Extension of the unified interpolation stencil for immersed boundary method for moving boundary problems

Hosnieh KOR*, Mehdi BADRI GHOMIZAD* and Koji FUKAGATA*

* Department of Mechanical Engineering, Keio University
Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan
E-mail: fukagata@mech.keio.ac.jp

Received: 4 April 2018; Revised: 31 May 2018; Accepted: 11 July 2018

Abstract

As an extension of the immersed boundary method with unified interpolation stencil (Kor, Badri Ghomizad, and Fukagata, J. Fluid Sci. Technol., Vol. 12, 2017, JFST0011), we propose an immersed boundary method that can handle moving boundary problems with a lower level of spurious force oscillation. The key modification to the previously proposed method, which was validated for fixed boundary problems, is to adopt the reconstruction method, in which the velocities outside the body are reconstructed, instead of the ghost-cell method, in which the velocities inside the body are set to satisfy the boundary conditions. From the comparison between the ghost-cell and reconstruction methods, both methods work equally well for a fixed boundary problem, but the reconstruction method is found to be effective in suppressing the spurious force oscillations that appear in moving boundary problems. The capability of the proposed method is demonstrated by numerical investigations of some typical problems. Both predefined motions, such an oscillating cylinder and a hovering flat plate, and interacting motions of rigid bodies are simulated to validate the method. For the latter, sedimentation of a single cylinder as well as a group of interacting cylinders under the gravitational force is examined to demonstrate the capability of the present method for fluid-structure interaction problems. The results show that the proposed method can properly handle the moving boundary problems, while preserving the simplicity of the unified interpolation stencil.

Keywords: Immersed boundary method, Ghost-cell method, Reconstruction method, Force oscillations, Moving boundary

1. Introduction

A panoply of fluid dynamics problems with geometrically complex boundaries, which may be fixed or moving, play a pivotal role not only in academic research but also in industrial applications including fluid-structure interactions phenomena and multi-phase flows, among others. A classical approach for simulating such flows is employing a body-fitted mesh. However, the prohibitive computational cost and memory requirements of these methods in addition to the overhead for re-meshing strategies in the case of moving boundaries have motivated researchers to adopt alternative means such as Cartesian methods.

Among the broad range of Cartesian methods, the immersed boundary method is a well-developed one, which efficaciously uses a Cartesian underlying grid for the Eulerian description of fluid flow as well as a set of markers to represent complex boundaries in the Lagrangian framework along with an appropriate interpolation technique to transfer data between these two datasets. The immersed boundary method was originally introduced by Peskin (1972). In Peskin’s method, also called continuous forcing method, some singular forces are added to the continuous form of momentum equation to satisfy the Dirichlet boundary conditions for velocity on the surface of the elastic bodies. Calculated from constitutive laws (e.g., Hooke’s law), these singular forces are described with discrete delta functions that smear the forcing effects over the neighboring Eulerian grid points. Although continuous forcing method well succeeded in solving problems with moving boundaries, it was somehow limited to flexible bodies and low Reynolds number problems. To ameliorate this situation, Goldstein et al. (1993) introduced the virtual boundary method utilizing a feedback approach.
to obtain the necessary forces. This method and its descendants proposed by Saiki and Biringen (1996) and Lai and Peskin (2000) were able to manage problems with rigid bodies. However, they were neither effective enough in capturing sharp boundaries nor were capable of fully suppressing the so-called spurious oscillations related to the moving boundary problems.

To sharply enforce the boundary conditions, Mohd-Yusof (1997) proposed an alternative method in which the boundary force is estimated by the momentum equation. Later, Fadlun et al. (2000) extended the method to a finite difference formulation on the staggered mesh. Mohd-Yusof’s method (Mohd-Yusof, 1997), also known as a discrete forcing method, was further developed in primitive and non-primitive (velocity-vorticity) forms of the Navier-Stokes equations by different research groups, i.e., Kim et al. (2001), Balaras (2004), Gilmanov and Sotiropoulos (2005), Kim et al. (2006), Zhang and Zheng (2007), Choi et al. (2007), Wang et al. (2009), and Soltani and Badri (2016), to name a few. In another point of view, discrete forcing method was extended by Tseng and Ferziger (2003) and Ghias et al. (2007) using alternative approach called the ghost-cell method. In this approach, the forcing point is considered inside the solid boundary and the proper value is extrapolated from the fluid domain. Later, Mittal et al. (2008) enhanced the ghost-cell method by introducing the concept of the image point and constructing the interpolation operators in a direction normal to the immersed boundary, which enabled the method to deal with the Neumann boundary conditions. However, another issue that emerges by using the image point entails finding an intercept point on the boundary. Mittal et al. (2008) adopted an algorithm to find the best intercept point (among all possibilities) in peculiar geometrical situations. It was, nevertheless, a complicated procedure due to the many unexpected geometrical circumstances with which they encountered. To address this problem, Berthelsen and Faltings (2008) developed a versatile ghost-cell method capable to handle highly irregular boundaries; however, the large stencil size (included seven grid points in each direction) practically prohibits the moving boundary implementation.

In addition to the interpolation accuracy and its flexibility, the spurious force oscillations (SFOs) should also be taken into account when discrete forcing methods are applied moving boundary problems (Uhlmann, 2005; Yang and Balaras, 2006). It has been reported that despite the adequate accuracy of the primary and secondary variables computed on immersed boundary, an intolerable amount of oscillations are observed in the surface stresses and thus in the non-dimensional forces, particularly in the drag force. Whereas some researchers reported this issue and suggested some remedies for it (Uhlmann, 2005; Yang and Balaras, 2006; Fadlun et al., 2000), it seems that the paper by Lee et al. (2011) was the first work which thoroughly examined and revealed the origin of SFOs in the discrete forcing methods. They reported that the main source of SFOs is the discontinuity in the pressure field near the boundary, which is enhanced by various descriptions of grid points at different time steps. When a boundary moves in the fluid region, some of the ghost points at the present time step may, at the next time step, turn into the fluid points used in the interpolation stencils for the other ghost points. Similarly, some fluid points may become ghost points at the next time step. These frequent changes in the definition leads to non-physical oscillations near the boundary. Although it is not possible to rigorously derive the magnitude of SFOs (since it is ultimately determined by the modification of pressure field, which is a non-local quantity), under some assumptions the magnitude of SFOs can be estimated as $O(\Delta x^2/\Delta t)$ (Lee et al., 2011; Seo and Mittal, 2011), and therefore it can be reduced by decreasing the grid size ($\Delta x$) and increasing the time-step ($\Delta t$). However, decreasing the grid size is not a preferable choice since it considerably drives up the computational cost.

