Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **13**, No. 1, pp. 83-100

DOI: 10.4208/aamm.OA-2019-0354
February 2021

A Novel Immersed Boundary Method Implemented by Imposing Reconstructed Velocity on Virtual Boundary

Jingyang Wang and Chunhua Zhou*

Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics,
Nanjing, Jiangsu 210016, China

Received 11 December 2019; Accepted (in revised version) 11 May 2020

Abstract. In this paper, a novel immersed boundary method is presented for simulating incompressible flows governed by Navier-Stokes equations. A virtual boundary is formed by the cell edges (for two-dimensional cases) in the vicinity of the immersed boundary. In the domain with the virtual boundary, the governing equations can be solved in the conventional way. Reconstructed velocity is imposed on the virtual boundary, which is determined via the interpolation along the direction normal to the wall and in conjunction with the no-slip condition for the actual boundary. For ”freshly cleared nodes” on the virtual boundary encountered in moving-boundary problems, pressure at the previous time step is reconstructed by solving the local simplified momentum equation. In the test case for an analytical solution, the local accuracy of pressure is verified to be of the second order. In order to further validate the present method, various flows over the stationary and/or moving circular cylinder and NACA0012 airfoil have been simulated. The obtained results agree well with the available numerical or experimental data in the published literatures.

AMS subject classifications: 76D05, 76M25

Key words: Immersed boundary method, boundary condition, convergence rate, solution reconstruction.

1 Introduction

The immersed boundary (IB) methods have attracted a lot of attention in the numerical simulation of moving-boundary flows. Using the IB methods, the governing equations for the flow of fluid can be solved on a fixed mesh and there is no need to update the mesh at each time step. Hence, the computational cost for mesh updating and solution transferring between the old mesh and the new one can be eliminated, and the errors due to solution transferring can also be avoided.

*Corresponding author.
Email: chzhou@nuaa.edu.cn (C. H. Zhou)

The IB method was firstly proposed by Peskin for simulating cardiac mechanics and associated blood flow [1]. The idea of his method is to transmit the effect of the solid boundary into the momentum equations as forcing terms, thereby the no-slip boundary condition is satisfied. Mittal and Iaccarino classified roughly the IB methods into two categories [2]: one is termed continuous forcing approach (diffuse interface approach) in which the forcing term is incorporated explicitly into the momentum equations before discretization and another is the discrete forcing approach (sharp interface approach) in which the forcing term is introduced explicitly or implicitly after discretization. The continuous forcing approach is very attractive for elastic solid boundaries while it inherently cannot provide a sharp representation of rigid boundaries.

The essence of many discrete forcing IB methods [3–5] is to reconstruct the solution at ″forcing points″ in the vicinity of the immersed boundary via the local approximate form of solution so that the spatial discrete form of the governing equations can be closured and the no-slip boundary condition can be enforced. In the ghost-cell IB method [3] and the local domain-free discretization method [5], some exterior cell-centers or exterior nodes near the immersed boundary are selected to be forcing points. In the hybrid Cartesian immersed boundary (HCIB) method [4, 6–8], some interior cell-centers or interior nodes near the immersed boundary are selected. When the second order spatial approximation is employed for the governing equations, the global and local accuracy of velocity is of the second order and the global accuracy of pressure is also of the second order [9,10]. However, in the work of [10,11], the local accuracy of pressure is reported to be around the first order. In many other published literatures, e.g., [3, 8, 12], the authors do not provide the result of the local accuracy of pressure.

In this work, a novel immersed boundary method is proposed, in which the corrected boundary conditions are enforced on a virtual boundary and then the governing equations can be solved in the same way as the conventional boundary-conforming methods. The virtual boundary consists of the cell-edges of background mesh and the correction of boundary condition is implemented by reconstructing the velocity at the nodes of virtual boundary. Our numerical experiment will show that the local accuracy of pressure obtained by the present method can reach the second order.

The remainder of this paper is arranged as follows. In Section 2, the governing equations and the basic numerical schemes are presented. Section 3 is the main part of the paper, in which the treatment of the immersed boundary is described. In Section 4, a numerical experiment for the verification of the convergence rate is carried out and flows past a circular cylinder and a NACA0012 airfoil are simulated to further validate the present IB method. Finally, in Section 5, summary and conclusion are given.

2 Governing equations and basic numerical schemes

In this work, two-dimensional incompressible flows of the fluid of constant density are governed by Navier-Stokes equations in the following form

$$\mathbf{I}^m \cdot \frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}_i}{\partial x_i} = \frac{\partial \mathbf{s}_i}{\partial x_i}, \quad i=1,2, \tag{2.1}$$

where $\mathbf{w}$ is the vector of flow variables, $\mathbf{f}_i$ and $\mathbf{s}_i$ are the convective and viscous flux vectors, respectively,

$$\mathbf{w} = \begin{bmatrix} p \\ u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{f}_i = \begin{bmatrix} u_i \\ u_1 u_i + p\delta_{1i} \\ u_2 u_i + p\delta_{2i} \end{bmatrix}, \quad \mathbf{s}_i = \begin{bmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \end{bmatrix}, \tag{2.2}$$

and $\mathbf{I}^m = diag(0,1,1)$ is the modified identity matrix. In (2.2), u_i represents the velocity components, p the pressure and τ_{ij} the viscous stress tensor. It is well known that the governing equations (2.1) have a unique solution when the no-slip condition is applied at the solid boundary.

We suppose that $\Omega \subset R^2$ is a connected open set containing one or more solid bodies. A triangulation of Ω for spatial discretization is denoted by T_h. In this work, the Garlerkin finite element approach proposed by Mavriplis and Jameson [13] is employed for spatial discretization.

