

## A Sharp-Interface Immersed Boundary Method for Simulating Incompressible Flows with Arbitrarily Deforming Smooth Boundaries

Zuo Cui\*,†, Zixuan Yang\* and Hong-Zhou Jiang†

\**Department of Mechanical Engineering, University of Minnesota  
Minneapolis 55455, USA*

†*Department of Mechatronics Engineering  
Harbin Institute of Technology  
Harbin 150001, P. R. China*

Wei-Xi Huang

*Department of Mechanical Engineering, Tsinghua University  
Beijing 100084, P. R. China*

Lian Shen

*Department of Mechanical Engineering and St. Anthony Falls Laboratory  
University of Minnesota, Minneapolis 55455, USA  
shen@umn.edu*

Received 6 November 2016

Accepted 4 January 2017

Published 16 February 2017

We develop a sharp interface immersed boundary (IB) method to simulate the interactions between fluid flows and deformable moving bodies. Fluid–solid interfaces are captured using a level-set (LS) function, which is updated at every time step by a reinitialization procedure. Motions of solid bodies are dynamically coupled with fluid flows by calculating the fluid forces exerted on solid bodies. The accuracy and robustness of the LS-based IB method are tested systematically in the context of several benchmark cases and self-propelled fish swimming. The effects of computational parameters on the accuracy of deformable body capturing are analyzed. It is found that the algorithm performs well in simulating the flow motions surrounding the deforming and moving bodies.

**Keywords:** Computational fluid dynamics; fluid–structure interaction; immersed boundary method; level-set method.

### 1. Introduction

The problem of fluid–structure interactions (FSI) is encountered in many scientific and engineering applications, such as the aero-elastic response of airplane wings [Kamakoti and Shyy (2004)], wind-excited vibration of turbine blades

[Löhner *et al.* (2015)], blood flows through heart valves [Sotiropoulos and Borazjani (2009)], and fish swimming [Kern and Koumoutsakos (2006)]. While analytical solutions for FSI problems are limited due to the complex nature of the problem, experiments and numerical simulations have been extensively conducted to investigate the flow physics associated with FSI. Numerical simulations are often employed to provide flow details in time and three-dimensional (3D) space and reveal the underlying flow dynamics. Because the flow fields are strongly affected by the presence of structures and the structure motions are coupled with the fluid flows, FSI problems pose considerable challenges to simulations in terms of numerical methods and computational costs [Dowell and Hall (2001)].

A key technical issue in numerical methods for simulating FSI problems is how to capture the geometries of structures that act as the boundaries of fluid flows. In general, boundary-capturing methods can be categorized into the conforming-mesh method and immersed boundary (IB) method [Mittal and Iaccarino (2005)]. In the conforming-mesh method, fluid–solid interfaces are explicitly treated as physical boundaries, and computational meshes are conformed to the interfaces. For moving or deformable solid bodies, it is usually expensive to regenerate meshes and interpolate flow fields onto new grid points at every time step [Dowell and Hall (2001); Mittal and Iaccarino (2005); Hou *et al.* (2012)]. In contrast, the IB method uses Eulerian grid to capture immersed solid bodies, an attractive feature for structures with complex geometries [Kim *et al.* (2001); Yang and Balaras (2006); Choi *et al.* (2007)].

A review of IB methods for simulating FSI problems is recently given by Sotiropoulos and Yang [2014]. In the IB method, the fluid–solid interfaces are captured by applying virtual body forces at IB points near the interfaces. The IB method can be further divided into the diffused-interface and sharp-interface methods according to the distribution of the IB points and forces. In the diffused-interface IB method [Peskin (1972); Liu and Vasilyev (2007)], the body forces are smeared over several grid points around the fluid–solid interfaces. An important advantage of the diffused-interface IB method is that there is no need to identify a node as a fluid node, solid node, or IB node. However, as indicated by Sotiropoulos and Yang [2014], the classic diffused-interface method is difficult to be applied to the simulation of FSI problems with rigid bodies due to numerical instabilities inherently associated with stiff systems. Originally developed in the context of sharp-interface IB methods [Fadlun *et al.* (2000)], the direct forcing method was implemented into diffused-interface IB methods [Uhlmann (2005); Yang *et al.* (2009)], which significantly improves the robustness of simulating FSI problems.

More related to this paper, the sharp-interface methods include but are not limited to the Cartesian-IB method [Mohd-Yusof (1997); Fadlun *et al.* (2000); Gilmanov *et al.* (2003); Balaras (2004); Gilmanov and Sotiropoulos (2005)], curvilinear-IB method [Ge and Sotiropoulos (2007)], and cut-cell method [Clarke *et al.* (1986); Cheny and Botella (2010); Meyer *et al.* (2010); Abgrall *et al.* (2014)]. In the Cartesian-IB method, a fixed Cartesian grid is used, and IB points are specified as one layer of grid points adjacent to the fluid–solid interface. The curvilinear-IB

method was developed to utilize curvilinear grid. The cut-cell method divides grid cells immersed by solid bodies into two subcells for the solid phase and fluid phase, respectively, and the continuity and momentum equations governing fluid flows are solved in the fluid phase.

The sharp-interface IB method has been successfully used to simulate many FSI problems [Borazjani *et al.* (2008); Yang *et al.* (2008); Calderer *et al.* (2014)]. A conventional method for identifying fluid, solid, and IB points is the ray-tracing algorithm [Gilmanov and Sotiropoulos (2005)]. The identification of background grid nodes using this method is computationally expensive for FSI problems, since it has to be carried out at every time step. Borazjani and Sotiropoulos [2009] developed an efficient classification algorithm using the idea of a bounding box, which significantly reduces the computational cost of the ray-tracing algorithm. Recently, a level-set (LS) method has been used together with the IB method for describing the geometries of moving bodies [Cheny and Botella (2010); Meyer *et al.* (2010)]. The LS function is defined as a signed distance to the fluid–solid interface, of which the value is positive and negative in the fluids and solid bodies, respectively. Shrivastava *et al.* [2013] developed a LS-based Cartesian-IB method to study fluid flows around rigid bodies, of which the motions are prescribed.

In this paper, we substantially improve the LS-based Cartesian-IB method by extending it to the simulation of the interaction between fluid flows and deformable bodies, such as swimming fish, together with detailed analyses on the computational parameters. Due to the body deformation, a reinitialization procedure [Sussman *et al.* (1994)] is used to update the LS function in time. The motion of the solid body is dynamically coupled with the fluid flow. The accuracy of the method is first tested in the context of several benchmark cases with rigid bodies, including the flow past a stationary circular cylinder, flow past a transversely oscillating circular cylinder, and vortex-induced vibration of a circular cylinder. To examine the performance of the new algorithm in capturing deformable bodies, we carry out simulations of self-propelled fish swimming. The effects of computational parameters on the accuracy of deformable body capturing are analyzed systematically.

The rest of this paper is organized as follows. In Sec. 2, the numerical method for solving fluid flows and the new boundary-capturing algorithm are introduced. In Sec. 3, results of test cases are discussed. Finally, conclusions are given in Sec. 4.

## 2. Computational Methodology

### 2.1. Governing equations and numerical methods for simulating fluid flows

The motions of incompressible, Newtonian fluid flows are governed by the following continuity and momentum equations:

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u}\mathbf{u}) - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} - \nabla \cdot \boldsymbol{\tau} + \mathbf{f}. \quad (2)$$

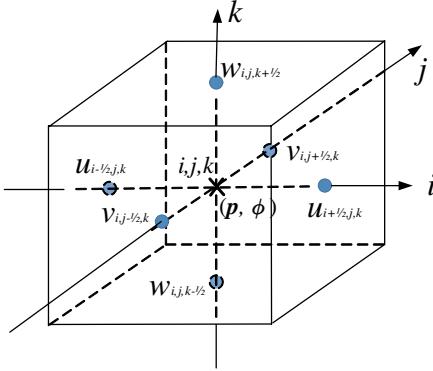


Fig. 1. Sketch of Cartesian grid cell.

Here, velocity  $\mathbf{u} = (u_1, u_2, u_3) = (u, v, w)$  and pressure  $p$  are functions of the Cartesian coordinates  $\mathbf{x} = (x_1, x_2, x_3) = (x, y, z)$  and time  $t$ ;  $\rho$  and  $\nu$  are the density and kinematic viscosity of fluids, respectively; subgrid-scale (SGS) stress tensor  $\tau$  is calculated using the dynamic Smagorinsky model [Germano *et al.* (1991); Lilly (1992)] for large-eddy simulations (LES) of turbulent flows, and is set to zero in simulations of laminar flows and direct numerical simulations (DNS) of turbulent flows;  $\mathbf{f} = (f_1, f_2, f_3)$  is the IB forcing, of which the details are introduced in Sec. 2.2.

A finite difference method is used to discretize Eq. (2). As shown in Fig. 1, velocity components are defined at the centers of grid cell surfaces and pressure is defined at grid cell centers. Each cell center is denoted by the index  $(i, j, k)$ , while the grid surfaces in  $x$ -,  $y$ -, and  $z$ -directions are denoted by the indices  $(i + 1/2, j, k)$ ,  $(i, j + 1/2, k)$ , and  $(i, j, k + 1/2)$ , respectively. The convection and viscous terms on the right-hand side (RHS) of Eq. (2) are calculated using a second-order central difference scheme. A second-order Runge–Kutta (RK2) method is used for time advancement. At each substep of the RK2 method, the fractional-step method of Kim and Moin [1985] is applied to ensure that the flow field satisfies the divergence-free condition given by Eq. (1). The semi-discretized equations for a complete RK2 step read

$$\hat{\mathbf{u}}^k = \mathbf{u}^{k-1} + \Delta t \left[ \alpha_k \mathbf{H}^{k-1} - \beta_k \left( \mathbf{H}^{k-2} + \mathbf{f}^{k-2} - \frac{1}{\rho} \nabla p^{k-2} \right) \right], \quad (3)$$

$$\tilde{\mathbf{u}}^k = \hat{\mathbf{u}}^k + \alpha_k \Delta t \mathbf{f}^k, \quad (4)$$

$$\mathbf{u}^k = \tilde{\mathbf{u}}^k - \frac{\alpha_k \Delta t}{\rho} \nabla p^{k-1}, \quad (5)$$

$$\nabla^2 p^{k-1} = \frac{\rho}{\alpha_k \Delta t} \nabla \cdot \tilde{\mathbf{u}}^k. \quad (6)$$

Here, the superscripts  $k = 1$  and  $2$  are the substep index;  $\mathbf{u}^0 = \mathbf{u}^N$  is the velocity field at step  $N$ ,  $\mathbf{u}^1$  is the intermediate velocity in the RK2 method, and  $\mathbf{u}^2 = \mathbf{u}^{N+1}$

gives the velocity field at step  $(N + 1)$ ;  $\Delta t$  is the time step; the values of coefficients  $\alpha_k$  and  $\beta_k$  are  $\alpha_1 = 1$ ,  $\beta_1 = 0$ , and  $\alpha_2 = \beta_2 = 0.5$ ;  $\mathbf{H}^k$  is the summation of the convection, viscous, and SGS stress terms at substep  $k$ , viz.

$$\mathbf{H}^k = -\nabla \cdot (\mathbf{u}^k \mathbf{u}^k) + \nu \nabla^2 \mathbf{u}^k - \nabla \cdot \boldsymbol{\tau}^k \quad \text{for } k = 0 \text{ and } 1. \quad (7)$$

Equation (3) gives a velocity field  $\tilde{\mathbf{u}}^k$  without considering the presence of solid bodies. The velocity is then updated at IB points according to Eq. (4) to obtain another velocity field  $\tilde{\mathbf{u}}^k$ , which satisfies the no-slip boundary condition at the fluid–solid interfaces. The continuity equation is then enforced by Eq. (5), in which the pressure field is given by the solution of the Poisson equation (6). The Poisson equation is solved by using the PETSc library [Balay *et al.* (1997)].

