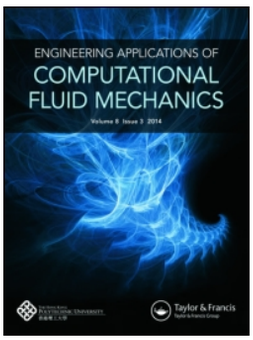

Taylor & Francis
Taylor & Francis Group

Engineering Applications of Computational Fluid Mechanics

ISSN: 1994-2060 (Print) 1997-003X (Online) Journal homepage: https://www.tandfonline.com/loi/tcfm20

Sharp interface immersed boundary method for simulating three-dimensional swimming fish

Zuo Cui, Zixuan Yang & Hongzhou Jiang

To cite this article: Zuo Cui, Zixuan Yang & Hongzhou Jiang (2020) Sharp interface immersed boundary method for simulating three-dimensional swimming fish, Engineering Applications of Computational Fluid Mechanics, 14:1, 534-544, DOI: 10.1080/19942060.2020.1724197

To link to this article: https://doi.org/10.1080/19942060.2020.1724197

Published online: 26 Feb 2020.

ENGINEERING APPLICATIONS OF COMPUTATIONAL FLUID MECHANICS
2020, VOL. 14, NO. 1, 534–544
https://doi.org/10.1080/19942060.2020.1724197

Taylor & Francis
Taylor & Francis Group

Sharp interface immersed boundary method for simulating three-dimensional swimming fish

Zuo Cui[a], Zixuan Yang[b] and Hongzhou Jiang[c]

[a]School of Aerospace Engineering, Guizhou Institute of Technology, Guiyang, People's Republic of China; [b]The State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, People's Republic of China; [c]School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, People's Republic of China

ABSTRACT

A second order finite-difference numerical method is used to solve the Navier–Stokes equations of incompressible flow, in which the solid body with complex geometry is immersed into the fluid domain with orthogonal Cartesian meshes. To account for influences of the solid body, interactive forces are applied as boundary conditions at Cartesian grid nodes located in the exterior but in the immediate vicinity of the solid body. Fluid flow velocities in these nodes are reconstructed to track and control the deformation of the solid body, in which the local direction normal to the body surface is employed using the level-set function. The capabilities of this method are demonstrated by the application to fish swimming, and the computed behaviors of swimming fish agree well with experimental ones. The results elucidate that the ability of swimming fish to produce more thrust and high efficiency is closely related to the Reynolds number. The single reverse Kármán street tends to appear when both the Strouhal number and tail-beating frequency are small, otherwise the double-row reverse Kármán street appears. The algorithm can capture the geometry of a deformable solid body accurately, and performs well in simulating interactions between fluid flow and the deforming and moving body.

ARTICLE HISTORY
Received 21 May 2019
Accepted 8 November 2019

KEYWORDS
Immersed boundary method; self-propelled fish; level-set function; vortex street; Strouhal number

1. Introduction

In many scientific research and engineering applications, problems of fluid–structure interaction (FSI) are inevitably encountered, such as the aeroelastic response of airplane wings (Gao, Zhang, & Tang, 2016), vibrations of wind turbine blades (Löhner et al., 2015; Mou, He, Zhao, & Chau, 2017), blood flow through heart valves (Zakaria et al., 2017) and flow past swimming fish or flying insects (Shrivastava, Malushte, Agrawal, & Sharma, 2017). Most of these take place in the fluid domain with complex immersed boundaries (IBs), and these FSI problems present significant challenges to numerical simulations (Akbarian et al., 2018; Ghalandari, Koohshahi, Mohamadian, Shamshirband, & Chau, 2019; Ramezanizadeh, Nazari, Ahmadi, & Chau, 2019) owing to their nonlinear and strongly coupled characteristics (Mhamed & David, 2013; Mosavi, Shamshirband, Salwana, Chau, & Tah, 2019).

In recent years, several numerical algorithms have been developed to solve FSI problems. Based on the treatment of IBs, they can be categorized into conforming mesh methods and non-conforming mesh methods (Mittal & Iaccarino, 2005). In conforming mesh methods, the physical boundary is regarded as the interface, and meshes are required to conform the interface. However, it is always time-consuming for remeshing or overlapping interpolation when the interface moves or deforms (Hou, Wang, & Layton, 2012). In contrast, the non-conforming mesh methods have advantages in dealing with a deforming or moving solid body. Among them, the IB method is an excellent approach, which was first proposed by Peskin (1972). In IB methods, the body forces are distributed over the nearby flow field of the solid boundary, and are smeared on the interface but not aligned with the mesh. IB methods can be further divided into two different types: (1) diffused interface IB methods, where the body forces are distributed to surrounding grid nodes by discrete delta functions or mask functions (Liu & Vasilyev, 2007); and (2) sharp interface IB methods, where the body forces can be directly applied on the grid points around the boundary (Fadlun, Verzicco, Orlandi, & Mohd-Yusof, 2000). In the present study, the sharp interface IB methods are adopted; these include but are not limited to the Cartesian-IB method (Gilmanov & Sotiropoulos, 2005; Yang & Balaras, 2006), the curvilinear-IB method (Ge & Sotiropoulos, 2007) and

CONTACT Zuo Cui cuizuo@yeah.net

the cut-cell method (Abgrall, Beaugendre, & Dobrzynski, 2014; Meyer, Devesa, Hickel, Hu, & Adams, 2010).

In the Cartesian-IB method, IB points are specified as Cartesian grid points located in the fluid but in the immediate vicinity of the solid body. In comparison, curvilinear grids are used in the curvilinear-IB method. The cut-cell methods use the interface of the solid body to divide the grid cells into two subcells, i.e. the solid phase and fluid phase. Combined with the Cartesian-IB method, Gilmanov and Sotiropoulos (2005) used the ray-tracing algorithm to identify the fluid, solid and IB points. Yang and Balaras (2006) applied a field-extension procedure to extrapolate the velocity and pressure fields of pseudo-fluid points after each deformation. Maitri, Das, Kuipers, Padding, and Peters (2018) used the sharp interface IB method to simulate the particle-laden flows, in which the ghost-cell approach was used to calculate the direct force. Zhu, Seo, and Mittal (2019) proposed a graph-partitioning framework for a sharp interface IB method, and it was suitable to simulate internal flows on large-scale parallel computers. In the study by Wang, Du, and Sun (2019), adaptive mesh refinement (AMR) was applied along with the sharp interface IB method, and the results showed that the AMR approach can increase the computational efficiency.

