A sharp interface direct-forcing immersed boundary method using the moving least square approximation

Mehdi BADRI GHOMIZAD*, Hosnieh KOR* 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: 7 January 2021; Revised: 27 February 2021; Accepted: 5 April 2021

Abstract
Based on the Moving Least Square (MLS) approximation, we propose a sharp interface direct-forcing immersed boundary method for incompressible fluid flows with fixed and moving boundaries. Since the domain of definition for the interpolation is highly flexible and the MLS approximation provides an accurate reconstructed approximation of the solution, the proposed method serves the precision and versatility required for a numerical framework to study the fluid-structure interaction problems. To alleviate the inherent spurious numerical oscillation that occurs in the calculated forces on moving boundary embedded objects, we use a two step predictor-corrector method in which the direct forcing terms are calculated after the predictor step and imposed on the whole solid domain as well as at the immediate vicinity of the solid boundary inside the fluid domain. To represent the arbitrary geometries, we adopt a signed distance function representation of the rigid body and an interpolation strategy to considerably reduce the computational cost of the re-initialization of the distance function at every time step. The potential capability of the method is demonstrated for both fixed and moving boundary problems. We also solve a sedimentation of a single cylinder to demonstrate the ability of the present method in solving fluid-structure interaction problems. These numerical experiments show that the proposed moving least square immersed boundary method can handle relatively complex moving problems while enjoying a versatile interpolation strategy and keeping the boundary conditions sharp with remarkable accuracy.

Keywords: Direct forcing, Immersed boundary method, Moving least square, Fluid-Structure interaction, Spurious force oscillation

1. Introduction

Immersed boundary methods as a well-established sub-category of Cartesian mesh methods are powerful approaches to study complex fluid flows and fluid-structure interaction problems in engineering, cardiovascular and biological discipline, to name but a few, due to their ability to handle with relative ease of multi-body and deformable objects. Based on the strategy that is used to consider the presence of immersed objects in the flow, these methods are broadly categorized into two general classes: diffuse interface (DI) and sharp interface (SI) methods.

The DI method was originally introduced in 1972 by Peskin (Peskin, 1972). In his pioneering work, Peskin dealt with the solid boundaries by adding a singular forcing term to the momentum equations in the vicinity of the body and distributed them onto the Cartesian grid points by convoluting the field with discrete delta functions. Since those discrete delta functions smear out the force among the layers of grid points, the name of diffused interface stems from this smearing process at the surface of the solid boundary. This method was extended later in several ways; among the others, Yuki et al. (2007) introduced the solid volume fraction instead of the Lagrangian markers which also leads to a smooth interface.

In the SI methods, the immersed boundary condition is implemented at the vicinity of the Lagrangian solid boundary either by changing the background grid cells around the immersed object with the so-called cut-cell methods (Ye et al., 1999) or by using a direct forcing method in which the velocity field is reconstructed according to the desired boundary condition (Mohd-Yusof, 1997; Uhlmann, 2005; Kor et al., 2017). Deploying a local unstructured mesh, the cut-cell
method has been able to accurately impose the needed boundary condition. Despite their accuracy, however, the cut-cell methods tend to be cumbersome in three-dimensional problems. In contrast to the cut-cell methods, direct forcing methods, also known as Hybrid Cartesian Immersed Boundary Methods (HCIB), use reconstructions on the Cartesian background mesh points where the points are at the immediate vicinity of the solid boundary (Balaras, 2004; Nishida and Tajiri, 2009; Liao et al., 2010; Kor et al., 2017; Kor, 2018; Nishikawa et al., 2020). Those hybrid Eulerian grid points are dubbed forcing (or immersed boundary) points when they are inside the fluid phase, and ghost points when they belong to the solid side (Kor, 2018; Kor et al., 2018). In this paper, we adopt the forcing and ghost point terminology for the hybrid nodes. All the reconstruction methods rely upon a sort of interpolation to obtain the proper value of the depending variables for either the forcing or ghost nodes (or in some cases on both of them) from the interior points on which the Navier-Stokes equation are solved and the value of the variable on the physical boundary obtained by the given boundary conditions. A wide range of the methods have been put forward using, among the others, interpolation in conjunction with ghost node methods, interpolation along the normal to the surface, or interpolation along the grid lines. All of these methods share a common characteristic: they employ a linear or quadratic polynomial function to calculate the required values for the forcing of ghost nodes. However, Peller et al. (2006) used the least square approximation to find the reconstructed values at the forcing point. They obtain the least square approximation along the grid lines and get the weighted average of them as a unique value for the forcing point. Combining the idea behind the Peskin’s classical immersed boundary with the SI formulation, Uhlmann (2005) proposed a DI method in which the forcing terms are calculated on the Lagrangian points rather than on the background nodes, and then by using the discrete delta functions, the forces are re-distributed onto the surrounding Eulerian grid nodes. Vanella and Balaras (2009) improved Uhlmann’s method and employed the moving least square (MLS) instead of the discrete delta function strategy to back-propagate the calculated forces on the Lagrangian markers. It should be mentioned that these methods are also categorized as DI methods since the boundary representation is not sharp.

The way of reconstructing the velocity at the hybrid points (either the forcing or the ghost point) have quite considerable impact on the overall accuracy of the solution (Kor, 2018). In both approaches to the immersed boundary method (SI or DI), the required values are usually interpolated along a specific line direction (albeit ad-hoc), which is not consistent with the strong elliptic feature of the incompressible flows. This feature of the incompressible flows demands a dispersed and relatively larger stencil for interpolation rather than those local and ad-hoc directional interpolations typically used. Although mesh-free interpolation methods can be used to obtain the values which better satisfy the requirements, the implementation of them along with the HCIB methods has been done just recently and is still in its early stage. Wang et al. (2010) and Ding et al. (2004) numerically solved the Navier-Stokes equations with moving boundaries by considering a cloud of nodes in the vicinity of the boundary. They applied the regular finite difference method to discretize the equations on the Cartesian nodes, while they employed the generalized finite difference (GFD) for the cloud surrounding the immersed boundary, which is computationally more expensive than typical finite difference methods. Soltani and Badri (2016) employed the Shepard method for heat transfer problems in conjunction with velocity-vorticity formulation of the Navier-Stokes equation discretized with pseudo-spectral techniques.

Besides the accuracy of the interpolation, another concern in immersed boundary methods is the inherent oscillations which emerges with moving boundary problems. Many researchers have shown that this matter can affect the accuracy and efficiency of the method (Seo and Mittal, 2011; Lee and You, 2013; Kor et al., 2018). In order to reduce the undesired numerical oscillations, Uhlmann (2005) showed that combining the delta function and direct forcing method could alleviate it though the boundary is not sharp anymore. Taira and Colonius (2007) proposed the so-called projection immersed boundary which can control those spurious forces, though with a prohibiting computational load and smearing the boundary. Later, detailed analysis in this regards resulted in a number of possible explanations on the cause and the remedies. The abrupt changes in the role of the Hybrid, fluid and solid nodes is deemed a reason for non-physical oscillations. In a series of studies (Yang and Stern, 2015; Yang and Balaras, 2006), a method grouped as field extension based on a consistent extrapolation was introduced by taking into account the correct velocity and pressure for each time step, and thus cure the problem. Luo et al. (2012) argued that the instant change in the reconstruction stencil during the movement of the object is the source of this issue and proposed a smooth transition of such stencil in time that reduced the temporal discontinuity in the velocities, and thus improved the quality of their solution. The lack of mass conservation was considered as another explanation for this problem as was suggested by Seo and Mittal (2011). This deficiency in mass conservation leads to accumulation of errors at the immersed boundary and consequently to the errors in immersed boundary forces. In the same way, Martins et al. (2017) addressed this issue and put forward a constrained interpolation method for it. Another solution for this problem was a predictor-corrector direct forcing with enforcing the whole solid
domain. Liao et al. (2010) proposed this approach with a linear interpolation along the line normal to the surface of the immersed object and revealed the superiority of the method for spurious oscillation reduction.

Aiming at reducing the inherent spurious oscillation of the immersed boundary method, introduced above, and increasing the accuracy of the force calculation in direct forcing immersed boundary methods, we propose a novel approach which retains the simplicity of standard predictor-corrector direct forcing method while eliminates the need for an ad-hoc directional interpolations, by utilizing the desired feature that comes by the mesh-free interpolation. Calculation of the required forces representing sharp solid-fluid interfaces, to the best of our knowledge, has not been proposed yet with a moving least square interpolation (approximation) methodology in the framework of a predictor-corrector direct forcing method, and our numerical experiments reveals its merits for the fluid-structure interaction problems in particular for moving boundary problems. Specifically, we point out that the present method not only enjoys the moving least square approximation for data transformation between Eulerian and Lagrangian nodes (in contrast to Vanella and Balaras’s (2009) approach, which results in a DI) but also it improves the quality of the solution comparing to the original direct forcing by employing a relatively simple technique to reduce the spurious oscillations. This technique closely resembles the seamless immersed boundary methods (Tajiri et al., 2014) though with a more accurate predictor-corrector strategy and interpolation which leads to a remarkable less spurious oscillation solution for moving boundary cases.