## 2.2. LS-based IB method

To force the velocity field to satisfy the boundary condition at the fluid–solid interfaces, the IB forcing needs to be given accurately at the IB points. The first step of the algorithm for calculating the IB forcing is to determine the value of the LS function at every grid point. Unlike the conventional method of evolving the governing equation of the LS function [Shrivastava *et al.* (2013)], we update the LS function using the reinitialization method. Figure 2 illustrates the process of generating the LS function schematically, using the swimming fish as an example. As shown in Fig. 2(a), an approximation of the LS function  $\phi$  is given as

$$\phi = \begin{cases} \Gamma_- - y & y < \Gamma_-, \\ -\min(\Gamma_+ - y, y - \Gamma_-) & \Gamma_- < y < \Gamma_+, \\ y - \Gamma_+ & y > \Gamma_+, \end{cases} \quad (8)$$

where the functions  $\Gamma_-(x)$  and  $\Gamma_+(x)$  define respectively the upper and lower boundaries of the solid body. The detailed expressions of  $\Gamma_-(x)$  and  $\Gamma_+(x)$  for a deformable swimming fish are given in Sec. 3.4. Figure 2(a) shows the isopleths of  $\phi$  given by Eq. (8). Note that the value of  $\phi$  shown in Fig. 2(a) does not possess the property of the LS function that  $|\nabla \phi| = 1$ . To correct the value of  $\phi$  so that  $|\nabla \phi| = 1$  is satisfied everywhere, a reinitialization procedure is needed. In the present study, we use the method of Sussman *et al.* [1994], which solves the following equation,

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi)(|\nabla \phi| - 1) = 0. \quad (9)$$

Here,  $\tau$  is a pseudo time and  $\text{sign}(\phi) = \phi / \sqrt{\phi^2 + \epsilon^2}$  is the smoothed sign function of  $\phi$ , where  $\epsilon$  is set to three times the minimal grid size. Equation (9) is integrated in  $\tau$  till the pseudo time step reaches a threshold value  $N_r$ . The effect of  $N_r$  on the computational results is discussed in Sec. 3.4. The solution of Eq. (9) ensures that  $|\nabla \phi| = 1$  without moving the isosurface of  $\phi = 0$ . Figure 2(b) shows that the isopleths of  $\phi$  become equal-spaced after applying the reinitialization procedure.

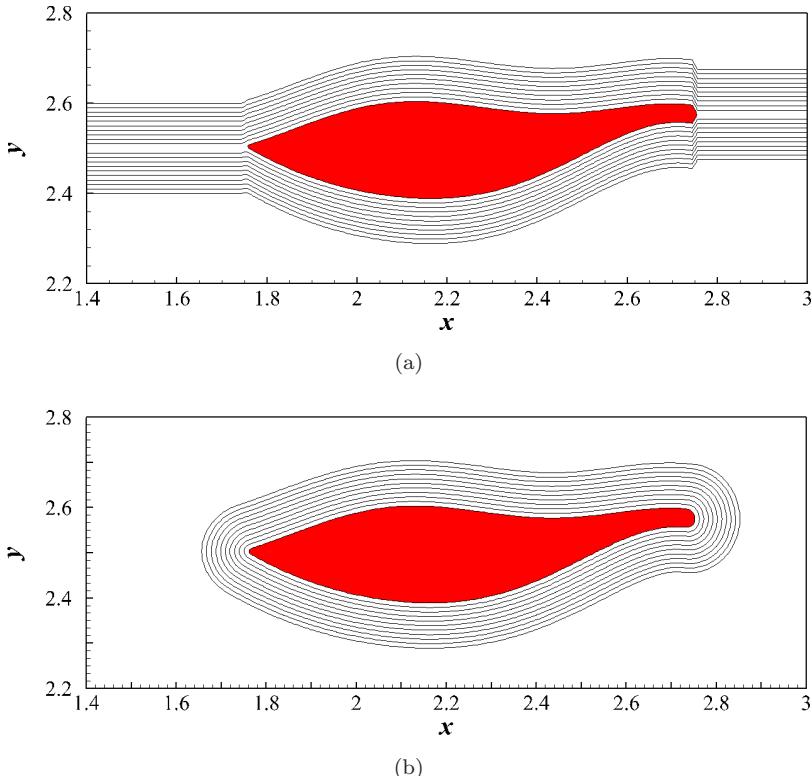


Fig. 2. Isopleths of the LS function  $\phi$  for capturing a swimming fish (a) before and (b) after applying the re-initialization procedure.

Once the LS function  $\phi$  is obtained, all grid points are categorized into solid points, fluid points, and IB points according to the value of  $\phi$ . As shown in Fig. 3,  $\phi < 0$  defines solid points. For a point with  $\phi > 0$ , if all of its four (six for 3D simulations) neighboring points have  $\phi > 0$ , then it is defined as a fluid point, otherwise it is an IB point. Note that due to the use of the staggered grid,  $\phi$  is defined at cell centers, whereas velocities are defined at cell surfaces (Fig. 1). To determine if a velocity point is a fluid point, solid point, or IB point, a linear interpolation is used to calculate the values of  $\phi$  at all cell surfaces based on those at two neighboring cell centers. In Fig. 3, point 0 denotes an IB point;  $\mathbf{n} = \nabla\phi$  is the wall-normal vector passing through point 0; point 1 is the interfacial point; and points 2 and 3 are the stencil fluid points. Note that two stencil fluid points are needed for 2D simulations as illustrated in Fig. 3, while three stencil points are needed for 3D simulations. We determine the points 1, 2, and 3 following Yang and Balaras [2006]. Let the coordinates and LS function at point 0 be denoted by  $[x_0, y_0]$  and  $\phi_0$ , respectively. The coordinates at point 1 are calculated as:

$$[x_1, y_1] = [x_0 - n_x \phi_0, y_0 - n_y \phi_0], \quad (10)$$

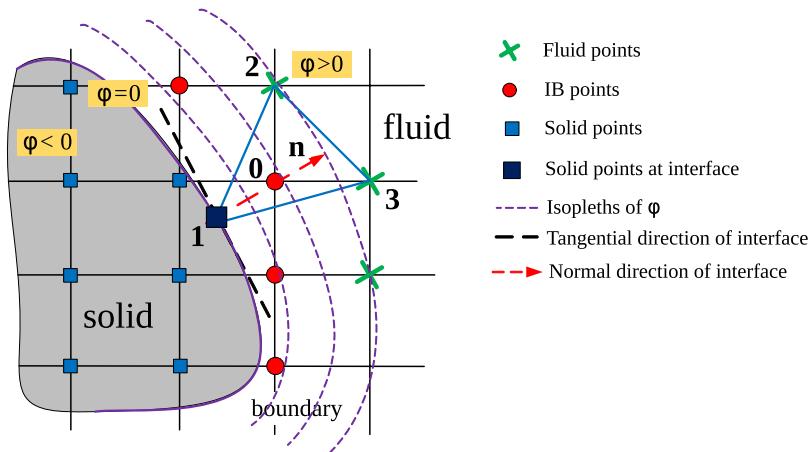


Fig. 3. Sketch of the sharp interface IB method used in the present study.

where  $n_x$  and  $n_y$  are components of  $\mathbf{n}$ . The coordinates at points 2 and 3 are given as:

$$\begin{cases} [x_2, y_2] = [x_0, y_0 + \text{sign}(n_y) \cdot \Delta y], \\ [x_3, y_3] = [x_0 + \text{sign}(n_x) \cdot \Delta x, y_0]. \end{cases} \quad (11)$$

However, if any of points 2 and 3 is also an IB point, then it should be replaced by:

$$[x_2, y_2] (\text{or} [x_3, y_3]) = [x_0 + \text{sign}(n_x) \cdot \Delta x, y_0 + \text{sign}(n_y) \cdot \Delta y]. \quad (12)$$

The velocity at point 1 is determined by the velocity of the solid body, and the velocities at points 2 and 3 are given by  $\hat{u}_i^k$  obtained from Eq. (3). A linear interpolation using points 1, 2, and 3 is then used to predict  $\tilde{u}_i^k$  at point 0. The interpolation uses the following general form

$$a = b_1 + b_2 x + b_3 y. \quad (13)$$

Let  $a_m$  denotes  $\hat{u}_i^k$  at point  $m$  for  $m = 1, 2$ , and 3. Substituting  $a_m$ ,  $x_m$ , and  $y_m$  into Eq. (13) yields the linear algebraic equations for coefficients  $b_1$ ,  $b_2$ , and  $b_3$  as

$$\begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}. \quad (14)$$

Once the coefficients  $b_1$ ,  $b_2$ , and  $b_3$  are obtained by solving Eq. (14),  $\tilde{u}_i^k$  at point 0 is obtained as

$$\tilde{u}_i^k = b_1 + b_2 x_0 + b_3 y_0. \quad (15)$$

The IB forcing  $f_i^k$  is then calculated as

$$f_i^k = \frac{\tilde{u}_i^k - \hat{u}_i^k}{\alpha_k \Delta t}. \quad (16)$$

Note that  $\hat{u}_i^k$  and  $\tilde{u}_i^k$  are different at the IB points only, i.e.,  $f_i^k$  is set to zero at the fluid points.

