INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:655–676
Published online 17 January 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.830

Challenge of CIP as a universal solver for solid, liquid and gas

T. Yabe[1,*,†], K. Takizawa[1], M. Chino[1], M. Imai[1] and C. C. Chu[2]

[1]*Department of Mechanical Engineering and Science, Tokyo Institute of Technology, 2-12-1 O-Okayama, Meguro-ku, Tokyo 152-8552, Japan*
[2]*Institute of Applied Mechanics, National Taiwan University, Taipei 106, Taiwan*

SUMMARY

We review some recent progress of the CIP method that is known as a general numerical solver for solid, liquid, gas and plasmas. This method is a kind of semi-Lagrangian scheme and has been extended to treat incompressible flow in the framework of compressible fluid. Since it uses primitive Euler representation, it is suitable for multi-phase analysis. Some applications to skimmer, swimming fish and laser cutting are presented. This method is recently extended to almost mesh-free system that is called 'soroban grid' that ensures the third-order accuracy both in time and space with the help of the CIP method. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: CIP; mesh-free; semi-Lagrangian schemes; stone-skipping; fish

1. INTRODUCTION

Over 20 years, we have been developing a universal numerical technique for all the differential equations by the method known as the cubic-interpolated propagation/constrained interpolation profile (CIP) method [1–5]. The effectiveness of the CIP has been demonstrated mainly in hyperbolic-type equations like fluid-dynamics, kinematic equations like the Fokker–Planck equation [6], the Maxwell equation, etc. Recently, a number of efforts are now going toward other type of equations: parabolic equations like diffusion equation and elliptic equations like the Poisson equation. The Shroedinger equation is also an interesting application area [7].

Another challenging extension of the CIP is to the mesh-free formulation in which the connectivity of the grid points are not important and the third-order accuracy is kept both in time and space. By this method, the complex shape as in Figure 1 of the material can be

*Correspondence to: Takashi Yabe, Department of Mechanical Engineering and Science, Tokyo Institute of Technology, 2-12-1 O-Okayama, Meguro-ku, Tokyo 152-8552, Japan.
†E-mail: yabe@mech.titech.ac.jp

Received 1 December 2003
Revised 18 August 2004

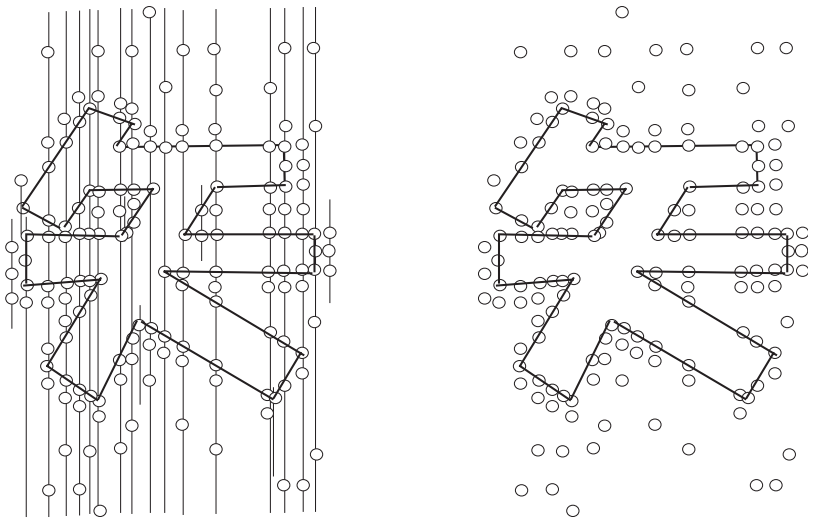

Figure 1. Adaptive mesh can be generated by the soroban grid. The vertical lines on the left figure are temporally used for interpolation and finding upstream points.

easily treated [8]. In this paper, we shall overview some of such movements with brief review of the CIP method.

2. CIP METHOD

2.1. CIP formulation in one dimension

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 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 Equation (1) gives a simple translational motion of wave with a velocity u. The initial profile (solid line of Figure 2(a)) 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 Figure 2(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 Figure 2(c). This process is called the first-order upwind scheme. On the other hand, if we use quadratic polynomial for

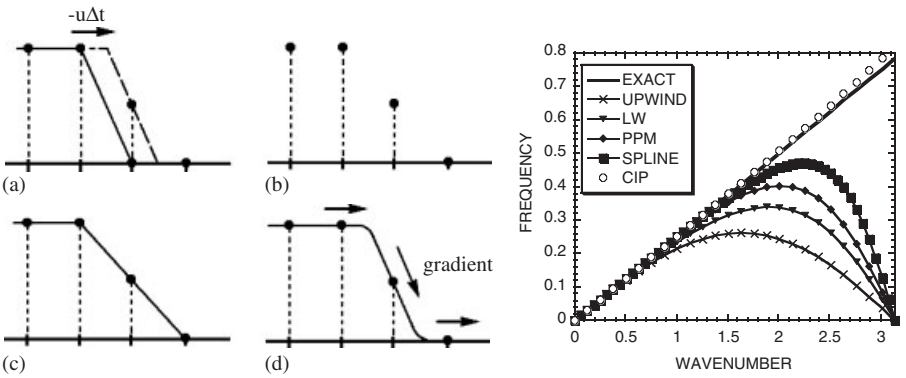

Figure 2. The principle of the CIP method. (a) Solid line is initial profile and dashed line is an exact solution after advection, whose solution (b) at discretized points. (c) When (b) is linearly interpolated numerical diffusion appears. (d) In the CIP, spatial derivative also propagates and the profile inside a grid cell is retrieved. (Right) Phase error of various schemes.

interpolation, it suffers from overshooting. This process is the Lax–Wendroff scheme or Leith scheme [9].

What made this solution worse? This is because we neglect the behaviour of the solution inside grid cell and merely follow after the smoothness of the solution. We proposed to approximate the profile as shown below. Let us differentiate Equation (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 stands for the spatial derivative of f, $\partial f/\partial x$. In the simplest case where the velocity u is constant, Equation (2) coincides with Equation (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 Equation (1). If g is predicted after propagation as shown by the arrows in Figure 2(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 much closer to the initial profile that is real solution. Most importantly, the solution thus created gives a profile consistent with Equation (1) even inside the grid cell. Importance of this constraint is demonstrated in the right figure of Figure 2 which shows the dispersion relation obtained by various numerical algorithms [10]. The mathematical analysis of the CIP method proves that the phase speed of all the waves is accurately reproduced up to the wavenumber of $k\Delta x = \pi$ [10].

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 $f^{n+1} = F(x - u\Delta t), g^{n+1} = \mathrm{d}F(x - u\Delta t)/\mathrm{d}x$.

$$\begin{aligned} a_i &= \frac{g_i + g_{iup}}{D^2} + \frac{2(f_i - f_{iup})}{D^3} \\ b_i &= \frac{3(f_{iup} - f_i)}{D^2} - \frac{2g_i + g_{iup}}{D} \end{aligned} \tag{3}$$

$$f_i^{n+1} = a_i\xi^3 + b_i\xi^2 + g_i^n\xi + f_i^n$$
$$g_i^{n+1} = 3a_i\xi^2 + 2b_i\xi + g_i^n \tag{4}$$

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

We should address the following points.

(1) The calculation costs do not increase even if an additional variable like derivative is introduced. For example, the CPU time required for a simple advection problem in one-dimension is Cubic Lagrange/CIP = 1.0, Spline/CIP = 1.68 (Thomas method is used to solve matrix), PPM [11]/CIP = 2.31.

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

(3) Adding to (2), in most of practical applications, we use additional variables like thermal conductivity, viscosity, temporal variables for matrix solution, etc. Let L_0 be the number of these variables, the memory requirement is then $(\alpha+1)L_t + L_0$ and $n^\alpha(L_t + L_0)$. Since $L_0 \gg L_t$ usually, memory requirement of the CIP is similar to other schemes even for un-refined mesh $n = 1$.