The flow variables are stored at the vertices of triangles and they are approximated by piecewise linear functions. Let $\mathbf{F}$ denote the convective flux tensor and $\mathbf{S}$ the viscous flux tensor. After applying the spatial discretization and employing the concept of a lumped mass matrix, the semi-discrete form of Navier-Stokes equations (for an interior node P) can be written as

$$\mathbf{I}^m \cdot \frac{\partial (\mathbf{\Omega}_\mathrm{P} \mathbf{w}_\mathrm{P})}{\partial t} = \sum_{e=1}^{n_e} \frac{\mathbf{F}^\mathrm{A} + \mathbf{F}^\mathrm{B}}{2} \cdot \mathbf{L}_\mathrm{AB} - \sum_{e=1}^{n_e} \frac{3}{2} \cdot \mathbf{S}^e \cdot \mathbf{L}_\mathrm{AB}, \tag{2.3}$$

where the summations are over all the triangles sharing node P and Ω_P represents the sum of area of all these triangles, as illustrated in Fig. 1. In formula (2.3), $\mathbf{L}_\mathrm{AB}$ denotes

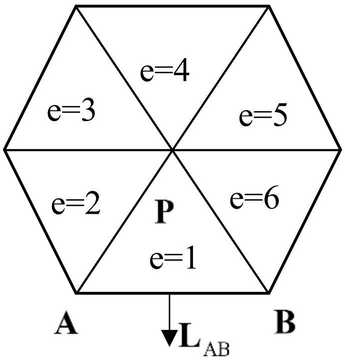


Figure 1: Influence domain of an interior node P.

the directed (outward normal) length of the triangle edge opposite to P, $\mathbf{F}^{\mathrm{A}}$ and $\mathbf{F}^{\mathrm{B}}$ are the convective fluxes at the both ends of this edge, and $\mathbf{S}^{e}$ is the viscous flux at each triangle.

An artificial dissipation constructed as a biharmonic operator [14] is adopted to prevent odd-even decoupling. At the far-field, the non-reflecting boundary conditions are constructed by the linearized characteristics approach [14]. The treatment of the immersed boundary will be discussed in the next section.

In the computation of unsteady flows, a dual time-stepping scheme [15] is employed for time integration of the semi-discrete equations (2.3). The third-order temporal discretization is adopted in physical time marching and the five-stage, point-implicit scheme proposed by Melon et al. [16] is used in pseudo time marching. To address the issue of "freshly cleared" nodes for moving-boundary problems [14], the physical time step must satisfy

$$\Delta t \leq \frac{h_{\min}}{3\max(|V_1^n|,|V_2^n|)}, \tag{2.4}$$

where V_i denotes the velocity components of the body motion and $h_{\min}$ the minimum mesh interval in the region near the solid wall.

3 Treatment of the immersed boundary

In the present IB method, the actual immersed boundary is replaced by a virtual boundary, which consists of the cell-edges of the background mesh. Instead of the no-slip condition, the corrected boundary condition is enforced on the virtual boundary. Then, the governing equations can be solved in the same way as the conventional boundary-conforming methods. Therefore, the local accuracy of pressure is expected to be of the second order.

3.1 Construction of virtual boundary

The first step in the treatment of the immersed boundary is to construct the virtual boundary, which consists of the selected cell-edges of the fixed background mesh. To reduce the interpolation error resulting from the correction of boundary condition which will be discussed later, the virtual boundary should be as close as possible to the actual one. As described below, the construction of virtual boundary can be implemented in a simple way.

For a given discretization of the actual immersed boundary (consisting of non-overlapping segments for two-dimensional case), all the mesh nodes are classified into two categories: fluid nodes (nodes in the fluid phase) and solid nodes (nodes in the solid phase). Here, a cell edge both ends of which are fluid nodes is called a fluid edge. If one of the two nodes opposite to a fluid edge is a fluid node and another is a solid node, this edge is selected to be a segment of the virtual boundary. In Fig. 2, the fluid edge D_1D_2

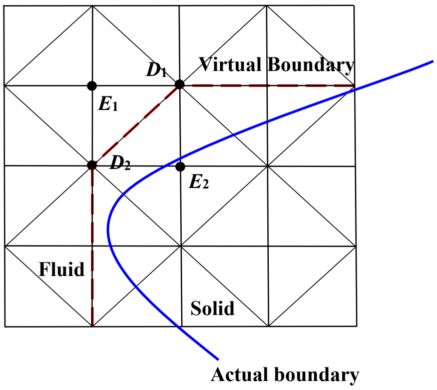


Figure 2: The virtual boundary consisting of mesh edges. Red dashed line: virtual boundary; blue solid line: actual boundary; D_1, D_2: virtual boundary nodes (the two end nodes of edge D_1D_2); E_1, E_2: the two opposite nodes of edge D_1D_2.

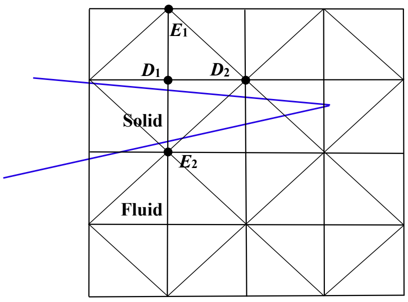


Figure 3: A special case near the thin portion of a solid body. Blue solid line: actual boundary; D_1, D_2: VB nodes (the two end nodes of edge D_1D_2); E_1, E_2: the two opposite nodes of edge D_1D_2.

is illustrated as an example, where D_1 and D_2 are the two end nodes of this edge and E_1 and E_2 are the two opposite nodes of this edge.

There exists a special case in the region near the thin portion of a solid body (for example, the trailing edge of an airfoil), the width of which is smaller than one mesh interval. In this special case, if the two end nodes of a fluid edge both are above or below the solid body, and if one of the two opposite nodes is above the body while the other is below the body, this edge is also selected to be a segment of the virtual boundary. In Fig. 3, the fluid edge D_1D_2 is illustrated as an example, where D_1 and D_2 (both above the solid body) are the two end nodes of this edge and E_1 (above the body) and E_2 (below the body) are the two opposite nodes of this edge.

