PHYSICAL REVIEW E **67**, 045701(R) (2003)

Numerical method for a moving solid object in flows

Kensuke Yokoi
Computer and Information Division, RIKEN (The Institute of Physical and Chemical Research), Wako 351-0198, Japan
(Received 10 January 2003; published 29 April 2003)

We propose a numerical method for dealing with a moving solid body that interacts with a complex liquid surface. The method is based on the level set method, the CIP method, and the ghost fluid method. The validity of the method was shown by applying it to Poiseuille and Couette flow problems. The method can precisely capture the boundary layer as well as a moving solid object.

DOI: 10.1103/PhysRevE.67.045701 PACS number(s): 83.85.Pt, 02.70.Bf, 46.15.−x, 95.75.Pq

Fluid phenomena, in which there is an interaction between a moving solid object and fluid are often observed in daily life. Although numerical studies on these phenomena are important for various research fields, the numerical studies are difficult because of complicated treatments for a moving boundary and its boundary condition. There is a need for a numerical method that can treat interaction of a moving solid body with fluid without complicated treatments such as mesh generation, marker particles, and cut cells for moving boundaries.

Numerical simulation of interaction between a rigid body and a complex interfacial flow has been conducted by using the TFT (tangent function transfer) method as an interface-capturing method in a finite difference framework on a Cartesian fixed grid system [1,2]. As different type of methods, the immersed boundary method [3] and the Cartesin grid projection method [4] have been widely used for the problem. Xiao's method can automatically satisfy the continuity conditions across the interface by solving a Poisson equation of pressure over the entire computational domain, including the solid region. The force and torque for a solid object are calculated by summing the force and torque over all of the grid points in the region of the solid. Although the method is useful, there are some problems with interface treatments. Figure 1 shows snapshots of the numerical simulation by the TFT method. The flat interface becomes bumpy because of the influence of the mesh in a small number of grids as shown in Fig. 1 ($t=1.8$). This is because the TFT method captures the interface as a sharp discontinuity. Sometimes the uneven interface generates an unreal pressure field. Another problem is that an appropriate no-slip condition is not imposed on the solid body. In this Rapid Communication, we modify the method [1,2] to overcome these problems by incorporating the level set method [5–7] and the ghost fluid method [8,9].

The level set method is an interface-capturing method and has been applied to various moving boundary problems [6,10,11]. This method expresses the surface of an $N-1$ dimension as the zero level (or contour) of an N-dimensional level set function ψ. In general, the signed distance function

$$\psi=0 \text{ at the interface,}$$

$$|\nabla\psi|=1 \quad \text{for the whole region} \tag{1}$$

is widely used as the level set function. To track the interface in the velocity field $\mathbf{u}$, the advection equation

$$\frac{\partial\psi}{\partial t}+\mathbf{u}\cdot\nabla\psi=0 \tag{2}$$

is used. To maintain the property of the sign distance function, reinitialization is done by solving the following problem to a steady state,

$$\frac{\partial\psi}{\partial t_l}=S(\psi_0)(1-|\nabla\psi|), \tag{3}$$

where t_l is a fictitious time and S is a smoothed sign function $S(\psi_0)=\psi_0/\sqrt{\psi_0^2+\varepsilon^2}$. See Ref. [7] for more details.

An advantage of the level set method is that the method can automatically treat topological change in interfaces without special treatment. Smoothing of the interface must be carried out to precisely capture interfaces in a Cartesian grid. In the level set formulation, the thickness of the interface can be easily maintained. The unit normal is always well defined from the level set function

$$\mathbf{n}=\frac{\nabla\psi}{|\nabla\psi|}. \tag{4}$$

The unit normal is useful for detecting the position of the interface by using the distance function and for estimating the curvature, which is used to calculate the surface tension force [12]. These properties play an important role in captur-

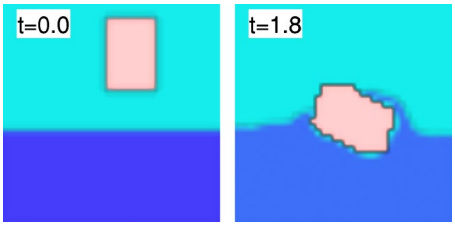

FIG. 1. Snapshots of the numerical simulation by the TFT method. The rectangular rigid body falls into a liquid surface. The left and right figures show the initial condition and the snapshot at $t=1.8$, respectively. The TFT method has been used only for the solid interface. For the liquid interface, the level set method has been used. 40×40 computational grids are used. The initial rectangular rigid body is represented by 8×12 meshes.

ing a flat interface in a Cartesian grid and in implementing the ghost fluid method for a boundary condition on the solid object.