This paper is organized as follows. In section 2, the governing equations for incompressible fluid flows as well as the solid body are introduced. The cell-centered finite difference discretization method used in the present study is also briefly discussed. Further, the details of the moving least square approximation is provided. We also describe the key aspect of the present two-step predictor-corrector direct forcing immersed boundary method, including the interface reconstruction and the geometrical considerations for the interpolation and its domain. In Section 3, we demonstrate the superiority of the present moving least square direct forcing formulation by applying it to the following benchmark problems: the flow past a circular cylinder at different Reynolds numbers, the flow over an oscillating cylinder in fluid at rest, and fluid-solid interaction between a falling cylinder and the surrounding fluid. Section 4 summarizes the research finding and discusses the future research direction.

2. Methodology of the immersed boundary technique

In this section, the general mathematical formulation of the fluid structure interaction problem is introduced. Then, the employed numerical method is discussed, followed by the details on the construction of the immersed boundary forces needed for representing the solid objects.

2.1. Governing equations

We consider the rigid body motion within an incompressible fluid, as schematically depicted in Fig. 1 with the parameters involved in this fluid-structure interaction problem. In Fig. 1, the whole domain is indicated by $\Omega$ while the fluid and the rigid body occupy the sub-domains $\Omega_f$ and $\Omega_s$, respectively. The whole system is assumed to be under the gravitational acceleration $g$ and the interaction occurs at the common boundary between the solid object and the fluid $\partial\Omega_i = \Omega_f \cap \Omega_s$. In addition, $\omega_s$ stands for the angular velocity, $v_s$ for the translational velocity and $\partial\Omega_f$ for the outer boundary of the system.
The fluid motion is governed by the incompressible continuity and Navier-Stokes equations,
\[ \nabla \cdot \mathbf{u} = 0 \]  
\[ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = \nabla \cdot \mathbf{\sigma}_f + g + f. \]  
where \( \mathbf{u} \) is the fluid velocity vector. The stress tensor \( \mathbf{\sigma}_f \) is given by
\[ \mathbf{\sigma}_f = -\frac{p}{\rho_f} \mathbf{I} + \nu \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right), \]  
where \( p \) is the pressure, \( \rho_f \) is the fluid density, and \( \nu \) is the kinematic viscosity. To make sure that the problem is well-posed, proper initial and boundary conditions are defined for each test case. In Eq. (1), the term \( f \) is added to account for the presence of the solid within the domain. Therefore, the fluid is simply simulated in the \( \Omega = \Omega_f(t) \cup \Omega_s(t) \). (See Fig. 1).

On the other hand, the rigid body motion for a two-dimensional problem is governed by the Newton-Euler equations, i.e.,
\[ m_s \frac{d\mathbf{v}_s}{dt} = \mathbf{F}_s + m_s \left( -\frac{\rho_f}{\rho_s} \right) g, \]  
\[ I_s \frac{d\omega_s}{dt} = \mathbf{T}_s, \]  
where \( m_s, \rho_s \), and \( I_s \) are the solid mass, density, and the moment of inertia, respectively. The force \( \mathbf{F}_s \) and torque \( \mathbf{T}_s \) acting from the fluid on the object are obtained as
\[ \mathbf{F}_s = \rho_f \int_{\partial \Omega_s} \mathbf{\sigma}_f \cdot \mathbf{n} dS, \]  
\[ \mathbf{T}_s = \rho_f \int_{\partial \Omega_s} \mathbf{r} \times (\mathbf{\sigma}_f \cdot \mathbf{n}) dS, \]  
where \( \mathbf{r} = \mathbf{x}_s - \mathbf{x}_c \) is the location vector of the surface point \( \mathbf{x}_s \) with respect to the center of the mass \( \mathbf{x}_c \). For this object, \( \mathbf{n} \) is the outward normal vector to the surface \( \partial \Omega_s \).

The surface forces can be recast to the body forces using the Gauss’ theorem and the immersed boundary forces (i.e., \( f \) in Eq. (1)) as (De, 2018)
\[ \mathbf{F}_s = -\rho_f \int_\Omega f dV, \]  
\[ \mathbf{T}_s = -\rho_f \int_\Omega (\mathbf{r} \times f) dV. \]  

The instantaneous location of each rigid object is obtained by integrating the ordinary differential equations for their kinetics
\[ \frac{d\mathbf{x}_c}{dt} = \mathbf{v}_s, \]  
\[ \frac{d\theta_c}{dt} = \omega_s, \]  
where \( \theta_c \) represents the rotation angle around the center of mass. To consider the effect of the fluid structure interaction, the boundary condition at the interface, i.e.,
\[ \mathbf{u} = \mathbf{v}_s + \omega_s \times \mathbf{r}. \]  
is also imposed.

2.2. Numerical schemes
2.2.1. Overview of the numerical procedure
The numerical procedure used here consists of a spatial discretization on a collocated grid and a fractional step method for the velocity-pressure coupling. In this study, all variables are evaluated at the cell centers of a Cartesian mesh, and the spatial derivatives are evaluated using the second-order central finite difference method. However, to cope with the checkerboard problem in velocity-pressure coupling, the contravariant component of velocity normal to each cell surface is also added to the list of variables which is updated during the time integration. In order to evolve
the system of equations (1) in time, they are discretized using a fractional step (projection) method being implemented with a combination of the Adams-Bashforth and Crank-Nicolson schemes for the advection and diffusion terms of the momentum equation, respectively. Several variants of the fractional step methods are available, such as Kim and Moin (1985), Bell et al. (1989), and Guermond et al. (2006) with the references there in, among the others. Here we employ the formulation of Kim and Moin (1985) due to its straightforward boundary condition enforcement.

The discrete form of the algorithm entails the following sequence:

\[
\frac{\tilde{u} - u^n}{\Delta t} = -\frac{3}{2} \nabla_h \cdot (Uu)^n + \frac{1}{2} \nabla_h \cdot (Uu)^{n-1} + \frac{1}{2} \nu \nabla_h^2 (u^n + \tilde{u}) - \nabla_h p^n + f^{n+1},
\]

(8)

where the superscripts \( n-1, n, \) and \( n+1 \) show the discrete times, and \( \tilde{u} \) is the first provisional velocity between time step \( n \) and \( n+1 \), which is a non-divergence-free approximation for velocity at time step \( n+1 \). The discrete spatial derivative operators (i.e., second-order central finite difference) are denoted by \( \nabla_h \), and \( U \) is the contravariant component of the velocity vector residing at the cell faces. The pressure normalized by the density is denoted as \( p = p/\rho_f \). In this formulation, the second provisional velocity \( u^* \) is computed by

\[
\frac{u^* - \tilde{u}}{\Delta t} = \nabla_h P_{cc}^n,
\]

(9)

where \( \nabla_h P_{cc}^n \) is the pressure gradient at the cell center.

Then, the provisional contravariant velocity are obtained from their cell center counterpart by an interpolation function, i.e.,

\[
U^* = F(u^*),
\]

(10)

in which \( F \) is the interpolation function from cell center variables to the cell face. Here, the interpolation is a second-order linear one. Adding the pressure gradient on the cell face to the interpolated velocity leads to a well-coupled pressure-velocity field:

\[
\frac{U^{**} - U^*}{\Delta t} = -\nabla_h P_{fc}^n.
\]

(11)

According to Lee et al. (2019), this treatment will also suppress the error in the kinetic energy. The pressure corrector term \( \phi \) is calculated by solving the Poisson equation

\[
\nabla_h^2 \phi_{cc} = \nabla_h \cdot U^{**}/\Delta t.
\]

(12)

under the homogeneous Neumann boundary condition on all boundaries.

Finally, the velocities are updated as

\[
\frac{u^{n+1} - u^n}{\Delta t} = -\nabla_h \phi_{cc},
\]

(13)

\[
\frac{U^{n+1} - U^*}{\Delta t} = -\nabla_h \phi_{fc},
\]

(14)

where \( \nabla_h \phi_{cc} \) and \( \nabla_h \phi_{fc} \) denote the gradients of \( \phi \) at the cell center and the cell face, respectively, and the pressure is updated as

\[
p^{n+1} = p^n + \frac{1}{2} \nu \nabla_h \cdot \tilde{u}.
\]

(15)

Having updated the fluid velocity and pressure field, the kinetics equation for the rigid body can be solved. Therefore, Eq. (3) is discretized as

\[
\frac{m_s v_{ss}^{n+1} - v_s^n}{\Delta t} = F_s^{\text{ext}} + m_s \left( 1 - \frac{\rho_s}{\rho_f} \right) g,
\]

(16a)

\[
\frac{I_s \omega_s^{n+1} - \omega_s^n}{\Delta t} = T_s^{\text{ext}},
\]

(16b)

where \( F_s^{\text{ext}} \) and \( T_s^{\text{ext}} \) are the exerted force and torque on the solid body at time step \( n+1 \).
2.2.2. Forcing strategy

The forces applied in the momentum equation (8) to represent the effect of the rigid body must be given before the procedure discussed in Sec. 2.2.1. To this end, the required force per unit mass is simply the Newtonian acceleration of all nodes near the solid body (Balaras, 2004; Uhlmann, 2005), that is

\[ f_{n+1} = \frac{u_F - \dot{u}}{\Delta t} - H_{n+1/2}, \]  

(17)

where the \( \dot{u} \) is an estimation of the velocity at the time step \( n + 1 \) in the neighborhood of the rigid body without considering the effect of the solid boundary, and \( u_F \) is the velocity at the same location while considering the presence of the body. The second term \( H_{n+1/2} \) represents the convective, viscous and pressure terms in momentum equations calculated in a time level between \( n \) and \( n + 1 \). It should be stressed that here the immersed boundary force is directly calculated on Cartesian mesh. Namely, \( u_F \) is set as the velocity of the rigid body for all nodes inside the solid domain \( \Omega_s \), while \( u_F \) is interpolated from the velocity field in the fluid domain and the velocity of three close nodes on the solid boundary for the nodes just at the vicinity of the boundary \( \partial \Omega_f \). This interpolated velocity along with \( H \) is used to calculate the required force. Thus, contrarily to other studies based on moving least square reconstruction (Vanella and Balaras, 2009), the velocity is not interpolated onto the surface. Furthermore, the force is not required to be interpolated from surface to the surrounding nodes smearing the boundary. Note that the this force merely affects the nodes in the solid domain and nodes very close to it and is set to zero in the fluid domain \( \Omega_f \). Details on how to choose the stencil for interpolation is discussed in the subsequent section.