After the virtual boundary has been defined, all fluid nodes can be divided into two groups: interior nodes and virtual boundary (VB) nodes. The virtual boundary will be updated at each time step when the actual boundary is moving. Obviously, in the present

IB method, all the solid nodes are blanked out of computation.

From the construction of the virtual boundary, we can see that compared to the ghost-cell IB method [3] or the local domain-free discretization method [5], where the forcing points are located inside the solid phase, the tedious task of handling the multi-valued points (usually located in the vicinity of thin portion of the body) can be avoided.

The main issue to be addressed in extending the present IB method into three dimensions is the construction of the virtual boundary. In three-dimensional case, the virtual boundary should consist of the cell faces in the fluid phase which are closest to the solid wall. For a cell face in the fluid phase, if one of its two opposite nodes is a fluid node and another is a solid node, this face can be selected to be a triangular element of the virtual boundary.

3.2 Corrected boundary condition for virtual boundary

Obviously, the no-slip condition cannot be applied directly at the virtual boundary and Dirichlet condition for velocity at the virtual boundary must be corrected. This correction is implemented by reconstructing the velocity at each node of the virtual boundary. In this work, the reconstructed velocity is obtained by the interpolation along the wall normal and between the velocity at a point in the flow field and that at the actual boundary. To conduct the interpolation for a given VB node, a reference point on the wall normal passing through the VB node must be defined. The reference point may not be a mesh node, at which the solution will be evaluated approximately in an appropriate way. To reduce the interpolation error, this reference point should be close to the actual boundary as possible. As illustrated in Fig. 4, we define the reference point F to be the intersection between the normal line and the cell-edge nearest to the actual boundary, the two end points (C_2 and C_3) of which are the interior nodes. Then, the value of any flow variable at the reference point can be obtained by the following linear interpolation over the segment C_2C_3

$$\phi^F = \frac{\phi^{C_2}|FC_3|+\phi^{C_3}|C_2F|}{|C_2C_3|}, \tag{3.1}$$

where ϕ^F, ϕ^{C_2} and ϕ^{C_3}, represent the flow variables at the points F, C_2 and C_3, respectively, and the absolute value is the distance between two points. From the definition of a reference point, we can see that for moving-boundary problems the reference point corresponding to a given VB node must be updated at each physical time step.

According to the no-slip condition, at the wall-intercept point W, the velocity of the fluid is equal to that of the body. Then, the velocity at the VB node D can be reconstructed by interpolation between the wall-intercept point W and the reference point F

$$\mathbf{u}^D = \frac{|FD|\mathbf{V}^W+|WD|\mathbf{u}^F}{|FW|}, \tag{3.2}$$

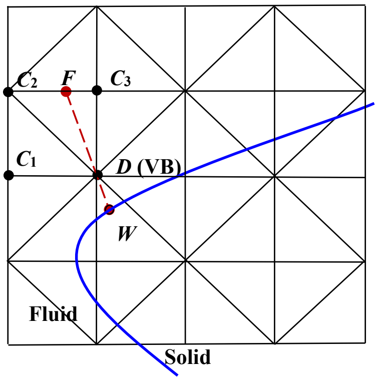


Figure 4: Schematic diagram of interpolation for correction of boundary condition. Blue solid line: solid boundary; red dashed line: wall normal; C_1, C_2, C_3: interior nodes; D: VB node; W: normal-wall intersection point; F: reference point.

where $\mathbf{u}^D$ and $\mathbf{u}^F$ denote the velocities at D and F respectively, and $\mathbf{V}^w$ denotes the velocity of the body motion at W. In Eq. (3.2), the velocity at the reference point $\mathbf{u}^F$ is taken from the velocity field at the previous pseudo time step. So, in the dual-time stepping scheme, the boundary conditions at the virtual boundary are treated explicitly in pseudo time. At each physical time step, the solution of governing equations is found by marching to a steady state in pseudo time [14,15].

After the velocities at all VB nodes are reconstructed, the corrected Dirichlet condition is specified. Thus, the governing equations are solved on the solution domain with a virtual boundary.

Obviously, the above approach for the correction of boundary conditions at the virtual boundary in two dimensions can be extended into three dimensions in a straight forward way.

3.3 Evaluation of the pressure at the previous time step for some VB nodes

In the dual time-stepping scheme, for the pseudo time iteration at the current physical time step, the initial values of solution are usually taken to be those at the previous time step. For moving-boundary problems, due to the limitation of the physical time step (2.4), any interior fluid nodes at the current physical time step must be a fluid node at the previous time step. But, some VB nodes at the current time step (e.g., D_1 and D_2 in Fig. 5) were in the solid phase at the previous time step. They are "freshly cleared" VB nodes. Therefore, the flow variables at these nodes have no physical values for the previous step. It means that the pressure at some VB nodes has no appropriate initial values for the pseudo time iteration at the current physical time step. Due to this issue, the convergence of the pseudo time iteration deteriorates greatly, as shown in our numerical experiments. On the other hand, the lack of physical values of pressure at these

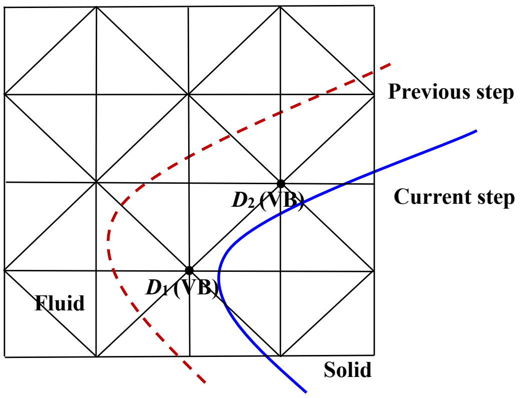


Figure 5: "Freshly cleared" VB nodes. Blue solid line: actual boundary at current step; red dashed line: actual boundary at previous step; D_1, D_2: VB nodes at current step.