### 2.3. Strong coupling for simulating FSI problems

To address FSI problems, the fluid forcing  $\mathbf{F} = (F_1, F_2, F_3)$  exerted on solid bodies needs to be calculated. We use the method of Lai and Peskin [2000] and Le *et al.* [2006], in which  $\mathbf{F}$  is calculated as

$$\mathbf{F} = - \sum_{m=1}^M \left[ \frac{\mathbf{f}^1 + \mathbf{f}^2}{2} \Delta_x \Delta_y \Delta_z \right]_m , \quad (17)$$

where the summation is performed over all of the IB points;  $M$  is the total number of the IB points;  $\Delta_x$ ,  $\Delta_y$ , and  $\Delta_z$  are the grid sizes in the  $x$ -,  $y$ -, and  $z$ -directions, respectively. Due to the use of the staggered grid, if non-uniform grid spacing is used, the grid size for an IB point is not that of a grid cell shown in Fig. 1. Instead, it is defined as the grid size of a virtual grid cell centered by this IB point. For example, for the IB point with index  $(i + 1/2, j, k)$ ,  $\Delta_x$ ,  $\Delta_y$ , and  $\Delta_z$  are calculated as  $x_{i+1} - x_i$ ,  $y_{j+1/2} - y_{j-1/2}$ , and  $z_{k+1/2} - z_{k-1/2}$ , respectively. The displacement  $\mathbf{X}_c = (X_c, Y_c, Z_c)$  and velocity  $\mathbf{U}_c = (U_c, V_c, W_c)$  of the mass center of the solid body is then solved according to their governing equations, of which the detailed form is introduced associated with the specific test cases in the following sections. The LS function is then updated using the method introduced in Sec. 2.2.

Thus far, a complete time step for simulating the interactions between fluid flows and deformable solid bodies has been introduced. The algorithm for advancing from time step  $N$  to  $N + 1$  is summarized in the following two-step procedure:

**Step 1.** Calculate the IB forcing  $\mathbf{f}^1$  and  $\mathbf{f}^2$  for respectively the first and second substeps of the RK2 method according to the geometry of the solid body at step  $N$  described by  $\phi^N$ , and meanwhile advances the flow field from step  $N$  to step  $(N + 1)$ .

**Step 2.** Advance the displacement and velocity of the mass center of the solid body from step  $N$  to step  $(N + 1)$  based on the IB forcing calculated in Step 1, and update the LS function to  $\phi^{N+1}$ .

In the above algorithm, the evolution of the flow field from step  $N$  to step  $(N + 1)$  depends on the solid body at time step  $N$ . This method is known as the weak coupling for simulating FSI problems. The weak coupling approach is robust for high-density-ratio cases, in which the density ratio between the solid body and fluid is usually higher than 10. However, if the density ratio is smaller than 10, a strong coupling approach is needed. A complete time step for the strong coupling approach is summarized as:

**Step 1.** Set the guess displacement  $\mathbf{X}_c^* = \mathbf{X}_c^N$  and the guess velocity  $\mathbf{U}_c^* = \mathbf{U}_c^N$  of the mass center of the solid body to be those at time step  $N$ , and set the inner iteration step  $n_{sc}$  of the strong coupling to be one.

**Step 2.** Calculate the guess LS function  $\phi^*$  using the algorithm introduced in Sec. 2.2 based on the guess displacement  $\mathbf{X}_c^*$  of the mass center.

**Step 3.** Calculate the IB forcing  $\mathbf{f}^1$  and  $\mathbf{f}^2$  for respectively the first and second substeps of the RK2 method according to the geometry of the solid body described by  $\phi^*$ , and meanwhile advances the flow field from step  $N$  to step  $(N + 1)$ .

**Step 4.** Advance the displacement and velocity of the reference point of the solid body from step  $N$  to step  $(N + 1)$ , and calculate the residual  $\epsilon_{sc} = |\mathbf{X}_c^{N+1} - \mathbf{X}_c^*|$ .

**Step 5.** If  $n_{sc} \geq N_{sc}$ , stop the iteration for the strong coupling; otherwise, increase  $n_{sc}$  by one, reset  $\mathbf{X}_c^* = \mathbf{X}_c^{N+1}$  and  $\mathbf{U}_c^* = \mathbf{U}_c^{N+1}$  to be respectively the displacement and velocity of the mass center of the solid body obtained in Step 4, and return to Step 2. Here,  $N_{sc}$  is the maximal number of inner steps for the strong coupling iteration.

We note here that if  $N_{sc}$  is set to one, the strong coupling approach degenerates to the weak coupling approach. The effect of  $N_{sc}$  on the simulation results is discussed in Sec. 3.4.

### 3. Test Results

#### 3.1. Flow past a circular cylinder

We first test the accuracy of the numerical code in capturing a stationary solid body by performing simulations of flows past a circular cylinder. Figure 4 shows a side view of the computational domain. Let  $x$ ,  $y$ , and  $z$  represent the streamwise, vertical, and spanwise directions, respectively. The flow is driven by a uniform inflow with  $u = U_\infty$  and  $v = w = 0$  at the left boundary of the computational

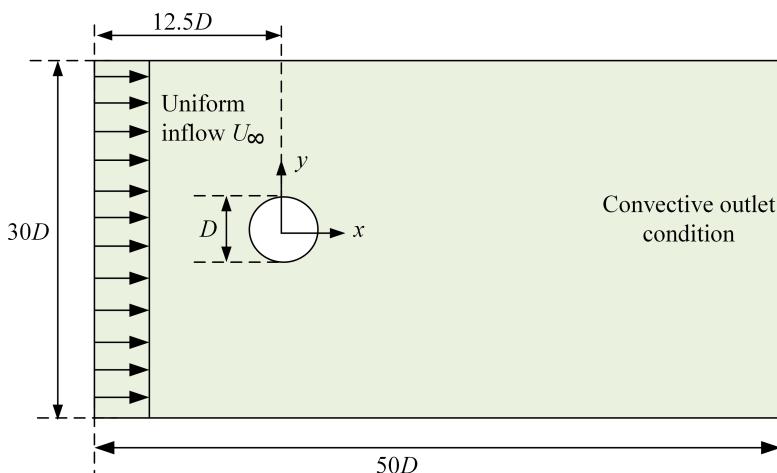


Fig. 4. Computational domain and coordinate system for simulation of flow past a circular cylinder.

domain, and a convective boundary condition is used at the outlet on the right. Periodic conditions are imposed in both the vertical and spanwise directions. We tested three Reynolds numbers, which are  $\text{Re} = U_\infty D/\nu = 40, 100$ , and 3900, respectively. Here,  $D$  represents the diameter of the cylinder. It is known from literature that the flows past a cylinder for  $\text{Re} = 40$  and 100 are two dimensional, and are steady and oscillatory, respectively. Therefore, we run 2D simulations for these two Reynolds numbers. The flow becomes turbulent for  $\text{Re} = 3900$ , for which we run a 3D simulation. As shown in Fig. 4, the origin of the coordinates is set at the center of the cylinder. The computational domain sizes in the streamwise and vertical directions are  $L_x \times L_y = 50D \times 30D$ , and the spanwise computational domain size for the 3D simulation is  $L_z = 2\pi D$ . The center of the cylinder is  $12.5D$  and  $15D$  from the inlet and top boundaries, respectively. The number of grid points is  $512 \times 384$  for 2D simulations, and 32 grid points are used in the spanwise direction for the 3D simulation. The grid is clustered near the cylinder for  $-D < x < D$  and  $-D < y < D$ . In this region, the grid resolution is  $0.025D$  in both  $x$ - and  $y$ -directions. The grid is stretched smoothly towards the boundaries. In the spanwise direction for the 3D simulation, the grid is evenly spaced with a resolution of  $0.2D$  approximately.

Figure 5 shows the streamlines for  $\text{Re} = 40$ . A steady circulation bubble is observed behind the cylinder. Table 1 compares the results of the drag coefficient  $C_d$  and the length of the recirculation bubble  $L_r$  for  $\text{Re} = 40$  obtained from the present simulation against those reported in literature. Here,  $C_d$  is defined as

$$C_d = -\frac{F_1}{\frac{1}{2}\rho U_\infty^2 D L}, \quad (18)$$

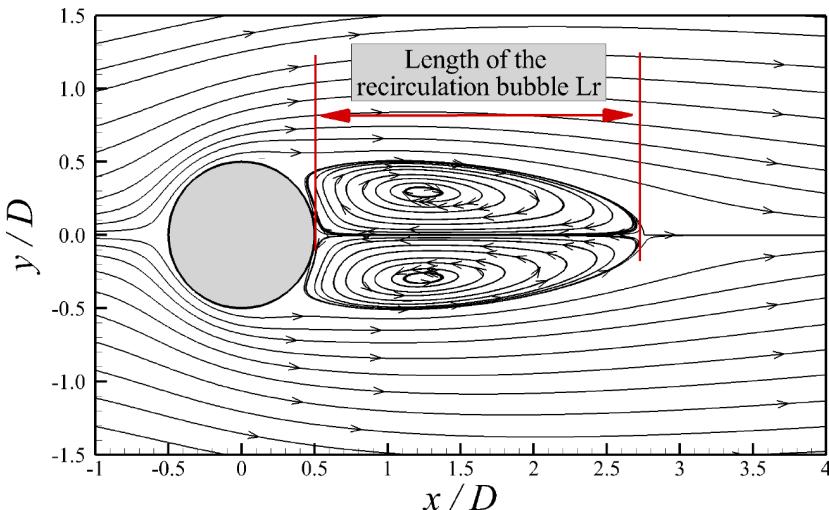


Fig. 5. Streamlines for flow past a circular cylinder at  $\text{Re} = 40$ .

Table 1. Key output parameters for flow past a circular cylinder at  $\text{Re} = 40$ .

Results from	$C_d$	$L_r/D$
Present method	1.53	2.25
Dennis and Chang [1970]	1.52	2.35
Fornberg [1980]	1.50	2.24
Park <i>et al.</i> [1998]	1.51	—
Ye <i>et al.</i> [1999]	1.52	2.27
Kim <i>et al.</i> [2001]	1.51	—
Tseng and Ferziger [2003]	1.53	2.21

where  $F_1$  is given by Eq. (17), and  $L$  represents the spanwise size of the cylinder. In 2D simulations,  $\Delta_z$  in Eq. (17) and  $L$  in Eq. (18) are set to unity. A good agreement between the present and previous results is evident from Table 1.