For FSI problems, the flow fields are strongly influenced by the presence of structures with moving boundaries, which poses a great challenge to the numerical method and computational cost. In this paper, we combine the level-set function with a Cartesian-IB method (Cui, Yang, Jiang, Huang, & Shen, 2018) to simulate the swimming fish. In our method, the level-set function is employed to identify the interface of the solid body. The velocity reconstruction relies on the normal direction of the solid boundary, which is computed from the level-set function and reinitialization process. This algorithm can deal easily with an arbitrarily solid body with a complex smooth surface, and there is no need to regenerate the mesh when the solid body moves or deforms. The capabilities of the method are demonstrated by simulating a swimming fish, in which fish swims in-line by undulating its body. In the present study, two-dimensional (2D) simulations are used to investigate the swimming performance of self-propelled fish. Owing to the computational cost, three-dimensional (3D) simulations of a tethered fish model are used to demonstrate the different patterns of vortex street, because 3D simulations of self-propelled fish would be computationally too expensive to perform several cases in a timely manner (it would take months on parallel computers).

The rest of the paper is organized as follows. The numerical methods, including the fluid solver, the level-set function and the IB method, are introduced in Section 2. A fish model with complex geometry and unique motions is presented in Section 3. Subsequently, several cases of swimming fish are simulated with different fluid viscosity and swimming gaits, and the vortex patterns are analyzed in Section 4. Finally, the conclusions are summarized in Section 5.

2. Computational methodology

2.1. The flow solver

In Cartesian coordinates, the incompressible Navier–Stokes equations of Newtonian fluid are presented as:

$$\frac{\partial u_i}{\partial t} + \frac{\partial \left(u_i u_j\right)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_{bi} \quad (1)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \ (i = 1, \ 2, \ 3) \quad (2)$$

where x_i are the Cartesian coordinates, x, y and z; u_i are the velocity components, u, v and w; p is the pressure, ρ is the density, ν is the kinematic viscosity, and f_{bi} are the components of body force f_b in different directions, which result from the interactions between the solid body and the fluid flow.

In our in-house developed code, the finite difference method is used for spatial discretization, and the second order Runge–Kutta (RK2) method is used for time integration. The pressure Poisson equation is solved by the Krylov-based multigrid solver from the PETSc library. The Navier–Stokes equations are solved by large eddy simulation (LES) methods, and the dynamic Smagorinsky model (Germano, Piomelli, Moin, & Cabot, 1991; Lilly, 1992) is adopted to calculate the turbulent flow. A detailed description can be seen in our previous work (Cui et al., 2018).

2.2. Treatments of the immersed solid boundary

In our code, the treatments of the immersed solid boundary consist of two key procedures: (1) tracking the IB by the level-set function; and (2) reconstructing the local velocity of the points near the IB. The detailed algorithm employed herein has already been presented in our previous work (Cui et al., 2018). For the sake of completeness, the boundary treatments are briefly introduced in this section.

2.2.1. Level-set function

In conventional IB methods, the normal direction of the solid body surface is calculated by generating a mesh on the body surface (Balaras, 2004; Choi, Oberoi, Edwards, & Rosati, 2007; Gilmanov & Sotiropoulos, 2005; Kim, Kim, & Choi, 2001). Differently from that, the level-set

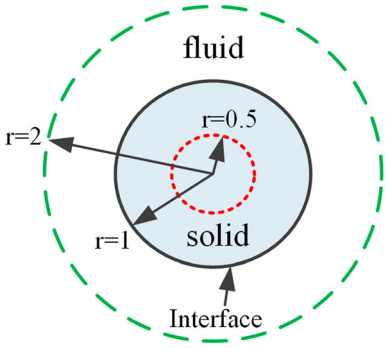

Figure 1. Schematic illustrating the level-set function of a sphere with radius of 1.

function is used to describe the boundary of the solid body, and is expressed as:

$$\phi(\vec{x},t)=\begin{cases}-\Gamma(\vec{x},t) & \text{in solid}\\ 0 & \text{interface}\\ \Gamma(\vec{x},t) & \text{in fluid}\end{cases} \tag{3}$$

where Γ is the distance from any point to the boundaries of the solid body. Taking a sphere as an example, when the radius is 1, the level-set function ϕ is $\sqrt{x^2+y^2+z^2}-1$. As shown in Figure 1, the distance of the points on the boundary is 0, i.e. $\phi(\vec{x},t)=0$. $\phi(\vec{x},t)$ is negative in the solid domain and positive in the fluid domain. The grid–interface relationship is then established by a given level-set function of the interface.

The level-set function of the IB is a signed distance function. Therefore, the unit normal vector $\vec{n}$ can be calculated by:

$$\vec{n}=\frac{\nabla\phi}{|\nabla\phi|} \tag{4}$$

Besides, for the complex geometry of the solid body, the reinitialization process may be needed to correct the level-set function (see the work of Cui et al., 2018).

2.2.2. Interpolation method

In the present IB method, the boundary of the solid body is treated as a sharp interface, and a force $\overrightarrow{f_b}$ is introduced into the Navier–Stokes equation to account for the influences of the solid body. First, the forcing points (denoted as IB points or IB nodes) should be found in the fluid domain immediately adjacent to the solid boundary. Then, the force $\overrightarrow{f_b}$ is calculated by the velocity at the forcing point, which is reconstructed via the bilinear interpolation scheme. The scheme of velocity reconstruction involves the velocities of the solid body and fluid flow, and determines the accuracy and adaptability of flow computations with complex boundaries.

Since the motions of the fish body are prescribed in this work, the location for each solid node is readily determined by the level-set function at every time step. The next step is to reconstruct the velocity vector of the IB points. As shown in Figure 2, point 1 is a solid point, and its variables are prescribed from the solid boundary conditions. The variables of points 2 and 3 in the fluid domain are computed from the previous iteration. Point 0 is the IB point or forcing point, and the boundary conditions are constructed by interpolating the variables along points 1, 2 and 3, where the dashed line $\vec{n}$ is normal to the interface passing through point 1. The normal direction is calculated by the gradient operators of the level-set function (Equation 4). After the boundary conditions interpolation of all IB points, the solution in all fluid points can be advanced to the next iteration.

In our developed code, the fluid and solid solvers need to be coupled to solve the FSI problems. The fluid solver calculates the force exerted on the solid body, and then transfers to the solid solver. Based on the forces, the solid solver updates the position of the solid body, and the fluid solver updates the flow field accordingly. As shown in Figure 3, at the beginning of step $n+1$, the position and velocity status of the solid body are given by an initial guess $S^n=S^*$, and the fluid and solid information, i.e. F^n and S^n, are advanced one step to yield F^{**} and S^{**}, respectively. After one inner iteration step, if the difference between S^* and S^{**} is sufficiently small (10^{-3} in the present study), the inner iteration converges in the dashed frame, and F^{n+1} and S^{n+1} are given as F^{**} and S^{**}, respectively. Otherwise, the new guess of S^* is set as S^{**} and a new inner iteration step starts.