To impose a boundary condition on a solid object, we use the ghost fluid method [8] or the zero-velocity fix technique [9]. The ghost fluid method is simple and effective for implementing a boundary condition on a solid body in finite difference framework. In this method, imaginary cells called the ghost cells are placed within a few grids from the interface in the solid region. The variables of fluid near the interface are extrapolated into the ghost cells to satisfy the boundary condition on the solid surface. We have used the ghost fluid method only for the calculation of velocity. The pressure boundary condition is computed by solving the pressure Poisson equation over all the computational domain, including the solid region like Xiao's work [1,2]. The zero-velocity fix technique to impose a no-slip condition for a stationary solid wall has been proposed. Therefore, we modified the method in order to be able to apply it to a moving solid object.

We use the following governing equations,

$$\nabla \cdot \mathbf{u} = \mathbf{0}, \tag{5}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \frac{\nabla \cdot \tau}{\rho} + \frac{\mathbf{f}}{\rho}, \tag{6}$$

where $\mathbf{u}$ is the velocity, p is the pressure, ρ is the density, τ is the viscous stress tensor, and $\mathbf{f}$ is the body force. The governing equations are solved by the CIP-CUP (cubic interpolated propagation combined unified procedure) method [1,2,13]. In this method, a fractional step approach on time is used. The governing equations are split into an advection step and a nonadvection step. The advection step and the nonadvection step are computed by the CIP method and the CUP method, respectively. The CIP method [16,17] is a less-diffusive and stable algorithm for solving the advection equation. The CUP method, which is based on a pressure Poisson equation such as the Mac method [14] and the projection method [4,15], is used for treating the multiphase flows.

To track the interface, we use a modified level set method [18] to avoid a reinitialization error that arises in the reinitialization procedure. In this method, a function ψ_{zero} that is not reinitialized is used to track the interface. The interface is tracked through the zero contour of ψ_{zero}, which is free from reinitialization errors. The time evolution of ψ_{zero} is calculated by advection equation (2). In this study, we use the CIP method [16,17] as an advection equation solver. Although the CIP method requires a relative large amount of computational memory, it achieves good performance. To capture the interface (i.e., to compute unit normal, smoothing, etc.), the level set function ψ is constructed from the zero-contour of ψ_{zero} by using the fast marching method [6,19] and the level set method [7]. A color (density) function ϕ that is used to define the physical properties for different materials can be generated from a smoothed Heaviside function. See Ref. [18] for details.

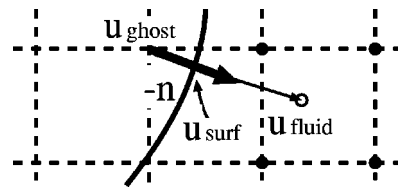

FIG. 2. Schematic figure of the ghost fluid method. The curve represents the solid interface. This is for $\psi_{ghost} > 0$. If $\psi_{ghost} < 0$, replace $-\mathbf{n}$ by $\mathbf{n}$.

To impose a no-slip condition on the solid interface, the velocity u_{ghost} on the ghost cells is estimated by a linear extrapolation using the level set function as follows:

$$\mathbf{u}_{ghost} = \frac{\psi_{ghost}}{\psi_{fluid}}\mathbf{u}_{fluid} + \left(1 - \frac{\psi_{ghost}}{\psi_{fluid}}\right)\mathbf{u}_{surf}, \tag{7}$$

here, u_{fluid} and u_{surf} are the velocity near the solid in the fluid and the velocity of the solid surface, respectively, as shown in Fig. 2. ψ_{ghost} is the level set function at the position where $\mathbf{u}_{ghost}$ is defined. From the definition of Eq. (7), the order of accuracy of the velocity boundary condition is of the first order. In this study, we used $\psi_{fluid} = \mp \Delta x$, where "$-$" sign is used for $\psi_{fluid} < 0$ and "$+$" is used for $\psi_{fluid} > 0$. $\mathbf{u}_{fluid}$ is derived by solving the following equation:

$$\frac{\partial \mathbf{u}}{\partial t_g} \pm \mathbf{n} \cdot \nabla \mathbf{u} = 0, \tag{8}$$

here "$+$" sign is for $\psi_{ghost} > 0$, "$-$" is for $\psi_{ghost} < 0$, and t_g is an artificial time, in this case, $\Delta t_g = |\psi_{ghost}| + |\psi_{fluid}|$.