2.2. *Variable velocity field*

Although the CIP has been proved to have the third-order accuracy in time and space for simple advection equation, it does not seem to be an easy task to guarantee the same accuracy for general equations that include non-advection term. Towards this goal, we move step by step. At first we start with the advection equation with variable advection velocity.

In this case, $\partial u/\partial x$ part on the right-hand side of Equation (2) remains. For movement of the profile, we must notice that the velocity should change during the advection. In testing this effect, we shall use the following example. When the advection velocity is $u(x) = 1/(1 + ax)$, we have an analytical solution at each grid point x_i.

$$f(x_i, t) = f(x_0, 0)$$
$$x_0 = \frac{\sqrt{1 + 2a(x_i + x_i^2/2 - t)} - 1}{a} \tag{5}$$

We shall compare the CIP solution with this analytical one. Here the system size is set to 1 and $a = 1$, initial profile is given by $f(x) = \exp[-(x - 0.2)^2/0.05^2]$, $g(x) = -2(x - 0.2)\exp[-(x - 0.2)^2/0.05^2]$. The time step and grid spacing are chosen so that $\Delta t/\Delta x = 0.2$ and $\Delta x = 1/NX$ with total mesh of $NX = 50$, 100, 200, 400, 800, 1600.

If we choose $\xi = -u_i\Delta t$ in Equation (4), where u_i is the velocity at the grid point x_i of concern, the solution becomes first-order accurate. Even if we use the velocity u^* at the upstream position $x_i - u_i\Delta t$, the accuracy still stays in the first order as shown in Figure 3.

Interestingly, if we use the average velocity $(u_i + u^*)/2$ for improved estimation of the upstream point as $\xi = (u_i + u^*)\Delta t/2$, then the third-order accuracy is realized. Since in the

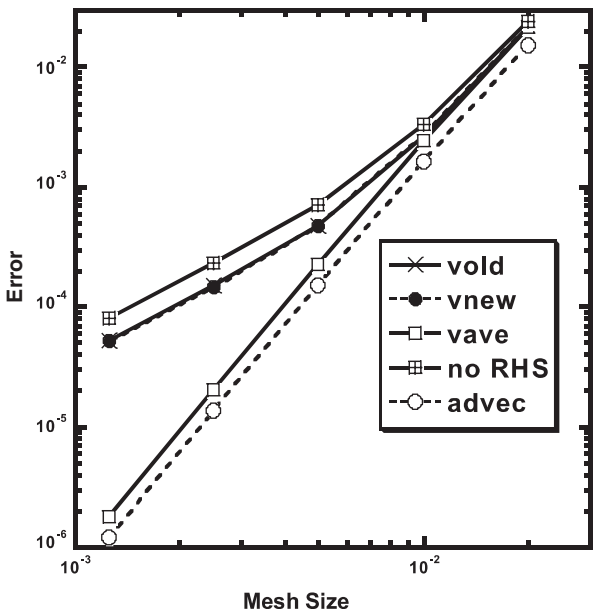

Figure 3. The numerical error at $t = 0.4$. v_{old}: $\xi_{\text{old}} = -u_i \Delta t$, v_{new}: $\xi_{\text{new}} = -u^* \Delta t$, v_{ave}: $\xi_{\text{ave}} = -(u_i + u^*)/2.\Delta t$, no RHS: vave but without non-advection term in g, advec: advection with constant velocity.

present example, the profile changes both in time and space, we can conclude that the average velocity procedure can guarantee the third-order accuracy both in time and space.

It is important to note that even in average velocity case, the neglection of the non-advection terms $-g\partial u/\partial x$ causes serious error. In all the calculations, we estimate the $\partial u/\partial x = -a/(1+ax)^2$ at x_i. Although we do not use the higher order scheme for this term, the accuracy is still very high.

2.3. Non-advection term

If the equation includes the non-advection term,

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = H \tag{6}$$

$$\frac{\partial g}{\partial t} + u\frac{\partial g}{\partial x} = -\frac{\partial u}{\partial x}g + \frac{\partial H}{\partial x} \tag{7}$$

we proposed to solve it by time splitting technique. At first, the CIP is applied to the advection term and updates the value and derivative to f^*, g^*. Then the non-advection term is solved as follows [2].

$$f_i^{n+1} = f_i^* + H_i \Delta t \tag{8}$$

$$\frac{g_i^{n+1}-g_i^*}{\Delta t}=\frac{f_{i+1}^{n+1}-f_{i-1}^{n+1}}{2\Delta x\Delta t}-\frac{f_{i+1}^*-f_{i-1}^*}{2\Delta x\Delta t}-\left(\frac{\partial u}{\partial x}\right)_i g_i^* \tag{9}$$

In the last equation, the differentiation of H that appeared in Equation (9) is no more needed. Since in the solution of f in Equation (8) H is already used, instead of the differentiation of H, we can take a differentiation of $(f^{n+1}-f^*)/\Delta t$. For example, if the non-advection term includes thermal conduction, we need to solve this term implicitly by matrix. It is not economical to solve the third-order spatial derivative in matrix form. Instead of this, we can include this effect by the above term after the thermal conduction is solved implicitly.

Let us apply this method to a conservative equation

$$\frac{\partial f}{\partial t}+\frac{\partial fu}{\partial x}=0 \tag{10}$$

In this equation, $h=-(\partial u/\partial x)f$. As in the previous case, we have an analytical solution,

$$f(x_i,t)=\frac{f(x_0,0)u(x_0)}{u(x_i)} \tag{11}$$

when the advection velocity $u(x)=1/(1+ax)$ is used. Here x_0 is the same as Equation (5).

Since u is not a function of time, non-advection term in Equation (8) can be easily integrated. The corresponding term in g equation is also solved as

$$\begin{aligned} f_i^{n+1}&=f_i^*\exp[-\partial u/\partial x\Delta t]\\ g_i^{**}&=g_i^*\exp[-\partial u/\partial x\Delta t]\end{aligned} \tag{12}$$

remaining terms are written as

$$g_i^{n+1}=g_i^{**}+\frac{f_{i+1}^{n+1}-f_{i-1}^{n+1}}{2\Delta x}-\frac{f_{i+1}^*-f_{i-1}^*}{2\Delta x} \tag{13}$$

The same initial condition as the previous section is used. Also in this case, average advection speed gives the third-order accuracy as shown in Figure 4. $\partial u/\partial x$ must be calculated at the present position and upstream position and then averaged. We should note that the Runge–Kutta method for time integration corresponding to Equation (12) gave the same result. This would be useful for general non-advection terms.

This is the surprising result. With non-advection term and variable velocity, the CIP can keep the third-order accuracy in time and space even if the time splitting technique is used.

We have another choice to calculate the non-advection term. Without using the finite difference, we can use the derivatives already obtained. The term corresponding to Equation (13) is only $\partial H/\partial x$ in Equation (7) which is now the spatial derivative of $-f_i\partial u/\partial x$. Then

$$g_i^{n+1}=g_i^{**}-\left(g\frac{\partial u}{\partial x}+f_i\frac{\partial^2 u}{\partial x^2}\right)\Delta t \tag{14}$$

Since f and g are known at the grid points, we can simply evaluate these terms. The result is shown in Figure 4 (right).