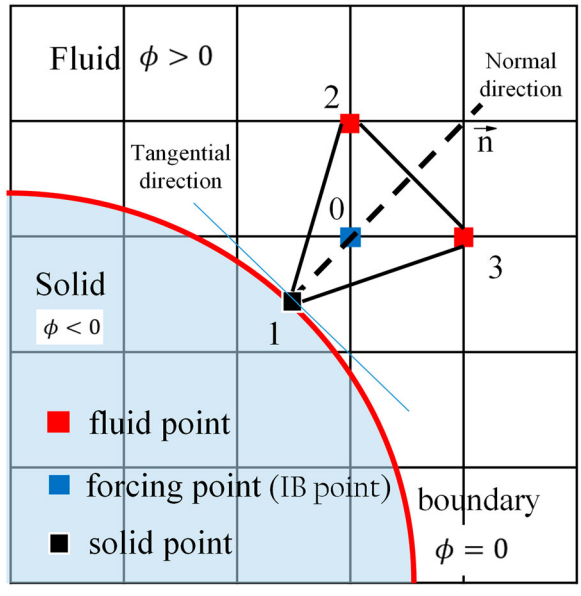

Figure 2. Schematic illustrating the level-set-based Cartesian-immersed boundary (IB) method.

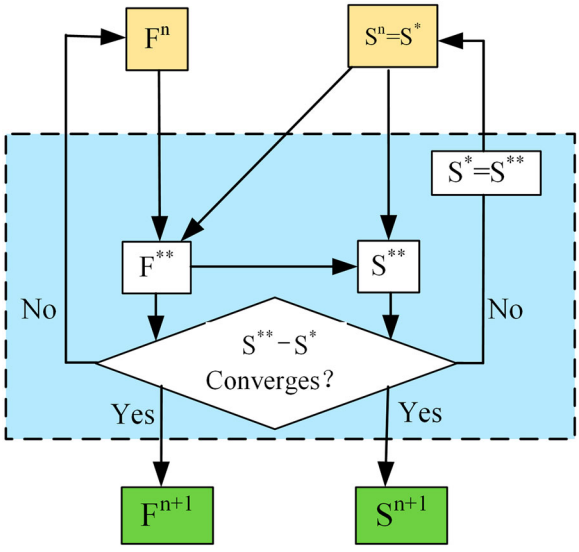

Figure 3. Flowchart of the algorithm of the level-set-based Cartesian-immersed boundary method.

The accuracy of the numerical method has been tested in the context of several benchmark cases, such as the flow past a fixed/oscillating cylinder, vortex-induced vibration of a cylinder, and a 2D self-propelled anguilliform fish. The results agree with previous experimental and numerical results. All these results can be found in our previous work (Cui et al., 2018).

3. Computational fluid dynamics fish model

3.1. Undulatory motions

Based on biological observations (Videler, 1993), the undulations of a fish body can be described by wave-like motions, and they are expressed as:

$$y(x,t) = A(x)\sin(2\pi ft + kx) \tag{5}$$

where f is the beating frequency of the fish tail, k is the wave number, and $A(x)$ is the amplitude function. $A(x)$ is

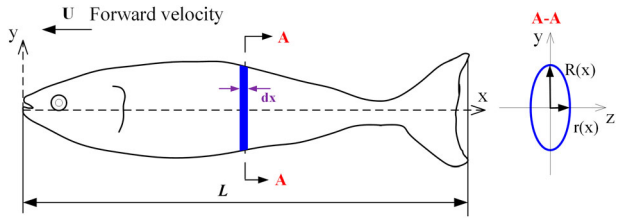

Figure 5. Shape of the fish body with elliptical cross-section.

varied along the fish's body, and described by a quadratic function:

$$A(x) = a_1 + a_2 x + a_3 x^2 \tag{6}$$

where a_1, a_2 and a_3 are the amplitude coefficients, which have been summarized by Alvarado (2007). For example, a set of parameters to describe these motions can be listed as: $k = 5.7/L$, $a_1 = 0.02L$, $a_2 = -0.12$ and $a_3 = 0.2/L$, where L is the full length of the fish body. When $f = 2$ Hz, the midline motions of the swimming fish in one tail-beat period are shown in Figure 4.

As shown in Figure 5, the shape of the fish body is described by the elliptical cross-section, which includes the caudal fin (excluding all other fins). The profiles of the cross-section are defined by $R(x)$ and $r(x)$, and they are expressed as (Alvarado, 2007):

$$R(x) = 0.14L\sin\left(\frac{2\pi x}{1.6L}\right) + 0.0008L(e^{2\pi x/1.1L} - 1) \tag{7}$$

$$r(x) = 0.045L\sin\left(\frac{2\pi x}{1.25L}\right) + 0.06L\sin\left(\frac{2\pi x}{3.14L}\right) \tag{8}$$

3.2. Self-propelled fish model

A self-propelled fish undulates its body with prescribed motions, and accelerates from a stationary state. At first, the thrust is generated by the interactions with fluid flow,

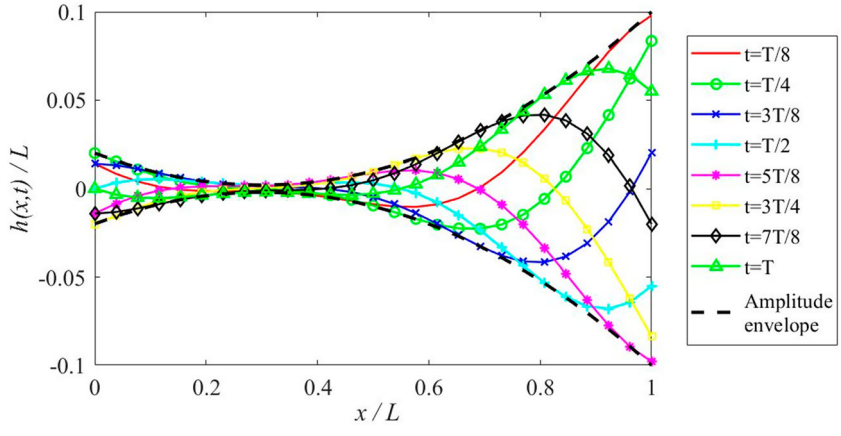

Figure 4. Midline motions in one tail-beat period.

and produces a forward movement. Then, a drag force is produced owing to the viscosity of fluid flow, which is smaller than the thrust at the beginning. Therefore, the forward speed increases, as well as the drag force, until the thrust and drag forces are balanced. At that time, the fish will swim with a constant average velocity, reaching a quasi-steady state. During this process, the governing equation is expressed as:

$$m_f \frac{\partial^2 x_c}{\partial t^2} = F_f \tag{9}$$

where m_f is the mass of the fish body; x_c is the gravity center position of the fish body in the x direction; and F_f is the net force.