To control the impact of SFOs, some techniques have been proposed in the past studies, which can be categorized in two main classes. In the first group, the immersed boundary condition in the momentum equation is treated so as to reduce the SFOs. Kajishima and Takiguchi (2002) used the solid volume fraction to stay away from SFOs. Kim and Choi (2006) eliminated SFOs for a single object by using the referential link to the solid. Luo et al. (2012) reduced SFOs by an implicit hybridization, and Chiu et al. (2010) applied a differential interpolation instead of algebraic one to tackle the problem. In second category, the immersed boundary conditions in the mass balance equation is also considered. Adding mass source is one of techniques in this class, which was utilized by Kim et al. (2001) and Lee et al. (2011) to minimize the pressure fluctuations arising from pressure discontinuity. Another means to properly cope with the mass balance, introduced by Seo and Mittal (2011), is to combine the direct forcing and the cut-cell method.

Here we present a unified interpolation approach within an immersed boundary framework — based on what we previously implemented for the fixed boundary problems with geometrically complex boundaries (Kor et al., 2017) — to study moving boundary problems. To seek for the possibility of reducing SFOs, we first make a comparison between the ghost-cell method and reconstruction-based direct forcing, which reveals the superiority of reconstruction method in suppressing SFOs. Then, we opt for the reconstruction approach combined with the unified interpolation stencil (Kor et al., 2017) as the basis of our immersed boundary method and validate the proposed method by some typical test cases in...
the moving boundary literature. Finally, the proposed immersed boundary method is applied to simulate a set of cylinders falling under the gravity force. This sedimentation process has been a highly challenging problem for moving immersed boundary methods (Kadapa et al., 2017) and the numerical results demonstrate the ability of the proposed method for real world problems and industrial applications.

2. Mathematical Formulation

In this section, a mathematical formulation of the direct forcing immersed boundary method for the incompressible fluid flows is presented. After adopting the unified interpolation stencil (Kor et al., 2017), two different interpolation strategies, namely, ghost-cell and fluid reconstruction, are implemented.

2.1. Governing Equations

To simulate a fluid flow interacting with a rigid obstacle, two sets of partitioned equations for each continuum field are considered. The mutual influence between them is taken into account by the stress boundary conditions.

2.1.1. Fluid Part

The fluid flow is governed by the incompressible Navier-Stokes equations. Figure 1 depicts the schematic configuration of the problem and some used nomenclatures. The whole domain is denoted by $\Omega$, being divided into two separated domains: $\Omega_F$ and $\Omega_S$ for the fluid region and the solid part, respectively. The boundary between these two domains is marked by $\Gamma$. The non-dimensional unsteady incompressible Navier-Stokes and continuity equations for constant viscosity fluid are given as

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{uu}) + \nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{f} \quad \text{in } \Omega_F$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_F$$

$$\mathbf{u} = \mathbf{u}_B \quad \text{on } \Gamma,$$

where $\mathbf{u}$ and $p$ are the velocity vector and the pressure, respectively, and $Re$ and $\mathbf{u}_B$ denote the Reynolds number and velocity on the boundary, respectively. All quantities without a superscript of # are made dimensionless by the fluid density $\rho^#$ as well as a representative length and a representative velocity, which will be defined later for each problem. When the quantities should be explicitly expressed in the dimensional form, they are indicated by a superscript of #. In this paper, the body force $\mathbf{f}$ represents solely the forces resulted from the presence of solid obstacles, which are imposed in the vicinity of the boundary, and the other body forces are not considered.

2.1.2. Rigid Body Part

The motion of a rigid body is governed by

$$\frac{d\mathbf{X}}{dt} = \mathbf{U},$$

$$m \frac{d\mathbf{U}}{dt} = \mathbf{F},$$

$$I \frac{d\omega}{dt} = \mathbf{T},$$

where $m$ and $I$ are the mass and moment of inertia of the body; $\mathbf{X}$, $\mathbf{U}$, and $\omega$ are the center of gravity of the body, its transnational velocity, and the angular velocity; $\mathbf{F}$ and $\mathbf{T}$ are the total force and torque acting on the body, respectively.
When the gravitational force is of interest, such as sedimentation of circular cylinders exemplified in the present study, the total force $F$ consists of the gravitational and buoyancy forces and the hydrodynamic force $F_h$, i.e.,

$$ F = \left(1 - \frac{\rho}{\rho_s}\right)mg + F_h, $$  \hspace{1cm} (5)

where $g$ denotes the gravitational acceleration, and $\rho$ and $\rho_s$ are densities of the fluid and the rigid body, respectively. The hydrodynamic force $F_h$ and torque $T$ are expressed as

$$ F_h = -\sum_k f_k \Delta s_k, $$  \hspace{1cm} (6)

$$ T = -\sum_k (X_k - X) \times f_k \Delta s_k, $$  \hspace{1cm} (7)

where $f_k$ denotes the boundary force acting on the fluid at the $k$-th Lagrangian boundary point, $X_k$, which represents the piecewise length of $\Delta s_k$ around it.

2.2. Numerical Schemes

To numerically solve Eq. (1), we employ the finite difference method on a Cartesian staggered mesh. The spatial derivatives are approximated by the energy-conservative second-order central difference scheme (Kajishima et al., 1998). The temporal integration is done by the low-storage third-order Runge-Kutta scheme (see, e.g., Spalart et al., 1991) for the advection term and the Crank-Nicolson scheme for the diffusion term. The velocity-pressure coupling is done similarly to the SMAC method (Amsden and Harlow, 1970) at each Runge-Kutta substep. Further, the direct forcing method is employed to enforce the desired boundary conditions. The discretized form of the algorithms at substep $l$ of the Runge-Kutta integration is summarized as follow.