In order to obtain the estimated field, \( \dot{u} \), a second-order explicit time step method is utilized to evolve the velocity field from time step \( n \) to the estimated one for the time step \( n + 1 \):

\[ \dot{u} - u^n = -\frac{3}{2} \left[ \nabla_h \cdot (uu) - \nu \nabla^2 u \right]^n + \frac{1}{2} \left[ \nabla_h \cdot (uu) - \nu \nabla^2 u \right]^{n-1} - \nabla_h p^n. \]  

(18)

In contrast to the Eq. (8), in Eq. (18) an explicit time step method is used for this prediction step since an explicit treatment for time integration is more efficient in terms of computational costs. Further, it has been shown (Nicolaou et al., 2014), for the corrector step with the immersed boundary force, a semi-implicit time step results in more accurate result. Therefore, while Eq. (8) is descretized using a combination of Adams-Bashforth and Crank-Nicolson schemes, we adopt the Adams-Bashforth explicit scheme for all terms in Eq. (18).

Then, we need to calculate the velocity field taking into account the boundary condition, namely \( u_F \), which is distributed around the boundary. Aiming at this, from the explicitly estimated field \( \dot{u} \), the velocity of the Eulerian nodes at the vicinity of the solid boundary residing inside the fluid domain are approximated by a moving least square method such that the boundary condition is satisfied at the physical solid boundary. It is worth mentioning that apart from those points inside the fluid domain at the neighbor of the physical boundary, \( u_F \) inside the body is also calculated. For the grid nodes inside the solid, the estimated velocity is set at the value consistent with both translational and rotational velocities of the body. Thus, the immersed boundary force is implemented on the nodes inside the body as well as on the fluid nodes close to it.

This procedure of velocity enforcement inside the body leads to improvement of accuracy in the results, particularly in reduction of the numerical oscillation of the variables (Liao et al., 2010; Tajiri et al., 2009). However, two issues should be resolved here: (1) the forcing location of the fluid nodes and (2) the specific details of calculating \( u_F \) for those nodes.

Fig. 2 Schematic representation of geometrical features: (a) definition of the normal to surface and the neighbors; (b) the role of the nodes.
2.2.3. Shape representation via a signed distance function

To represent the shape of the object, a signed distance function from the set of markers put on the solid-fluid interface, denoted as \( \Gamma \), is described as

\[
\phi(x) = \text{sgn}(\mathbf{n}_i \cdot (x_i - x_f)) \min \left( \|x_i - x_f\| \right),
\]

where \( x_f \in \Omega \) indicates the coordinate of Lagrangian markers laid on the interface, \( \Omega \), and \( \mathbf{n}_i \) denotes the surface-normal vector on them, as depicted in Fig. 2(a).

For regular geometries with analytical expression, this function can be obtained readily from the mathematical definition. For more complex shapes, without a closed-form expression, one can produce the function by combining the atomic shape (regular geometry) (Günther et al., 2014) or from a Standard Triangle Language (STL) representation of the surface (Luo et al., 2014) via explicitly calculating the distance from each node to the surface elements. Aiming at increasing the versatility of the method, we initialize the distance function by calculating the distance of the each Eulerian node from the nearest Lagrangian marker. When the system evolves and the location of the boundary changes, the distance function on the Eulerian nodes must be updated to reflect the new position of objects. Here, without re-initialization, we update it according to the motion of the boundary using a radial basis function interpolation as we discuss later in this section.

Figure 2(a) schematically shows how the Lagrangian markers (blue circles) are distributed on the surface of the rigid body. It is worth mentioning that the distance between markers is approximately kept equal to the resolution of the background mesh. In addition, the signed distance function \( \phi \) is positive if the \( x_i \) is outside the object and is negative when it is inside, while \( \phi(x_i) = 0 \) indicates the surface itself.

The normal vector to the surface \( \mathbf{n}_i \) on the Lagrangian marker point can be calculated from the normal vectors on the neighboring elements \( \mathbf{n}_k \) \( (k = 1, \ldots, m) \), which can easily be computed from the coordinates, as

\[
\mathbf{n}_i = \frac{\sum_{k=1}^{m} \alpha_k \mathbf{n}_k}{\sum_{k=1}^{m} \alpha_k},
\]

where \( \alpha_k \) denotes the weight for each element meeting at the Lagrangian markers. Those elements in two-dimensional cases are the line segments connecting the neighbor Lagrangian markers, and their weights are considered as their length. For example, in Fig. 2(a) the vector \( \mathbf{n}_1 \) is obtained from \( \mathbf{n}_1^1 \) and \( \mathbf{n}_1^2 \), while their weights are \( \|x_1^1 - x_f\| \) and \( \|x_1^2 - x_f\| \), respectively.

2.2.4. Node stencil for the reconstruction of the velocity

In this study, the immersed boundary force is applied on the forcing nodes. As exemplified by the point \( F \) in Fig. 2(b), the forcing node are Eulerian nodes belonging to the fluid domain and have at least one neighbor in the solid domain. Associated to each forcing node, a physical boundary point, which is defined as the closest point on the fluid-solid interface, is located by

\[
x_s = x_f - n_f \phi_f,
\]

where \( x_s \) is the location of the physical boundary point corresponding to the forcing nodes located at \( x_f \), and \( \phi_f \) and \( n_f \) are the distance function and the unit normal vector to the surface at the forcing point, respectively. The unit normal vector to the surface from the forcing point can be obtained from the distance function as

\[
n_f = \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right)_f,
\]

where \( \nabla \phi \) is computed using the second-order central finite difference method. In Fig. 2(b), for example, the point \( S \) is the physical boundary point associated with this specific forcing node \( F \) and the normal vector to the surface is illustrated by \( n_f \). It worth nothing that the location of physical boundary points calculated by Eq. (21) may or may not coincide with the Lagrangian markers that were used to obtain the distance function.

Thanks to the versatile strategy in the proposed MLS method (section 2.2.6), an approximation of the velocity can be obtained even with one stencil point. However, in this work, the stencil of velocity reconstruction consist of three to eight points depending on the geometrical constraints such as proximity of the objects and walls. It always have at
least three points from the physical boundary points and at most five fluid nodes. Thus, contrasting with traditional sharp immersed boundary method, the effect of the physical boundary is taken into account with more points, which yields sharper representation of the boundary and a higher accuracy in capturing the boundary curvature.

The algorithm for finding the required stencil points is as follow.

(1) For each forcing node, the first stencil point is obviously its associated physical boundary point. This point is tagged with $S_1$, as schematically shown in Fig. 2(b).

(2) Having found $S_1$, we pick two auxiliary nodes, $A_2$ and $A_3$, in the domain surrounding the forcing point along the direction of the normal vector to surface. To spot them, we first define the unit normal vector $\mathbf{n}_f$ constrained on the grid, $\mathbf{e}_f$, at the forcing point as

$$ e_i = \text{sgn}(n_i), $$

where $e_i$ and $n_i$ denote the $i$-th component of $\mathbf{e}_f$ and $\mathbf{n}_f$ ($i = 1, 2$). Then, supposing that the indices for the forcing point $F$ is $(i, j)$, the indices of the auxiliary nodes would be $A_1(i + e_1, j + e_2), A_2(i + e_1, j), \text{and } A_3(i, j + e_2)$.

(3) By using Eq. (21), the location of corresponding physical boundary points to $A_1, A_2$ and $A_3$ are sought out and tagged as $S_1, S_2$ and $S_3$, respectively. The points $S_1$ and $S_2$, and $S_3$ are always the first three stencil points for each forcing node.

(4) Two other auxiliary nodes should be chosen (i.e $A_4$ and $A_5$). To do so, assuming the forcing node index, $(i, j)$, the indices for those would be $A_4(i + 2e_1, j), A_5(i, j + 2e_2)$. Each of these auxiliary points might be a different forcing node, or may be inside the solid phase of the other objects (when the objects are very close to each other) thus we must assess all points in the set of auxiliary points $\mathcal{A} = \{A_1, A_2, A_3, A_4, A_5\}$ and select the acceptable ones from the $\mathcal{A}$. At this stage, however, we just record the indices of all potential stencil nodes and later we decide which one is used for the calculation. Note that $A_1$ and $A_2$ become identical in the cases where the surface perfectly perpendicular to an axis (i.e., either $n_1$ or $n_2$ is smaller than the machine epsilon). In such exceptional cases, we seek for the nodes with a different algorithm so that $A_1 = (i + e_1, j + e_2), A_2 = (i + e_2, j + e_1), A_3 = (i - e_2, j - e_1), A_4 = (i + 2e_1, j + 2e_2) \text{ and } A_5 = (i + 3e_1, j + 3e_2)$.