In this paper, m_f can be calculated by $\rho_f V_f$, ρ_f is the density of the fish body, equal to that of fluid flow ρ; and V_f is the volume of the fish body. The net force $F_f = (F_x, F_y, F_z)$ is calculated by summing the forces of all the IB points ($m = 1, 2, \ldots, M$), and the formulae are shown as (Lai & Peskin, 2000; Le, Khoo, & Peraire, 2006):

$$F_x = -\sum_{m=1}^{M} f_{ix}, \quad F_y = -\sum_{m=1}^{M} f_{iy}, \quad F_z = -\sum_{m=1}^{M} f_{iz} \tag{10}$$

where f_{ix}, f_{iy} and f_{iz} are the force of the ith IB point in the x, y and z directions, respectively.

The consumed power P_c is calculated as:

$$P_c = \sum_{m=1}^{M} (f_{xi}u_i + f_{yi}v_i + f_{zi}w_i) \tag{11}$$

where u_i, v_i and w_i are the velocity of ith IB point in the x, y and z directions, respectively.

Additional parameters used to describe the self-propelled fish model are listed as follows:

(1) Reynolds number (Re): this is defined by the fish length and the reference velocity (fish length times tail-beat frequency, body length/s), and expressed as:

$$\mathrm{Re_f} = \frac{L^2}{(T\nu)} \tag{12}$$

where T is the tail-beat period. For example, when $\nu = 6.25e-4\,(\mathrm{N}\cdots)/\mathrm{m}^2$, $T = 0.4\,\mathrm{s}$, $L = 1\,\mathrm{m}$, the calculated Re_f is 4000.

(2) Strouhal number (St): in swimming fish, this is defined as:

$$\mathrm{St} = \frac{f A_{\max}}{U} \tag{13}$$

where $A_{\max}$ is the maximum amplitude, and U is the forward speed.

(3) Slip ratio β: this is defined as (Borazjani & Sotiropoulos, 2010; Eloy, 2012; Piñeirua, Godoy-Diana, & Thiria, 2015):

$$\beta = \frac{U}{V} \tag{14}$$

where V is the phase velocity of the undulatory motion.

(4) Elongated body theory (EBT) efficiency: this is introduced to quantify swimming efficiency, and expressed as (Cheng & Blickhan, 1994):

$$\eta_{\mathrm{EBT}} = \frac{1}{2}(1+\beta) - \frac{1}{2}\left(\frac{\lambda}{2\pi}\frac{h'(L)}{h(L)}\right)^2 \frac{\beta^2}{1+\beta} \tag{15}$$

where λ is the wavelength of the fish's undulatory motions, and $h(L)$ and $h'(L)$ are the amplitude and its derivative, respectively, at the fish tail.

4. Results and discussion

In this section, the numerical method is applied to simulate a self-propelled swimming fish. As shown in Figure 6, the fish is placed at $48.5L$ from the inlet plane in the x direction, and centered in the y and z directions. A refined uniform mesh with a smaller grid size is used around the fish. The uniform mesh is used in the x direction, while the non-uniform mesh is adopted in the y and z directions, in which a hyperbolic tangent function is used. In the present study, the effects of grid resolution and time step dt have been tested, and the results indicate that a resolution of the grid near the fish body $\Delta x \times \Delta y \times \Delta z = 0.02L \times 0.005L \times 0.005L$ achieves resolution independency, and $dt = 0.005$ is used to warrant numerical iterations. The set-up parameters are shown in Table 1.

For the boundary conditions, the periodic boundary condition is used in the x direction and the no-slip condition is applied in the other boundaries. The boundary conditions are shown in Table 2. It is notable that a $0.5L$ sponge layer is adopted at the end of the outlet

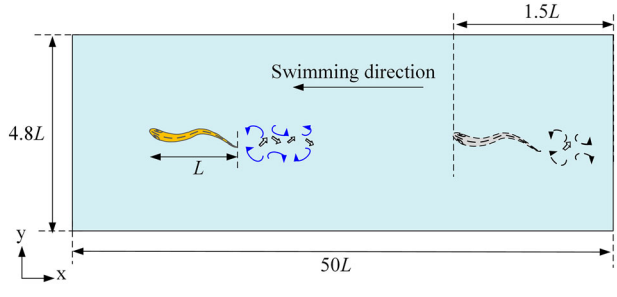

Figure 6. Computational domain of a self-propelled swimming fish.

Table 1. Set-up parameters of the self-propelled fish model.

Case	Computational domain	Grid mesh	CPU number	Mesh size round fish body
2D	$50L \times 4.8L$	2048×256	128	$0.02L \times 0.002L$
3D	$50L \times 5L \times 5L$	$2048 \times 160 \times 160$	512	$0.02L \times 0.005L \times 0.005L$

Note: 2D = two-dimensional; 3D = three-dimensional; CPU = central processing unit.

Table 2. List of boundary conditions in the cases of swimming fish.

Boundaries	Formulae
Inlet	$\partial p/\partial y = 0$, $u = u_b$, $v = v_b$
Outlet	$\partial u/\partial t + U_{\text{conv}}(\partial u/\partial x) = 0$, U_{conv} is the convection outlet velocity
Periodic	$u_{\text{in}} = u_{\text{out}}$, $p_{\text{in}} = p_{\text{out}}$ u_{in} is the inlet velocity, u_{out} is the outlet velocity; p_{in} is the inlet pressure, p_{out} is the outlet pressure
No-slip	$\mu(\partial u/\partial y + \partial v/\partial x) = 0$, $\mu(\partial w/\partial y + \partial v/\partial z) = 0$
Solid wall	$u = v = w = 0$

boundary, which is used to eliminate the effect of the periodic boundary on the fish's swimming performance. To reduce the computational time, our code is parallelized using the message passing interface (MPI) based on domain decomposition. For 2D cases of swimming fish, it would take almost one week on parallel computers using 128 central processing units (CPUs). For 3D cases, the grid number is around 52.43 million, and it would take weeks on parallel computers using 512 CPUs.