"freshly cleared" VB nodes for the previous time step will result in the spurious force oscillations [10]. Therefore, to improve the convergence of the pseudo time iteration and reduce the nonphysical oscillations, the pressure at the "freshly cleared" VB nodes for the previous time step must be reconstructed.

In this work, the value of pressure at a "freshly cleared" VB node for the previous time step is obtained by using the local simplified momentum equation. Ignoring the viscous term and utilizing the no-slip boundary condition, the simplified momentum equation in the region near the wall can be written as

$$\frac{\partial p}{\partial n} = -\left(\frac{d\mathbf{V}}{dt}\right)_{\gamma} \cdot \mathbf{n}, \tag{3.3}$$

where n denotes the outward normal direction of the wall, $\mathbf{V}$ the known velocity vector of the body motion, and γ the solid boundary. After approximating the partial differential derivative at the left hand side of (3.3), p_D, the pressure at the VB node D, is obtained as following (with reference to Fig. 4)

$$p_D = p_F + |FD|\left(\frac{d\mathbf{V}}{dt}\right)_{\gamma} \cdot \mathbf{n}, \tag{3.4}$$

where p_F is the pressure at the reference point F.

The neighbor nodes of a reference point (C_2 and C_3 in Fig. 4) are fluid nodes at the current physical time step. Due to the limitation of the time step (2.4), the solid object never transverses an entire mesh interval within one step. So, the two neighbor nodes must also be fluid nodes at the previous time step. Therefore, the neighbor nodes and then the reference point can always have the physical values at the previous time step.

4 Numerical experiments

In this section, the convergence rate is firstly verified for the present method. Then, numerical experiments for the flows past stationary and moving bodies are carried out to validate the feasibility of the method. To enhance the local resolution, the meshes are refined in the vicinity of the immersed boundary except for the test cases of the verification of convergence rate.

4.1 Verification of convergence rate

We utilize the exact solution of Navier-Stokes equations with a forcing term to verify the convergence rate of the present method. The exact solution is provided by Frutos and Bosco in [17] and is described as follows

$$\begin{cases} u_1 = 2\pi\sin^2(\pi x_1)\sin(\pi x_2)\cos(\pi x_2), \\ u_2 = -2\pi\sin^2(\pi x_2)\sin(\pi x_1)\cos(\pi x_1), \\ p = 20{x_1}^2 x_2. \end{cases} \tag{4.1}$$

The forcing term and Dirichlet boundary conditions are derived from the above analytical solution. As shown in Fig. 6, the solution domain is chosen to be $\Omega\backslash\omega$, where $\Omega = [-0.5,0.5]\times[-0.5,0.5]$ and ω is a circle with the radius of 0.25 and the center at $(0,0)$. We measure the errors in terms of L_∞, L_1 and L_2 norms. L_1 and L_2 can represent the global error and L_∞, which effectively captures the error around the immersed boundary, can represent the local error. Four, uniform and successively refined meshes are used to perform error analysis and verify the convergence rate. The initial (first level) mesh is illustrated in Fig. 6.

The exact solution (4.1) is denoted by ϕ. The mesh interval, the discrete solution and the error of discretization on the kth level mesh are denoted by h_k, ϕ_h^k and e^k, respectively.

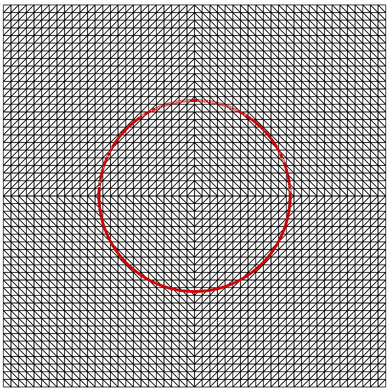


Figure 6: Initial mesh for verification of convergence rate.

Table 1: Errors and orders of convergence rates for u_1.

Mesh interval	Global error				Local error	
	L_1 error	order	L_2 error	order	L_∞ error	order
2.0×10^{-2}	1.84×10^{-3}	-	2.61×10^{-3}	-	1.73×10^{-2}	-
1.0×10^{-2}	4.33×10^{-4}	2.09	6.17×10^{-4}	2.08	4.78×10^{-3}	1.86
5.0×10^{-3}	1.09×10^{-4}	1.99	1.53×10^{-4}	2.01	1.38×10^{-3}	1.78
2.5×10^{-3}	2.95×10^{-5}	1.89	4.20×10^{-5}	1.87	3.86×10^{-4}	1.84

Table 2: Errors and orders of convergence rates for p.

Mesh interval	Global error				Local error	
	L_1 error	order	L_2 error	order	L_∞ error	order
2.0×10^{-2}	8.21×10^{-3}	-	1.06×10^{-2}	-	7.20×10^{-2}	-
1.0×10^{-2}	2.10×10^{-3}	1.97	2.65×10^{-3}	2.00	1.86×10^{-2}	1.95
5.0×10^{-3}	5.15×10^{-4}	2.03	6.41×10^{-4}	2.05	5.29×10^{-3}	1.81
2.5×10^{-3}	1.12×10^{-4}	2.20	1.40×10^{-4}	2.20	1.55×10^{-3}	1.78

Assume the discrete solution is a γth-order approximation to the exact one. Then, we have

$$\phi_h^k-\phi=e^k=Ch_k^\gamma, \tag{4.2}$$

where C is a constant independent of mesh interval. Therefore, the order of accuracy or convergence rate can be calculated as following

$$\gamma=\frac{\log\left(\|e^k\|/\|e^{k-1}\|\right)}{\log 2}. \tag{4.3}$$

The errors and the orders of convergence rates measured by different norms for velocity u_1 (identical to u_2) and pressure p are presented in Table 1 and Table 2, respectively.

It can be seen from the tables that both velocity and pressure are globally and locally second order accurate. Especially, by using the present discrete-forcing IB method, the local accuracy of pressure can reach the second order. It should be noticed that the local accuracy of pressure was reported to be around the first order in [10, 11] and was not mentioned in [3, 8, 12]. Therefore, the advantage of the present method is validated.