(5) Form the set of all auxiliary points, $\mathcal{A}$ a subset of them belonging to the fluid domain are selected as the stencil points, while those are forcing points or the nodes inside another object(s) are ruled out. Therefore, in this study, the total number of stencil points for each forcing node is not fixed and may vary between three to eight.

It should be pointed out that in order to fully exploit the proposed moving least square approximation, one needs, at least six, distinct nodes in the stencil and typically we fulfill this requirement in our algorithm. Even in the case where both auxiliary points $A_2$ and $A_3$ are forcing points and excluded from the interpolation, we still have six authentic nodes satisfying the minimum requirement for fully accurate approximation.

However, in a highly anomalous situations, the number of stencil points of very few forcing nodes might be smaller than six. Even in those cases, the algorithm is stable and the overall accuracy is not compromised because in the worst case scenario our algorithm guarantees the existence of at least three distinct stencil nodes which is enough for acceptable accuracy (Mirzaie, 2015). Moreover, these cases hardly occur; even in their occurrence, for a typical object the number of forcing points with limited stencil are much smaller than those with normal stencil (who has at least six nodes in their stencil), thus their end result is almost negligible. For example, when an object collides the wall, for very few time steps just before and after the collision, few forcing nodes may have less than six stencil points while we have much more forcing nodes around the object with full stencil nodes; therefore, the effect of those small-stencil interpolation on the calculation of immersed boundary or physical force (e.g., lift and drag) is virtually imperceptible.
2.2.5. Method for computing the signed distance for moving objects

For moving objects, frequent re-initialization of the signed distance function is mandatory. To alleviate this costly procedure, Liu and Hu (2014) suggested a method for finding the signed distance without re-initialization at every time step. We adopt this method with a slight change; here we provide a brief explanation of it for the sake of completeness.

In this strategy, a local Cartesian mesh covering the object is considered, as shown in Fig. 3. For this mesh we consider a local coordinate system \((x, y)\), which initially coincides with the global coordinate system, denoted here as \((X, Y)\). Note that the resolution of this local mesh is not necessarily the same as the global Cartesian mesh on which the governing equations are solved. Before time-stepping, the distance function is calculated on the local Cartesian grid. During the time integration, the object is moved together with the local coordinate system. Then, the value of the signed distance function for nodes on the global coordinate system can be interpolated from the surrounding nodes of the local mesh. While in their work, Liu and Hu (2014) used a bi-linear interpolation to obtain the desired value for the global mesh, we use a localized radial basis function (Biancolini, 2017), which is more flexible and accurate. As illustrated in Fig. 3(b), to find the value of distance function for an arbitrary node on the global mesh, such as the black circle node in the figure, a block of \(3 \times 3\) grid nodes from the local system \((x, y)\) is selected so that the block covers the desired node on the global mesh.

By using the displacement of mass center \(d\) and the rotation matrix \(R\) of the object, we can obtain the coordinate of the global grid node in the local coordinate system. The rotation matrix for two-dimensional cases is given as

\[
R = \begin{bmatrix}
    \cos \theta & -\sin \theta \\
    \sin \theta & \cos \theta
\end{bmatrix},
\]

where \(\theta\) is the rotation angle of the object around the \(z\) axis. The global coordinate is mapped to the local coordinate by

\[
x = R^{-1} X - d,
\]

where \(x\) is the mapped coordinate of the target node in the local system attached to the object.

Then, the distance function \(\phi\) of the desired node in the local coordinate is calculated by a radial basis interpolation as

\[
\phi(x) = \sum_{i=1}^{M} \beta_i \cdot \psi(||r_i||) + P(x),
\]

where \(||r_i||\) indicates the Euclidean distance between the desired node and the node \(i\) on the local system selected for interpolation, \(\beta_i\) are the weight coefficients that must be calculated, and \(P\) is a linear polynomial which stabilizes the system (Biancolini, 2017) and for two-dimensional problems reads \(P(x) = p_0 + p_1x + p_2y\). Further, \(\psi\) is the basis function. In the present study we employed an identity function for those basis functions (i.e. \(\psi(||r_i||) = ||r_i||\)); therefore in two-dimensional cases we have

\[
\psi(||r_i||) = \sqrt{(x - x_i)^2 + (y - y_i)^2}.
\]

The weight coefficients in Eq. (26) can be obtained by solving the following system:

\[
\begin{bmatrix}
    \phi(r_{11}) & \phi(r_{12}) & \cdots & \phi(r_{1M}) \\
    \phi(r_{21}) & \phi(r_{22}) & \cdots & \phi(r_{2M}) \\
    \vdots & \vdots & \ddots & \vdots \\
    \phi(r_{M1}) & \phi(r_{M2}) & \cdots & \phi(r_{MM})
\end{bmatrix} \begin{bmatrix}
    1 & x_1 & y_1 \\
    1 & x_2 & y_2 \\
    \vdots & \vdots & \vdots \\
    1 & x_M & y_M
\end{bmatrix} = \begin{bmatrix}
    \beta_1 \\
    \beta_2 \\
    \vdots \\
    \beta_M
\end{bmatrix},
\]

\[
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_M
\end{bmatrix} = \begin{bmatrix}
    \phi_1 \\
    \phi_2 \\
    \vdots \\
    \phi_M
\end{bmatrix}.
\]

where \(M\) is the number of grid points used for this interpolation (i.e., the green points in Fig. 3(b)); in this study, \(M = 9\). Having determined the interface, the forcing nodes and their corresponding interpolation stencil, the moving least square method for this study is introduced in the next subsection.
2.2.6. Moving least square approximation

The moving least square method is an approximation method that reconstructs a continuous function around the point on which we need the reconstructed value, via the calculation of the weighted least squares (WLS) measure biased towards the region of interest. The idea is to take an WLS formulation for an arbitrary point, move it over the domain of computation, and compute the WLS for every point individually during this move. Here we present the classical moving least squares for which the notation is adopted from Liu (2009). Given \( n \) nodes located at \( x_j \in \mathbb{R}^d \) (\( j = 1, \ldots, n \)), the WLS approximation for a continuous function \( u(x) \) is written as limited series, i.e.,

\[
    u(x) \approx \sum_{i=1}^{m} p_i(x)a_i(x) = P^T(x)a(x), \tag{29}
\]

where \( p_i(x) \) are basis functions with \( m \) terms. In the present study, we adopt the monomial basis functions, which are a function of the spatial coordinate. The bases and the coefficient matrices can be written as

\[
    p(x)^T = \begin{bmatrix} 1, x, y, x^2, y^2 \end{bmatrix}, \tag{30a}
\]

\[
    a(x)^T = \begin{bmatrix} a_0, a_x, a_y, a_{x^2}, a_{xy}, a_{y^2} \end{bmatrix}. \tag{30b}
\]

The coefficients \( a(x) \) in Eq. (29) can be obtained by minimizing the error functional,

\[
    J(x) = \sum_{j=1}^{n} \left( u^d(x_j) - u_j \right)^2 w(\|x - x_j\|), \tag{31}
\]

where \( u_j = u(x_j) \) and \( u^d(x_j) \) is its discrete approximation that we want to obtain. The error between the desired approximation and the given function is weighted using a positive weight function \( w \), which is a function of Euclidean distance between the evaluation point and the location of the given nodes, as detailed later. The error functional can be written in the matrix form,

\[
    J = (Pa - u)^T W (Pa - u), \tag{32}
\]

where

\[
    P = \begin{bmatrix}
        p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\
        p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\
        \vdots & \vdots & \ddots & \vdots \\
        p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n)
    \end{bmatrix},
\]

\[
    W = \begin{bmatrix}
        w(\|x - x_1\|) & 0 & \cdots & 0 \\
        0 & w(\|x - x_2\|) & \cdots & 0 \\
        \vdots & \vdots & \ddots & \vdots \\
        0 & 0 & \cdots & w(\|x - x_n\|)
    \end{bmatrix}, \tag{34}
\]

and the vector of the given values is

\[
    u^T = [u_1, u_2, \ldots, u_n]. \tag{35}
\]

Equation (31) can be minimized by setting \( \nabla J = 0 \), i.e.,

\[
    \frac{\partial J}{\partial a} = P^T WPa - P^T Wu = 0. \tag{36}
\]

If the so-called momentum matrix, \( M = P^T WP \), is not singular, the value of the coefficient is obtained with

\[
    a(x) = \left( P^T WP \right)^{-1} PWu. \tag{37}
\]

Therefore, the approximated function is computed as

\[
    u^d(x) = P^T \left( P^T WP \right)^{-1} P^T Wu. \tag{38}
\]

The weight function has an important role in the formulation of the moving least square method. These weights ensure that the approximated function is globally continuous by ensuring that nodes enter and leave the compact domain...
smoothly, and thus the shape function compatibility is satisfied. In the literature, a wide variety of weight functions have been proposed (Levin, 2004; Li and Liu, 2002; Biancolini, 2017). Here we use the following weight cubic spline function (Biancolini, 2017) due to its compactness and straightforward formulation for two and three dimensional problems:

\[
w(\tilde{r}) = \begin{cases} 
\frac{2}{3} - 4\tilde{r}^2 + 4\tilde{r}^3 & \text{for} \quad \frac{1}{2} \leq \tilde{r} \leq 1 \\
\frac{2}{3} - 4\tilde{r} + 4\tilde{r}^2 & \text{for} \quad \frac{1}{2} < \tilde{r} \leq 1 \\
0 & \text{for} \quad \tilde{r} > 1 
\end{cases}
\]  

(39)

where \( \tilde{r} \) is the distance from point \( x \) made dimensionless with respect the separation distance, that is, \( \tilde{r} = \|x - x_i\| / r_w \) and \( r_w \) is the maximum distance between the forcing node and the nodes in its interpolation stencil.