4.1. Forward speed

For the 2D self-propelled fish, it swims forward in a straight line with undulatory motions, which are described as in Section 3.1. The fish starts at rest and accelerates to achieve a larger forward speed. When the Reynolds number is varied from 600 to 1500, the changing patterns of forward speed are shown in Figure 7. The unit BL/s means the body length per second, which is used to describe the forward speed. As a whole, when the fish swims in a steady state, the mean forward speed increases with Reynolds number and the instantaneous speed fluctuates periodically, matching the tail-beat period.

Other parameters used to evaluate swimming performance are also listed in Table 3. In general, the kinematic viscosity of fluid flow decreases with Reynolds number. For a self-propelled fish, when it swims in a steady state with constant forward speed, the drag force will also decrease with Reynolds number. Therefore, a smaller thrust force is required to overcome this drag force, and the net thrust force will be increased. As a result, the forward speed of the swimming fish would be increased to reach the force balance, and more consumed power is needed owing to the larger speed and

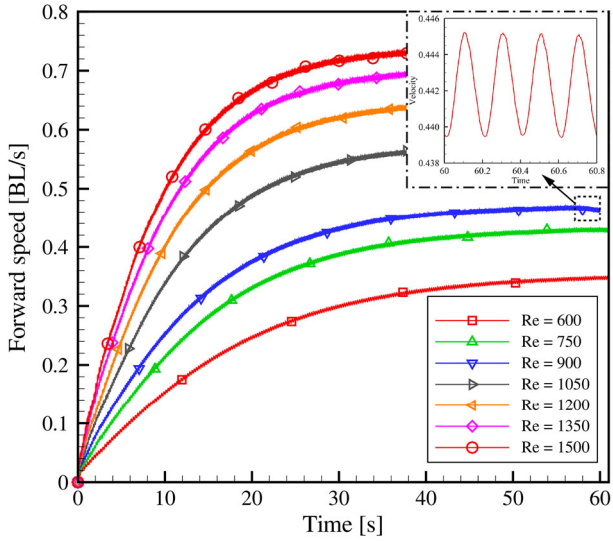

Figure 7. Forward speed of a self-propelled fish varies with Reynolds number (Re).

thrust force. Therefore, the Strouhal number and the slip ratio increase with the Reynolds number (or the forward speed). The consumed power increases with the Reynolds number, while the EBT swimming efficiency gradually increases. These results elucidate that the ability of a swimming fish to produce more thrust and high efficiency is closely related to the Reynolds number.

When the Reynolds number is large enough, the fluid flow becomes inviscid. The fish body undulates and pushes the surrounding water backwards, and pairs of counter-directional rolling vortices are released, producing a jet flow or thrust force along the swimming direction. On both sides of this jet, vortices start to stretch vertically and break into weaker vortices as they are advected axially and vertically (see Section 4.2).

4.2. Vortex analysis

To demonstrate the capabilities of the numerical method, the flow past a 3D tethered fish body is simulated with different body motions. Experiments involving flapping flat plates, as well as swimming fish, have shown the important effects of Strouhal number on the vortex structure of the wake flow (Eloy, 2012). More importantly, the studies of Borazjani and Sotiropoulos (2008, 2009, 2010) showed that the single-row and double-row wake structures of

Table 3. Swimming performance of the self-propelled fish model.

Case no.	Re	Speed (BL/s)	Thrust (N)	Slip ratio	EBT efficiency (%)	St	Consumed power (NàĹŹ m/s)
1	600	0.350	2.432e−3	0.3175	65.9	1.143	1.057e−2
2	750	0.432	3.603e−3	0.3919	69.6	1.157	2.058e−2
3	900	0.492	4.780e−3	0.4246	71.2	1.282	3.358e−2
4	1050	0.605	5.391e−3	0.5189	75.9	1.224	4.880e−2
5	1200	0.675	6.328e−3	0.5851	79.3	1.240	7.334e−2
6	1350	0.715	7.936e−3	0.6305	81.5	1.295	1.023e−1
7	1500	0.740	9.199e−3	0.6623	83.1	1.370	1.373e−1

Note: Re = Reynolds number; EBT = elongated body theory; St = Strouhal number.

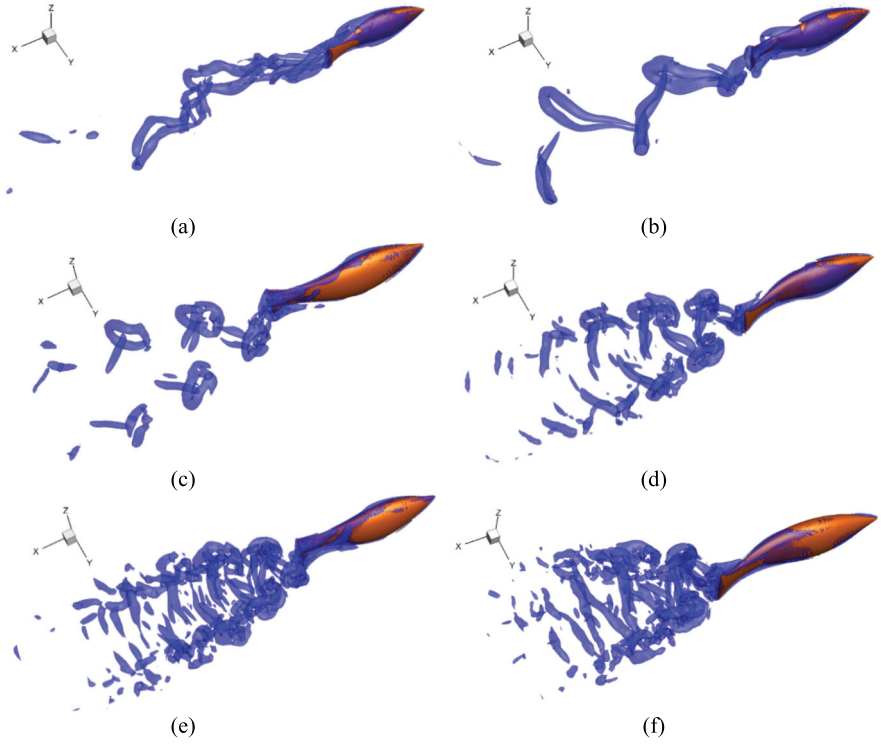

Figure 8. Three-dimensional wake structures of a swimming fish varied with tail-beat frequency: isosurface of vorticity magnitude visualized using the λ^2 criterion. (a) $f = 0.5$ Hz (St = 0.1); (b) $f = 1.0$ Hz (St = 0.2); (c) $f = 2.0$ Hz (St = 0.4); (d) $f = 3.0$ Hz (St = 0.6); (e) $f = 4.0$ Hz (St = 0.8); (f) $f = 5.0$ Hz (St = 1.0).

swimming fish depend on the Strouhal number. In contrast, we focus on the study of the influences of kinematic parameters (i.e. amplitude, tail-beat frequency and wave number) on the pattern of the wave structure. The results are shown in Figures 8–10, respectively.