4.2 Flows over a stationary circular cylinder

To validate the present method for practical problems, we firstly simulate the flows over a stationary circular cylinder. The definition of Reynolds number is based on the diameter of the cylinder D. For steady flows, Reynolds numbers are set to be 10, 20 and 40. For unsteady flows, Reynolds numbers are 100 and 200. The computation domain is $[-15D,35D]\times[-15D,15D]$ and the center of the cylinder is located at (0,0). The mesh

Table 3: Comparison of drag coefficient and reattachment length for steady flows.

	Re$=$10		Re$=$20		Re$=$40	
	C_D	L/D	C_D	L/D	C_D	L/D
Present work	2.83	0.24	2.05	0.91	1.53	2.23
Park et al. [18] (BFD)	2.78		2.01		1.51	
Dennis et al. [19] (BFD)	2.85	0.27	2.05	0.94	1.52	2.35
Choi et al. [11] (BFD)			1.98	0.92	1.49	2.21
Choi et al. [11] (IB)			2.02	0.90	1.52	2.25
Cai et al. [20] (IB)					1.59	2.36
Taira et al. [21] (IB)			2.06	0.94	1.54	2.30
Tritton [22] (EXP)	2.86		2.08		1.59	

interval in the vicinity of the solid boundary is about $1.2\times10^{-2}D$. The non-dimensional size of time step is set to be 10^{-2}.

The results for steady flows at Re$=$10, 20 and 40, including drag coefficient and reattachment length, are shown in Table 3. Some published experimental data and numerical results of body-fitted grid (BFG) and IB methods are also given in the table for comparison.

Fig. 7 shows the variations of pressure coefficient and spanwise vorticity along the cylinder surface for Re$=$40. Numerical results of Cai et al. [20], Choi et al. [11] with the IB methods and Braza et al. [23] with the body-fitted grid method are also plotted in Fig. 7 for comparison. Following the work of Braza et al. the pressure coefficient shown in this figure is defined as

$$C_p=\Big(p-p_0+\frac{1}{2}\rho_\infty U_\infty^2\Big)\Big/\Big(\frac{1}{2}\rho_\infty U_\infty^2\Big),$$

where p_0 is the pressure at the front stagnation point.

Together with the published experimental data and numerical results of body-fitted grid and IB methods, the results for unsteady flows at Re$=$100,200, including drag and

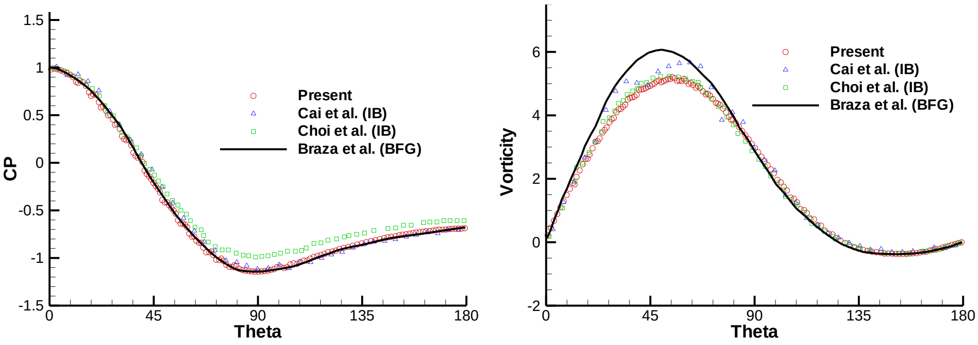


Figure 7: Flow variables along the cylinder surface for Re$=$40 (left: pressure coefficient; right: vorticity).

Table 4: Comparison of drag and lift coefficients and Strouhal number for unsteady flows.

	Re = 100			Re = 200		
	C_D	C_L	St	C_D	C_L	St
Present work	1.30±0.006	±0.315	0.159	1.29±0.040	±0.64	0.192
Liu et al. [24] (BFD)	1.35±0.012	±0.339	0.164	1.31±0.049	±0.69	0.192
Rosenfeld et al. [25] (BFD)				1.31±0.040	±0.65	0.20
Stalberg et al. [26] (BFD)	1.32±0.009	±0.33	0.166			
Taira et al. [21] (IB)				1.35±0.048	±0.68	0.196
Cai et al. [20] (IB)	1.38±0.010	±0.35	0.160	1.37±0.046	±0.70	0.200
Choi et al. [11] (IB)	1.34±0.011	±0.315	0.164	1.36±0.048	±0.64	0.191
Tritton [22] (Exp)	1.27					
Wille [27] (Exp)				1.30		

Table 5: Global errors and orders of convergence rates for Re = 20.

Mesh interval	u_1		u_2		p	
	L_1 error	order	L_1 error	order	L_1 error	order
4.0×10^{-2}	5.39×10^{-2}	-	3.56×10^{-2}	-	1.29×10^{-1}	-
2.0×10^{-2}	1.66×10^{-2}	1.71	1.14×10^{-2}	1.65	4.06×10^{-2}	1.68
1.0×10^{-2}	3.69×10^{-3}	2.17	2.81×10^{-3}	2.03	1.33×10^{-2}	1.62