It should be emphasized that according to Eq. (38) the shape function and approximation would be available only when the momentum matrix is non-singular. The singularity of this matrix depends on the used basis. For the linear basis (not used in this study), the sufficient condition is that there must be at least three non-co-linear nodes in the support domain for two-dimensional problems. However, for quadratic basis the condition will be more complicated, and at least six nodes are needed in the support domain. Even with more nodes in the support domain, the invertibility of the matrix may not be guaranteed. To circumvent this issue one can enlarge the supporting domain (Li and Liu, 2002), use a modified version of the linear basis (Joldes et al., 2015) or matrix triangularization (Li and Liu, 2002). Although one might solve the singularity problem by an enlarged support domain or modified basis, the system might still be ill-conditioned specifically when the nodes are very close to each other, which usually happens in the field of immersed boundary method. Therefore, further tricks (Wang et al., 2018) must be taken into account for these cases if a very large supporting domain is not favorable or possible. In present study, we have used a singular-value decomposition (SVD)-based pseudo-inverse method, where both singular and ill-conditioned problems are bypassed by truncating the small eigenvalue values (smaller than \( 10^{-10} \)) akin to triangularization techniques. Applying the other methods would be possible in this framework and could be the subject of the future studies.

3. Numerical Results

We numerically carry out a benchmark test, namely the flow inside two concentric cylinders to reveal the spatial and temporal convergence rate of the proposed method. Also, flows around a circular cylinder are considered to validate the present method in steady-state and transient flows. Then, two moving boundary examples are solved. The flow around a predefined oscillation cylinder in a fluid at rest is an illustrative test to study the capability of the method for dealing with the inherent spurious oscillation in immersed boundary problems. Sedimentation of a cylinder in fluid is the last test case we consider to demonstrate the capability for fluid-structure interactions.

3.1. Flow inside two concentric cylinders

Here we simulate the flow between two concentric hollow cylinders as illustrated in Fig. 4(a). This test case is one of the rare cases bearing the exact solution (Taira and Colonius, 2007). The inner cylinder has radius \( r_1 = 0.5 \) whereas the
outer is \( r = 1 \). The angular velocity of the rotating cylinder, i.e. the inner one, is \( \omega_1 \) and is changed smoothly from zero so that we can avoid the sudden initial move which might pose some adverse effects on the time accuracy of the solution, i.e.,

\[
\omega_1 = \frac{u_0(r_1)}{r_1} = 1 + \tanh \left( \frac{t - 0.2}{0.05} \right),
\]

(40)

where the \( u_0 \) is the tangential velocity component and \( t \) represents the time.

The outer cylinder is kept fixed and the initial condition is deemed a quiescent fluid. A solid wall boundary condition is imposed on the outer boundaries of the computational domain whose size is \([0, 2.1] \times [0, 2.1]\). We compute the azimuthal velocity along a radial line \( u_\theta \) \((i = 1, \cdots, N \text{ in } r_1 \leq r \leq r_2)\) and report the convergence rate of its \( L_2 \) and \( L_\infty \) error norms, i.e.,

\[
\| u_\theta - U_\theta \|_2 = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (u_\theta(r) - U_\theta(r))^2},
\]

(41)

\[
\| u_\theta - U_\theta \|_\infty = \max_i |u_\theta(r) - U_\theta(r)|,
\]

(42)

where the analytical solution \( U_\theta(r) \) is given by

\[
U_\theta(r) = \frac{1}{r_2^2 - r_1^2} \left( -\omega_1 r_1^2 r + \frac{r_1^2 r_2^2}{r} \omega_1 \right).
\]

Figure 4(b) shows the spatial convergence rate which is slightly smaller than two for both \( L_2 \) and \( L_\infty \). To obtain this rate, we fixed the time step at \( \Delta t = 4.0 \times 10^{-6} \) and varied space resolution uniformly from \( \Delta x = \Delta y = 1.6 \times 10^{-2} \) to \( 0.41 \times 10^{-2} \) in five steps. According to this figure, the rate of decay for the spatial errors to be 1.5 and about 1.8 for \( L_\infty \) and the \( L_2 \), respectively. Although this rate of convergence is better than diffusive direct forcing methods (Uhlmann, 2005; Taira and Colonius, 2007), which use discrete delta functions to transfer data, further investigation is required to explain why the \( L_2 \)-norm of the error is smaller than two. The impact of the splitting error in the fractional step method as well as time integration is demonstrated in Fig. 4(c) by calculating the temporal convergence rate for a spatial resolution of \( \Delta x = \Delta y = 0.41 \times 10^{-2} \), which confirms that the temporal convergence rate is first order as designed.

3.2. Flow over a fixed circular cylinder

Next, a flow over a fixed circular cylinder is considered as a canonical test case, since a great amount of numerical and experimental studies are available for comparison. The Reynolds number \( Re = u_\infty D/\nu \) based on the inflow velocity \( u_\infty \), the cylinder diameter \( D \), and the kinematic viscosity \( \nu \) determines the flow characters. The simulation is performed in a rectangular domain with a fluid flow from left to right (see, Fig. 5). The uniform flow velocity \( u_\infty = 1 \) in the streamwise \((x)\) direction is imposed at the inlet, the slip boundary condition is imposed at the bottom and top boundaries, and the following convective outlet boundary condition is adopted at the outlet:

\[
\frac{\partial u}{\partial t} = -u_\infty \frac{\partial u}{\partial x}.
\]

(44)

For the pressure, a Neumann boundary condition, \( \partial p/\partial n = 0 \), is imposed on all boundaries. The cylinder is placed 20\( D \) downstream of the inlet. While a uniform coarse mesh, \( \Delta x = \Delta y = 0.04 \), is employed for the whole domain, the grid is refined in a rectangular domain with \([-10D, 10D] \times [-10D, 10D]\) around the cylinder so that \( \Delta x = \Delta y = h \) as shown in Table 1. Note that the coarse and refined meshes are smoothly connected at their interfaces by placing computational cells of intermediate sizes.

For the sake of comparison, the drag and lift coefficients are defined as

\[
C_D = \frac{F_D}{\frac{1}{2} \rho u_\infty^2 D}, \quad C_L = \frac{F_L}{\frac{1}{2} \rho u_\infty^2 D},
\]

(45)

where \( F_L \) and \( F_D \) are the exerted lift and drag force form the fluid, respectively, which can be calculated using the immersed boundary forces in Eq. (4) both on the solid domain and forcing nodes.
3.2.1. Steady case at $Re = 40$

The flow at Reynolds numbers $Re = 40$ is a steady flow having a re-circulation region in the wake. This region is characterized by the length of the circulation $l$, the streamwise distance $a$ from the nearest point on the body surface to the vortex center, the transverse distance $b$ between the pair of vortices, and the angle of separation $\theta$, as depicted in Fig. 5. We perform the simulation with various domain sizes as well as different mesh resolutions to check the grid convergence and ensure that the boundary confinement effect does not impact the solution. The time step is selected small enough ($\Delta t = 0.001$) to satisfy the CFL condition (i.e., the Courant number, $C < 1$). Our numerical experiments revealed that the present method remains stable even with the Courant number close to one.

Figure 6 shows the pressure and vorticity contours and the streamlines, which are observed to be in close agreement with those reported in the literature (Uhlmann, 2005; Taira and Colonius 2007). Table 1 shows the comparison of the wake dimension (i.e., $l$, $a$, $b$, and $\theta$ defined in Fig. 5) and the drag coefficient $C_D$ against those reported in the experimental and other numerical studies. The result shows a good agreement with those reference values. Also, one could conclude that the narrow domain leads to a slightly larger drag coefficient, which was also observed in the previous studies (Uhlmann, 2005; Taira and Colonius 2007).