Step 1: The first provisional velocity $u^*$ is obtained, i.e.,

$$ u^* = u^l + \Delta u, $$  \hspace{1cm} (8)

where $\Delta u$ is obtained by solving (see, e.g., Verzicco and Orlandi, 1996):

$$ \left(I - \frac{\Delta t}{2 Re} L\right) \delta u^* = \Delta t \left[\gamma^l A^l + \zeta^l A^{l-1} - \alpha^l G p^l + \alpha^l \frac{1}{Re} Lu^l\right]. $$  \hspace{1cm} (9)

Here, $I$ is the unit diagonal matrix and $L$ denotes the discrete Laplacian operator. The coefficients $\gamma^l$, $\zeta^l$, and $\alpha^l$ are the Runge-Kutta integration coefficients at substep $l$. Moreover, $A$ represents the discretized advection term and $G$ is the discretized gradient operator.

Step 2: The force $f$ due to the boundary is calculated via direct forcing method, i.e.,

$$ f = u_m - u^* \frac{1}{\alpha^l \Delta t}, $$  \hspace{1cm} (10)

where $u_m$ represents the velocity on the forcing point, which will be discussed in the next section, and the second provisional velocity $u^{**}$ is obtained by

$$ u^{**} = u^* + \alpha^l \Delta t \ f. $$  \hspace{1cm} (11)

Step 3: The second provisional velocity is corrected to be divergence-free, i.e.,

$$ u^{l+1} = u^{**} - \alpha \Delta t \ G \phi, $$

$$ p^{l+1} = p^l + \phi, $$

where the pressure correction $\phi = p^{l+1} - p^l$ is obtained by solving the pressure Poisson equation:

$$ L \phi = \frac{1}{\alpha^l \Delta t} D u^{**}, $$  \hspace{1cm} (13)

with $D$ being the discretized divergence operator.

It should be noted that the velocity and pressure after the last Runge-Kutta substep is taken as the velocity and pressure at the next time step, $n + 1$. 

[DOI: 10.1299/jfst.2018jfst0008] © 2018 The Japan Society of Mechanical Engineers
2.3. Forcing Strategy

In order to meet the required boundary conditions, we have to appropriately calculate the force in Eq. (10). This force is simply the Newtonian acceleration of the fluid on any node near the solid body. As mentioned in Step 2 above, the force is proportional to the difference of the first provisional velocity, \( \mathbf{u}^* \), and the velocity interpolated onto the nodes adjacent to the solid boundary, \( \mathbf{u}_m \). Therefore, two issues should be addressed here: on which grid point the force should be exerted and how the velocity should be interpolated. In this paper, we have investigated two different strategies introduced below.

As a common prerequisite, the boundary shape is defined by using the predefined boundary points (i.e., purple points \( P \) in Fig. 2). With a spline approximation using three neighboring points, we can compute the outward unit normal vector, \( \mathbf{n} \), and tangential vector, \( \mathbf{t} \), at any point on the boundary, i.e., even in-between these predefined boundary points. For every ghost point \( G \) or forcing point \( F \), a boundary point \( B \) used in the following computation can be newly set on the line normal to the boundary so that \((x_B - x_G) \cdot \mathbf{t} = 0\) or \((x_F - x_B) \cdot \mathbf{t} = 0\), where \(x_B, x_G,\) and \(x_F\) are the locations of the newly set boundary point \( B \), the corresponding ghost point \( G \) for ghost-cell method, and forcing point \( F \) for the reconstruction method, respectively.

Note that the grid lines as shown in Fig. 2 are different for \( u \) and \( v \) components on the staggered grid system, and in our simulation we perform interpolation separately for each component; but for simplicity, we explain here using a single set of grid lines, and the computational point where the velocity is defined is referred to as the grid node.

2.3.1. Ghost-Cell Method

In ghost-cell method, the forcing point is inside the solid domain on the so-called ghost point and the interpolation is undertaken along the normal direction through the surface. As illustrated in Fig. 2(a), a ghost point (i.e., point \( G \) in the figure) is defined as a grid point inside \( \Omega_S \) which has at least one neighbor point in \( \Omega_F \). Besides, as a direct extrapolation tends to produce large values on the ghost point close to the physical boundary, Mittal et al. (2008) put forward the concept of image point (point \( I \)), which is defined as the reflection of the ghost point with respect to the corresponding physical boundary into the fluid domain. After finding the velocity on image point, \( \mathbf{u}_I \), it is simply reflected back to compute the velocity on the ghost point, \( \mathbf{u}_G (= \mathbf{u}_m) \), i.e.,

\[
\mathbf{u}_G = 2\mathbf{u}_B - \mathbf{u}_I. \tag{14}
\]

The velocity on the image point, \( \mathbf{u}_I \), is computed here using the unified interpolation stencil (Kor et al., 2017). Namely, we consider one auxiliary point inside \( \Omega_F \) along the normal line (point \( A \)) so that a linear interpolation from the two neighboring grid points can be used to obtain the value at this auxiliary point \( A \). The details of this procedure can be found in Kor et al. (2017); the procedure for a ghost point \( G \) and the corresponding boundary point \( B \) is summarized as follows.

1) The image point \( I \) is defined by mirror-reflecting the ghost point, i.e., \( x_I = x_B + \delta \mathbf{n} \). The unit normal vector \( \mathbf{n} \) has already been computed as described above, and \( \delta = (x_B - x_G) \cdot \mathbf{n} \).

2) The first outer intersection between the normal line and grid lines (point \( A \)) is identified, and the value at point \( A \) (i.e., \( \mathbf{u}_A \)) is computed by a linear interpolation from the two neighboring grid nodes on the same grid line. For instance, in the situation exemplified in Fig. 2(a), the velocity is interpolated from the left and right neighbors to point \( A \) along the horizontal grid line.
3. Numerical Experiments

Before studying moving boundary problems, we examine the accuracy of the present reconstruction method for a circular cylinder in uniform flow at $Re = U^\infty_0 D/\nu = 100$ (where $\nu$ is the kinematic viscosity), as an example of fixed boundary problems. The quantities are non-dimensionalized by the cylinder diameter, $D^\infty$, and the uniform velocity imposed at the inlet, $U^\infty_0$. Following Kim et al. (2001) and also our previous study using the ghost-cell method (Kor et al., 2017), the size of computational domain is set to be 70$D$ in the streamwise ($x$) direction and 100$D$ in the transverse ($y$) direction. The center of the cylinder is located at 30$D$ downstream of the inlet. The computational grid is uniform in $x$ direction and nonuniform in $y$ direction. The numbers of the computational cells are 2048 in $x$ direction and 500 in $y$ direction, whereof about 30 grid points are uniformly distributed within the cylinder diameter in every direction. A convective boundary condition is imposed at the outlet of the computational domain.