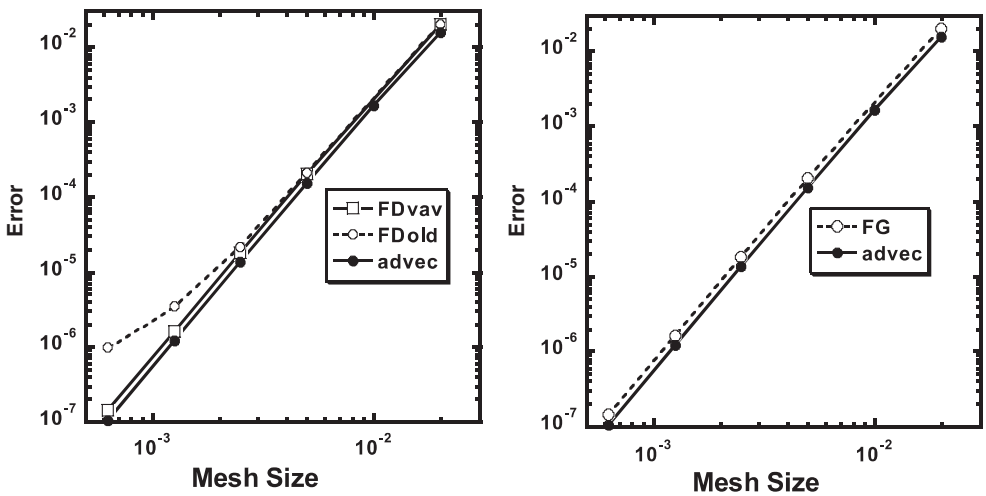

Figure 4. (Left) FD_{vav}: In Equation (12), derivative of velocity $(u_i + u^*)/2$ is used. FD_{old}: only u_i. (Right) FG: Equation (14) is used.

2.4. Interface tracking: a sharpness preserving method

Treatment of interface that lies between materials of different properties remains a formidable challenge to the computation of multi-phase fluid dynamics. Eulerian methods have proven robust in simulating flows with interfaces of complex topology. Generally, Eulerian methods use colour functions to distinguish the regions where different materials fall in. To accurately reproduce the physical processes across the interface transition region, keeping the compact thickness of the interface is of great importance. The finite difference schemes constructed on an Eulerian grid, however, intrinsically produce numerical diffusions to the solution of advection equation by which the interface is predicted temporally. Thus, the direct implementation of finite difference schemes (even of high order) cannot maintain the compactness of the interface.

Various kinds of methods have been developed so far to achieve a compact and correctly defined interface by introducing extra programming. Among those mostly used algorithms are the volume of fluid (VOF) methods [12, 13] the level set methods [14] for front capturing, and others for front tracking [15, 16]. Level set method gets around the computation of interfacial discontinuity by evaluating the field in higher dimensions. The interface of interest is then recovered by taking a subset of the field. Practically, the interface is defined as the zero level set of a distance function from the interface.

In References [17, 18], we devised an interface tracking technique which appears efficient, geometrically faithful and diffusionless. The method is a combination of the CIP advection solver and a tangent function transformation.

Consider K kinds of impermeable materials occupying closed areas $\{\Omega_k(t), k = 1, 2, \ldots, K\}$ in computational domain $D \in \mathbf{R}^3(x, y, z)$, we identify them with colour functions or density functions $\{\phi_k(x, y, z, t), k = 1, 2, \ldots, K\}$ by the following definition:

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

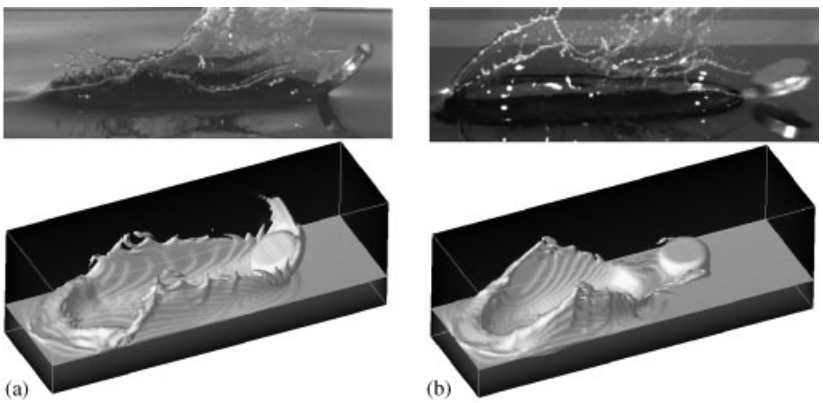

Figure 5. Two examples of skimmer with different attack angles of aluminium disk over the water surface (top: experiment, bottom: simulation).

Suppose these materials move at the local speed, the colour functions evolve then according to the following advection equation:

$$\frac{\partial \phi_k}{\partial t} + \mathbf{u} \cdot \nabla \phi_k = 0, \quad k = 1, 2, \ldots, K \tag{15}$$

where $\mathbf{u}$ is the local velocity.

It is known that solving the above equation by finite difference schemes in an Eulerian representation will produce numerical diffusion and tend to smear the initial sharpness of the interfaces. In our method, rather than the original variable ϕ_k itself, its transformation, say $F(\phi_k)$, is calculated by the CIP method. We specify $F(\phi_k)$ to be a function of ϕ_k only, which means that the new function $F(\phi_k)$ is also governed by the same equation as (15). Hence, we have

$$\frac{\partial F(\phi_k)}{\partial t} + \mathbf{u} \cdot \nabla F(\phi_k) = 0 \tag{16}$$

and all the algorithms proposed for ϕ_k (schemes for advection equation) can be used to $F(\phi_k)$. Hopefully, by the considerable simplicity, this kind of techniques would be very attractive for practical implementation. We here use a transformation of a tangent function for $F(\phi_k)$, that is,

$$F(\phi_k) = \tan[(1 - \varepsilon)\pi(\phi_k - 1/2)] \tag{17}$$

$$\phi_k = \tan^{-1} F(\phi_k)/[(1 - \varepsilon)\pi] + 1/2 \tag{18}$$

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

Figure 5 shows an example of this treatment in fluid dynamics. This example is known as the skimmer and was at first investigated by the authors with experiments and simulations [19]. An aluminium disk is thrown over the water surface. By the reaction force

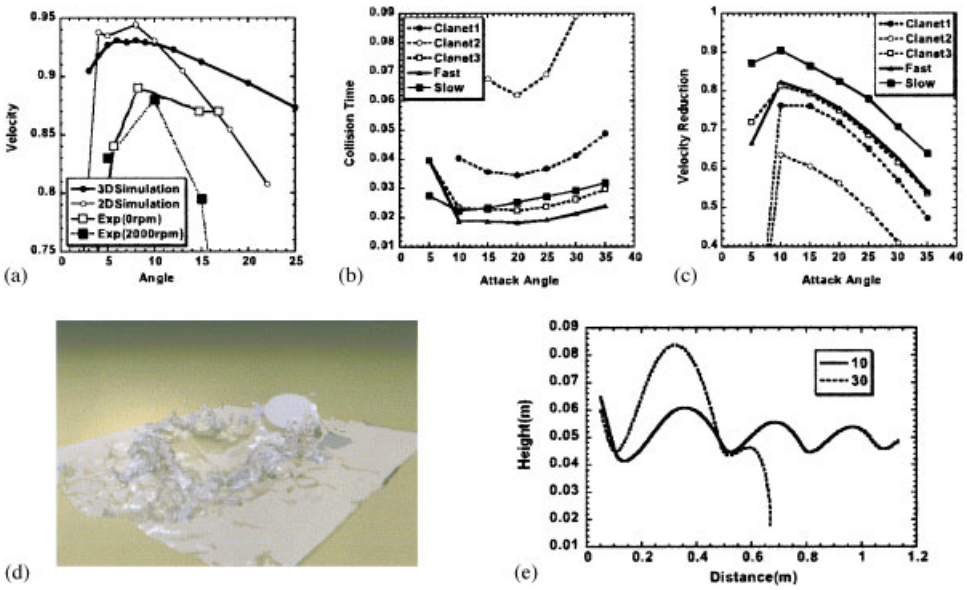

Figure 6. (a) Velocity reduction for various attack angles. injection speed $= U = 5-6$ m/s and injection angle $= \gamma$ is 9.5–11.3. (b) Collision time (c) Velocity reduction. Dashed lines are the results with the parameters similar to Clanet's experiment. Clanet1: $U = 3.5$, $\gamma = 20$, Clanet2: $U = 3.5$, $\gamma = 30$, Clanet3: $U = 5, \gamma = 20$, Fast: $U = 6$, $\gamma = 20$, Slow: $U = 3.5$, $\gamma = 10$. (d) A snapshot of 3D simulation. (e) Trajectory of mass centre. Water surface is located at 0.04 m. $U = 3.5$, $\gamma = 20$ and $\beta = 10, 30$.

from the water at the impact, this disk gains the lifting force. For two examples of different attack angles, the simulation replicated the experimental results very well. The initial sharpness of the disk is well preserved and the discontinuities are advected with a correct speed.