3.2.2. Periodic vortex shedding cases at $Re = 100$ and 200

The flow over a fixed cylinder becomes unsteady and a periodic vortex shedding emerges when the Reynolds number is increased further, as can be observed in Fig. 7. In Fig. 8, the time evolution of the drag and lift for those Reynolds number are plotted. Note that the oscillating frequency of the lift is indeed the vortex shedding frequency, $f_s$, and the

Table 1 Comparison of the steady-state wake dimension and the force coefficient for flow over a circular cylinder at $Re = 40$. The results with * are experimental ones.

|                 | $l/D$ | $a/D$ | $b/D$ | $\theta(\degree)$ | $C_D$ |
|-----------------|-------|-------|-------|-------------------|-------|
| Coutanceau and Bauard (1977)* | 2.13  | 0.76  | 0.59  | 53.5              | -     |
| Wang and Zhang (2011)     | 2.36  | 0.72  | 0.6   | 53.8              | 1.54  |
| Taira and Colonius (2007) | 2.30  | 0.73  | 0.60  | 53.7              | 1.54  |
| Present ($\Omega = 70D \times 30D$, $h = 0.04$) | 2.38  | 0.77  | 0.59  | 52.0              | 1.58  |
| ($\Omega = 70D \times 30D$, $h = 0.02$)    | 2.34  | 0.76  | 0.62  | 52.0              | 1.58  |
| ($\Omega = 70D \times 30D$, $h = 0.02$)    | 2.36  | 0.77  | 0.60  | 53.1              | 1.59  |
| ($\Omega = 70D \times 40D$, $h = 0.02$)    | 2.36  | 0.75  | 0.62  | 54.5              | 1.54  |
| ($\Omega = 70D \times 60D$, $h = 0.02$)    | 2.34  | 0.76  | 0.62  | 54.5              | 1.54  |
frequency of the drag is twice that of the lift coefficient (Balaras, 2004).

Table 2 summarizes the mean drag coefficient $C_D$, the amplitude of drag coefficient $C'_D$ and lift coefficient $C'_L$ as well as the Strohal number, $St = f_s D/U_\infty$. For the sake of a better comparison, the result of the experimental and numerical studies have been provided. In the numerical section, both body-fitted and immersed boundary methods are considered. A good agreement has been obtained against those well-established results. With the reference values from a body-fitted method published by Liu et al. (1998), the mean drag coefficient shows approximately merely 3% over-prediction for the smaller domain while other IBM methods such as Uhlmann (2005) overpredicts it around 11%. From Table 2, it can be seen that the accuracy will be improved by enlarging the computational domain. Further, since our method is not iterative unlike the methods of Breugem (2012) and Ji et al. (2012), the present method can be much more efficient in terms of computational costs.

3.3. In-line oscillating cylinder in fluid at rest

A predefined oscillatory motion of a cylinder in a fluid at rest, that has been frequently reported in the literature, was studied with both experimental (Dütsch et al., 1998) and numerical methods (Kor, 2017; Kor et al., 2018; Liao et al., 2010). In particular, this test case can reveal spurious oscillations in the force imposed by the immersed boundary method and is utilized to investigate the effect of treatments for those oscillations (Seo and Mittal, 2013).

The key dimensionless parameters in this flow are the Reynolds number, $Re = U_{max} D/\nu$ and the Keulegan-Carpenter number, $KC = U_{max}/(fD)$, where $U_{max}$ and $f$ are maximum velocity of the cylinder and the characteristic frequency of it. In the present work, we utilized the set up employed in experimental and numerical study with a non-inertial reference
Table 2 Comparison of the Strouhal number and force coefficients for the flow over the circular cylinder at Re = 100 and Re = 200. The results with * has been obtained from experimental study.

| Re = 100 | Williamson (1989)* | - | - | - | 0.164 |
|----------|-------------------|---|---|---|-------|
|          | Uhlmann (2005)     | 1.453 ± 0.011 | ± 0.339 | 0.169 |
|          | Ji et al. (2012)   | 1.376 ± 0.010 | ± 0.339 | 0.169 |
|          | Braza et al. (1986)| 1.359 ± 0.019 | ± 0.293 | 0.16 |
|          | Liu et al. (1998)  | 1.350 ± 0.012 | ± 0.339 | 0.165 |
|          | Mineau et al. (2015)| 1.40 ± 0.010 | ± 0.32 | 0.165 |
|          | Xu and Wang (2006) | 1.423 ± 0.013 | ± 0.34 | 0.171 |
|          | Present (Ω = 70D x 30D, h = 0.04) | 1.401 ± 0.011 | ± 0.344 | 0.162 |
|          | (Ω = 70D x 30D, h = 0.02) | 1.410 ± 0.011 | ± 0.339 | 0.162 |
|          | (Ω = 70D x 30D, h = 0.01) | 1.421 ± 0.011 | ± 0.348 | 0.162 |
|          | (Ω = 70D x 40D, h = 0.02) | 1.389 ± 0.011 | ± 0.344 | 0.162 |
|          | (Ω = 70D x 60D, h = 0.02) | 1.378 ± 0.011 | ± 0.337 | 0.162 |

| Re = 200 | Williamson* | - | - | - | 0.197 |
|----------|-------------|---|---|---|-------|
|          | Taira and Colonius (2007) | 1.35 ± 0.048 | ± 0.68 | 0.196 |
|          | Ji et al. (2012) | 1.354 ± 0.044 | ± 0.682 | 0.20 |
|          | Braza et al. (1986) | 1.386 ± 0.040 | ± 0.766 | 0.20 |
|          | Liu et al. (1998) | 1.31 ± 0.049 | ± 0.69 | 0.192 |
|          | Mineau et al. (2015) | 1.44 ± 0.05 | ± 0.75 | 0.200 |
|          | Xu and Wang (2006) | 1.42 ± 0.04 | ± 0.66 | 0.202 |
|          | Present (Ω = 70D x 30D, h = 0.04) | 1.365 ± 0.044 | ± 0.687 | 0.201 |
|          | (Ω = 70D x 30D, h = 0.02) | 1.375 ± 0.046 | ± 0.702 | 0.201 |
|          | (Ω = 70D x 30D, h = 0.01) | 1.382 ± 0.048 | ± 0.713 | 0.201 |
|          | (Ω = 70D x 40D, h = 0.02) | 1.368 ± 0.044 | ± 0.692 | 0.201 |
|          | (Ω = 70D x 60D, h = 0.02) | 1.355 ± 0.045 | ± 0.692 | 0.201 |

reported by Dütsch et al. (1998). The motion of the cylinder is predefined by its center location $X_C$, as

$$X_C(t) = -U_{max} \sin(2\pi ft).$$

The Reynolds number and Keulegan-Carpenter number are set at Re = 100 and KC = 5, respectively.

The computational domain has a size of $40D \times 40D$ in $x$ and $y$ direction, and the cylinder is initially located at the center. A uniform grid with resolution of $\Delta x = \Delta y = 0.025$ is used to discretize the domain. An appropriate time step is chosen to keep the Courant number about $C = 0.55$ (Liao et al., 2010). Started from the static condition, the simulation continues until periodic vortex shedding is fully developed.

Figure 9 shows the pressure and vorticity contours at four different phase angle ($\varphi = 2\pi ft = 0^\circ, 96^\circ, 192^\circ, 288^\circ$) after vortex shedding begins. The vortices contour are drawn from −3 to 3 with an increment of 0.4 displaying the same structure as Dütsch et al. (1998). As the cylinder moves, upper and lower boundary layers are developed at the leading face of the cylinder and separated while the cylinder moves through them. When cylinder goes in the reverse direction, it splits the already stirred flow (i.e., vortices remaining in the field). These are consistent with the experimental observation (Dütsch et al., 1998).

In Fig. 10, the profile of velocity components are depicted in three different angle phase ($\varphi = 2\pi ft = 180^\circ, 210^\circ, 330^\circ$) at four different streamwise location ($x = -0.6D, 0D, 0.6D, 1.2D$). The experimental results from Dütsch et al. (1998) are also plotted for comparison. The present results agree well with those from Dütsch et al. (1998) in the fluid domain. It is worth mentioning that according to this figure, the forcing treatment inside the solid phase does not influence the fluid phase and just provides a smooth transition between the fluid and the solid domains.

Figure 11 shows the evolution of the drag coefficient ($C_D = F_D / [(1/2) \rho U_{max}^2 D]$) versus dimensionless time (with the period of the oscillation) at the grid resolution of $\Delta x = \Delta y = 0.02$. In this figure, the present method is compared with reconstruction direct forcing methods, and the body-fitted numerical result of the Guilmineau and Queutey (2002). The proposed method shows a much smoother drag coefficient evolution with a better agreement with the body-fitted method than the original direct forcing methods. In fact, this is the great merit of the present method, which motivated us to develop it.

3.4. A freely falling circular cylinder in the fluid at rest

Finally, the dynamics of free falling solid circular cylinder is simulated to demonstrate the ability of the present method to solve two-way fluid-structure interaction without pre-defined velocity. In this regard, the falling of a two-dimensional cylinder in a rectangular tank filled with a Newtonian fluid is considered. The configuration of this test case

[DOI: 10.1299/jfst.2021jfst0013] © 2021 The Japan Society of Mechanical Engineers
is same as those of Glowinski et al. (2001). That means, the computational domain is a rectangle, (2 cm × 6 cm), and the diameter of the cylinder is $D = 0.25$ cm while the initial location for the center of the cylinder center is $(x, y) = (1$ cm, $4$ cm). The density of the fluid is $\rho_f = 1$ g/cm$^3$ and that of the cylinder is $\rho_s = 1.25$ g/cm$^3$. We also take the viscosity as $\nu = 0.1$ cm$^2$/s. The mesh sizes are 1/36 and 1/64 of the cylinder diameter, that is $\Delta x = \Delta y = (1/144)$ cm and $\Delta x = \Delta y = (1/256)$ cm. For all of these cases, 300 Lagrangian markers are considered on the interface; thus, the geometric effects are similar for all cases.