Figure 6 displays the contours of an instantaneous vorticity field for  $\text{Re} = 100$ . Vortex shedding is observed. Let  $f$  denote the frequency of the vortex shedding. Table 2 shows that the time-averaged drag coefficient  $\bar{C}_d$ , root mean square (RMS) of the lift coefficient fluctuations  $C'_l$ , and the Strouhal number  $St = fD/U_\infty$  are

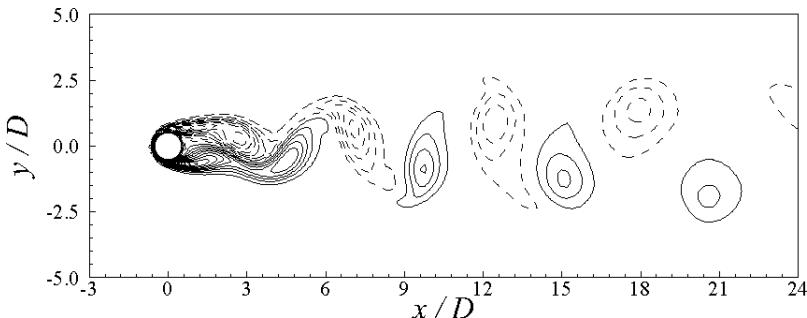


Fig. 6. Contours of an instantaneous spanwise vorticity field for flow past a circular cylinder at  $\text{Re} = 100$ . The contour interval is  $0.32U_\infty/D$ . Dashed contour lines denote negative values.

Table 2. Key output parameters for flow past a circular cylinder at  $\text{Re} = 100$ .

Results from	$\bar{C}_d$	$C'_l$	$St$
Present method	1.34	0.31	0.165
Liu <i>et al.</i> [1998]	1.35	0.34	0.165
Park <i>et al.</i> [1998]	1.33	0.33	0.165
Lai and Peskin [2000]	1.45	0.33	0.165
Kim <i>et al.</i> [2001]	1.33	0.32	0.165
Tseng and Ferziger [2003]	1.42	0.29	0.164
Uhlmann [2005]	1.45	0.34	0.169



Fig. 7. Vortex structures visualized using isosurfaces of  $\lambda_2 = -0.7U_\infty^2/D^2$  for flow past a circular cylinder at  $Re = 3900$ .

well validated against the results in literature. Here, the lift coefficient is calculated as

$$C_l = -\frac{F_2}{\frac{1}{2}\rho U_\infty^2 DL}, \quad (19)$$

with  $F_2$  defined in Eq. (17).

Figure 7 demonstrates the instantaneous vortex structures visualized using the  $\lambda_2$ -criterion of Jeong and Hussain [1995] for  $Re = 3900$ . At this Reynolds number, the flow is turbulent, and 3D vortex structures can be observed. In Table 3, time-averaged drag coefficient  $\bar{C}_d$ , time-averaged length of recirculation bubbles  $\bar{L}_r$ , separation angle  $\theta_s$ , and Strouhal number  $St$  obtained from the present simulation are compared with previous experimental [Ong and Wallace (1996)], DNS [Tremblay (2002)], and LES results [Kravchenko and Moin (2000); Tremblay (2002)]. The present results are consistent with the reference data.

### 3.2. Flow past a transversely oscillating 2D circular cylinder

In Sec. 3.1, the accuracy of our method for capturing a stationary body is examined in the test of flows past stationary circular cylinders. In this section, we further test

Table 3. Key output parameters for flow past a circular cylinder at  $Re = 3900$ .

Results from	$\bar{C}_d$	$\bar{L}_r/D$	$\theta_s$	$St$
Present method	1.03	1.37	87.6	0.22
Experiment [Ong and Wallace (1996)]	0.99	1.40	86.0	0.22
DNS [Tremblay (2002)]	1.03	1.30	85.7	0.22
LES [Kravchenko and Moin (2000)]	1.04	1.35	88.0	0.21
LES Smagorinsky [Tremblay (2002)]	1.14	1.04	87.3	0.21
LES Dynamic [Tremblay (2002)]	1.15	1.02	86.5	0.22

the flow past a transversely oscillating circular cylinder to examine the capability of the method for capturing moving bodies. The computational domain, grid resolution, and boundary conditions are set to be the same as those for the flow past a stationary cylinder, as shown in Fig. 4. However, the cylinder oscillates in the vertical direction in this test case. The motion of the cylinder center  $[X_c, Y_c]$  is prescribed as  $X_c = 0$  and  $Y_c(t) = A \cos(2\pi f_e t)$ , where  $A = 0.2D$  and  $f_e$  represent the amplitude and frequency of the oscillation, respectively. In total, six cases are tested. The normalized oscillating frequency  $f_e/f_0$  is 0.8, 0.9, 1.0, 1.1, 1.12, and 1.2 for the six cases, respectively, and the Reynolds number is fixed at  $\text{Re} = U_\infty D/\nu = 185$ . Here,  $f_0$  denotes the frequency of vortex shedding for the flow past a stationary cylinder at  $\text{Re} = 185$ . The Strouhal number based on  $f_0$  is found to be  $\text{St} = f_0 D/U_\infty = 0.194$ .

Figures 8(a)–8(f) show the contours of the spanwise vorticity for the six cases when  $Y_c = A$ , i.e. the cylinder reaches its extreme upper position. For  $f_e/f_0 \leq 1.0$ , the size of the attached negative vortex denoted by “A” decreases as  $f_e/f_0$  increases (Figs. 8(a)–8(c)). At  $f_e/f_0 = 1.1$ , a sudden change of the vortex pattern is observed (Fig. 8(d)). The positive attached vortex denoted by “B” becomes dominant for  $f_e/f_0 \geq 1.1$  (Figs. 8(d)–8(f)), and vortex A is detached. As  $f_e/f_0$  increases from 1.1 to 1.2, vortex B moves upwards, and a small negative attached vortex denoted by “C” is observed. The sudden change of the vortex pattern at  $f_e/f_0 = 1.1$  has been observed in literature using different numerical methods [Guilmineau and Queutey (2002); Yang and Balaras (2006)].

Figure 9 compares the mean drag coefficient  $\bar{C}_d$  and RMS of the lift coefficient fluctuations  $C'_l$  with those obtained by Guilmineau and Queutey [2002] and Yang and Balaras [2006]. Both  $\bar{C}_d$  and  $C'_l$  obtained in the present study agree with those given by Yang and Balaras [2006]. Although the present results of  $\bar{C}_d$  are slightly higher than those of Guilmineau and Queutey [2002], the difference is smaller than 3% for all test cases. The results demonstrated in Fig. 9 indicate that the present algorithm captures moving bodies accurately.

### 3.3. Vortex-induced vibrations of a 2D circular cylinder

In the test cases introduced above, the motion of the solid body is not coupled to the fluid flow. In this section, we further test the performance of our algorithm for simulating FSI problems. Vortex-induced vibration (VIV) of a circular cylinder has been studied extensively in experiments [see for example Anagnostopoulos and Bearman (1992)], and has been used as a benchmark test case for evaluating the accuracy of numerical methods for simulating FSI problems [Schulz and Kallinderis (1998); Li *et al.* (2002); Williamson and Govardhan (2004); Yang *et al.* (2008)]. In our tests, the computational domain is similar to that shown in Fig. 4, but the domain size is expanded to  $L_x \times L_y = 100D \times 70D$ . The number of the grid points is  $448 \times 512$ . The grid resolution is  $0.025D$  in the box for  $-D < x < D$  and  $-D < y < D$ , and the grid is stretched gradually to the four boundaries. The

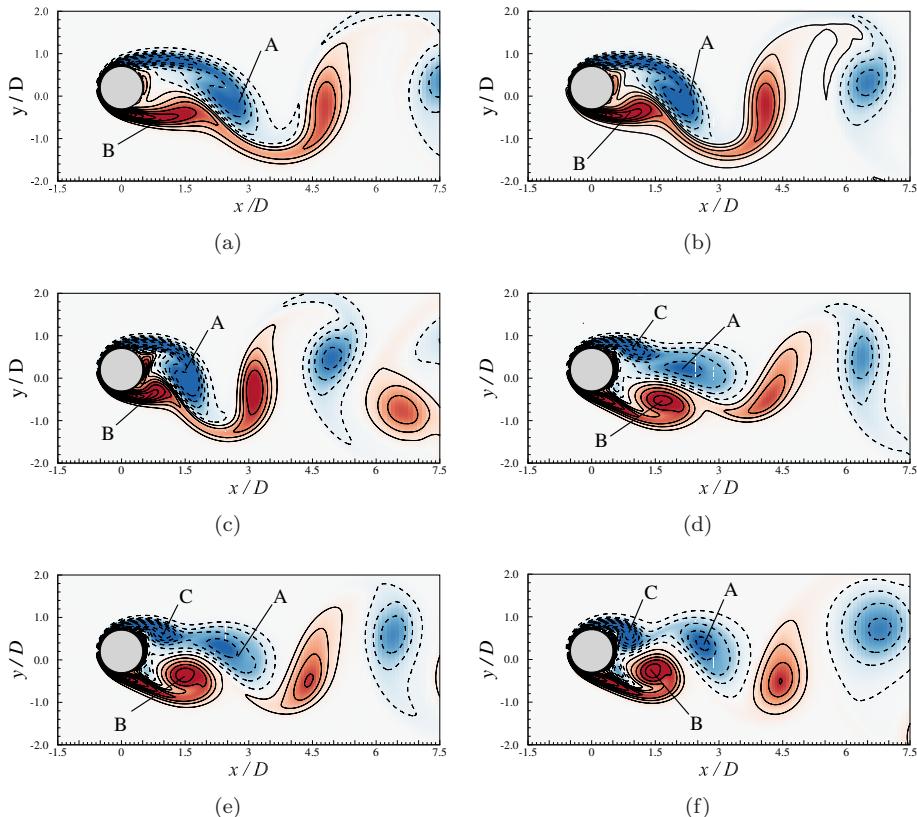


Fig. 8. Contours of instantaneous spanwise vorticity  $\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$  when the oscillating cylinder reaches its extreme upper position for (a)  $f_e/f_0 = 0.8$ , (b)  $f_e/f_0 = 0.9$ , (c)  $f_e/f_0 = 1.0$ , (d)  $f_e/f_0 = 1.1$ , (e)  $f_e/f_0 = 1.12$ , and (f)  $f_e/f_0 = 1.2$ . Solid and dashed lines represent positive and negative vorticity, respectively. The contour interval is  $0.75U_\infty/D$ .