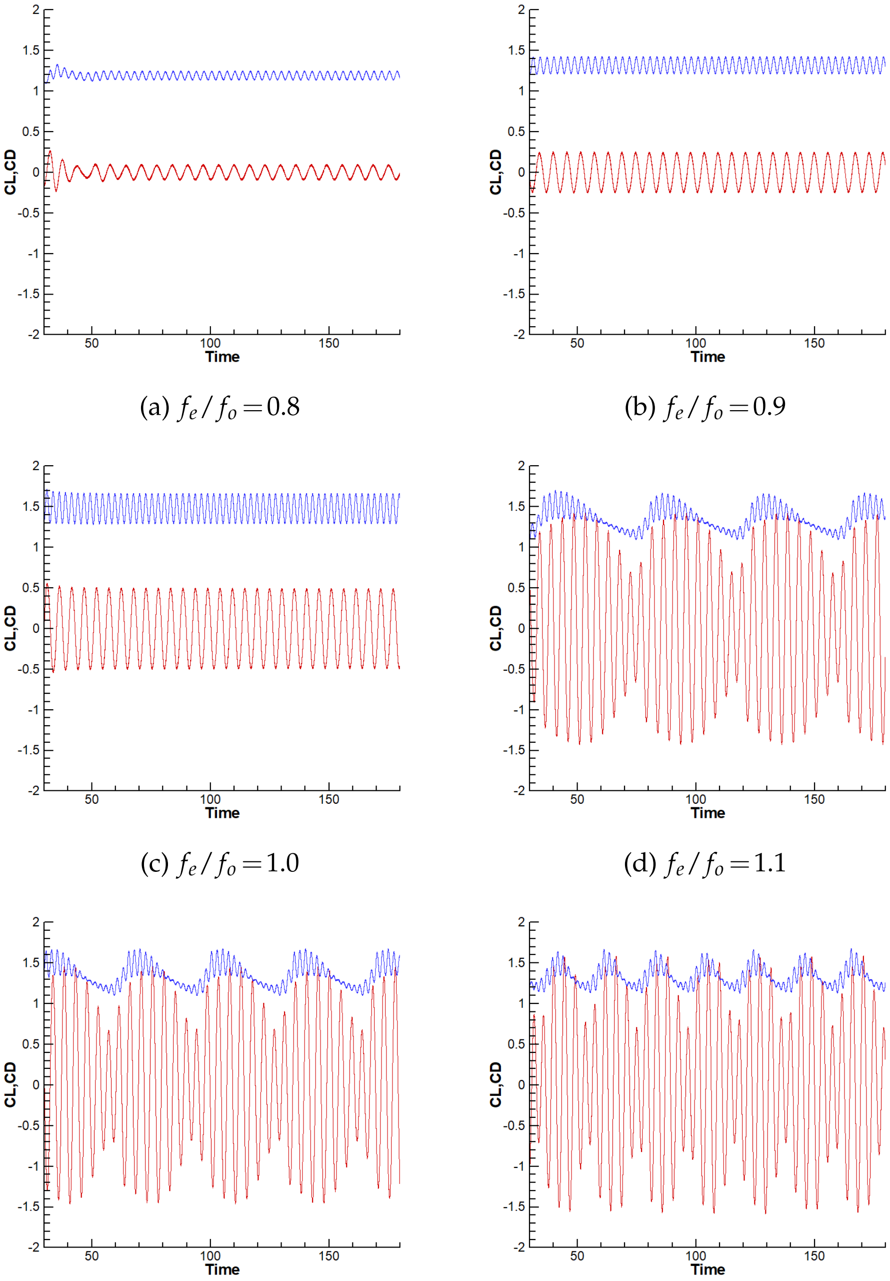
lift coefficients and Strouhal number, are shown in Table 4.

The comparisons in Fig. 7, Table 3 and Table 4 show that the results obtained by the present method fall into a good agreement with the published numerical results or experimental data.

We also examine the accuracy of the present method for the flow past a circular cylinder at Re = 20. For this test, to reduce the computational cost, we employ a small computational domain with the size of $4D\times4D$. The flow is then computed on a hierarchy of uniform meshes (101×101, 201×201, 401×401 and 801×801). Since an exact solution for this case does not exist, the numerical solution computed on the highly resolved 801×801 mesh is used as a baseline for computing the truncation error. The global errors measured in L_1 norm and the orders of convergence rates for two velocity components and pressure are presented in Table 5. Nearly second order convergence of the global errors can be observed. Some deviations from strict second-order convergence should be attributed to the lack of a true exact solution of the governing equations and the small computational domain.

4.3 Flows over a transversely oscillating circular cylinder

In this subsection, flows over a circular cylinder oscillating transversely in a free stream are simulated to demonstrate the ability of the present method for moving-boundary problems. The motion of the cylinder is described by the following harmonic function of



(a) $f_e/f_o=0.8$

(b) $f_e/f_o=0.9$

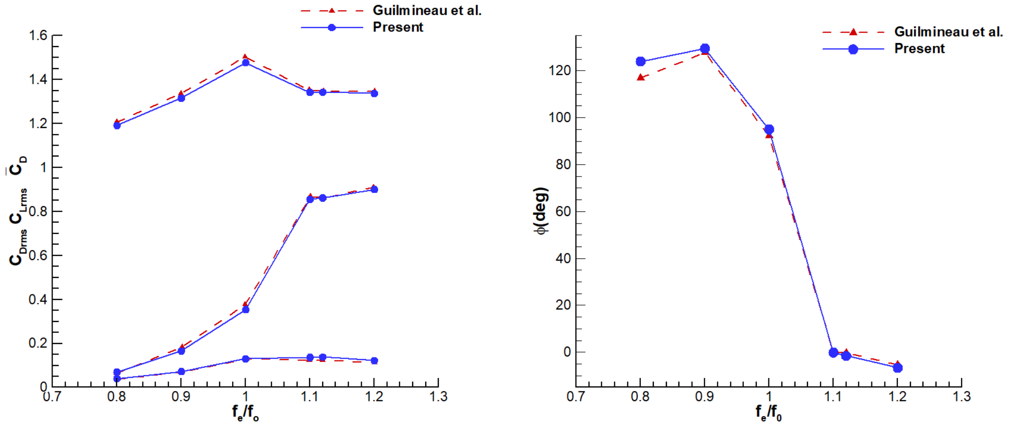
(c) $f_e/f_o=1.0$

(d) $f_e/f_o=1.1$

(e) $f_e/f_o=1.12$

(f) $f_e/f_o=1.2$

Figure 8: Drag (blue) and lift (red) coefficients versus time for an oscillating cylinder.