We use a semi-Lagrangian approach to solve Eq. (8) because $\mathbf{n}\Delta t_g$ is usually more than Δx (i.e., CFL number > 1) as shown in Fig. 2. We use a first-order semi-Lagrangian scheme. Although we also tried a high-order semi-Lagrangian scheme based on the CIP method [20], there was almost no difference in the results. Therefore, the first-order scheme seems to be sufficient for the calculation of Eq. (8).

In the first-order semi-Lagrangian formulation, the interpolation function is constructed using the grid points surrounding the position where $\mathbf{u}_{fluid}$ is defined, marked by the black circles in Fig. 2. We use the following interpolation function:

$$F(x,y) = a_{11}XY + f_x X + f_y Y + f_{i',j'}, \tag{9}$$

$$X = x - x_{i',j'}, \quad Y = y - y_{i',j'},$$

$$f_x = \frac{f_{iup',j'} - f_{i',j'}}{\Delta x}, \quad f_y = \frac{f_{i',jup'} - f_{i',j'}}{\Delta y},$$

$$a_{11} = \frac{f_{iup',jup'} - f_{i',j'}}{\Delta x \Delta y} - \frac{f_x \Delta x + f_y \Delta y}{\Delta x \Delta y},$$

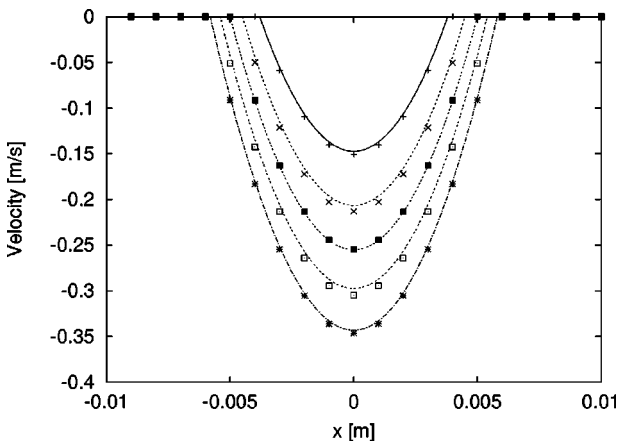

FIG. 3. Comparison between the numerical and theoretical velocity profiles for various wall widths. The dots and lines represent the numerical and theoretical solutions, respectively. The viscosity coefficient $\mu=4.9\times10^{-3}$ Pa s, $dp/dy=100$ Pa/m, and $\Delta x=\Delta y=1$ mm are used.

here $i'=i+\mathrm{int}(-n_x\Delta t_g/|\Delta x|)$, $iup'=i'+\mathrm{sgn}(-n_x\Delta t_g)$, $\Delta x=x_{iup',j'}-x_{i',j'}$, $\mathrm{int}(a)$ means the integer part of a. $\mathbf{u}_{fluid}$ can be computed from $F(x_{i,j}-n_x\Delta t_g, y_{i,j}-n_y\Delta t_g)$ or $F(x_{i',j'}-\mathrm{mod}(n_x\Delta t_g/|\Delta x|),\quad y_{i',j'}-\mathrm{mod}(n_y\Delta t_g/|\Delta y|)$, where $\mathrm{mod}(a/b)$ means the reminder of a/b.

To check the validity of our method, we used it for solving the two-dimensional Poiseuille and Couette flow problems.

The theoretical solution of the Poiseuille flow is

$$u_x=\frac{a^2}{2\mu}\left(-\frac{dp}{dy}\right)\left(1-\frac{x^2}{a^2}\right),$$

where a is a wall width. Figure 3 shows a comparison between the numerical results and theoretical solutions for various wall widths. The velocity profiles agree well with the analytical solutions even in a small number of grids.

To validate Eq. (7) for a moving wall and the influence of a rectangular grid for a circular shape, the method was applied for a Couette flow problem. The configuration is shown

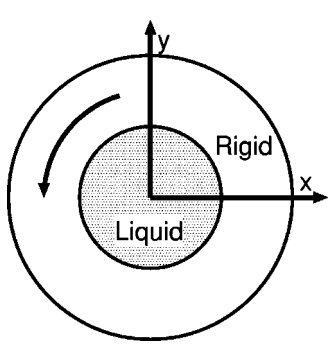

FIG. 4. Configuration of a Couette flow.