By solving equations (8)–(12) and (16), the location of the cylinder is updated and the corresponding immersed
Fig. 11 The comparison of the time evolution of the drag force for a cylinder oscillating in a fluid at rest. The solid black lines obtained with the present method, the red line is obtained from the reconstruction method (Kor et al., 2018). The orange circular symbols indicate the body-fitted result of Guilmineau and Queutey (2002).

Table 3 Comparison of the maximum Reynolds number during the sedimentation of a cylinder ($\rho_s = 1.25 \text{ g/cm}^3$, $\nu = 0.1 \text{ cm}^2/\text{s}$).

| Present       | Glowinski et al. (2001) | Wan and Turek (2007) |
|---------------|-------------------------|----------------------|
| $h$ (cm)      | 1/72                    | 1/144                |
| $Re_{\text{max}}$ | 16.962                  | 17.216               |

boundary forces at each time step is calculated. We also considered a modeled collision force between cylinder and wall to prevent it from interpenetration to the wall. Aiming at keeping the physical distance between objects, Glowinski et al. (2001) proposed a short-range repulsive force model. Singh et al. (2003) modified that model by allowing a little overlap between objects. Combining these two models, Wan and Turek (2007) suggested their own repulsive model, which is utilized in the present study, i.e.,

$$F_w = \begin{cases} 
\frac{1}{\epsilon'} (X_i - X'_i) (2R_i - d'_i) & d_{i,j} < 2R_i \\
\frac{1}{\epsilon} (X_i - X'_i) (2R_i + \xi - d'_i) & 2R_i \leq d_{i,j} < 2R_i + \xi \\
0 & 2R_i + \xi \leq d_{i,j} 
\end{cases}$$

(47)

where $R_i$ is the radius of cylinder, $X_i$ is its center, $X'_i$ is the coordinate vector of the imaginary cylinder located on the physical boundary, and $d_{i,j} = |X_i - X'_j|$. The parameter $\xi$ is the range of the repulsive force, while $\epsilon$ and $\epsilon'$ represent the small positive parameters for the collision. For the results provided here we took $\epsilon' = (\Delta x)^2$, $\epsilon = \Delta x$, and $\xi = \Delta x/10$. Further, the maximum Reynolds number is defined as

$$Re_{\text{max}} = \max(Re(t)) = \max \left[ \frac{\sqrt{u_x(t)^2 + v_y(t)^2}}{\nu} \cdot D \right]$$

(48)

where $u_x(t)$ and $v_y(t)$ are the $x$ and $y$ components of the cylinder velocity.

The comparison of the maximum Reynolds number during the settling in the fluid with previous numerical studies is given in Table 3. The maximum Reynolds numbers predicted under different grid resolution agree well with those of the other well-established methods (Glowinski et al., 2001; Wan and Turek, 2007). Table 4 shows the maximum Reynolds number obtained at a fixed resolution, $\Delta x = (1/256) \text{ cm}$, with different time step size $\Delta t$, suggesting that the convergence rate for $\Delta t$ is approximately one.

The vorticity contour at four different time steps in Fig. 12 show that the flow field is symmetric and virtually no rotation happens. Also, the vorticity contours are in good agreement with the previous numerical results (Glowinski et al., 2001).
Table 4  Dependence of the maximum Reynolds number on the time step size $\Delta t$ ($\rho_s = 1.25\, \text{g/cm}^3$, $\nu = 0.1\, \text{cm}^2/\text{s}$ and $(1/256)$ cm.)

| $\Delta t$ (s) | $\Re_{\text{max}}$ |
|----------------|---------------------|
| $5.0 \times 10^{-5}$ | 17.323 |
| $6.25 \times 10^{-6}$ | 17.316 |
| $3.125 \times 10^{-6}$ | 17.307 |

Figure 12  Vorticity isolines around the falling cylinder with $\rho_s = 1.25\, \text{g/cm}^3$ and $\nu = 0.1\, \text{cm}^2/\text{s}$ at different time instants.

Figure 13 quantitatively compares the vertical and horizontal components for the center of the cylinder as well as the vertical location of it during the settlement with two different resolution. These results indicate that the numerical solution are well-converged and are in good agreement with those reported by the other researchers (Glowinski et al., 2001; Kor et al., 2018). In addition, Fig. 14 shows two other important quantities, i.e., the transnational kinetic energy $E_t = (1/2)M_s(u(t)^2 + v(t)^2)$ and the rotational kinetic energy $E_r = (1/2)I_s\omega_s(t)^2$, where $M_s$ and $I_s$ are the mass and the momentum of the inertia of the cylinder, and $\omega_s(t)$ stands for instantaneous angular velocity. It can be seen that the result computed on two different resolutions are essentially the same except for the rotational kinetic energy due to its very small amount ($O(10^{-4})$), which is attributed to the numerical errors.

3.5. Flow induced by a hovering elliptical wing

The hovering of an elliptical wing in a fluid initially at rest is considered to demonstrate the capability of the proposed algorithm for the cases with rotational movement. The wing undergoes a prescribed motion determined by the following horizontal movement of the center of mass and the rotation of the body:

$$X(t) = \frac{1}{2}X_{\text{max}} \cos (2\pi ft), \quad (49a)$$

$$\alpha(t) = \alpha_0 + \alpha_{\text{max}} \sin (2\pi ft), \quad (49b)$$

where $X(t)$ is the time dependent location of the mass center in $x$ direction and $\alpha$ is the counter-clockwise angle of the wing with respect to the positive $x$ axis. For the sake of comparison, the parameters are set similar to Eldredge (2007): the initial angle is $\alpha_0 = \pi/4$, the rotational amplitude is $\alpha_{\text{max}} = \pi/4$, and the translational amplitude is $X_{\text{max}} = 2.8\, c$, where $c$ denotes the chord length of the wing. The period of the oscillation is set at $T = 1/f = 4$ and the maximum translational velocity is $8.8\, c/T$. The Reynolds number is set at $\Re = 75$ based on the maximum velocity and the chord length. Also, the $y$ coordinate of the mass center is kept zero ($Y(t) = 0$), and the ratio of the chord length to short axis of the wing ($c/b$) is 10. No mean flow is assumed.

Figure 15 shows snapshots of the vorticity field during $2 \leq t/T \leq 4$. In comparison with other studies such as Eldredge (2007), our results indicate that the proposed algorithm can properly cope with problems with moving and rotating boundary. It should be noted that Eldredge (2007) obtained the results by a vortex particle method with the same configuration.
Fig. 13 Three representative quantities of a falling cylinder dynamics: (a) x-coordinate for the center of the cylinder during the sedimentation; (b) y-coordinate for the center of the cylinder; (c) y-component of the cylinder velocity. The green lines are for the result with $\Delta x = (1/256)$ cm, while the red ones are for $\Delta x = (1/144)$ cm.

Fig. 14 Kinetic energy of the cylinder with different resolution; the green dash line is for $\Delta x = \Delta y = (D/64)$ cm and the red solid one is for $\Delta x = \Delta y = (D/36)$ cm: (a) translational kinetic energy; (b) rotational kinetic energy. The green lines are for the result with $\Delta x = (1/256)$ cm, while the red ones are for $\Delta x = (1/144)$ cm.

Figure 16 compares the computed lift and drag coefficients with those of Eldredge (2007). The present results are obtained with $\Delta x = \Delta y = 0.009$ using the proposed algorithm, which show a good agreement with the reference despite the differences in the methods.

4. Conclusions

In this paper, we have proposed a more accurate and relatively simple sharp interface direct forcing immersed boundary method based on a two-step predictor-corrector projection strategy for solving the system of the governing equations. For the consistency consideration, we adopted the second order in time Kim-Moin scheme for the projection in the proposed method. Regarding the direct immersed boundary force term, it is calculated after prediction step, in which a time step of the projection method is taken without considering the immersed object, and distributed on the whole solid domain while the data is transferred between the Eulerian and Lagrangian nodes by using a direct moving least square approximation. In contrast to the traditional moving least square immersed boundary methods, this approach succeeded in keeping the boundary conditions sharp while reducing the spurious numerical oscillations.

We have performed several numerical experiments to illustrate the accuracy and capability of the proposed method, including flow induced by rotating cylinder inside a larger stationary hollow cylinder for convergence test, the flow past a stationary cylinder with different Reynolds number, the flow over an oscillating cylinder with predefined motion and finally the sedimentation of a cylinder due to its weight in a fluid at rest. In particular, our numerical results shows that the convergence rate of the velocity field, being close to the second order, is apparently better than those of the first order projection methods. It has been showed that our numerical is capable enough to be considered as an accurate and relatively simple method for the fluid structure interaction problem and has high potential for the other field such as heat...
Fig. 15  Vorticity field of a hovering elliptical cylinder. The contours are drawn with 40 discrete levels between \(-20\) and \(20\).