Let us show some universal behaviour seen in the experimental results with rotation and without rotation of the disk [20], and two- and three-dimensional simulations [19] as shown in Figure 6(a). In all the results, the minimum deceleration occurs around attack angle $= \beta = 10$ independent of the rotation. The three-dimensional simulation with rotation of 2000 rpm and the two-dimensional simulation but with fixed attack angle all through the computation also give similar tendency although the velocity reduction is slightly smaller in the simulation. Since all the results give the minimum velocity reduction around $\beta = 5-10$, we are encouraged to use only two-dimensional simulation for further parameter study by economical reason.

After three years from our first experiment [19] on skimmer, Clanet *et al.* [21] argued that shortest collision time at the attack angle of $\beta = 20^\circ$ is beneficial to successful stone-skipping. However, the key issue is how to sustain the kinetic energy and thus Figure 6(a) indicates the inadequacy of their conclusion because large deceleration can be exerted as well as larger lifting force leading to shorter collision time. If the attack angle is A, lifting force is proportional to $\cos A$ while the drag force is to $\sin A$ and thus larger angle leads to larger

deceleration. In addition, short collision time with large force gives the same deceleration as the long collision time with small force since (force)×(duration time) determines overall deceleration. Therefore the discussion based only on the collision time leads to incorrect conclusion for the optimal skipping.

In order to prove the correctness of the present simulation, we compared the collision time with similar parameters as in the Clanet's experiment as shown in Figure 6(b) which well replicates the tendency in the experiments [21]: the collision time is shortest at $\beta = 20$ and three dashed curves very resemble the experiments. The deceleration of the disk, however, is minimum at $\beta = 10$ as in Figure 6(c) that also coincides with Figure 6(a).

Now a question arises whether the collision time is really a key issue for the skipping or not. The answer is no. We performed calculation allowing number of hoppings until the disk sinks as shown in Figure 6(e). Interestingly, the disk for $\beta = 30$ failed to continue skipping, while number of hoppings is much larger for $\beta = 10$ as the deceleration results imply although the collision time at $\beta = 10$ and 30 for Clanet1 in Figure 6(b) is the same.

In this example, a non-slip condition is used to impose the solid motion onto the velocity field of fluid which in turn is driven through the velocity coupling like

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -v \sum_l \phi_l(\mathbf{u} - \bar{\mathbf{u}}_{b(l)}) \tag{19}$$

Slip boundary can also be treated similarly but with a little complication [22] like

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -v \sum_l \nabla\phi_l(\mathbf{u} - \bar{\mathbf{u}}_{b(l)}) \tag{20}$$

Difference is that the relaxation occurs only at the surface recognized by the spatial gradient of density function. Then, $\bar{\mathbf{u}}_{b(l)}$ is determined by

$$\begin{gathered} \bar{\mathbf{u}}_{b(l)} \cdot \mathbf{n} = 0, \quad \bar{\mathbf{u}}_{b(l)}^2 = \mathbf{u}^2 \\ \frac{\bar{\mathbf{u}}_{b(l)} \times \mathbf{n}}{|\bar{\mathbf{u}}_{b(l)} \times \mathbf{n}|} = \frac{\mathbf{u} \times \mathbf{n}}{|\mathbf{u} \times \mathbf{n}|} \end{gathered} \tag{21}$$

Such an accurate simulation can sometimes reveal important physics. One of such interesting examples is the quick turn of a killifish. Figure 7 (right) shows the time evolution of the killifish movement. In order to shed light on the physics of the motion, we idealized its motion to clarify the driving force in the quick turn. We imposed a simple folding and unfolding motion around the mass centre of the killifish. Then we compared two different shapes. One shape has an equal mass along the body. Another shape is set to be close enough to the real figure.

As shown in Figure 7, equal-mass fish returns to its original direction without turn, while measured shape shows a turn by 90°. It is too early to simply attribute this turn to a large drag in the tail side because the force in folding phase will be cancelled out by that in unfolding phase. Therefore, the quick turn of the killifish can be achieved by small difference of water response in folding and unfolding motion.

2.5. Treatment of surface tension

Although the tangent transformation can improve the sharpness of the interface, such sharpness may cause another serious problem in calculating the curvature of the liquid. In calculating

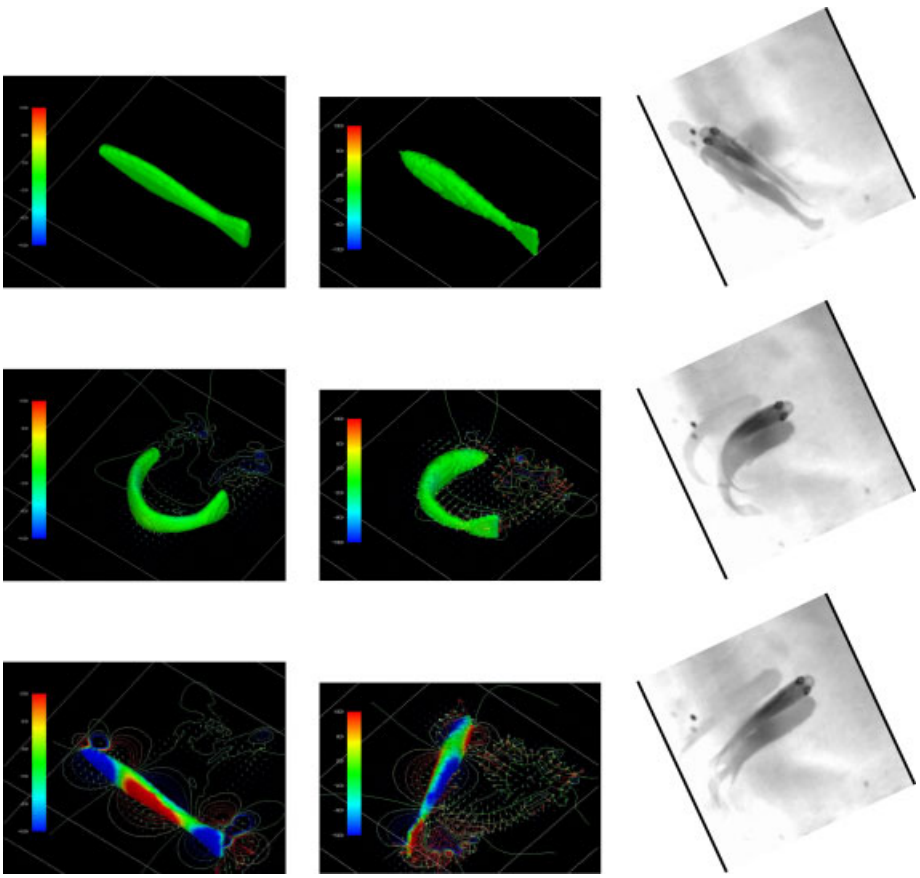

Figure 7. Physics of killifish turn. (Left) constant thickness, (centre) real figure, (right) observation.

the curvature by $\nabla \cdot \mathbf{n}$ where $\mathbf{n} \equiv \nabla\phi$ is the normal vector, the level set [14] has been one of the best methods because $|\nabla\phi|$ is designed to be 1 by re-initialization procedure and thus the calculation of the curvature is easy and accurate.