All of the cases in Figure 8 have the same maximum amplitude (0.2) and the same wave number (5.7). When the tail-beat frequency is varied from 0.5 to 5 Hz, two types of vortex street structures appear behind the fish tail. When the tail-beat frequency is 0.5 Hz (St = 0.1) and 1.0 Hz (St = 0.2), a single-row vortex street structure with staggered braided hairpin vortices is observed. But if the tail-beat frequency increases, the structure of the single-row vortex street turns into a double-row structure and the vorticity also increases. In Figure 8, the structures of the vortex ring are clearly visualized using the λ^2 criterion, and the jet-like flow induced by vortex rings is directed away, producing thrust force that helps the fish to swim forward.

In the present study, it was found that the single-row and double-row vortex structures of the wake flow are dependent on tail-beat frequency or Strouhal number. Dong, Mittal, and Najjar (2006) also reported that single-row and double-row vortex structures can appear in flow past a flapping ellipsoidal foil. Similarly, Buchholz and Smits (2005) observed a single row of vortices when the Strouhal number is 0.2–0.25, and a double vortex row with two separate trains when the Strouhal number is large. In Figure 8, it is clear that the vortex rings are linked with each other. This is consistent with experimental results using live fish, which revealed that these staggered vortices can further improve the abilities of thrust production or swimming maneuvers.

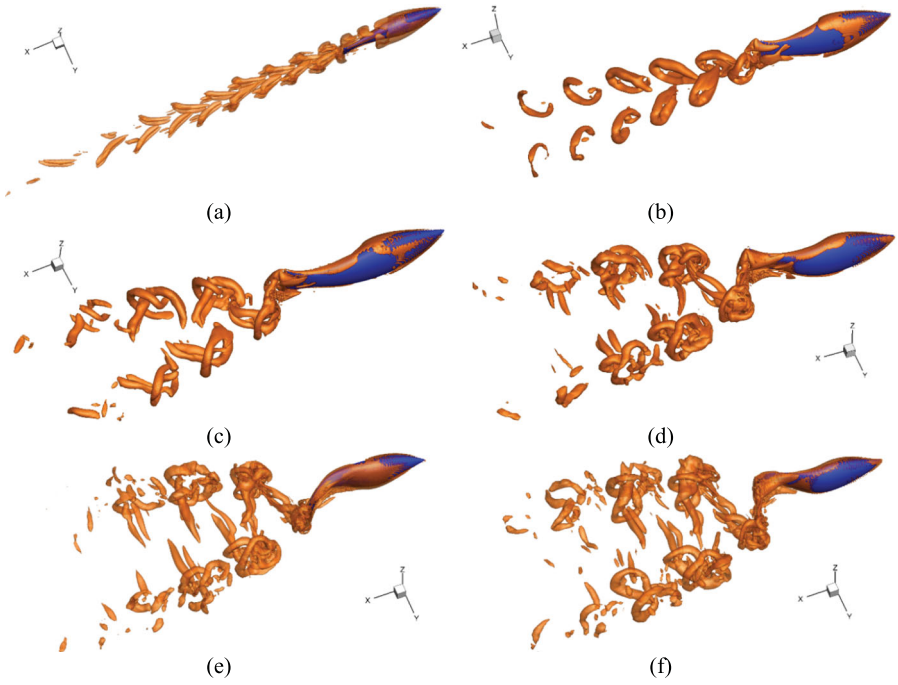

Figure 9. Three-dimensional wake structures of a swimming fish varied with beating amplitude: isosurface of vorticity magnitude visualized using the λ^2 criterion. (a) $A_c = 0.25$, St $= 0.11$; (b) $A_c = 0.5$, St $= 0.22$; (c) $A_c = 0.75$, St $= 0.33$; (d) $A_c = 1.5$, St $= 0.66$; (e) $A_c = 2.0$, St $= 0.88$; (f) $A_c = 2.5$, St $= 1.1$.

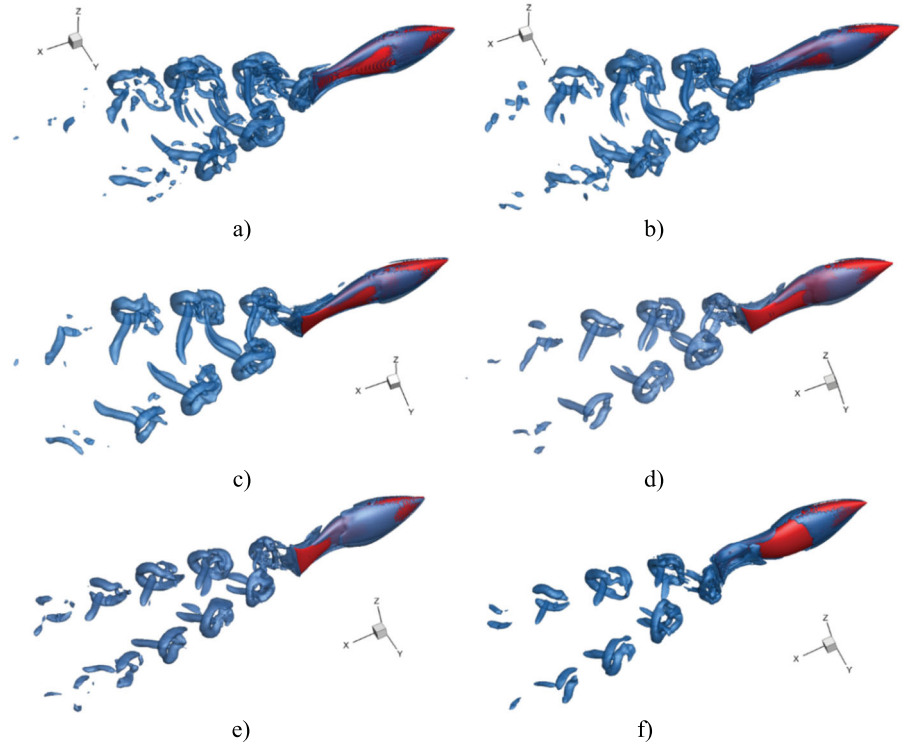

Figure 10. Three-dimensional wake structures of a swimming fish varied with wave number: isosurface of vorticity magnitude visualized using the λ^2 criterion. (a) $k = 1.5$, St $= 0.44$; (b) $k = 3.0$, St $= 0.44$; (c) $k = 4.5$, St $= 0.44$; (d) $k = 6.0$, St $= 0.44$; (e) $k = 7.5$, St $= 0.44$; (f) $k = 10.5$, St $= 0.44$.