(a) (b)

Figure 9: Force coefficients (a) and phase angle (b) versus frequency f_e/f_o.

time

$$x_2^c(t) = -A\sin(2\pi f_e t), \tag{4.4}$$

where x_2^c denotes the location of the cylinder center, A the amplitude of oscillation, and f_e the frequency of oscillation. The simulations are performed at Re$=$185, $A/D=0.2$ and $0.8 \leq f_e/f_o \leq 1.2$, where f_o is the natural shedding frequency. The used mesh is the same as that for the stationary circular cylinder and the non-dimensional time step is set to be 5×10^{-3}.

Fig. 8 shows the evolution of the drag and lift coefficients. The results agree well with those in [28,29].

To make a quantitative comparison, together with the body-fitted grid result obtained by Guilmineau et al. [29], the variations of the time-averaged drag coefficient ($\bar{C}_D$) and the root mean square values of drag and lift coefficients ($C_{D\text{rms}}$ and $C_{L\text{rms}}$) with f_e/f_o are exhibited in Fig. 9(a), and the phase angle (ϕ) between the lift coefficient and the vertical displacement of the cylinder is also presented in Fig. 9(b). We can see that the two numerical results agree well with each other.

4.4 Flows over a heaving-pitching airfoil

To validate the applicability of the present IB method for complex moving-boundary flows, we consider a NACA0012 airfoil heaving and pitching in a free-steam of velocity U_∞. The heave motion $h(t)$ and pitch motion $\theta(t)$ are described by harmonic functions as below

$$h(t) = h_0\sin(\omega t), \quad \theta(t) = \theta_0\sin(\omega t + \psi). \tag{4.5}$$

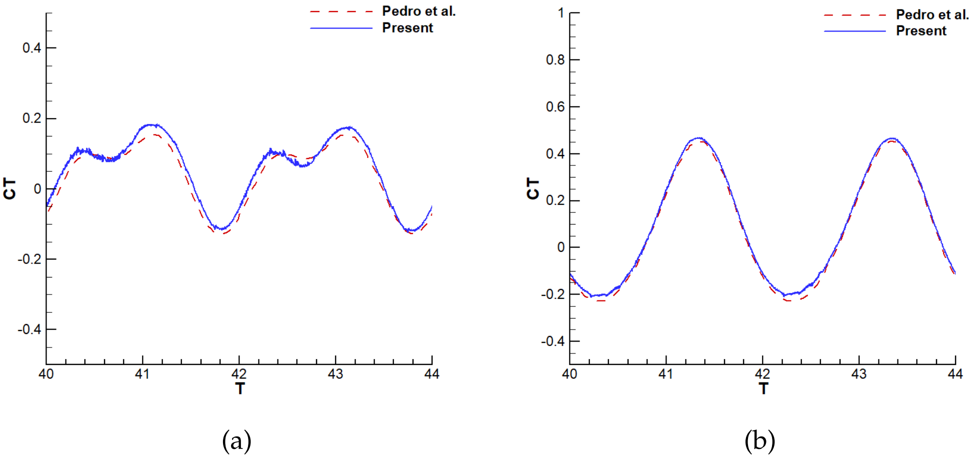


Figure 10: Thrust coefficient versus time (a): $\psi=70^\circ$; (b): $\psi=90^\circ$.

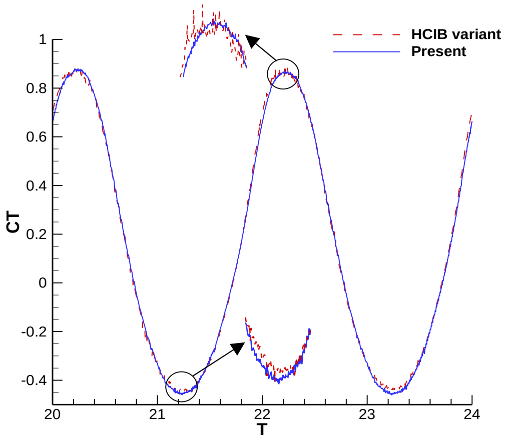


Figure 11: Comparison of the histories of thrust coefficient ($\psi=115^\circ$).

In the above formula, h_0 denotes the amplitude of heaving, θ_0 the amplitude of pitching, ω the frequency of heaving and pitching and ψ the phase difference between heaving and pitching. The chord length b of the airfoil is chosen to be the reference length. Five principal parameters of computation are fixed: (1) $h_0/b=0.50$, (2) $\theta_0=30^\circ$, (3) the dimensionless distance of the pivot point from the leading edge $d/b=0.25$, (4) $St=\frac{h_0\omega}{\pi U_\infty}=0.25$ and (5) $\mathrm{Re}=1100$. The mesh spacing is about $0.0063b$ in the vicinity of the airfoil and the non-dimensional time step is set to be 0.0025.

In Fig. 10, the time histories of thrust coefficient for $\psi=70^\circ$ and $\psi=90^\circ$ are compared with the body-fitted grid result of Pedro et al. [30]. The agreement between the two numerical results is satisfactory. Some small oscillations occur in the present result, which is the inherent feature of the sharp-interface IB methods [10].

For comparison, we construct an IB method by incorporating the treatment of an immersed boundary in HCIB [8] into the present spatial discretization, which can be regarded as a HCIB variant. In Fig. 11, the histories of thrust coefficient for $\psi=115^\circ$ obtained by the present method and the HCIB variant are plotted together. It can be observed from the partial view that, when using the present method, the non-physical oscillations are reduced considerably and the variation of thrust coefficient with time becomes smoother.

5 Summary and conclusions

Imposing the reconstructed velocity on the virtual boundary formed by the cell edges of background mesh, a novel immersed boundary method is proposed in this work. The reconstructed velocity is obtained via the interpolation along the direction normal to the wall and in conjunction with the actual no-slip condition. By the present method, the governing equations can be solved in the conventional way. Compared to the ghost-cell IB method [3] or the local domain-free discretization method [5], the tedious task of handling the multi-valued points can be avoided.