In other words, the re-initialization of the level set provides the smooth surface for the calculation of $\nabla\phi$. In this sense, we may choose alternative method because we know already correct location of interface by the CIP method and most importantly the re-initialization in the level set method sometimes gives rise to incorrect movement of the 0 level leading to volume error. Thus some researchers tried to use the CIP method for this re-initialization to correctly calculate the surface movement.

We prefer to use the CIP method throughout the process and thus we choose to diffuse ϕ after it is calculated by the CIP method in Equation (16) only for the calculation of the curvature. We call this diffused profile $\bar{\phi}$. It should be noticed that we always keep tracing the sharp interface, all the calculation is done based on this profile. The diffusion starts from

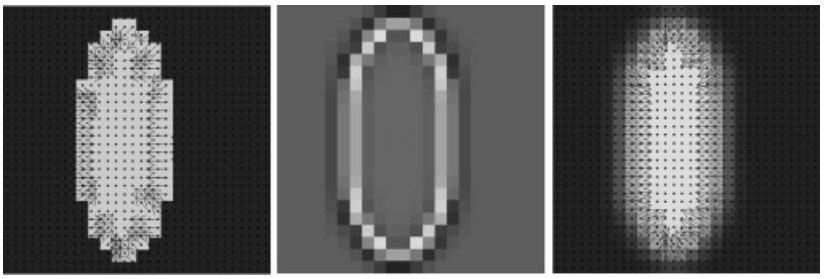

Figure 8. (Left) The original profile ϕ calculated by tangent transformation. Horizontal diameter is only 8 grids. The arrows show the direction of the normal vector. (Centre) ϕ is smoothed by procedure (22). (Right) The direction of normal vector of smoothed surface.

ϕ calculated by Equation (16) at every time step and we use

$$\begin{aligned}\phi^* &= \phi \\ \bar{\phi} &= \frac{1}{2}\phi^*_{i,j} + \frac{1}{2}\frac{1}{1+4C_1+4C_2}[\phi^*_{i,j} + C_1(\phi^*_{i-1,j} + \phi^*_{i+1,j} + \phi^*_{i,j-1} + \phi^*_{i,j+1}) \\ &\quad + C_2(\phi^*_{i-1,j-1} + \phi^*_{i-1,j+1} + \phi^*_{i+1,j-1} + \phi^*_{i+1,j+1})]\end{aligned} \tag{22}$$

This process is repeated by five times at each time step replacing ϕ^* by $\bar{\phi}$ so that this leads to the diffusion of about two grid sizes. The coefficient C_1, C_2 depends on the distance from the centre (i, j) and then

$$C_1 = 1/\left(1 + 1/\sqrt{2}\right)$$

$$C_2 = C_1/\sqrt{2}$$

Although the finite difference form of diffusion equation does not include the term of C_2, we used this expression in order to avoid the effect of the Cartesian grid. We confirmed that this procedure is very stable for the calculation of the curvature even for the perturbation of one grid size.

Figure 8 (left) shows the elliptic liquid drop whose horizontal diameter is 8 grid size. The arrow shows the normal vector calculated by this surface. The middle figure is the contour of diffused surface $\bar{\phi}$ by Equation (22). The arrows in Figure 8 (right) are calculated by this diffused surface.

Figure 9 shows the time evolution of the liquid drop with the above two calculation methods of the normal vector coupled with the hydrodynamic calculation. On the top figure, smoothing is not used and thus because of incorrect force, the oscillation does not occur and the shape monotonically changes. On the contrary, if we use the diffused surface, the surface approaches the circle oscillating as is expected even if such coarse mesh is used.

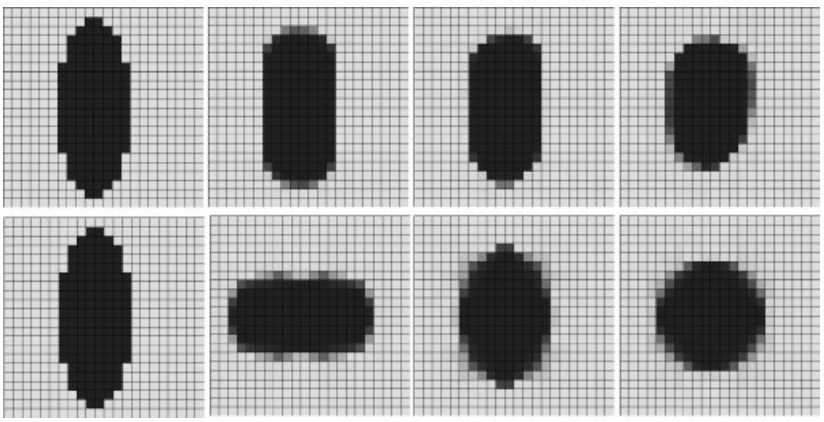

Figure 9. Time evolution of the liquid surface calculated with (top) non-smoothed surface and (bottom) smoothed surface.

2.6. *Pressure-based algorithm in primitive Euler representation*

We apply the above-mentioned strategy to general hydrodynamic equations including elastic motion of solid by the following set of hydrodynamic-type equations:

$$\frac{\partial \mathbf{f}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{f} = \mathbf{S} \tag{23}$$

Here, $\mathbf{f} = (\rho, \mathbf{u}, T)$, $\mathbf{S} = (-\rho \nabla \cdot \mathbf{u} + Q_m, -\nabla p/\rho + Q_u, -P_{TH} \nabla \cdot \mathbf{u}/\rho C_v + Q_E)$, and ρ is the density, $\mathbf{u}$ the velocity, p the pressure, T the temperature, Q_m represents the mass source term, Q_u represents viscosity, elastic stress tensor, surface tension, etc., and Q_E represents viscous heating, thermal conduction and heat source.

Here, C_v is the specific heat for constant volume and we define $P_{TH} = T(\partial p/\partial T)_\rho$ which is derived from the first principle of thermodynamics and the Helmholtz free energy.

The CIP method uses the primitive Euler method to solve Equation (23), thus the formulation into a simultaneous solution of incompressible and compressible fluid is readily obtained. In order to get an idea of this strategy, we shall start at first examining how it has been difficult to solve them together. In ordinary compressible fluid, the density ρ is solved by the mass conservation equation and then the temperature T is obtained by energy equation. After that, from the equation of state (EOS) schematically shown in Figure 10, the pressure $p = p(\rho, T)$ is calculated. In the low-density side, $p \propto \rho T$ like ideal fluid and dependence is relatively weak, but at solid or liquid density p steeply rises as the density. This means that extremely high pressure is needed to compress solid or liquid even slightly. In other words, for solid or liquid, the sound speed $C_s = (\partial p/\partial \rho)^{1/2}$ is very large. Therefore, if we choose the process in which density is calculated at first, only a small amount of error on density of 10%, for example, causes a large pressure pulse by 3–4 orders of magnitude.

In such a situation, incompressible approximation is normally adopted, that is, pressure equation to ensure $\nabla \cdot \mathbf{u} = 0$ is derived from equation of motion and mass conservation. This scheme is called pressure-based scheme [23].

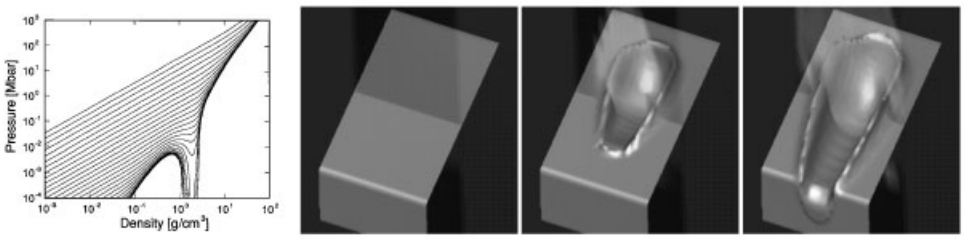

Figure 10. (Left) The schematics of EOS. (Right) Three-dimensional simulation of laser cutting. Laser is scanning over a surface of acrylic. In the central plane of each figure, contour of gas flow is depicted. In the right figure, melting flow can be seen.