Table 1 summarizes the time-averaged value of drag coefficient, $C_D$, the root-mean-square (rms) value of lift coefficient, $C_L_{rms}$, and the Strouhal number, St, computed using the present reconstruction method, together with the values reported in literature. The drag and lift coefficients are defined here as $C_D = 2F_D/\rho U^\infty_0 D^2$ and $C_L = 2F_L/\rho U^\infty_0 D^2$, where $F_D$ and $F_L$ are the drag and lift forces, respectively. The Strouhal number is defined as $St = f_{shed} D^2/U^\infty_0$, where $f_{shed}$ is the frequency of vortex shedding. Although there are some differences in the values, $C_D$ and $C_L_{rms}$ obtained using the present reconstruction method is within the range of values reported in literature. To be more detail, $C_D$ computed using the reconstruction method is closer to those computed using boundary fitted coordinates (Park et al., 1998; Naito et al., 2017), the size of computational domain is set to be 70$D$ in the streamwise ($x$) direction and 100$D$ in the transverse ($y$) direction. The center of the cylinder is located at 30$D$ downstream of the inlet. The computational grid is uniform in $x$ direction and nonuniform in $y$ direction. The numbers of the computational cells are 2048 in $x$ direction and 500 in $y$ direction, whereof about 30 grid points are uniformly distributed within the cylinder diameter in every direction. A convective boundary condition is imposed at the outlet of the computational domain.

Table 1 Comparison of the results on the flow around a circular cylinder at Re = 100.

| Method                  | $C_D$  | $C_L_{rms}$ | St    |
|-------------------------|--------|-------------|-------|
| Present reconstruction  | 1.37   | 0.30        | 0.164 |
| Kor et al. (2017)       | 1.43   | 0.29        | 0.160 |
| Tseng and Ferziger (2003)| 1.42   | 0.29        | 0.164 |
| Lai and Peskin (2000)   | 1.44   | 0.32        | 0.165 |
| Kim et al. (2001)       | 1.33   | 0.32        | 0.165 |
| Park et al. (1998)      | 1.33   | 0.33        | 0.165 |
| Naito and Fukagata (2012)| 1.33   | 0.23        | 0.165 |
| Norberg (2003)          | -      | 0.23        | 0.164 |

[DOI: 10.1299/jfst.2018jfst0008] © 2018 The Japan Society of Mechanical Engineers
and Fukagata, 2012); in contrast, $C_{L,max}$ computed using the present ghost-cell method (Kor et al., 2017) is closer to those computed using boundary fitted coordinates and the empirical correlation by Norberg (2003). Therefore, it is fair to state that there is no clear superiority or inferiority between these two methods as far as the fixed boundary problem is concerned.

In the rest of this section, we study different types of moving boundary problems. We begin by illustrating the accuracy of the ghost-cell and reconstruction methods with a predefined moving boundary test case. For all simulations, the Courant number is kept smaller than 0.5.

### 3.1. In-line Oscillations of a Circular Cylinder

For the first test case, a periodic oscillation of a circular cylinder in fluid at rest is considered, which has a wide range of practical engineering applications such as marine technology. For the present study, the motion of the cylinder is defined as

$$X = -A \sin(2\pi f t) e_x,$$

where $X$ is the center of the cylinder, and $A = A^\# / D^\#$ and $f = f^\# / U_{\text{max}}^\#$ denote the amplitude and frequency of the oscillation, respectively. All quantities are made dimensionless by the cylinder diameter $D^\#$ and the maximum velocity of the oscillation $U_{\text{max}}^\# = 2\pi f^\# A^\#$ such that non-dimensional diameter and maximum velocity become $D = 1$ and $U_{\text{max}} = 1$, respectively. The Reynolds number is defined as $Re = U_{\text{max}}^\#/D^\# \nu$. The non-dimensional parameter for oscillation frequency, called the Keulegan-Carpenter number, is defined as $KC = U_{\text{max}}^\#/f^\# D^\#$ = $1/f$. Small Keulegan-Carpenter number indicates that the inertia is dominant and a two-dimensional simulation makes sense.

Figure 3 shows the configuration of the moving cylinder. In order to compare our results with the numerical and experimental data obtained by Dütsch et al. (1998), the computation is carried out at $Re = 100$ and $KC = 5$; namely, $A = KC / (2\pi) = 0.8$ and $f = 1/KC = 0.2$ in Eq. (16). The size of the computational domain is $40D \times 40D$. The Neumann boundary conditions ($\partial u / \partial n = 0$) are implemented to the all outer boundaries. As mentioned, direct forcing methods are generally suffered from the numerical oscillations in moving boundary problems. Here, we examine the ghost-cell and reconstruction approaches at different grid resolutions to study their effects in suppressing the force oscillations. The number of the computational cells is 500 in $y$ direction (nonuniform), while we examine two different resolutions in $x$ direction (uniform), i.e., 1024 cells ($\Delta x = 0.04$) and 2048 cells ($\Delta x = 0.02$).

Figure 4(a) shows a time trace of the drag coefficient, $C_D = 2F_D^\#/ \rho U_{\text{max}}^2 D^\#$, computed using the ghost-cell method with $\Delta x = 0.04$. As seen, the drag coefficient is significantly affected by the oscillations. In another attempt, the flow is solved with a finer mesh, $\Delta x = 0.02$, as shown in Fig 4(b). Obviously, the amplitude of the oscillation for finer grid is lower than what it is obtained with coarser one.