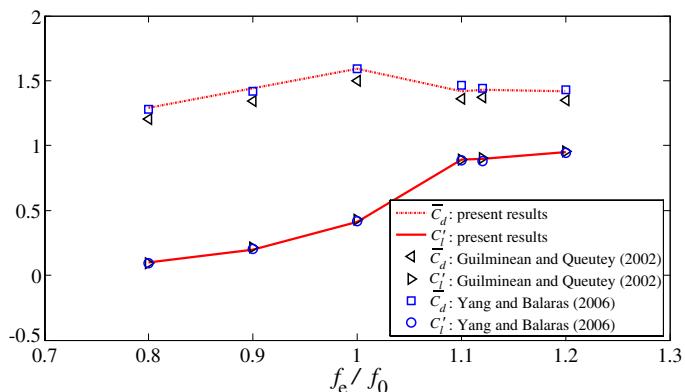


Fig. 9. Mean drag coefficient  $\bar{C}_d$  and RMS of lift coefficient  $C'_l$  for flow past a transversely oscillating 2D circular cylinder at different  $f_e/f_0$ .

inflow is uniform, with its magnitude denoted by  $U_\infty$ . The convection condition is used at the outlet, and the periodic condition is imposed in the vertical direction. The original location of the cylinder center is  $30D$  and  $35D$  from the inlet and upper boundaries, respectively. Let  $[X_c, Y_c]$  denote the coordinates of the cylinder center. We enforce  $X_c = 0$  following the setup of VIV problems in literatures [Schulz and Kallinderis (1998); Williamson and Govardhan (2004); Yang *et al.* (2008)]. The transverse motion of the cylinder is described as a forced vibration with damping, of which the governing equation reads

$$\frac{d^2Y_c^*}{dt^2} + \frac{4\pi\zeta}{U_{\text{red}}} \frac{dY_c^*}{dt} + \left( \frac{2\pi}{U_{\text{red}}} \right)^2 Y_c^* = \frac{2C_l}{\pi\rho_c^*}. \quad (20)$$

Here,  $Y_c^* = Y_c/D$  is the dimensionless displacement of the cylinder center in the  $y$ -direction;  $\zeta = 0.5c/\sqrt{km}$  is the damping ratio, where  $c$  is the damping coefficient;  $k$  is the stiffness of the spring;  $m$  is the mass of the cylinder;  $U_{\text{red}} = \text{Re}/f_n^*$  is the reduced velocity, where  $\text{Re} = U_\infty D/\nu$ , and  $f_n^* = D^2 \sqrt{k/m}/2\pi\nu$  is the dimensionless natural frequency of free vibration;  $C_l$  is the lift coefficient given by Eq. (19);  $\rho_c^* = \rho_c/\rho_f$  is the dimensionless density of the cylinder, where  $\rho_c$  and  $\rho_f$  are the densities of the cylinder and fluid, respectively. Equation (20) is discretized as

$$\begin{aligned} & \frac{(Y_c^*)^{N+1} - 2(Y_c^*)^N + (Y_c^*)^{N-1}}{\Delta t^2} \\ & + \frac{4\pi\zeta}{U_{\text{red}}} \frac{(Y_c^*)^{N+1} - (Y_c^*)^{N-1}}{2\Delta t} + \left( \frac{2\pi}{U_{\text{red}}} \right)^2 (Y_c^*)^N = \frac{2C_l^N}{\pi\rho_c^*}, \end{aligned} \quad (21)$$

from which it is known that the values of  $Y_c^*$  at steps  $N$  and  $(N-1)$  are needed to calculate that at step  $(N+1)$ .

We have tested eight cases, for which  $\zeta = 0.0012$ ,  $\rho_c^* = 149.10$ , and  $f_n^* = 17.9$  are fixed. The Reynolds numbers for the eight test cases are set to  $\text{Re} = 90, 95, 100, 110, 115, 120, 130$ , and  $140$ , and correspondingly, the values of  $U_{\text{red}}$  is  $5.02, 5.31, 5.587, 6.15, 6.42, 6.70, 7.26$ , and  $7.82$ , respectively. To generate the initial flow field, the cylinder is fixed till the frequency of the vortex shedding becomes steady. The cylinder is then released to move in the  $y$ -direction according to Eq. (20).

Figure 10 demonstrates the time series of the dimensionless displacement  $Y_c^*$  of the cylinder center when the vibration reaches a steady state. We show in Fig. 10 the results for three representative Reynolds numbers,  $\text{Re} = 90, 100$ , and  $120$ , respectively. The frequency  $f_{\text{VIV}}$  and amplitude  $A_{\text{VIV}}$  of the vibration of the cylinder at all eight Reynolds numbers tested are summarized in Fig. 11. The results measured in experiments [Anagnostopoulos and Bearman (1992)] and those obtained from numerical simulations based on various algorithms [Schulz and Kallinderis (1998); Li *et al.* (2002); Williamson and Govardhan (2004); Yang *et al.* (2008)] are also shown in Fig. 11 for comparison. As shown in Fig. 11(a), at  $\text{Re} = 90$ , the ratio  $f_{\text{VIV}}/f_n$  between the frequency of VIV and the natural frequency is about 0.8, and the amplitude of VIV is relatively small as shown in Figs. 10(a) and 11(b). The Reynolds numbers between 95 and 115 fall into the lock-in regime, for which

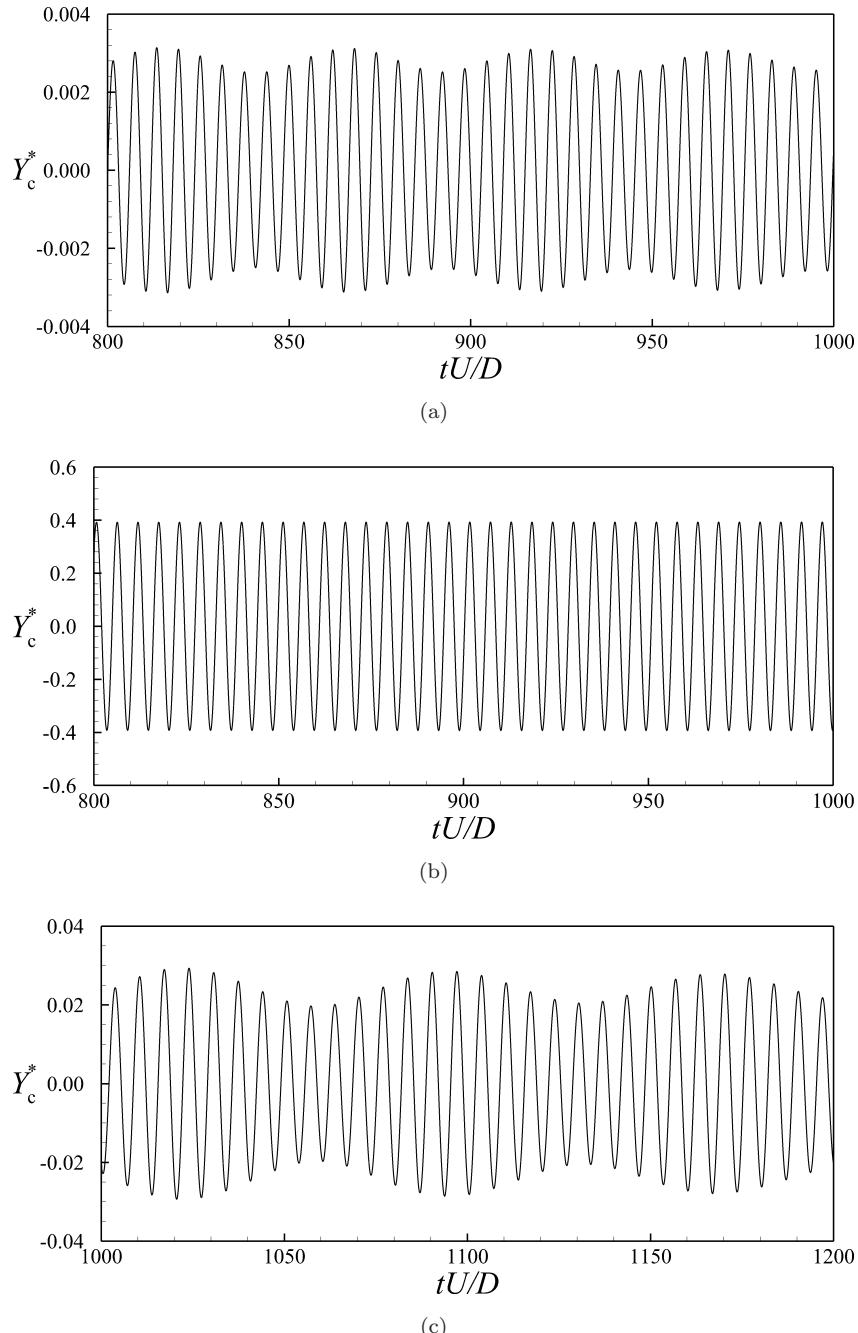


Fig. 10. Time history of the dimensionless displacement of the cylinder  $Y_c^*$ , when the vibration reaches a steady state. The Reynolds number is (a) 90, (b) 100, and (c) 120.

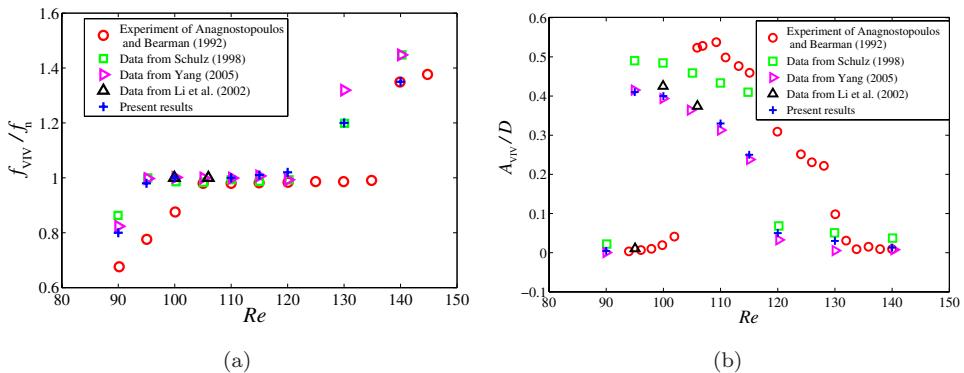


Fig. 11. Dimensionless (a) frequency and (b) amplitude for various Reynolds numbers. The results measured in experiments and those obtained from previous numerical simulations are superposed to validate the present results.

$f_{VIV}/f_n$  is close to 1.0 and the amplitude of VIV is large. As shown in Figs. 10(b) and 11(b), the dimensionless amplitude of VIV is about  $A_{VIV}/D = 0.4$ , which is significantly larger than that for  $Re = 90$ . The value of  $f_{VIV}/f_n$  becomes larger than 1.0 for  $Re = 120, 130$ , and  $140$ , which are higher than the upper bound of the lock-in regime. As shown in Figs. 10(c) and 11(b), the amplitude for  $Re = 120$  is about  $A_{VIV}/D = 0.03$ , which is one order smaller than that for  $Re = 100$ .