In order to extend this idea to compressible fluid, we had better modify the EOS. Let us rotate the EOS by 90°, then the steep pressure curve becomes now flat density curve. This means that if we could solve the pressure at first and then estimate the density later on by this EOS in terms of $\rho(p, T)$, the problem at the liquid density will be removed. Adding to this, since the EOS in lower density gas depends linearly on other quantities, no problem occur there by this reverse procedure.

Then how we realize this reverse procedure? For this purpose, we should predict how the pressure reacts to the change of density and temperature. Such a unified procedure to incorporate compressible fluid with incompressible fluid has been initiated by Harlow as the implicit continuous Eulerian (ICE) [24] that used conservative Euler equations.

On the other hand, the CCUP [25] uses primitive Euler equations and splits the advection term from the other terms related to sound waves instead of conservative form. By this simplification, pressure equation becomes simple and the ability to attack the multi-phase flow has been greatly improved.

For small change of density and temperature, the pressure change can be linearly proportional to them as [26]

$$\Delta p = \left(\frac{\partial p}{\partial \rho}\right)_T \Delta\rho + \left(\frac{\partial p}{\partial T}\right)_\rho \Delta T \tag{24}$$

where Δp means the pressure change $p^{n+1} - p^*$ during one time step and * is the profile after advection. This applies also to ρ, T. From this relation, once $\Delta\rho, \Delta T$ are predicted, Δp will be predicted based on Equation (24). Needless to say, $\partial p/\partial\rho, \partial p/\partial T$ are given by EOS.

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

$$\begin{aligned} \Delta\rho &= -\rho^* \nabla \cdot \mathbf{u}^{n+1} \Delta t \\ \rho^* C_v \Delta T &= -P_{TH} \nabla \cdot \mathbf{u}^{n+1} \Delta t \end{aligned} \tag{25}$$

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

$$\Delta\mathbf{u} = -\frac{\nabla p^{n+1}}{\rho^*} \Delta t \tag{26}$$

Since $\Delta\mathbf{u} = \mathbf{u}^{n+1} - \mathbf{u}^*$, Equations (24)–(26) leads to a pressure equation [25, 26]

$$\nabla\left(\frac{1}{\rho^*}\nabla p^{n+1}\right) = \frac{p^{n+1} - p^*}{\Delta t^2\left(\rho^* C_s^2 + \dfrac{P_{TH}^2}{\rho C_v T}\right)} + \frac{\nabla\cdot\mathbf{u}^*}{\Delta t} \tag{27}$$

Then substituting the given p^{n+1} into Equation (26), we obtain the velocity $\mathbf{u}^{n+1}$ and then density ρ^{n+1} from Equation (25). From this procedure, density can be solved in terms of pressure which is analogous to rotate EOS by 90°. Equation (27) has many important features in the following points. This equation shows that, at the sharp discontinuity, $\mathbf{n}\cdot(\nabla p/\rho)$ is continuous. Since $\nabla p/\rho$ is the acceleration, it is essential that this term is continuous since the density changes by several orders of magnitude at the boundary between liquid and gas. In this case, the denominator of $\nabla p/\rho$ changes by several orders and pressure gradient must be calculated accurately enough to ensure the continuous change of acceleration. The pressure equations derived by the ICE seems to be similar to Equation (27) but the continuity of $\nabla p/\rho$ in the formers is not guaranteed. However, Equation (27) works robustly even with a density ratio larger that 1000 and enables us to treat both compressible and incompressible fluids.

One typical example is the laser processing since the melting of metal and evaporation of liquid must be incorporated together. By the use of correct EOS, we are able to predict the ablation process after the heating by laser [10, 27]. Figure 10 shows the three-dimensional simulation of this process.

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

3. BODY-FITTED SOROBAN GRID FOR HIGHER ORDER SOLUTION

The direct application of the CIP to non-uniform grid system can achieve the third-order accuracy in time and space as well as in uniform grid system in one dimension. However, the application of the method to multi-dimensions is not possible in general. Therefore we used the co-ordinate-transformed CIP in the previous paper [28]. However, it is well known that the system becomes the lower order when the curvilinear co-ordinate is discretized even if the higher order scheme is used to describe the spatial derivative including metrics. 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 CIP method.

In the CIP, the time evolution of f at the grid point x_i is realized by transferring f at the upstream departure point given by $x_i - u\Delta t$. Retrieving the information at this departure point between two 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 [5]. There exists another choice to construct the polynomial by splitting scheme that

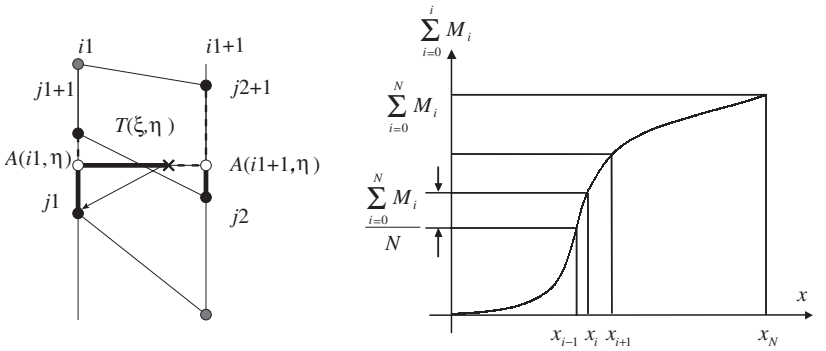

Figure 11. (Left) Schematics of the soroban grid. All the grid points move along the vertical lines. The number of grid points in each line can be different. (Right) Accumulated monitor function is divided into equal pieces. The x boundary of each piece gives the grid point.

uses the one-dimensional CIP sequentially in all directions [2]. In this paper, we use the latter method.

Let us consider a special mesh system shown in Figure 11, where the vertical mesh (y direction) is the straight line, while the grid points move along each line. Since this mesh system looks like an abacus, we call this system 'soroban' which is the Japanese name of abacus. In this system, the CIP is solved as follows. If the upstream departure point is given as $(\xi,\eta)=(x_i-u\Delta t, y_i-v\Delta t)$, at first the upstream grid x_{i1} 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 CIP. This means that f and $\partial_y f$ are transported as shown in Equation (4) along y direction. 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 CIP method in the x direction. In this step, A and $\partial_x A$ must be transported. However, in the first step, only $\partial_y A$ is obtained and we need some method to estimate $\partial_x A$ from $\partial_x f$.

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

3.1. Type-M CIP

In the previous paper [2], we proposed to use the first-order scheme 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 Reference [2] which we call the 'Type-M CIP' scheme, while the direct multi-dimensional CIP commonly used is called the 'Type-A CIP' [5].