Verification for the convergence rate of solution has been performed. Both velocity and pressure are globally and locally second order accurate. Numerical experiments for benchmark problems are carried out for the further validation of the present IB method. The results show good agreements with the published experimental or numerical data and a slight improvement has also been observed in the simulation of moving-boundary problems.

Acknowledgements

This work has been supported by National Natural Science Foundation of China under Grant No. 11972191.

References

[1] C. S. Peskin, *Flow patterns around heart valves: a numerical method,* J. Comput. Phys., 10 (1972), pp. 220–252.

[2] R. Mittal and G. Iaccarino, *Immersed boundary methods,* Ann. Rev. Fluid Mech., 37 (2005), pp. 239–261.

[3] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas and A. von Loebbeck, *A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries,* J. Comput. Phys., 227 (2008), pp. 4825–4852.

[4] Y. Zhang and C. H. Zhou, *An immersed boundary method for simulation of inviscid compressible flows,* Int. J. Numer. Methods Fluids, 74 (2014), pp. 775–793.

[5] C. H. Zhou and C. Shu, *A local domain-free discretization method to simulate three-dimensional compressible inviscid flows,* Int. J. Numer. Methods Fluids, 61 (2009), pp. 970–986.

[6] J. Mohd-Yusof, *Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries,* Ctr Annual Research Briefs, Stanford University, NASA Ames, 1997.

[7] E. A. Fadlun, R. Verzicco, P. Orlandi and J. Mohd-Yusof, *Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations,* J. Comput. Phys., 161 (2000), pp. 35–60.

[8] A. Gilmanov and F. Sotiropoulos, *A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies,* J. Comput. Phys., 207 (2005), pp. 457–492.

[9] P. H. Chiu, R. K. Lin and W. H. Sheu Tony, *A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier–Stokes equations in time-varying complex geometries,* J. Comput. Phys., 229 (2010), pp. 4476–4500.

[10] Y. Zhang and C. H. Zhou, *Reduction of numerical oscillations in simulating moving-boundary problems by the local DFD method,* Adv. Appl. Math. Mech., 8 (2016), pp. 145–165.

[11] J. I. Choi, R. C. Oberoi, J. R. Edwards and J. A. Rosati, *An immersed boundary method for complex incompressible flows,* J. Comput. Phys., 224 (2007), pp. 757–784.

[12] C. H. Zhou, C. Shu and Y. Z. Wu, *Extension of domain-free discretization method to simulate compressible flows over fixed and moving bodies,* Int. J. Numer. Methods Fluids, 53 (2007), pp. 176–199.

[13] D. J. Mavriplis and A. Jameson, *Multigrid solution of Navier–Stokes equations on triangular meshes,* AIAA J., 28(8) (1990), pp. 1415–1425.

[14] C. H. Zhou and C. Shu, *A local domain-free discretization method for simulation of incompressible flows over moving bodies,* Int. J. Numer. Methods Fluids, 66 (2011), pp. 162–182.

[15] A. Belov, L. Martinelli and A. Jameson, *A new implicit algorithm with multigrid for unsteady incompressible flow calculations,* 33rd Aerospace Sciences Meeting and Exhibit, Reno NV, (1995), 9–12.

[16] N. D. Melson, M. D. Sanetrik and H. L. Atkins, *Time-accurate Navier–Stokes calculations with multigrid acceleration,* Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, Cooper Mountain, CO, USA, April 1993.

[17] Javier de Frutos and G. A. Bosco, *A posteriori error estimations for mixed finite-element approximations to the Navier-Stokes equations,* J. Comput. Appl. Math., 236 (2011), pp. 1103–1122.

[18] J. Park, K. Kwon and H. Choi, *Numerical solutions of flow past a circular cylinder at Reynolds number up to 160,* KSME Int. J., 12 (1998), pp. 1–20.

[19] S. C. R. Dennis and G. Z. Chang, *Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100,* J. Fluid Mech., 42 (1970), pp. 471–489.

[20] S. G. Cai, A. Ouahsine, J. Favier and Y. Hoarau, *Moving immersed boundary method,* Int. J. Numer. Methods Fluids, 85 (2017), pp. 288–323.

[21] K. Taira and T. Colonius, *The immersed boundary method: A projection approach,* J. Comput. Phys., 225 (2007), pp. 2118–2137.

[22] D. J. Tritton, *Experiments on the flow past a circular cylinder at low Reynolds numbers,* J. Fluid Mech., 6 (1959), pp. 547–567.

[23] M. Braza, P. Chassaing and H. Haminh, *Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder,* J. Fluid Mech., 166 (1986), pp.

79–130.

[24] C. Liu, X. Sheng and C. H. Sung, *Preconditioned multigrid methods for unsteady incompressible flows,* J. Comput. Phys., 139 (1998), pp. 35–57.

[25] M. Rosenfeld, D. Kwak and M. Vinokur, *A fractional step solution method for the unsteady incompressible Navier–Stokes equations in generalized coordinate systems,* J. Comput. Phys., 94 (1991), pp. 102–137.

[26] E. Stalberg, A. Brüger, P. Lötstedt, A. V. Johansson and D. S. Henningson, *High order accurate solution of flow past a circular cylinder,* J. Sci. Comput., 27 (2006), pp. 431–441.

[27] R. Wille, *Karman vortex streets,* Adv. Appl. Mech., 6 (1960), pp. 273–291.

[28] J. Yang and E. Balaras, *An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries,* J. Comput. Phys., 215 (2006), pp. 12–40.

[29] E. Guilmineau and P. Queutey, *A numerical simulation of vortex shedding from an oscillating cylinder,* J. Fluids Structures, 16(6) (2002), pp. 773–794.

[30] G. Pedro, A. Suleman and N. Djilali, *A numerical study of the propulsive efficiency of a flapping hydrofoil,* Int. J. Numer. Methods Fluids, 42 (2003), pp. 493–526.