Figure 5 shows the evolution of the drag coefficient when the reconstruction method is used. The oscillations in the drag coefficient are substantially suppressed. Figure 6 compares the temporal variations of pressure at a computational point near the boundary between the ghost-cell and reconstruction methods. The selected point belongs the fluid region for a certain time period, while it is inside the solid region for the rest. It is clearly seen that the oscillation is significantly suppressed.
Fig. 4 Drag coefficient computed using the ghost-cell method: (a) $\Delta x = 0.04$; (b) $\Delta x = 0.02$.

Fig. 5 Drag coefficient computed using the reconstruction method with second-order interpolation: (a) $\Delta x = 0.04$; (b) $\Delta x = 0.02$.

Fig. 6 Comparison of time traces of the pressure at a computational point close to the boundary, $(x, y) = (-0.492, 0.008)$, computed using the ghost-cell and reconstruction methods.

lower in the reconstruction method than in the ghost-cell method. The oscillation in the ghost-cell method is particularly strong in the time period of $5 < t < 7.5$ when the selected point is located inside the solid region.

Figure 7 compares the velocity profiles obtained by the present reconstruction method with the experimental results by Dütsch et al. (1998) at four streamwise locations, $x/D = -0.6, 0, 0.6$, and 1.2, and for three different phase angles, $180^\circ, 210^\circ$, and $330^\circ$. As seen in the figure, the present results are also quantitatively in good agreement with those of the
reference. Based on the results above, in what follows, we adopt the reconstruction approach.

3.2. Flow Induced by a Hovering Flat Plate

As the second problem with prescribed motion, we consider a rigid flat plate undergoing a sinusoidal translational and rotational motion, described by

\[ X(t) = \frac{A}{2} \cos(2\pi ft)x, \]
\[ \theta(t) = \frac{\pi}{2} + \theta_m \sin(2\pi ft), \]

where \( X(t) \) is the center of gravity of the plate and \( \theta(t) \) is the angle between the plate and horizontal axis. The non-dimensionalization is based on the chord length, \( c^* \), and the maximum velocity, \( \frac{U_{max}^*}{\nu^*} = \pi A^* f^* \). The non-dimensional translational amplitude, \( A^* \), the rotational amplitude, \( \theta_m^* \), and the frequency, \( f^* \), are set at \( A = 2.8 \), \( \theta_m = \pi/4 \), and \( f = 0.25 \), to compare with literature. The Reynolds number is \( \text{Re} = \frac{U_{max}^* c^*}{\nu^*} = 75 \).

We perform the simulation in a square domain of \( 20c \times 20c \). The numbers of the computational cells are 2048 in \( x \) direction (uniform, \( \Delta x = 0.0098 \)) and 500 in \( y \) direction (nonuniform, \( \Delta y = 0.0098 \) around the plate) for the base case, while twice finer and coarser grid resolutions in both directions (\( \Delta x = \Delta y = 0.0049 \) and \( \Delta x = \Delta y = 0.0195 \) around the plate) are also examined for verification. A no-slip boundary condition for the flat plate and a zero normal derivative for the velocity at the outer boundaries are implemented.

In order to implement the boundary conditions, we set the forcing points on both sides of the plate, as schematically shown in Fig. 8. Since the distance between the end point and the closest forcing point should influence the accuracy, sufficiently fine grid should be used for these points to be close enough. A moving thin object is usually difficult to handle using discrete forcing immersed boundary methods or ghost-cell methods, while it has been successfully simulated using continuous forcing methods (see, e.g., Park and Sung, 2018). However, the present reconstruction method has a merit that it can directly be applied to this kind of problem in contrast to the ghost-cell method requiring ghost points inside the body.
Figure 9 depicts the vorticity field around the flat plate from $t/T = 2$ to 4, where $T = 1/f = 4$ is the period of oscillation. Comparison of the instantaneous vorticity with the results obtained by Eldredge (2007) and Luo et al. (2012) shows that our simulation properly captures the flow features. Note that Eldredge (2007) considered an elliptic thin airfoil with the aspect ratio of 10, and Luo et al. (2012) considered the thickness 7% of the chord length. In contrast, the flat plate in the present case does not have any thickness.

Figure 10 shows the force coefficients obtained in our study compared with those of references. Here, the lift and drag coefficients are defined as $C_D = 2F_x^*/(\rho U_{\text{max}}^2 c^*)$ and $C_L = 2F_y^*/(\rho U_{\text{max}}^2 c^*)$, where $F_x$ and $F_y$ are the horizontal and
vertical forces acting on the plate, respectively. Despite the slight difference in the considered geometry, the drag and lift coefficients show good agreement. The figure also verifies that the grid resolution for the base case (Δx = Δy = 0.0098) is sufficiently fine, although the peak values are observed slightly influenced by the grid resolution. For both coefficients, the number of the peaks and their locations are similar to the previous studies.

It is worth noting that it is not possible to directly apply the ghost-cell method for this problem since the ghost points should be set inside the body. It might be possible to develop a new method equivalent to the ghost-cell approach if we combine the present interpolation stencil with, for instance, the idea of direct discretization approach proposed by Sato et al. (2013). However, this is not straightforward and therefore left for the future work.

3.3. Single Cylinder Falling in a Channel Under the Gravity Force

There are many applications in the industry that involve liquids which have interaction with the particles such as mining extraction, slurry flows and fluidization of the catalyst beds, to name but a few. In order to examine the reliability of our method, we solve the flow of a falling cylinder in a vertical channel.

The quantities are made dimensionless by the fluid density, ρ\# = 1.00 g/cm³, the cylinder diameter, D\# = 0.25 cm, and the characteristic velocity V_s\# expressed as (Koblitz et al., 2017)

\[ V_s\# = \sqrt{\frac{\pi D^2}{2} \left( \rho_s^\# - \rho^\# \right) g^\#}, \quad (19) \]

where the solid density is ρ_s\# = 1.25 g/cm³ and the gravitational acceleration is g\# = 9.81 m/s². The channel size is 8D × 24D, and the numbers of the computational cells are 512 in x direction and 1563 in y direction (both uniform). No-slip boundary condition is applied on all wall boundaries. The cylinder is initially located at (x, y) = (4D, 16D) and the initial velocity is zero.