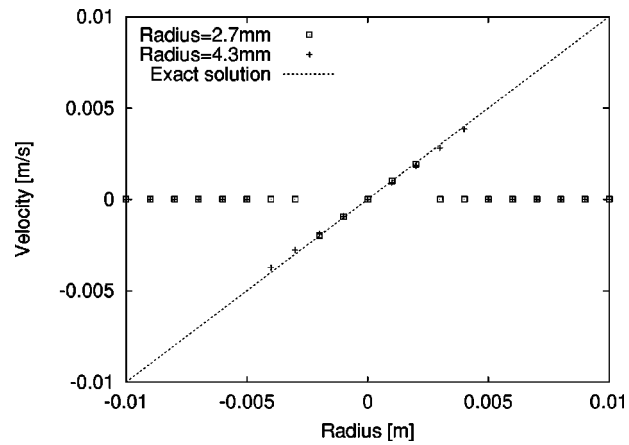

FIG. 5. The dots and lines represent the numerical and theoretical solutions, respectively. As the radii of the cylinder, 2.7 and 4.3 mm are used. The angular velocity $\omega_{cyl}=1.0$.

in Fig. 4. Liquid is put in a cylinder and the cylinder is rotated at a constant angular velocity ω_{cyl}. Then the theoretical solution of the velocity profile is $v(r)=\omega_{cyl}r_{cyl}$, here r_{cyl} is the radius of the cylinder. In this simulation, the shape of the cylinder is imposed through the level set function analytically $\psi(x,y)=\sqrt{x^2+y^2}-r_{cyl}$. Therefore, the shape is represented well in the grid system. The rotation of the rigid body is expressed through $\mathbf{u}_{surf}$ on the ghost cells, $u_{surf,x}=-\omega_{cyl}y_{cyl}=-\omega_{cyl}r_{cyl}n_y$ and $u_{surf,y}=\omega_{cyl}x_{cyl}=\omega_{cyl}r_{cyl}n_x$ here, $\psi_{ghost}>0$. If $\psi_{ghost}<0$, replace n_x and n_y with $-n_x$ and $-n_y$. Figure 5 shows a comparison between the numerical and theoretical solutions. The dots show the y velocity component on the x axis. The results show that Eq. (7) is appropriate and that the method is appropriate for treating a circular shape in a Cartesian grid.

Numerical convergence studies are performed for the Poiseuille problem of the wall width of 3.8 mm and the Couette problem of the cylinder of radius 4.3 mm. The error is defined as $Error=|u_{sim}-u_{exact}|/|u_{exact}|$, here u_{exact} and u_{sim} are the velocities of the exact solution and the simulation, respectively. These errors are declined linearly as shown in Table I. The results show that the method has first-order accuracy.

We applied the method to a fluid problem in which a rigid body interacts with a complex liquid interface in a gravitational field. The motion of a solid body suspended in a flow is determined by $\mathbf{u}_{solid}=\mathbf{u}_{center}+\mathbf{r}\times\omega$, where $\mathbf{u}_{center}$ is the

TABLE I. Results of the numerical convergence studies. $Error_p$ and $Error_c$ are the errors for the velocity at $x=0$ in the Poiseuille flow problem and at *radius* of about 4.0 mm in the Couette flow problem, respectively.

Grid spacing	1 mm	0.5 mm	0.25 mm
$Error_p$	2.16×10^{-2}	1.03×10^{-2}	4.67×10^{-3}
$Error_c$	3.75×10^{-2}	2.00×10^{-2}	1.23×10^{-2}

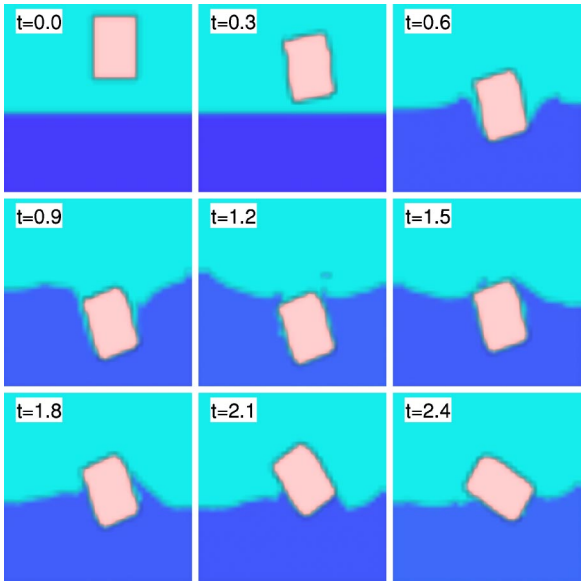

FIG. 6. Time series of the numerical simulation of interaction between the rigid body and the liquid surface. The density ratio of gas:liquid:solid is 1.25:1000:500. The movies are available on Ref. [21].