Fig. 16  Lift and drag coefficients for a hovering elliptical wing.

transfer and two phase problems. As this method utilizes a versatile stencil which is not defined along any arbitrary line, it can be straightforwardly extended to the three dimensional problems. Also, the proposed method takes into account more than one nodes on the solid-fluid interface, which yields more accurate boundary capturing capability, particularly in three dimensional problems. Extension to three dimensional problem is one of our future research plans. Besides,
the cell center strategy adopted in this work and the implicit interface representation pave the way for the future works on more efficient methods such as adaptive mesh refinement solvers equipped with immersed boundary method for fluid solid interaction problems.

References

Balaras, 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.
Bell, J. B., Colella, P., and Glaz, H. M., A second-order projection method for the incompressible Navier-Stokes equations, Journal of Computational Physics, Vol. 85 (1989), pp. 257–283.
Breugem, W. P., A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows, Journal of Computational Physics, Vol. 231 (2012), pp. 4469–4498.
Biancolini, M. E., Fast Radial Basis Functions for Engineering Applications, Springer International Publishing (2017).
Borazjani, I., Ge, L., and Sotiropoulos, F., Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies, Journal of Computational Physics, Vol. 227 (2008), pp. 7587–7620.
Braza, M., Chassaing, P. H., and Minh, H. H., Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, Journal of Fluid Mechanics, Vol. 165 (1986), pp. 79–130.
Coutanceau, M. and Bouard, R., Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. steady flow, Journal of Fluid Mechanics, Vol. 79 (1977), pp. 231–256.
De, A. K., A diffuse interface immersed boundary method for complex moving boundary problems. Journal of Computational Physics, Vol. 366 (2018), pp. 226–251.
Ding, H., Shu, C., Yeo, K. S., and Xu, D., Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method, Computer Methods in Applied Mechanics and Engineering, Vol. 193 (2004), pp. 727–744.
Eldredge, J. D., 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.
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.
Glowinski, R., Pan, T. W., Hasla, T. I., Joseph, D. D., Periaux 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.
Guermond, J. L., Minev, P., and Shen, J., An overview of projection methods for incompressible flows, Computer Methods in Applied Mechanics and Engineering, Vol. 195 (2006), pp. 6011–6045.
Guilmineau, E. and Queutey, P., A numerical simulation of vortex shedding from an oscillating circular cylinder, Journal of Fluids and Structures, Vol. 16 (2002), pp. 773–794.
Günther, C., Meinke, M., and Schröder, W., A flexible level-set approach for tracking multiple interacting interfaces in embedded boundary methods, Computers & Fluids, Vol. 102 (2014), pp. 182–202.
Ji, C., Munjiza, A., and Williams, J. J. R., A novel iterative direct-forcing immersed boundary method and its finite volume applications, Journal of Computational Physics, Vol. 231 (2012), pp. 1797–1821.
Joldes, G. R., Chowdhury, H. A., Wittek, A., Doyle, B., and Miller, K., Modified moving least squares with polynomial bases for scattered data approximation, Applied Mathematics and Computation, Vol. 266 (2015), pp. 893–902.
Kim, J. and Moin, P., Application of a fractional-step method to incompressible Navier-Stokes equations, Journal of Computational Physics, Vol. 59 (1985), pp. 308–323.
Kor, H., Development of an efficient immersed boundary method for simulation of flows around stationary and moving bodies, PhD Thesis, Keio University (2018).
Kor, H., Badri Ghomizad, M., and Fukagata, K., Extension of the unified interpolation stencil for immersed boundary method for moving boundary problems, Journal of Fluid Science and Technology, Vol. 13, No. 2 (2018), DOI:10.1299/jfst.2018jfst0008.
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, No. 1 (2017), DOI:10.1299/jfst.2017jfst0011.
Lee, W., Jung, E., Kang, S., and Hur, N., On the momentum interpolation scheme for collocated meshes with improved
discrete energy conservation, Journal of Mechanical Science and Technology, Vol. 33 (2019), pp. 2761–2768.
Lee, J. and You, D., An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations, Journal of Computational Physics, Vol. 15 (2013), pp. 295–314.
Levin, D., Mesh-independent surface interpolation, Geometric Modeling for Scientific Visualization, Springer (2004), p. 37.
Li, S. and Liu, W. K., Meshfree and particle methods and their applications, Applied Mechanics Reviews, Vol. 55 (2002), pp. 1–34.
Liao, C. C., Chang, Y. W., Lin, C. A., and McDonough, J. M., Simulating flows with moving rigid boundary using immersed-boundary method, Computers & Fluids, Vol. 39 (2010), pp. 152–167.
Liu, G. R., Meshfree Methods: Moving Beyond the Finite Element Method, Second Edition, Taylor & Francis (2009).
Liu, C. and Hu, C., An efficient immersed boundary treatment for complex moving object, Journal of Computational Physics, Vol. 274 (2014), pp. 654–680.
Liu, C., Zheng, X., and Sung, C. H., Preconditioned multigrid methods for unsteady incompressible flows, Journal of Computational Physics, Vol. 139 (1998), pp. 35–57.
Luo, H., Dai, H., de Sousa, P. J., and Yin, B., On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries, Computers & Fluids, Vol. 15 (2012), pp. 61–76.
Martins, D. M., Albuquerque, D. M., and Pereira, J. C., Continuity constrained least-squares interpolation for SFO suppression in immersed boundary methods, Journal of Computational Physics, Vol. 336 (2017), pp. 608–626.
Mimeau, C., Gallizio, F., Cottet, G. H., and Mortazavi, I., Vortex penalization method for bluff body flows, International Journal for Numerical Methods in Fluids, Vol. 79 (2015), pp. 55–83.
Mirzaie, D., Analysis of moving least squares approximation revisited, Journal of Computational and Applied Mathematics, Vol. 282 (2015), pp. 237–250.
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.
Mohd-Yusof, J., Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, Center for Turbulence Research Annual Research Briefs, Vol. 161 (1997), pp. 317–327.
Nicolaou, L., Jung, S. Y., and Zaki, T. A., A robust direct-forcing immersed boundary method with enhanced stability for moving body problems in curvilinear coordinates, Computers & Fluids, Vol. 119 (2015), pp. 101–114.
Nishida, H. and Tajiri, K., Numerical simulation of incompressible flows around a fish model at low Reynolds number using seamless virtual boundary method, Journal of Fluid Science and Technology, Vol. 4, No. 3 (2009), DOI:10.1299/jfst.4.500.
Nishikawa, K., Konno, K., and Hattori, Y., Assessment of immersed boundary method as a tool for direct numerical simulation of aeroacoustic sound, Journal of Fluid Science and Technology, Vol. 15, No. 1 (2020), DOI:10.1299/jfst.2020jfst0004.
Peller, N., Duc, A. L., Tremblay, F., and Manhart M., High-order stable interpolations for immersed boundary methods, International Journal for Numerical Methods in Fluids, Vol. 52 (2006), pp. 1175–1193.
Peskin, C. S., Flow patterns around heart valves: a numerical method, Journal of Computational Physics, Vol. 10 (1972), pp. 252–271.
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.
Singh, P., Hesla, T. I., and Joseph D. D., Distributed Lagrange multiplier method for particulate flows with collisions, International Journal of Multiphase Flow, Vol. 29 (2003), pp. 495–509.
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.
Taira, K. and Colonius, T., The immersed boundary method: a projection approach, Journal of Computational Physics, Vol. 225 (2007), pp. 2118–2137.
Tajiri, K., Nishida, H., and Tanaka, M., Property of seamless immersed boundary method for large eddy simulation of incompressible turbulent flows, Journal of Fluid Science and Technology. Vol. 9 No. 2 (2014), DOI:10.1299/jfst.2014jfst0020.
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.
Vanella, M. and Balaras, E., A moving-least-squares reconstruction for embedded-boundary formulations, Journal of computational physics, Vol. 228 (2009), pp. 6617–6628.

Wan, D. and Turek, S., An efficient multigrid-FEM method for the simulation of solid–liquid two phase flows, Journal of Computational and Applied Mathematics, Vol. 203 (2007), pp. 561–580.

Wang, X. Y., Yu, P., Yeo, K. S., and Khoo, B. C., SVD–GFD scheme to simulate complex moving body problems in 3D space, Journal of Computational Physics, Vol. 229 (2010), pp. 2314–2338.

Wang, S. and Zhang, X., An immersed boundary method based on discrete stream function formulation for two-and three-dimensional incompressible flows, Journal of Computational Physics, Vol. 230 (2011), pp. 3479–3499.

Wang, Q., Zhou, W., Cheng, Y., Ma, G., Chang, X., Miao, Y., and Chen, E., Regularized moving least-square method and regularized improved interpolating moving least-square method with nonsingular moment matrices, Applied Mathematics and Computation, Vol. 325 (2018), pp. 120–145.

Williamson, C. H. K., Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers, Journal of Fluid Mechanics, Vol. 206 (2006), pp. 579–627.

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.

Yang, J. and Stern, F., A non-iterative direct forcing immersed boundary method for strongly-coupled fluid–solid interactions, Journal of Computational Physics, Vol. 295 (2015), pp. 779–804.

Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W., An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, Journal of Computational Physics, Vol. 156 (1999), pp. 209–240.

Yuki, Y., Takeuchi, S., and Kajishima, T., Efficient immersed boundary method for strong interaction problem of arbitrary shape object with the self-induced flow, Journal of Fluid Science and Technology, Vol. 2, No. 1 (2007), DOI:10.1299/jfst.2.1.