The vorticity contours of the cylinder at different time steps are shown in Fig. 11. Figure 12 shows the temporal evolution of the vertical position and velocity of the center of cylinder. As shown, the results are in quantitative agreement with the previous studies (Glowinski et al., 2001; Hu et al., 2015; Koblitz et al., 2017; Wang et al., 2009). The computed terminal Reynolds number is Re_T = V_T^\# D^\# / ν^\# = 17.18 (where V_T^\# is the resultant terminal velocity, and ν^\# = 0.100 cm²/s) is consistent with the literature.

3.4. Two Falling Cylinders — Drafting, Kissing, and Tumbling

The other test case is sedimentation of two circular cylinders in a vertical channel. It has been shown that when two cylinders fall close to each other, three physical phenomena, namely drafting, kissing and tumbling, take place. Although two cylinders receive the same gravitational acceleration, the upper cylinder falls faster than the lower cylinder because of lower drag; this stage is called drafting. The distance between two cylinders decreases until they touch each other, called the kissing. However, since the movement of cylinders is unstable, the upper cylinder starts to tumble on the lower cylinder, triggered by a small amount of asymmetricity, and eventually two cylinders separate.

To avoid cylinders from inter-penetrating each other, a collision model is implemented. If the distance between two bodies becomes less than a certain threshold, a repulsive force, \( F_{\text{col}} = F_{\text{p-p}} + F_{\text{p-w}} \) (where p-p and p-w denote particle-
particle and particle-wall collisions, respectively), acts on the bodies and makes them move to the opposite direction. With this repulsive force, the total force $F$ in Eq. (3) can be re-written as

$$F = \left(1 - \frac{\rho_f}{\rho_s}\right)mg + F_h + F_{\text{col}}. \quad (20)$$

The collision force is modeled similarly to Wu et al. (2016). The particle-particle collision force for $i$-th cylinder, $F_{i}^{p-p}$, is computed as

$$F_{i}^{p-p} = \begin{cases} 0, & \text{if}\ X_{i,j} > (R_i + R_j) + \zeta, \\ 2.4\epsilon \sum_{j=1}^{N} \frac{2(\frac{R_i + R_j}{X_{i,j}})^{14} - (\frac{R_i + R_j}{X_{i,j}})^{8}}{X_i - X_j} \frac{X_i - X_j}{(R_i + R_j)^2}, & \text{if}\ X_{i,j} \leq (R_i + R_j) + \zeta. \end{cases} \quad (21)$$

where $R_i$ represents the radius of $i$-th cylinder, $X_{i,j} = |X_i - X_j|$, $\epsilon = [2R_iR_j/(R_i + R_j)]^2$, and $\zeta$ is the threshold set here to $\zeta = 3\Delta x$. Similarly, $F_{i}^{p-w}$ is computed as

$$F_{i}^{p-w} = \begin{cases} 0, & \text{if}\ X_{i,k} > 2R_i + \zeta, \\ 2.4\epsilon \sum_{k} \frac{2(\frac{R_i}{X_{i,k}})^{14} - (\frac{R_i}{X_{i,k}})^{8}}{X_i - x_k} \frac{X_i - x_k}{(R_i)^2}, & \text{if}\ X_{i,k} \leq 2R_i + \zeta, \end{cases} \quad (22)$$

where $x_k$ represents the position of wall element and $X_{i,k} = |X_i - x_k|$.

Note that the distance between two cylinders can become smaller than $\Delta x$ even with this collision force. Although in such situation the interpolation stencil and the corresponding intersection points for one cylinder may be located inside...
Fig. 13  Vorticity contours around the falling two cylinders at different time instants. Vorticity is also dimensional (s$^{-1}$).

Fig. 14  Time history of falling two circular cylinders: (a) horizontal position; (b) vertical position.

another cylinder, we still use them for interpolation to avoid additional complexities. This treatment can be justified by the fact that the velocities inside another cylinder are smoothly connected across the boundary and they are not much different from the velocity on the surface. Moreover, in such situation, the fluid dynamic force is negligibly small as compared to the repulsive force and the gravitational and buoyancy forces; namely, small inaccuracy of interpolation hardly affects the total force. It is also worth noting that, if two boundaries are close to a forcing point, the boundary closer to the forcing point is always considered for the definition of the interpolation stencil.

The parameters are set exactly the same as Patankar et al. (1999), who studied this problem using a Lagrange-multiplier-based fictitious-domain method, as well as Jafari et al. (2011) and Wang et al. (2014), who used the lattice Boltzmann method. The parameters are given as dimensional values. The channel size is 2 cm $\times$ 8 cm, and the diameter of the cylinders is $D_h = 0.2$ cm. The numbers of the computational cells are 512 in $x$ direction and 2048 in $y$ direction (both uniform). The fluid viscosity is $\mu_h = 1.00$ g/cm s and the density of fluid and cylinder are $\rho_h = 1.00$ g/cm$^3$ and $\rho_s = 1.01$ g/cm$^3$, respectively. Initially, two cylinders are located exactly on the centerline ($x^h = 1$ cm) with a vertical separation of 0.4 cm. Note that the results for this problem are presented in dimensional form so as to simplify the comparison with literature (also given in dimensional form), while the quantities in our simulation are made dimensionless based on $D_h$ and $g_h$.

Figure 13 shows the vorticity contours at different time instants, and Fig. 14 compares the time history of the positions of two cylinders with the results of Jafari et al. (2011). As can be observed, four stages of the interaction between two cylinders, i.e., drafting, kissing ($t^h = 1.50$ s of Fig. 13), tumbling ($t^h = 1.88$ s), and separating ($t^h = 2.75$ s and 3.5 s)
processes, are properly captured by our method. Although relatively large difference is observed between the present results and on and after tumbling process, it should be noted that the tumbling is initiated by a contingent asymmetry stemming solely from numerical errors, because two cylinders are perfectly aligned without any horizontal offset at the initial time instant; namely, no physical asymmetry is contained in this problem setting (in that sense, this problem setting is not well-posed, although it has been studied by many researchers). Namely, without any numerical errors, the flow will never become asymmetric and two cylinders will never depart from kissing stage (Uhlmann, 2005). Such a tiny numerical difference is amplified to make the trajectories significantly different on and after tumbling stage, as has also been observed in the previous study (Patankar et al., 1999).