Here we represent the cubic polynomial of the CIP procedure by

$$\begin{aligned} dd &= YG(j1+1) - YG(j1) \\ yy &= ynew - YG(j1) \\ a01 &= ((GY(j1) + GY(j1+1)) * dd + 2.0 * (F(j1) - F(j1+1)))/dd * *3 \\ a11 &= (3.0 * (F(j1+1) - F(j1)) - (GY(j1+1) + 2. * GY(j1)) * dd)/(dd * dd) \\ A(n) &= ((a01 * yy + a11) * yy + GY(j1)) * yy + F(j1) \\ AY(n) &= (3. * a01 * yy + 2. * a11) * yy + GY(j1) \\ AX(n) &= GX(j1) + yy/dd * (GX(j1+1) - GX(j1)) \end{aligned} \tag{28}$$

The same procedure is repeated also for $i1+1$ and $j2$. Then two points $A(i1+1,\eta) = A(2)$ and $A(i1,\eta) = A(1)$ are interpolated by the CIP method to get $T(\xi,\eta)$ as follows:

$$\begin{aligned} dd &= XG(iline+1) - XG(iline) \\ xx &= xnew - XG(iline) \\ a01 &= ((AX(1) + AX(2)) * dd + 2.0 * (A(1) - A(2)))/dd * *3 \\ a11 &= (3.0 * (A(2) - A(1)) - (AX(2) + 2. * AX(1)) * dd)/(dd * dd) \\ FN(igrid) &= ((a01 * xx + a11) * xx + AX(1)) * xx + A(1) \\ GXN(igrid) &= (3. * a01 * xx + 2. * a11) * xx + AX(1) \\ GYN(igrid) &= AY(1) + xx/dd * (AY(2) - AY(1)) \end{aligned} \tag{29}$$

where *igrid*, *iline* represent the grid and line indices shown in Figure 12, and $FN(igrid) \equiv f^{n+1}_{i1,j1}$, $GXN(igrid) \equiv \partial_x f^{n+1}_{i1,j1}$, $GYN(igrid) \equiv \partial_y f^{n+1}_{i1,j1}$.

Although the Type-M CIP is sufficient for many applications, a little more accurate scheme is possible at the price of memory requirement. This scheme is proposed by Aoki [29] and we call it 'Type-C CIP' hereafter. In this scheme, independent variables are now $f, \partial_x f, \partial_y f, \partial_{xy} f$ in two dimensions. In estimating $\partial_y A$, the CIP is applied to the advection of $(\partial_y f)$ and $\partial_x(\partial_y f)$.

3.2. Soroban grid and accuracy

It is very important to note that the number of grid points along each j line can be different. Corresponding 4 points $j1, j1+1, j2, j2+1$ including (ξ,η) point can be numerically searched. Most efficient method for finding the grid points in non-uniform mesh was invented by one of the authors [30]. By this method, the grid belonging to the upstream departure point is readily found with only one indexing procedure. Let us introduce one-dimensional array $IS(100000)$ which is designed so that IS keeps the grid number. As shown in Figure 12, IS denotes the fine grid with a constant grid spacing Δx, and $IS(ix) = 1$ if the fine grids belong to a grid cell between x_1, x_2 and $IS(ix) = 2$ if the fine grids belong to a grid cell between x_2, x_3, etc. Thus the grid that the point x belongs to can be found by at first taking $ix = \text{Integer}(x/\Delta x)$ and then resulting $IS(ix)$ means that x falls in between x_{IS}, x_{IS+1}.

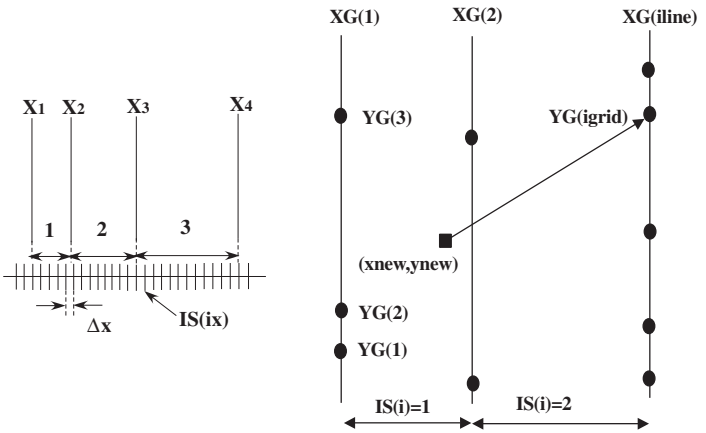

Figure 12. (Left) The method to find the grid number in non-uniformly spaced grid system. (Right) The grid arrangement and indexing. (*xnew*, *ynew*) is the upstream point and *YG*(*igrid*) is the present location of concern.The number of grid points along each line can be different. Even the line length can be different.

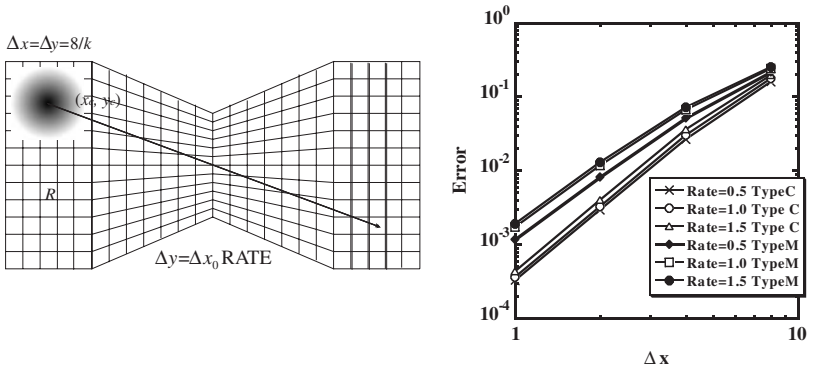

Figure 13. (Left) Grid arrangement and initial profile for benchmark test. (Right) The numerical error ε for different grid shape with Type-M and Type-C CIP.

For the benchmark test, we adopted a mesh system shown in Figure 13. This mesh is symmetrical at the centre 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 $\Delta y = RATE \times \Delta x_0$ at the centre. The benchmark test was done for three cases $RATE = 0.5, 1.0, 1.5$ by changing the mesh size as $\Delta x_0 = 8, 4, 2, 1$ for measuring accuracy. $RATE = 1.0$ corresponds to the uniform region and $RATE = 0.5$ gives a shape like Figure 13.

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

$$\begin{aligned}
&\Delta y = \Delta x_0, \quad i = 0, \ldots, 5k \\
&\Delta y = \Delta x_0 \times (1 + (RATE - 1) \times (i - 5k)/7k), \quad i = 5k, \ldots, 12k \\
&\Delta y = \Delta x_0 \times (1 + (RATE - 1) \times (19k - i)/7k), \quad i = 12k, \ldots, 19k \\
&\Delta y = \Delta x_0, \quad i = 19k, \ldots, 24k
\end{aligned} \tag{30}$$

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 centre.

The initial profile is

$$f(x, y) = \begin{cases} \dfrac{1 + \cos\left(\pi\sqrt{(x - x_c)^2 + (y - y_c)^2}/R\right)}{2}, & (x - x_c)^2 + (y - y_c)^2 < R^2 \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

and $R = 15, (x_c, y_c) = (16, 80)$, $u = 0.1$, $v = -0.04$, $\Delta t/\Delta x_0 = 0.2$ are used for the test run. The initial value for the derivatives are estimated by analytically differentiating Equation (31). At $t = 1600$, the initial profile moves to the opposite side of Figure 13.

The numerical error is shown in Figure 13. 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 the second-order.

3.3. Moving adaptive soroban grid

Finally, 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:

$$M(x, t) \equiv \left(1 + \alpha\left(\frac{\partial f}{\partial x}\right)^2\right)^{1/2} + \beta\left(\frac{\partial^2 f}{\partial x^2}\right)^2 \tag{32}$$

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 Figure 11. If we divide the accumulated function into equal pieces and get the x co-ordinate to get such integration

$$M_i = \int_{x_i}^{x_{i+1}} M(x, t)\,\mathrm{d}x = \frac{1}{N}\int_{x_1}^{x_N} M(x, t)\,\mathrm{d}x \quad (i = 1, \ldots, N) \tag{33}$$

then these points can be used as grid points.

