.E. Leith, Numerical simulation of the earth's atmosphere, in: B. Alder et al. (Eds.), Methods in Computational Physics, vol. 4, Academic Press, New York, 1965, pp. 1–28.
[15] R. Tanaka, T. Nakamura, T. Yabe, Constructing exactly conservative scheme in a non-conservative form, Comput. Phys. Commun. 126 (2000) 232.
[16] T. Yabe, R. Tanaka, T. Nakamura, F. Xiao, An exactly conservative semi-Lagrangian scheme, Mon. Weather Rev. 129 (2001) 332.
[17] T. Nakamura, R. Tanaka, T. Yabe, K. Takizawa, Exactly conservative semi-Lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique, J. Comput. Phys. 173 (2002) 171.
[18] T. Utsumi, T. Kunugi, T. Aoki, Stability and accuracy of the cubic interpolated propagation scheme, Comput. Phys. Commun. 101 (1996) 9.
[19] T. Aoki, Multi-dimensional advection of CIP (cubic-interpolate propagation) scheme, CFD J. 4 (1995) 279.
[20] A. Nishiguchi, T. Yabe, Second order fluid particle scheme, J. Comput. Phys. 52 (1983) 390.
[21] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithm for fluids, J. Comput. Phys. 31 (1979) 335.
[22] M.J. Berger, J. Oliver, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484.
[23] T. Utsumi, Differential algebraic hydrodynamics solver with cubic-polynomial interpolation, CFD J. 4 (1995) 225.
[24] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov–Poisson equation in phase space, Comput. Phys. Commun. 120 (1999) 122.