3.5. Falling Multiple Cylinders

In the last example, we simulate a flow around nine cylinders falling under gravity. The purpose of this simulation is to simply demonstrate the capability of the method to handle more complex configuration. In this example, the physical and simulation parameters are similar to those for sedimentation of the single cylinder, but we consider two different sizes of cylinder, \( D_1^* = D^* = 0.25 \text{ cm} \) and \( D_2^* = 0.8D^* = 0.20 \text{ cm} \). The quantities are made dimensionless by \( D^* \) and \( V_s^* \) defined in Eq. (19). The size of the channel is \( 8D \times 22D \), and the numbers of the computational cells are 512 in \( x \) direction and 1563 in \( y \) direction (both uniform). The initial positions of the cylinders and the distance between them are as shown in the left column (\( t = 0 \)) of Fig. 15.

Figure 15 shows the vorticity contours of the sedimentation of the multiple cylinders at different time instants. As seen, a complicated sedimentation process involving interaction among cylinders and surrounding fluid is well captured, which demonstrates that the present method can be used for simulations of similar complex problems.

4. Conclusions

Two distinct but interwoven issues has been explored in this work. First, the implementation of an immersed boundary method for moving boundary problems equipped with the unified and versatile interpolation scheme. Second, trying to decrease the amplitude of the spurious force oscillation for the proposed method and to alleviate this inherent kink of all direct forcing methods. From the comparison between the ghost-cell and reconstruction methods, both methods have been found to work equally well for a flow around a fixed cylinder, which was chosen as an example of fixed boundary problems. However, for moving boundary problems, the reconstruction method has been shown to have a strong merit in suppressing the spurious force oscillations. Therefore, use of the reconstruction method is recommended for moving boundary problems.

Based on the output of the numerical results, we chose the reconstruction approach and a series of well-known test cases were considered to assess the accuracy of the method. Compared with the other numerical results and an experimental study, the accuracy in captured velocity profile for a transversely oscillating cylinder in a fluid at rest indicates the effectiveness of our method in the calculation of the primary variables for the forced rigid moving boundary problems.

![Fig. 15 Initial geometry and vorticity contours around multiple cylinders at different time instants (t = \( t^* V_s^*/D^* \)).](image_url)
Moreover, the results for the hovering flat plate illustrates that apart from considering a flat plate without any thickness, we are able to perfectly capture the features of the flow. On the other hand, the precise results for a falling cylinder under its own gravity force, revealed another merit of this simple direct forcing method, namely its potential for fluid-structure interaction problems. Finally, the sedimentation of some cylinders were simulated to demonstrate the capability of the present method for fluid-solid interactions in a more realistic problem. The proposed method managed the computational complexity of this problem in terms of many body motions, interactions and complicated fluid flow phenomena.

Acknowledgments

The authors are grateful to Drs. Shinnosuke Obi and Keita Ando (Keio University), Dr. Yosuke Hasegawa (The University of Tokyo), and Dr. Kaoru Iwamoto (Tokyo University of Agriculture and Technology) for fruitful discussion. H. Kor and M. Badri Ghomizad acknowledge the financial supports from Top Global University MEXT Scholarship and Otsuka Toshimi Foundation. This work was partly supported through Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), “Innovative Combustion Technology” (Funding agency: JST).

References

Balabas, E., Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. Computers & Fluids, Vol. 33 (2004), pp. 375–404.
Berthelsen, P. A. and Faltinsen, O. M., A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. Journal of Computational Physics, Vol. 227 (2008), pp. 4354–4397.
Belliard, M., Chandresris, M., Dumas, J., Gross, Y., Jamet, D., Josserand, C., and Mathieu, B., An analysis and an affordable regularization technique for the spurious force oscillation in context of direct-forcing immersed boundary methods. Computer and Mathematics with Applications, Vol. 71 (2016), pp. 1089–1113.
Chiu, P. H., Lin, R. K., and Sheu, T. W. H., A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier-stokes equations in time-varying complex geometries. Journal of Computational Physics, Vol. 229 (2010), pp. 4476–4500.
Choi, J. I., Oberoi, R. C., Edwards, J. R., and Rosati, J. A., An immersed boundary method for complex incompressible flows. Journal of Computational Physics, Vol. 224 (2007), pp. 757–784.
Dukowicz, J. K. and Dvinsky, A. S., Approximate factorization as high-order splitting for the implicit incompressible flow equations. Journal of Computational Physics, Vol. 102 (1992), pp. 336–347.
Dütsch, H., Durst, F., Becker, S., and Lienhart, H., Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers. Journal of Fluid Mechanics, Vol. 360 (1998), pp. 249–271.
Eldredge, J., Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method. Journal of Computational Physics, Vol. 221 (2007), pp. 626–648.
Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., Combined immersed-boundary finite-difference for three-dimensional complex flow simulations. Journal of Computational Physics, Vol. 161 (2000), pp. 35–60.
Ghias, R., Mittal, R., and Dong, H., A sharp interface immersed boundary method for compressible viscous flows. Journal of Computational Physics, Vol. 225 (2007) pp. 528–553.
Gilmanov A. and Sotiropoulos F., A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. Journal of Computational Physics, Vol. 207 (2005), pp. 457–492.
Glowinski, R., Pan, T. W., Hesla, T. I., Joseph, D. D, and Périaux, J., A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. Journal of Computational Physics, Vol. 169 (2001), pp. 363–426.
Goldstein, D., Handler, R., and Sirovich, L., Modeling a no-slip flow boundary with an external force field. Journal of Computational Physics, Vol. 105 (1993), pp. 354–366.
Guermond, J. L., Minev, P., and Shen, J., An overview of projection method for incompressible flows. Computer Methods in Applied Mechanics and Engineering, Vol. 195 (2006), pp. 6011–6045.
Hu, Y., Li, D., Shu, S., and Niu, X., Modified momentum exchange method for fluid-particle interactions in the lattice Boltzmann method. Physical Review E, Vol. 91 (2015), Paper No. 033301.
Jafari, S., Yamamoto, R., and Rahnama, M., Lattice-Boltzmann method combined with smoothed-profile method for particulate suspensions. Physical Review E, Vol. 83 (2011), Paper No. 026702.