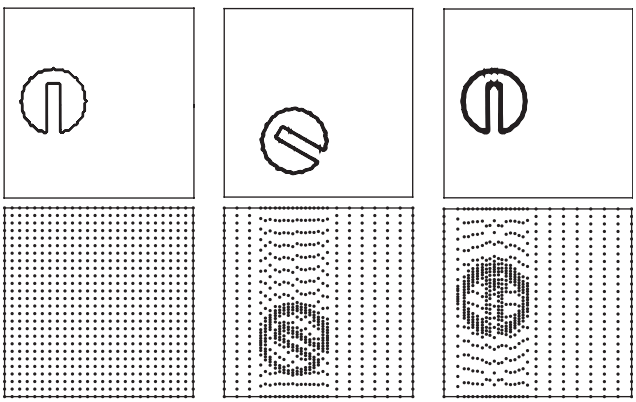

Figure 14. Solid body revolution. Density contour and grid points: (a) $t=0$; (b) $t=133$; and (c) $t=800$.

In the two-dimensional mesh shown in Figure 11, 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 [31]. The mesh of 101×101 is used and the rotation centre is located at $(x_c, y_c)=(50,50)$. The initial profile is

$$f(x,y)=\begin{cases}1, & R<17 \text{ and } (|x-26|>3 \text{ or } y>60)\\ 0 & \text{otherwise}\end{cases} \tag{34}$$

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.

$$\begin{aligned} v&=-2\pi(y-y_c)/800\\ v&=2\pi(x-x_c)/800\end{aligned} \tag{35}$$

Figure 14 shows the contour where increment is 0.1 and the corresponding grid points are depicted. All the figures are only one quarter of the whole figure. 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.

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 Figure 1. Most importantly, this mesh system can

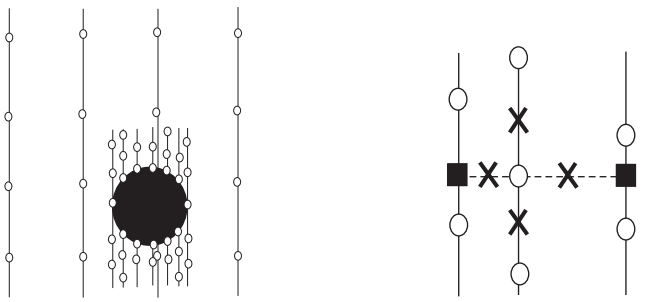

Figure 15. (Left) Coarse and fine grid systems are combined easily in soroban grids. (Right) The spatial gradient in the non-advection term can be estimated by the value denoted by $\times$.

also be used for adaptive mesh refinement as shown in Figure 15 (left). We have two grid system: coarse and fine grid. The latter is used to capture the complex shape. The fine mesh system can easily move and after movement we can reorganize the grid numbering and the resultant grid is the same as soroban grid discussed in this paper.

4. 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 with the indexing proposed in Reference [30]
- High accuracy.
- CFL can be large.

For non-advection part, we need $\nabla \cdot \mathbf{u}$ and ∇p at the grid point. The CIP interpolation can be used to generate the values at $\times$ points shown in Figure 15. Then simple finite difference scheme as in the Cartesian grid can be employed. The application to hydrodynamic system will be given elsewhere in near future.

REFERENCES

1. Takewaki H, Nishiguchi A, Yabe T. The cubic-interpolated pseudo-particle (CIP) method for solving hyperbolic-type equations. *Journal of Computational Physics* 1985; **61**:261.
2. Takewaki H, Yabe T. Cubic-interpolated pseudo particle (CIP) method—application to nonlinear or multi-dimensional problems. *Journal of Computational Physics* 1987; **70**:355.
3. Yabe T, Takei E. A new higher-order Godunov method for general hyperbolic equations. *Journal of the Physical Society of Japan* 1988; **57**:2598.
4. Yabe T, Aoki T. A universal solver for hyperbolic-equations by cubic-polynomial interpolation. I. One-dimensional solver. *Computer Physics Communications* 1991; **66**:219.
5. Yabe T, Ishikawa T, Wang PY, Aoki T, Kadota Y, Ikeda F. A universal solver for hyperbolic-equations by cubic-polynomial interpolation. II. Two- and three-dimensional solvers. *Computer Physics Communications* 1991; **66**:233.

6. Nakamura T, Yabe T. Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space. *Computer Physics Communications* 1999; **120**:122.
7. Utsumi T, Yabe T, Koga J, Aoki T, Ogata Y, Matsunaga E, Sekine M. A note on the basis set approach in the constrained interpolation profile method. *Journal of Computational Physics* 2004; **196**:1.
8. Yabe T, Mizoe H, Takizawa K, Moriki H, Im HN, Ogata Y. Higher-order schemes with CIP method and adaptive Soroban grid towards mesh-free scheme. *Journal of Computational Physics* 2004; **194**:57.
9. Leith CE. Numerical simulation of the Earth's atmosphere. In *Methods in Computational Physics*, Alder B *et al.* (eds), vol. 4. Academic Press: New York, 1965; 1–28.
10. Yabe T, Xiao F, Utsumi T. Constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics* 2001; **169**:556.
11. Colella P, Woodward PR. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics* 1984; **54**:174.
12. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201.
13. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluids and Dynamics*, Morton KW, Baines MJ (eds). 1982; 273.
14. Osher S, Sethian JA. Front propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12.
15. Unverdi SO, Tryggvasson GA. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25.
16. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**:83.
17. Yabe T, Xiao F. Description of complex and sharp interface during shock wave interaction with liquid drop. *Journal of the Physical Society of Japan* 1993; **62**:2537.
18. Yabe T, Xiao F. Description of complex and sharp interface with fixed grids in incompressible and compressible fluid. *Computers and Mathematics with Applications* 1995; **29**:15.
19. Yabe T *et al.* The next generation CIP as a conservative semi-Lagrangian solver for solid, liquid and gas. *Journal of Computational and Applied Mathematics* 2002; **149**:267.
20. Chino M, Takizawa K, Yabe T. Experimental research on rotating simmer. *Proceedings of FEDSM'03*, *4th ASME-JSME Joint Fluids Engineering Conference FEDSM2003-45171*, Honolulu, Hawaii, USA, 6–11 July 2003.
21. Clanet C, Hersen F, Bocquet L. Secrets of successful stone-skipping. *Nature* 2004; **427**:29.
22. Yabe T, Xiao F, Zhang Y. Strategy for unified solution of solid, liquid, gas and plasmas, *AIAA Paper No.99-3509*, *30th AIAA Fluid Dynamics Conference*, Norfolk, USA, 28 June–1 July 1999.
23. Harlow FH, Welch JE. Numerical calculation of time dependent viscous incompressible flow with free surface. *Physics of Fluids* 1965; **8**:2182.
24. Harlow FH, Amsden AA. Numerical simulation of almost incompressible flow. *Journal of Computational Physics* 1968; **3**:80.
25. Yabe T, Wang PW. Unified numerical procedure for compressible and incompressible fluid. *Journal of the Physical Society of Japan* 1991; **60**:2105.
26. Xiao F *et al.* An efficient model for driven flow and application to a gas circuit breaker. *Computer Modelling and Simulation Engineering* 1996; **1**:235.
27. Yabe T. Simulation of laser-induced melting and evaporation dynamics by the unified solver CIP for solid, liquid and gas. In *Mathematical Modeling of Weld Phenomena 4*, Cerjak H (ed.). The University Press: Cambridge, 1998; 26–36.
28. Wang PY, Yabe T, Aoki T. A general hyperbolic solver—the CIP method—applied to curvilinear coordinate. *Journal of the Physical Society of Japan* 1993; **62**:1865.
29. Aoki T. Multi-dimensional advection of CIP (cubic-interpolate propagation) scheme. *CFD Journal* 1995; **4**:279.
30. Nishiguchi A, Yabe T. Second order fluid particle scheme. *Journal of Computational Physics* 1983; **52**:390.
31. Zalesak ST. Fully multidimensional flux-corrected transport algorithm for fluids. *Journal of Computational Physics* 1979; **31**:335.
32. Berger MJ, Oliver J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**:484.