Figure 11(a) shows that the frequency of VIV in the present study is consistent with that in Yang *et al.* [2008] for  $Re \leq 120$ . It is slightly lower in the present study than in Yang *et al.* [2008] for  $Re = 130$  and  $140$ . However, our results agree with the numerical results of Schulz and Kallinderis [1998] and experimental results of Anagnostopoulos and Bearman [1992] for  $Re = 130$  and  $140$ , respectively. Furthermore, it can be seen from Fig. 11(b) that the amplitude of VIV obtained from the present simulations agree with that given by Yang *et al.* [2008]. The upper and lower bounds of the lock-in regime obtained from the present simulations are  $Re = 95$  and  $115$ , respectively. These two critical Reynolds numbers are consistent with those obtained from numerical simulations of Yang *et al.* [2008] and Schulz and Kallinderis [1998]. However, the values obtained from various numerical simulations are lower than those obtained from experiments. Schulz *et al.* [1998] attributed the reason for the discrepancy between the numerical and experimental results to the 3D effects in the experiments.

### 3.4. Self-propelled anguilliform fish

Fish swim forward by undulating their body and/or caudal fins (BCF). The body wave propagates from the anterior to the posterior of the body [Sfakiotakis *et al.* (1999)]. This wave can be described as a traveling sinusoidal wave with a growing amplitude, which provides propulsions for fish swimming. The capability of the present algorithm for capturing deformable bodies is tested in the context of

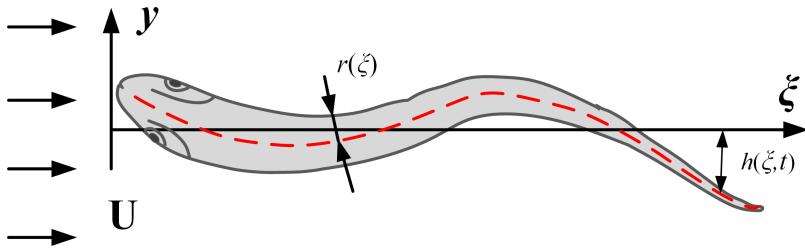


Fig. 12. Top view of the geometry of a swimming anguilliform fish.

swimming anguilliform fish, which refer to long and slender fish, such as eels and lampreys. Figure 12 shows the swimming anguilliform fish considered in the present study. Its geometry can be described by using the half width of the body  $r(\xi)$ , where  $\xi$  represents the coordinate with its origin point located at the front of the fish. Following Carling *et al.* [1998] and Kern and Koumoutsakos [2006],  $r(\xi)$  is given as

$$r(\xi) = \begin{cases} \sqrt{2r_h\xi - \xi^2} & 0 \leq \xi \leq \xi_b, \\ r_h - (r_h - r_t) \left( \frac{\xi - \xi_b}{\xi_t - \xi_b} \right)^2 & \xi_b \leq \xi \leq \xi_t, \\ r_t \frac{L - \xi}{L - \xi_t} & \xi_t \leq \xi \leq L, \end{cases} \quad (22)$$

where  $L$  is the fish length, and constants  $r_h$ ,  $r_t$ ,  $\xi_b$ , and  $\xi_t$  are  $0.04L$ ,  $0.01L$ ,  $0.04L$ , and  $0.95L$ , respectively.

Based on the observations of anguilliform fish reported in literature, the body deformation for self-propelling fish can be described as [Kern and Koumoutsakos (2006)]:

$$H(\xi, t) = 0.125L \frac{0.03125 + \xi/L}{1.03125} \sin \left[ 2\pi \left( \frac{\xi}{L} - \frac{t}{T} \right) \right], \quad (23)$$

where  $H(\xi, t)$  represents the lateral displacement of the midline of the fish body, and  $T$  is the period of body beating. Combining Eqs. (22) and (23) gives the functional expressions of  $\Gamma_+$  and  $\Gamma_-$  in Eq. (8) as:

$$\Gamma_{\pm}(x, t) = \begin{cases} H(0, t), & \text{if } x < X_h, \\ H(x - X_h, t) \pm r(x - X_h, t), & \text{if } X_h \leq x \leq X_t, \\ H(L, t), & \text{if } x > X_t. \end{cases} \quad (24)$$

Here,  $X_h$  and  $X_t$  represents the  $x$  coordinates of the front point and end point of the fish, respectively. At the beginning of the simulation, the mass center of the fish is located at  $[X_c, Y_c] = [0, 0]$ . Following Kern and Koumoutsakos [2006], Gazzola *et al.* [2011], and Ghaffari *et al.* [2015], the motion of the fish is confined in the  $x$ -direction, such that  $Y_c$  is fixed to 0. The fish accelerates and then reaches a steady speed. The displacement of fish is governed by Newton's second law as  $d^2 X_c/dt^2 = F_1$  with  $F_1$  calculated from Eq. (17). The value of  $X_c$  at step ( $N+1$ ) is

Table 4. Computational parameters and results for the resolution-dependency study of a 2D self-propelled anguilliform fish.

Case	$N_x \times N_y$	$\Delta_x/L \times \Delta_y/L^a$	$U_0 T/L$
A1	$1024 \times 236$	$0.0078 \times 0.0042$	0.48
A2	$2048 \times 256$	$0.0039 \times 0.0025$	0.52
A3	$4096 \times 288$	$0.0020 \times 0.0013$	0.55
A4	$8192 \times 352$	$0.0010 \times 0.0010$	0.55
Kern and Koumoutsakos [2006]	$2.1 \times 10^4$ grid cells <sup>b</sup>	—	0.54
Gazzola <i>et al.</i> [2011]	$4096 \times 2048$	$0.0020 \times 0.0013$	0.55
Ghaffari <i>et al.</i> [2015]	$2048 \times 1024$	$0.0039 \times 0.0039$	0.53

Notes: <sup>a</sup>Grid resolution near the fish body.

<sup>b</sup>C-type nonstructural grid is used in Kern and Koumoutsakos [2006].

updated based on those at steps  $N$  and  $(N - 1)$  as  $X_c^{N+1} = F_1^N \Delta t^2 - X_c^{N-1} + 2X_c^N$ . It is assumed that the deformation of the fish in the  $x$ -direction is negligible, such that  $X_h - X_c$  and  $X_l - X_c$  remain constant during the simulation. Under this assumption,  $X_h$  and  $X_l$  can be updated with  $X_c$ , and then the LS function can be obtained following the procedure introduced in Sec. 2.2. The computational domain for the self-propelled 2D anguilliform fish is a rectangle with  $L_x \times L_y = 10L \times 5L$ . In the  $x$ - and  $y$ -directions, periodic boundary conditions are imposed.

We perform simulations using four sets of grid to test the resolution-dependency of the computational results. The grid is evenly spaced in the  $x$ -direction, while it is refined near the fish body between  $y = -0.125L$  and  $y = 0.125L$  in the  $y$ -direction. Table 4 summarizes the number of the grid points and the grid resolution near the fish body for all test cases. The computational results of the final forward speed  $U_0$  reached by the fish are also listed in the table. The Reynolds number is fixed at  $fL^2/\nu = 7143$  in all cases, where  $f = 1/T$  is the frequency of the body beating of the anguilliform fish. The threshold values of the iteration step for reinitialization and for strong coupling are set to  $N_r = 5$  and  $N_{sc} = 3$ , respectively. According to the analysis presented next, it has been ensured that these two threshold values are sufficient such that the results have negligible changes if more iterations are performed. The computational parameters and results for the same Reynolds number obtained by Kern and Koumoutsakos [2006], Gazzola *et al.* [2011], and Ghaffari *et al.* [2015] are also summarized in Table 4 for comparison. It is shown that the values of  $U_0$  for cases A3 and A4, which have high grid resolution, are the same, and they are close to the results in literature [Kern and Koumoutsakos (2006); Gazzola *et al.* (2011); Ghaffari *et al.* (2015)]. Particularly, the values of  $U_0$  for cases A3 and A4 agree with that in Ghaffari *et al.* [2015], in which the grid resolution is the same as in case A3. The results indicate that the present algorithm captures deformable bodies accurately, and the grid resolution in case A3 is necessary and sufficient for capturing the swimming anguilliform fish.

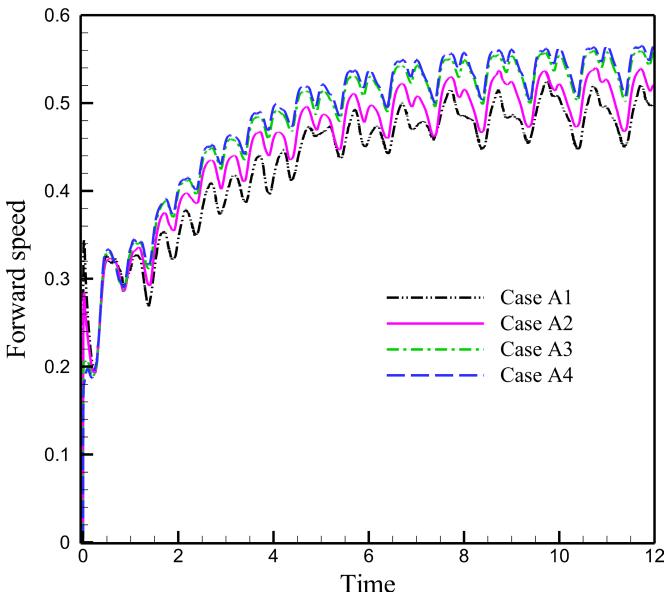


Fig. 13. Time history of the forward speed of the swimming anguilliform fish based on different grid resolutions.

Figure 13 compares the time series of the forward speed of the fish for cases A1–A4. Discrepancies can be observed between the results of cases A1, A2, and A3, during both the accelerating and steady stages, while the results for cases A3 and A4 agree with each other. The results demonstrated in Fig. 13 indicate that the grid resolution in case A3 is needed to achieve resolution-independency. This conclusion is consistent with that drawn from Table 4 as discussed above.

Figure 14 compares the time series of the forward speed of the swimming anguilliform fish based on different threshold value  $N_r$  of the iteration step for the re-initialization procedure introduced in Sec. 2.2. We have tested four cases with  $N_r = 1, 3, 5$ , and 10, respectively. The grid resolution in case A3 is used to test the effect of  $N_r$ . As demonstrated in Fig. 14, cases with  $N_r = 5$  and 10 have indiscernible difference, whereas the forward speed is underpredicted if  $N_r = 1$  or 3. The above test results indicate that  $N_r = 5$  is needed for re-initializing the LS function at each time step in our algorithm.