Kadapa, C., Dettmer W. G., and Perić D., A stabilised immersed boundary method on hierarchical b-spline grids for fluid-rigid body interaction with solid-solid contact. Computer Methods in Applied Mechanics and Engineering, Vol. 318 (2017), pp. 242–269.

Kajishima, T., Finite-difference method for convective terms using non-uniform grid. Transactions of the JSME Series B, Vol. 65, No. 633 (1999), pp. 1607–1612. (in Japanese)

Kajishima, T. and Takiguchi, S., Interaction between particle clusters and particle-induced turbulence. International Journal of Heat and Fluid Flow, Vol. 23 (2002), pp. 639–646.

Kang, S., Iaccarino, G., Ham, F., and Moin, P., Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method. Journal of Computational Physics, Vol. 228 (2009), pp. 6753–6772.

Kim, D. and Choi, H., Immersed boundary method for flow around an arbitrarily moving body. Journal of Computational Physics, Vol. 212 (2006), pp. 662–680.

Kim, J., Kim, D., and Choi, H., An immersed-boundary finite volume method for simulations of flow in complex geometries. Journal of Computational Physics, Vol. 171 (2001), pp. 132–150.

Kobritz, A. R., Lovett, S., Nikiforakis, N., and Henshaw, W. D. Direct numerical simulation of particulate flows with an overset grid method. Journal of Computational Physics, Vol. 343 (2017), pp. 414–431.

Kor, H., Badri Ghomizad, M., and Fukagata, K., A unified interpolation stencil for ghost-cell immersed boundary method for flow around complex geometries. Journal of Fluid Science and Technology, Vol. 12 (2017), Paper JSFT0011.

Lai, M. C. and Peskin, C. S., An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. Journal of Computational Physics, Vol. 160 (2000), pp. 705–719.

Lee, J., Kim, J., Choi, H., and Yang, K.-S., Sources of spurious force oscillations from an immersed boundary method for moving-body problems. Journal of Computational Physics, Vol. 230 (2011), pp. 2677–2695.

Luo, H., Dai, H., Ferreira de Sousa, P. J. S. A., and Yin, B., On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. Computers & Fluids, Vol. 56 (2012), pp. 61–76.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F. M., Vargas, A., and von Loebbecke, A., A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. Journal of Computational Physics, Vol. 227 (2008), pp. 4825–4852.

Mittal, R. and Iaccarino, G., Immersed boundary methods. Annu. Rev. Fluid Mech., Vol. 37 (2005), pp. 239–261.

Mohd-Yusof, J., Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries. CTR Annual Research Briefs, NASA Ames/Stanford University (1997).

Naito, H. and Fukagata, K., Numerical simulation of flow around a circular cylinder having porous surface. Physics of Fluids, Vol. 24 (2012), Paper 117102.

Norberg, C., Fluctuating lift on a circular cylinder: review and new measurements. Journal of Fluids and Structures, Vol. 17 (2003), pp. 57–96.

Park, S. G. and Sung, H. J., Hydrodynamics of a self-propelled flexible fin in perturbed flows. Mechanical Engineering Reviews, Vol. 5, Paper 17-00286.

Patankar, N. A., Singh, P., Joseph, D. D., Glowinski, R., and Pan, T. W., A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. International Journal of Multiphase Flow, Vo. 26 (2000), pp. 1509–1524.

Peskin, C., Numerical Analysis of blood flow in heart. Journal of Computational Physics, Vol. 5 (1972), pp. 220–252.

Russell D. and Wang, Z. J., A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. Journal of Computational Physics, Vol. 191 (2003), pp. 177–205.

Saiki, E. M. and Biringen, S., Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. Journal of Computational Physics, Vol. 123(1996), pp. 450–465.

Sato, N., Kajishima, T., Takeuchi, S., Inagaki, M., and Horinouchi, N., A direct discretization approach near wall boundaries for a Cartesian grid method (Consideration of consistency between velocity and pressure fields). Transactions of the JSME Series B, Vol. 79, No. 800 (2013), pp. 605–621. (in Japanese)

Seo, J. H. and Mittal, R., A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. Journal of Computational Physics, Vol. 230 (2011), pp. 7347–7363.

Soltani, E. and Badri Ghomizad, M., Immersed boundary method for the simulation of heat transfer and fluid flow based on vorticity-velocity formulation. Numerical Heat transfer, Part B: Fundamentals, Vol. 70 (2016), pp. 25–46.
Spalart, P. R., Moser, R. D., and Rogers, M. M., Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. Journal of Computational Physics, Vol. 96 (1991), pp. 297–324.
Tseng, Y. and Ferziger, J. H., A ghost-cell immersed boundary method for flow in complex geometry. Journal of Computational Physics, Vol. 192 (2003), pp. 593–623.
Uhlmann, M., An immersed boundary method with direct forcing for the simulation of particulate flows. Journal of Computational Physics, Vol. 209 (2005), pp. 448–476.
Wan, D. and Turek, S., Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method. International Journal for Numerical Methods in Fluids, Vol. 51 (2006), pp. 531–566.
Wang, L., Guo, Z. L., and Mi, J. C., Drafting, kissing and tumbling process of two particles with different sizes. Computers & Fluids, Vol. 96 (2014), pp. 20–34.
Wang, Z., Fan, J., and Cen, K., Immersed boundary method for the simulation of 2D viscous flow based on vorticity-velocity formulations. Journal of Computational Physics, Vol. 228 (2009), pp. 1504–1520.
Wu, J., Wu, J., Zhan, J., and Zhao, N., A robust immersed boundary-lattice Boltzmann method for simulation of fluid-structure interaction problems. Communication in Computational Physics, Vol. 20 (2016), pp. 156–178.
Xu S. and Wang, Z. J., An immersed interface method for simulating the interaction of a fluid with moving boundaries. Journal of Computational Physics, Vol. 216 (2006), pp. 454–493.
Yang, J. and Balaras, E., An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. Journal of Computational Physics, Vol. 215 (2006), pp. 12–40.
Zhang, N. and Zheng, Z. C., An improved direct-forcing immersed-boundary method for finite difference applications. Journal of Computational Physics, Vol. 221 (2007), pp. 250–268.