For the cases in Figure 9, the amplitude is changed by multiplying by a constant A_c, i.e. $A_c A(x)$, and A_c is changed from 0.25 to 2.5. Meanwhile, the tail-beat frequency is the same (2.2 Hz) and the wave number is also the same ($k = 5.7$). With different amplitude, the Strouhal number varies from 0.11 to 1.1, and the 3D wake structures visualized by the instantaneous isosurfaces of the λ^2 criterion are shown in Figure 9. In Figure 9(a), the Strouhal number is 0.11, but the wake pattern is featured as a double-row wake. At the beginning, the vortex is attached to the fish surface, and then the vortices are shed from the tail in a wedge-like arrangement. Owing to the undulation of the fish body, more than one bump regions exists on the fish surface, which results in the vortices detaching in a staggered way.

For the cases in Figure 10, the amplitude and tail-beat frequency are the same ($A = 0.2, f = 2.2$), and the wave number is varied from 1.5 to 10.5. The structures of the vortex street are double-row, and there are no obvious changes in Figure 10. This result is consistent with the conclusion in our previous study (Cui, Gu, Li, & Jiang, 2017), in which we found that the wave number of body motions has little effect on swimming performance.

In Figures 8–10, the vortex structures of the wake shed from the fish tail for all cases are fairly similar at high Strouhal numbers, but the main difference is seen at low Strouhal numbers. At high Strouhal numbers, the wake behind the fish tail grows vertically as well as axially owing to high vertical or lateral velocity, and a double-row structure can be observed. It should be noted that the single-row vortical structure is observed at low Strouhal number (St = 0.1, $f = 0.5$ Hz) in Figure 8(a), but a double-row vortical structure is also observed at low Strouhal number (St = 0.11, $A = 0.25$) in Figure 9(a). For these two cases, the vortices are not strong enough to create a strong vortex wake similar to the other cases. In Figure 8(a), the formed vortex rings are partially linked to the immediately released one, but become separated completely before detaching from the tail and shedding into the wake. This merging of the single-row vortex street occurs because of the relatively slow vertical velocity. For this case in Figure 9(a), a double row consisting of two reverse Kármán vortex streets has formed, and these vortices propagate along the axial and lateral directions to form two rows of vortices due to the high tail-beat frequency (compared with the case in Figure 8a).

Borazjani and Sotiropoulos (2008, 2009) found that the vortex structure of the fish wake was related to the Strouhal number. When the Strouhal number is small, the vortex street structure of the fish wake is a single-row vortex street structure, whereas when the Strouhal number is large, it corresponds to a double-row vortical structure. The premise of this conclusion is to assume that the tail-beat amplitude remains unchanged, while the frequency is changed to obtain different Strouhal numbers. In contrast, the Strouhal number in the present study is changed by either amplitude or tail-beat frequency. Our results show that: (1) the single-row vortex street structure will appear only when both Strouhal number and tail-beat frequency are small; (2) the vortex street structure of the fish body wake is mainly determined by the tail-beat frequency and Strouhal number; and (3) it is independent of the tail-beat amplitude and wave number. These results further revise the findings in previous research on the wake vortex street structure (Borazjani & Sotiropoulos, 2008, 2009, 2010).

The vertical or lateral distances of the adjacent vortices are also varied in Figures 8–10. In Figure 8, when $f = 0.5$ Hz, the vertical length of the single-row wake structure is about one length of the fish body, and the lateral length is about $0.3L$. As the tail-beat frequency increases, the vertical length reduces, but the lateral length increases. When $f = 4.0$ Hz or 5.0 Hz, the vertical length becomes very small and the lateral length is about $1.0L$. In contrast, when the amplitude increases, the lateral distances of the vortex streets expand gradually, from $0.2L$ (Figure 9a) to $1.2L$ (Figure 9f), but the vertical distances show no obvious changes. In Figure 10, the changes in wave number have very a small influence on the vertical and lateral distances.

5. Conclusions

In this paper, a sharp interface Cartesian-IB method has been presented to simulate the interactions between fluid flow and a solid body with complex deformed boundaries. In this method, the interface of the solid body is treated as a level-set function, and its normal direction is used to reconstruct the velocity of forcing points. Further, the numerical method is applied to simulate fish swimming, and the 2D simulations of a self-propelled fish show that the computed behaviors are in good agreement with experimental studies on live fish. The computational results of swimming fish indicate that the level-set function is suitable to describe the complex geometry of the fish body, and the level-set-based IB method captures the interactions between a deformable body and the surrounding fluid flow accurately. The vortex street behind the fish tail is also analyzed when the motions of the fish body are varied with the amplitude, frequency and wave number. The numerical results of the 3D fish model show that the single reverse Kármán street is prone to appear when both the St number and tail-beat frequency are small; otherwise, the double-row reverse Kármán street appears. Therefore, we predict that the double-row

reverse Kármán street appears for most swimming fish, to optimize their integrative swimming performance.

It should be noted that we assumed that the weight of self-propelled fish is balanced with the buoyancy, and neglected rotations (yaw, pitch and roll) in the dynamics of motion. In addition, we did not investigate the performances under different swimming patterns, such as fast starts, C-turning and maneuvering. In our simulations, the motions of the fish body are prescribed, which occasionally happen in the real world. Therefore, many more numerical cases of swimming fish with optimized kinematic parameters should be investigated in the future.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by a Scientific Start-up Project of GuiZhou Institute of Technology [grant number XJGC20190956]; and the fund of the Research Cultivation and Technology Exploration Program of GuiZhou Institute of Technology [grant number [2017]5789-20].

References

Abgrall, R., Beaugendre, H., & Dobrzynski, C. (2014). An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, *257*, 83–101.

Akbarian, E., Najafi, B., Jafari, M., Ardabili, S. F., Shamshirband, S., & Chau, K. (2018). Experimental and computational fluid dynamics-based numerical simulation of using natural gas in a dual-fueled diesel engine. *Engineering Applications of Computational Fluid Mechanics*, *12*(1), 517–534.

Alvarado, P. V. Y. (2007). *Design of biomimetic compliant devices for locomotion in liquid environments*. Cambridge, MA: Massachusetts Institute of Technology.

Balaras, E. (2004). Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Computers & Fluids*, *33*, 375–404.

Borazjani, I., & Sotiropoulos, F. (2008). Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*, *211*, 1541–1558.