To further investigate the effect of  $N_r$  on capturing the anguilliform fish, we compare in Fig. 15 the isopleths of the LS function  $\phi$  near the fish head based on different threshold values of  $N_r$ . It is shown that the re-initialization procedure does not move the isopleth of  $\phi = 0$  [Sussman *et al.* (1994)], indicating that the captured geometry of the anguilliform fish is not influenced by the value of  $N_r$ . However, the isopleths of  $\phi = \pm 0.01L$  are sensitive to  $N_r$ , such that the outer-normal of the surface calculated as  $\nabla\phi$  is influenced by the value of  $N_r$ . Figure 15 shows that the isopleths of  $\phi$  with  $N_r = 5$  and 10 collapse, which explains why in Fig. 14 the time

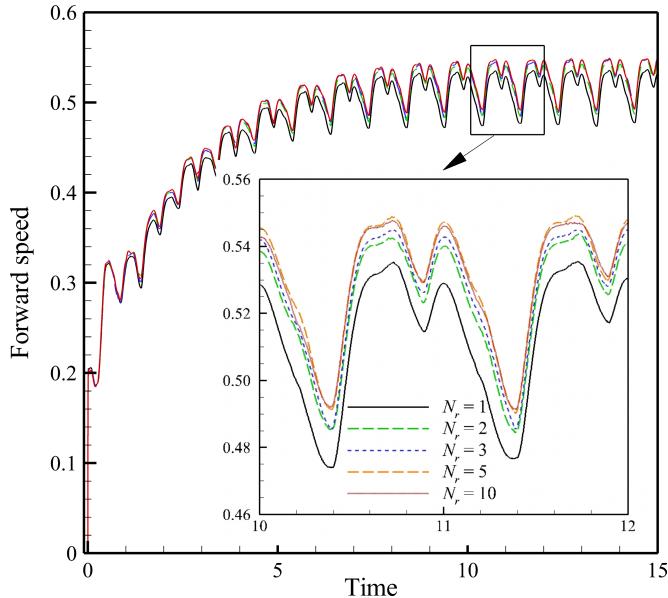


Fig. 14. Time history of the forward speed of the swimming anguilliform fish based on different threshold values of the iteration step for the re-initialization procedure.

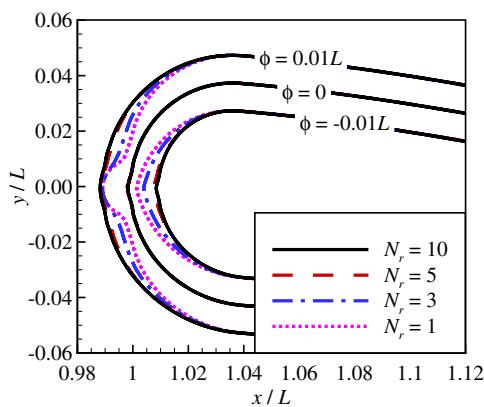


Fig. 15. Isopleths of the LS function near the fish head based on different threshold values of the iteration step for the re-initialization procedure.

histories of the forward speed based on  $N_r = 5$  and 10 are close to each other. In contrast, the isopleths of  $\phi$  with  $N_r = 1$  and 3 deviate from those with  $N_r = 5$  and 10, and as a result, the forward speed is not accurately predicted if  $N_r = 1$  or 3 is used for performing the re-initialization procedure.

Figure 16 compares the time series of the forward speed of the swimming anguilliform fish using different threshold values of the iteration step for the strong-coupling

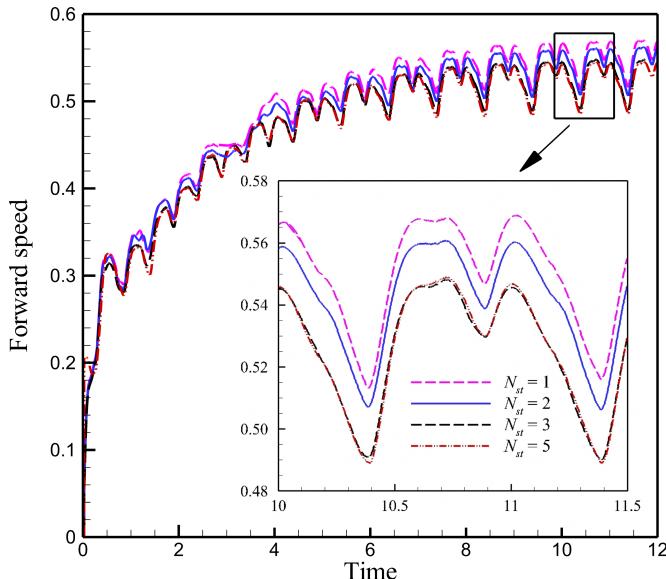


Fig. 16. Time history of the forward speed of the swimming anguilliform fish based on different threshold values of the iteration step for the strong-coupling simulation.

simulation introduced in Sec. 2.3. Four test cases with  $N_{sc} = 1, 2, 3$ , and 5 are conducted, with the grid resolution being the same as that in case A3 and  $N_r = 5$ . Figure 16 shows that the results for  $N_{sc} = 3$  and 5 agree with each other. The simulations with  $N_{sc} = 1$  and 2 overpredict the forward speed. The above results lead to a conclusion that  $N_{sc} = 3$  is needed in the strong-coupling in our algorithm.

Based on the test results discussed above, we further perform five simulation cases with  $Re = 500, 1000, 2000, 3000$ , and 7000 to study the effect of the Reynolds number on the swimming performance of the anguilliform fish. Table 5 summarizes the results in terms of the Strouhal number  $St = fA/U_0$ , slip ratio  $S_l = U_0/V_0$ , and swimming efficiency  $\eta_{EBT} = (1 + S_l)/2$  based on the Elongated Body Theory (EBT) [Lighthill (1969)]. Here,  $A$  represents the amplitude of the transverse motion of the fish tail,  $f$  is the tail-beating frequency, and  $V_0 = L/T$  is the phase speed

Table 5. Swimming performance of anguilliform fish at different Reynolds numbers.

Re	St	$S_l$	$\eta_{EBT}$
500	0.909	0.274	63.7%
1000	0.753	0.332	66.6%
2000	0.665	0.376	68.8%
3000	0.625	0.400	70.0%
7000	0.481	0.520	76.0%

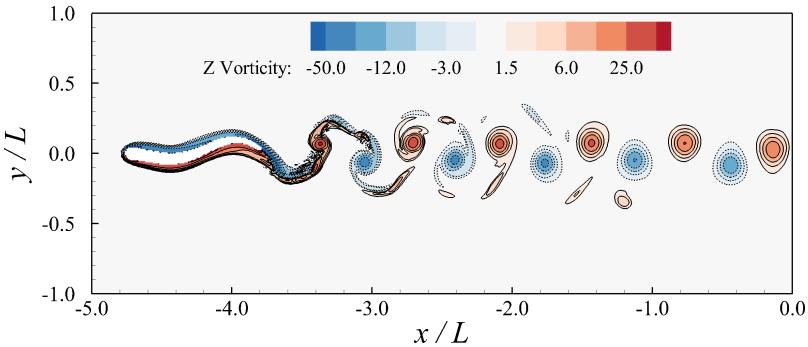


Fig. 17. Contours of vorticity around and behind the anguilliform swimming fish, normalized by  $1/T$ . The results is for  $Re = 3000$ . Reverse Kármán vortex street is observed. Solid and dashed lines represent positive and negative values of vorticity, respectively.

of the body wave traveling from the head to the tail of the fish. Low value of St and high values of  $S_l$  and  $\eta_{EBT}$  are indicators of high efficiency of fish swimming. As shown in Table 5, as the Reynolds number increases, St decreases, while  $S_l$  and  $\eta_{EBT}$  increase. The results obtained from the present simulations indicate that the swimming efficiency of a anguilliform fish is higher when the Reynolds number is higher. This conclusion is consistent with the findings of Borazjani and Sotiropoulos [2009].

Figure 17 shows the contours of the spanwise vorticity generated by the swimming anguilliform fish for  $Re = 3000$ . Positive and negative vorticity can be observed in the wake above and below the center line, respectively. Such pattern of the vorticity field is different from that observed in flows past stationary cylinders, where negative and positive vorticity occur above and below the horizontal center line of the cylinder, respectively (see Fig. 6). The vorticity pattern shown in Fig. 17 is called the reversed Kármán vortex street [Triantafyllou *et al.* (1993)].

#### 4. Conclusions and Final Remarks

In this paper, a LS-based sharp interface IB method is developed to simulate interactions between fluid flows and deformable solid bodies. A reinitialization procedure is applied to update the LS function at every time step. The value of the LS function is then used to categorize the grid points into solid points, fluid points, and IB points. The outer normal direction of the body surface needed for determining stencil fluid points is calculated based on the gradients of the LS function. Once the IB points and the corresponding stencil fluid points are determined, the IB forcing is calculated based on the flow field at these points. To account for FSI, the fluid force exerted on solid bodies is calculated as the summation of IB forcing, and the strong coupling approach is used.

The LS-based IB method is implemented into a finite difference code. The accuracy of the IB method is tested in the context of several benchmark cases, including

flow past a fixed circular cylinder, flow past a transversely oscillating cylinder, and vibration of cylinder induced by vortex shedding. The results agree with previous experimental and numerical results. To further examine the capability of the algorithm for capturing deformable bodies with more complex geometries, we have performed simulations of self-propelled fish swimming. We have tested the effects of the grid resolution, the threshold value  $N_r$  for re-initializing the LS function, and the threshold value  $N_{sc}$  for the strong coupling in FSI. The test results indicate that a resolution of  $\Delta_x \times \Delta_y = 0.002L \times 0.0013L$  near the fish body achieves resolution-independency, and it is necessary to use  $N_r = 5$  and  $N_{sc} = 3$  to warrant sufficient iterations.

Finally, we remark that the computational results shown in this paper indicate that the LS-based IB method captures the geometry of a deformable body accurately. More degrees of freedom can be considered in the future to obtain more details of the motion of deformable bodies. The technique of adaptive mesh refinement can be implemented to improve the accuracy and robustness of the simulation. Furthermore, a coupled computational fluid dynamics and computational solid dynamics method will be developed in the future based on the present algorithm.

## Acknowledgments

ZC and HJ would like to thank the Natural Science Foundation of China for its support (Award number 51275127) to this research. ZC also acknowledges the China Scholarship Council for sponsoring his visit to the University of Minnesota from October 2014 to October 2016.