velocity at the mass center, $\mathbf{r}$ is the vector from the mass center, and ω is the angular velocity. The dynamics is determined by the equations

$$\frac{d}{dt}(M\mathbf{u}_{center})=\mathbf{F}=\int \rho \frac{d\mathbf{u}}{dt}\phi_s dV \tag{10}$$

and

$$\frac{d}{dt}(\mathbf{I}\cdot\omega)=\mathbf{T}=\int \left(\mathbf{r}\times\rho\frac{d\mathbf{u}}{dt}\right)\phi_s dV, \tag{11}$$

here M is the mass of the solid, $\mathbf{F}$ is the force for the solid body, ϕ_s is the color function of the solid, $\mathbf{I}$ is the tensor of inertia moment, and $\mathbf{T}$ is the torque. The force and the torque are calculated by a volume force formulation. See Refs. [1,2] for more details. In this calculation, $\mathbf{u}_{surf}$ of Eq. (7) is calculated from $\mathbf{u}_{surf}=\mathbf{u}_{center}+[\mathbf{r}+|\psi_{ghost}|(-\mathbf{n})]\times\omega$. The time series of the results are shown in Fig. 6. The rectangular rigid body is fixed on the liquid surface [21] initially and is released with a perturbation in angular speed. This simulation was done on 40×40 rectangular fixed grids. The initial rigid body is represented by 8×12 grids.

We have proposed a numerical method based on the level set method, CIP method, and the ghost fluid method for capturing a moving solid object that interacts with fluid in a Cartesian fixed grid. The validity of the method has been shown by the test problems of the Couette and Poiseuille flows. The results of simulation of a rigid falling into the liquid show that the method precisely captures a moving solid object and liquid surface in a small number of grids and that the model is robust.

I would like to thank Feng Xiao and Hao Liu for their useful discussions and comments. Numerical computation for this work was carried out at the Computer and Information Division, RIKEN and the Yukawa Institute for Theoretical Physics, Kyoto University. I am grateful to RIKEN for financial support and the Basic Science Grant.

[1] F. Xiao *et al.*, Comput. Phys. Commun. **102**, 147 (1997).
[2] F. Xiao, J. Comput. Phys. **155**, 348 (1999).
[3] C.S. Peskin, J. Comput. Phys. **25**, 220 (1977).
[4] A.S. Almgren *et al.*, SIAM J. Sci. Comput. (USA) **18**, 1289 (1997).
[5] S. Osher and J.A. Sethian, J. Comput. Phys. **79**, 12 (1988).
[6] J.A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge University Press, Cambridge, 1999).
[7] M. Sussman, P. Smereka, and S. Osher, J. Comput. Phys. **114**, 146 (1994).
[8] R. Fedkiw *et al.*, J. Comput. Phys. **152**, 457 (1999).
[9] M. Watanabe, R. Kikinis, and C.F. Westin, *Lecture Notes in Computer Science 2489* (Springer, Berlin, 2002), p. 405.
[10] K. Yokoi and F. Xiao, Phys. Rev. E **61**, R1016 (2000).
[11] K. Yokoi and F. Xiao, Physica D **161**, 202 (2002).
[12] J.U. Brackbill, D.B. Kothe, and C. Zemach, J. Comput. Phys. **100**, 335 (1992).
[13] T. Yabe and P.Y. Wang, J. Phys. Soc. Jpn. **60**, 2105 (1991).
[14] F.H. Harlow and J.E. Welch, Phys. Fluids **8**, 2182 (1965).
[15] A.J. Chorin, J. Math. Comput. **22**, 745 (1968).
[16] T. Yabe *et al.*, Comput. Phys. Commun. **66**, 233 (1991).
[17] T. Yabe, F. Xiao, and T. Utsumi, J. Comput. Phys. **169**, 2 (2001).
[18] K. Yokoi, Phys. Rev. E **65**, 055701(R) (2002).
[19] D. Adalsteinsson and J.A. Sethian, J. Comput. Phys. **148**, 2 (1999).
[20] K. Sakurai, T. Aoki, and T. Yabe, CFD Journal, **10**, 76 (2001).
[21] http://www.es.que.jp/kensuke/