Borazjani, I., & Sotiropoulos, F. (2009). Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*, *212*, 576–592.

Borazjani, I., & Sotiropoulos, F. (2010). On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming. *Journal of Experimental Biology*, *213*, 89–107.

Buchholz, H. J., & Smits, A. J. (2005). Wake of a low aspect ratio pitching plate. *Physics of Fluids*, *17*, 091102.

Cheng, J. Y., & Blickhan, R. (1994). Note on the calculation of propeller efficiency using elongated body theory. *Journal of Experimental Biology*, *192*, 169–177.

Choi, J. I., Oberoi, R. C., Edwards, J. R., & Rosati, J. A. (2007). An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, *224*, 757–784.

Cui, Z., Gu, X., Li, K., & Jiang, H. (2017). CFD studies of the effects of waveform on swimming performance of carangiform fish. *Applied Sciences*, *7*(2), 149.

Cui, Z., Yang, Z., Jiang, H. Z., Huang, W. X., & Shen, L. (2018). A sharp-interface immersed boundary method for simulating incompressible flows with arbitrarily deforming smooth boundaries. *International Journal of Computational Methods*, *15*(1), 1750080.

Dong, H., Mittal, R., & Najjar, F. M. (2006). Wake topology and hydrodynamic performance of low-aspect-ratio flapping foils. *Journal of Fluid Mechanics*, *566*, 309.

Eloy, C. (2012). Optimal Strouhal number for swimming animals. *Journal of Fluids and Structures*, *30*, 205–218.

Fadlun, E. A., Verzicco, R., Orlandi, P., & Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, *161*, 35–60.

Gao, X., Zhang, Q., & Tang, Q. (2016). Fluid-structure interaction analysis of parachute finite mass inflation. *International Journal of Aerospace Engineering*, *1*, 1–8.

Ge, L., & Sotiropoulos, F. (2007). A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. *Journal of Computational Physics*, *225*, 1782–1809.

Germano, M., Piomelli, U., Moin, P., & Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, *3*, 1760–1765.

Ghalandari, M., Koohshahi, E. M., Mohamadian, F., Shamshirband, S., & Chau, K. W. (2019). Numerical simulation of nanofluid flow inside a root canal. *Engineering Applications of Computational Fluid Mechanics*, *13*(1), 254–264.

Gilmanov, A., & Sotiropoulos, F. (2005). A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, *207*, 457–492.

Hou, G., Wang, J., & Layton, A. (2012). Numerical methods for fluid-structure interaction – A review. *Communications in Computational Physics*, *12*, 337–377.

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

Lai, M. C., & Peskin, C. S. (2000). An immersed boundary method with formal secondorder accuracy and reduced numerical viscosity. *Journal of Computational Physics*, *160*, 705–719.

Le, D. V., Khoo, B. C., & Peraire, J. (2006). An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *Journal of Computational Physics*, *220*, 109–138.

Lilly, D. K. (1992). A proposed modification of the Germano subgrid-scale closure method. *Physics of Fluids A: Fluid Dynamics*, *4*, 633–635.

Liu, Q., & Vasilyev, O. V. (2007). A Brinkman penalization method for compressible flows in complex geometries. *Journal of Computational Physics*, *227*, 946–966.

Löhner, R., Haug, E., Michalski, A., Muhammad, B., Drego, A., Nanjundaiah, R., & Zarfam, R. (2015). Recent advances in

computational wind engineering and fluid–structure interaction. *Journal of Wind Engineering and Industrial Aerodynamics*, *144*, 14–23.

Maitri, R. V., Das, S., Kuipers, J. A. M., Padding, J. T., & Peters, E. A. J. F. (2018). An improved ghost-cell sharp interface immersed boundary method with direct forcing for particle laden flows. *Computers & Fluids*, *175*, 111–128.

Meyer, M., Devesa, A., Hickel, S., Hu, X., & Adams, N. (2010). A conservative immersed interface method for large-eddy simulation of incompressible flows. *Journal of Computational Physics*, *229*, 6300–6317.

Mittal, R., & Iaccarino, G. (2005). Immersed boundary methods. *Annual Review of Fluid Mechanics*, *37*, 239–261.

Mosavi, A., Shamshirband, S., Salwana, E., Chau, K., & Tah, J. H. M. (2019). Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning. *Engineering Applications of Computational Fluid Mechanics*, *13*(1), 482–492.

Mou, B., He, B.-J., Zhao, D.-X., & Chau, K. (2017). Numerical simulation of the effects of building dimensional variation on wind pressure distribution. *Engineering Applications of Computational Fluid Mechanics*, *11*(1), 293–309.

Peskin, C. S. (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, *10*, 252–271.

Piñeirua, M., Godoy-Diana, R., & Thiria, B. (2015). Resistive thrust production can be as crucial as added mass mechanisms for inertial undulatory swimmers. *Physical Review E*, *92*(2), 021001.

Ramezanizadeh, M., Nazari, M. A., Ahmadi, M. H., & Chau, K. (2019). Experimental and numerical analysis of a nanofluidic thermosyphon heat exchanger. *Engineering Applications of Computational Fluid Mechanics*, *13*(1), 40–47.

Shrivastava, M., Malushte, M., Agrawal, A., & Sharma, A. (2017). CFD study on hydrodynamics of three fish-like undulating hydrofoils in side-by-side arrangement. In *Fluid mechanics and fluid power–contemporary research*. New Delhi: Springer.

Souli, M. H., & Benson, D. J. (2013). *Arbitrary Lagrangian Eulerian and fluid-structure interaction: Numerical simulation*. Hoboken, New Jersey: John Wiley & Sons.

Videler, J. J. (1993). *Fish swimming*. London, UK: Chapman and Hall.

Wang, Z., Du, L., & Sun, X. (2019). Adaptive mesh refinement for simulating fluid-structure interaction using a sharp interface immersed boundary method. *Computers & Fluids*. In press. https://doi.org/10.1016/j.compfluid.2019.06.002.

Yang, J., & Balaras, E. (2006). An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, *215*, 12–40.

Zakaria, M. S., Ismail, F., Tamagawa, M., Aziz, A. F. A., Wiriadidjaja, S., Basri, A. A., & Ahmad, K. A. (2017). Review of numerical methods for simulation of mechanical heart valves and the potential for blood clotting. *Medical & Biological Engineering & Computing*, *55*(9), 1519–1548.

Zhu, C., Seo, J. H., & Mittal, R. (2019). A graph-partitioned sharp-interface immersed boundary solver for efficient solution of internal flows. *Journal of Computational Physics*, *386*, 37–46.