## References

- Abgrall, R., Beaugendre, H. and Dobrzynski, C. [2014] “An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques,” *J. Comput. Phys.* **257**, 83–101.
- Anagnostopoulos, P. and Bearman, P. W. [1992] “Response characteristics of a vortex excited cylinder at low Reynolds numbers,” *J. Fluid Struct.* **6**, 39–50.
- Bahmani, M. H. and Akbari, M. H. [2011] “Response characteristics of a vortex-excited circular cylinder in laminar flow,” *J. Mech. Sci. Technol.* **25**, 125–133.
- Balay, S., Gropp, W. D., McInnes, L. C. and Smith, B. F. [1997] “Efficient management of parallelism in object oriented numerical software libraries,” in *Modern Software Tools in Scientific Computing*, eds. Arge, E., Bruaset, A. M. and Langtangen, H. P. (Birkhäuser Press, Boston, USA), pp. 163–202.
- Balaras, E. [2004] “Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations,” *Comput. Fluids* **33**, 375–404.
- Borazjani, I., Ge, L. and Sotiropoulos, F. [2008] “Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies,” *J. Comput. Phys.* **227**, 7587–7620.
- Borazjani, I. and Sotiropoulos, F. [2009] “Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes,” *J. Exp. Biol.* **212**, 576–592.

- Calderer, A., Kang, S. and Sotiropoulos, F. [2014] "Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures," *J. Comput. Phys.* **277**, 201–227.
- Carling, J., Williams, T. L. and Bowtell, G. [1998] "Self-propelled anguilliform swimming: Simultaneous solution of the two-dimensional Navier-Stokes equations and Newton's laws of motion," *J. Exp. Biol.* **201**, 3143–3166.
- Cheny, Y. and Botella, O. [2010] "The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties," *J. Comput. Phys.* **229**, 1043–1076.
- Choi, J. I., Oberoi, R. C., Edwards, J. R. and Rosati, J. A. [2007] "An immersed boundary method for complex incompressible flows," *J. Comput. Phys.* **224**, 757–784.
- Clarke, D., Salas, M. and Hassan, H. [1986] "Euler calculations for multi-element airfoils using Cartesian grids," *AIAA J.* **24**, 353–358.
- Dennis, S. C. R. and Chang, G. Z. [1970] "Numerical solution for steady flow past a circular cylinder at Reynolds numbers up to 100," *J. Fluid Mech.* **42**, 471–489.
- Dowell, E. H. and Hall, K. C. [2001] "Modeling of fluid–structure interaction," *Annu. Rev. Fluid Mech.* **33**, 445–490.
- Fadlun, E. A., Verzicco, R., Orlandi, P. and Mohd-Yusof, J. [2000] "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations," *J. Comput. Phys.* **161**, 35–60.
- Fornberg, B. [1980] "A numerical study of steady viscous flow past a circular cylinder," *J. Fluid Mech.* **98**, 819–855.
- Gazzola, M., Chatelain, P., van Rees, W. M. and Koumoutsakos, P. [2011] "Simulations of single and multiple swimmers with non-divergence free deforming geometries," *J. Comput. Phys.* **230**, 7093–7114.
- Ge, L. and Sotiropoulos, F. [2007] "A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries," *J. Comput. Phys.* **225**, 1782–1809.
- Germano, M., Piomelli, U., Moin, P. and Cabot, W. H. [1991] "A dynamic subgrid-scale eddy viscosity model," *Phys. Fluids* **3**, 1760–1765.
- Ghaffari, S. A., Viazzo, S., Schneider, K. and Bontoux, P. [2015] "Simulation of forced deformable bodies interacting with two-dimensional incompressible flows: Application to fish-like swimming," *Int. J. Heat Fluid Flow* **51**, 88–109.
- Gilmanov, A., Sotiropoulos, F. and Balaras, E. [2003] "A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids," *J. Comput. Phys.* **191**, 660–669.
- Gilmanov, A. and Sotiropoulos, F. [2005] "A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies," *J. Comput. Phys.* **207**, 457–492.
- Guilmineau, E. and Queutey, P. [2002] "A numerical simulation of vortex shedding from an oscillating circular cylinder," *J. Fluid Struct.* **16**, 773–794.
- Hou, G., Wang, J. and Layton, A. [2012] "Numerical methods for fluid-structure interaction — A review," *Commun. Comput. Phys.* **12**, 337–377.
- Jeong, J. and Hussain, F. [1995] "On the identification of a vortex," *J. Fluid Mech.* **285**, 69–94.
- Kamakoti, R. and Shyy, W. [2004] "Fluid–structure interaction for aeroelastic applications," *Prog. Aerosp. Sci.* **40**, 535–558.
- Kern, S. and Koumoutsakos, P. [2006] "Simulations of optimized anguilliform swimming," *J. Exp. Biol.* **209**, 4841–4857.

- Kim, J., Kim, D. and Choi, H. [2001] “An immersed boundary finite volume method for simulations of flow in complex geometries,” *J. Comput. Phys.* **171**, 132–150.
- Kim, J. and Moin, P. [1985] “Application of a fractional-step method to incompressible Navier-Stokes equations,” *J. Comput. Phys.* **59**, 308–323.
- Kravchenko, A. G. and Moin, P. [2000] “Numerical studies of flow over a circular cylinder at  $Re = 3900$ ,” *Phys. Fluids* **12**, 403–417.
- Lai, M. C. and Peskin, C. S. [2000] “An immersed boundary method with formal second-order accuracy and reduced numerical viscosity,” *J. Comput. Phys.* **160**, 705–719.
- Le, D. V., Khoo, B. C. and Peraire, J. [2006] “An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries,” *J. Comput. Phys.* **220**, 109–138.
- Li, L., Sherwin, S. J. and Bearman, P. W. [2002] “A moving frame of reference algorithm for fluid/structure interaction of rotating and translating bodies,” *Int. J. Numer. Methods Fluids* **38**, 187–206.
- Lighthill, M. J. [1969] “Hydromechanics of aquatic animal propulsion,” *Annu. Rev. Fluid Mech.* **1**, 413–446.
- Lilly, D. K. [1992] “A proposed modification of the Germano subgrid-scale closure method,” *Phys. Fluids* **4**, 633–635.
- Liu, Q. and Vasilyev, O. V. [2007] “A Brinkman penalization method for compressible flows in complex geometries,” *J. Comput. Phys.* **227**, 946–966.
- Liu, C., Zheng, X. and Sung, C. H. [1998] “Preconditioned multigrid methods for unsteady incompressible flows,” *J. Comput. Phys.* **139**, 35–57.
- Löhner, R., Haug, E., Michalski, A., Muhammad, B., Drego, A., Nanjundaiah, R. and Zarfam, R. [2015] “Recent advances in computational wind engineering and fluid-structure interaction,” *J. Wind Eng. Ind. Aerod.* **144**, 14–23.
- Meyer, M., Devesa, A., Hickel, S., Hu, X. and Adams, N. [2010] “A conservative immersed interface method for large-eddy simulation of incompressible flows,” *J. Comput. Phys.* **229**, 6300–6317.
- Mittal, R. and Iaccarino, G. [2005] “Immersed boundary methods,” *Annu. Rev. Fluid Mech.* **37**, 239–261.
- Mohd-Yusof, J. [1997] “Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries,” CTR, Annual Research Briefs, pp. 317–327.
- Ong, L. and Wallace, J. [1996] “The velocity field of the turbulent very near wake of a circular cylinder,” *Exp. Fluids* **20**, 441–453.
- Park, J., Kwon, K. and Choi, H. [1998] “Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160,” *KSME Int. J.* **12**, 1200.
- Peskin, C. S. [1972] “Flow patterns around heart valves: A numerical method,” *J. Comput. Phys.* **10**, 252–271.
- Schulz, K. W. and Kallinderis, Y. [1998] “Unsteady flow structure interaction for incompressible flows using deformable hybrid grids,” *J. Comput. Phys.* **143**, 569–597.
- Sfakiotakis, M., Lane, D. M. and Davies, J. B. C. [1999] “Review of fish swimming modes for aquatic locomotion,” *IEEE J. Ocean. Eng.* **24**, 237–252.
- Shrivastava, M., Agrawal, A. and Sharma, A. [2013] “A novel level set-based immersed-boundary method for CFD simulation of moving-boundary problems,” *Numer. Heat Transfer, Part B* **63**, 304–326.
- Sotiropoulos, F. and Borazjani, I. [2009] “A review of state-of-the-art numerical methods for simulating flow through mechanical heart valves,” *Med. Biol. Eng. Comput.* **47**, 245–256.
- Sotiropoulos, F. and Yang, X. [2014] “Immersed boundary methods for simulating fluid-structure interaction,” *Prog. Aerosp. Sci.* **65**, 1–21.

- Sussman, M., Smereka, P. and Osher, S. [1994] “A level set approach for computing solutions to incompressible two-phase flow,” *J. Comput. Phys.* **114**, 146–159.
- Tremblay, F. [2002] “Direct and large-eddy simulation of flow around a circular cylinder at subcritical Reynolds numbers,” Dissertation Technische Universität München.
- Triantafyllou, G. S., Triantafyllou, M. S. and Grosenbaugh, M. A. [1993] “Optimal thrust development in oscillating foils with application to fish propulsion,” *J. Fluid Struct.* **7**, 205–224.
- Tseng, Y. H. and Ferziger, J. H. [2003] “A ghost-cell immersed boundary method for flow in complex geometry,” *J. Comput. Phys.* **192**, 593–623.
- Uhlmann, M. [2005] “An immersed boundary method with direct forcing for the simulation of particulate flows,” *J. Comput. Phys.* **209**, 448–476.
- Williamson, C. H. K. and Govardhan, R. [2004] “Vortex-induced vibrations,” *Annu. Rev. Fluid Mech.* **36**, 413–455.
- Yang, J. and Balaras, E. [2006] “An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries,” *J. Comput. Phys.* **215**, 12–40.
- Yang, J., Preidikman, S. and Balaras, E. [2008] “A strongly coupled, embedded-boundary method for fluid-structure interactions of elastically mounted rigid bodies,” *J. Fluid Struct.* **24**, 167–182.
- Yang, X., Zhang, X., Li, Z. and He, G.-W. [2009] “A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations,” *J. Comput. Phys.* **228**, 7821–7836.
- Ye, T., Mittal, R., Udaykumar, H. S. and Shyy, W. [1999] “An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries,” *J. Comput. Phys.* **156**, 209–240.