Applied Mathematical Modelling 114 (2023) 627–645

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

An efficient explicit immersed boundary-reconstructed lattice Boltzmann flux solver for isothermal fluid-structure interaction problems with large deformations and complex geometries

Buchen Wu [a,b], Jinhua Lu [e], HsuChew Lee [c], Chang Shu [b], Minping Wan [a,c,d,*]

[a] National Center for Applied Mathematics Shenzhen (NCAMS), Southern University of Science and Technology, Shenzhen, Guangdong 518055, China
[b] Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore
[c] Guangdong Provincial Key Laboratory of Turbulence Research and Applications, Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China
[d] Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China
[e] Department of Mechanical Engineering, Chair of Aerodynamics and Fluid Mechanics, Technical University of Munich, Boltzmannstraße 15, Garching 85748, Germany

ARTICLE INFO

Article history:
Received 19 April 2022
Revised 6 October 2022
Accepted 10 October 2022
Available online 15 October 2022

Keywords:
Reconstructed lattice Boltzmann flux solver
Explicit immersed boundary method
Arbitrary Lagrangian-Eulerian method
Rigid/flexible structure-fluid interaction

ABSTRACT

In this paper, a novel approach for isothermal fluid-structure interaction (FSI) problems is proposed that not only inherits the advantages of a reconstructed lattice Boltzmann flux solver (RLBFS) in solving the fluid field, but also retains the efficiency found in the explicit boundary condition-enforced immersed boundary method (EIB) for implementation of boundary conditions on the solid wall. Hence, the new approach (EIB-RLBFS) is highly suitable for complex FSI problems such as large deformations and complex geometries. Furthermore, the arbitrary Lagrangian-Eulerian (ALE) approach is integrated with EIB-RLBFS to solve moving boundary problems efficiently. The structural dynamics are solved by a finite difference method for moving objects and flexible structures. The overall framework of EIB-RLBFS is simple yet robust, where a fractional method is applied to split the FSI process into a predictor step and a corrector step. The RLBFS is applied in the predictor step to predict the intermediate flow field, while the EIB is invoked in the corrector step to impose the no-slip boundary conditions within the flow field, thereby, correcting the predicted flow field to satisfy the no-slip boundary conditions. The proposed approach (EIB-RLBFS) has been tested and evaluated extensively in terms of accuracy and efficiency on several benchmark cases such as, a 2D self-propelled unconstrained flapping/undulatory foil, a 2D filament, a 3D streamwise rotating sphere, and a 3D flapping flag. Results obtained using the proposed approach are in good agreement with previous studies, substantiating the capability and flexibility of EIB-RLBFS for solving FSI problems.

* Corresponding author.
E-mail address: wanmp@sustech.edu.cn (M. Wan).

https://doi.org/10.1016/j.apm.2022.10.017
0307-904X/

1. Introduction

Fluid-structure interaction (FSI) problems related to the natural environment have always been fascinating yet challenging. Typical FSI problems include natural swimmer propulsion [1–8] and flapping flexible structures [9–15].

There have been several accurate and efficient numerical methods put forward to solve complex FSI problems, and most attention [16–24] has been placed on the immersed boundary method (IBM) introduced by Peskin [25], owning to its simple yet robust method for simulating complex FSI problems. In contrast to conventional numerical methods employing body fitted meshes, IBMs allow the exchange of boundary information between Lagrangian points (solid structure) and Eulerian points (surrounding flows), and the implementation of no-slip boundary conditions. Hence, IBMs avoid the tedious grid regeneration when simulating flexible structures or moving objects, allowing it to be superior to conventional numerical methods in terms of efficiency.

Many variants of IBMs have been proposed, for example, the penalty forcing [19,25], the feedback forcing [21,22], the direct forcing [23,26,27], momentum exchange [28,29], and implicit boundary condition-enforced schemes [30]. Among those listed IBMs, the implicit boundary condition-enforced scheme proposed by Wu and Shu [30] is a straightforward approach, which obtains the velocity correction by accurately satisfying the no-slip boundary condition. However, the implicit boundary condition-enforced IBM involves a tedious computational process, namely, the assembly of a large matrix and its inverse. Recently, the explicit boundary condition-enforced scheme [31] was proposed to remove these drawbacks of the previous implicit boundary condition-enforced IBM. Due to its high efficiency and accuracy, the explicit boundary condition-enforced scheme [31] is adopted in the present work.

The most time intensive process in simulating FSI problems is the coupling of solid and fluid phases, where a flow-field update process is initiated at every time-step. Although IBM greatly simplifies the coupling process, an efficient flow solver is also needed to further improve the overall efficiency. The popular hybrid approaches can be systematically divided into two groups. In the first approach, the IBM is coupled with the Navier-Stokes (N-S) equation solver [18,21,25]. The second approach is to couple the IBM method with the lattice Boltzmann method (LBM) [14,15,30,32–35]. However, the major disadvantage of N-S solvers lies with the governing equation for the velocity-pressure coupling that is not as computationally efficient as the LBM. In the LBM, there is a special discretization scheme for the space and particle velocity, thereby, further increasing the computational efficiency [36,37]. Therefore, the IBM coupled with LBM has a high potential to efficiently solve FSI problems, accelerating the development of IB-LBM [14,29,30,38–40]. Although the IB-LBM absorbs the advantages of LBM, it also retains the drawbacks of LBM, such as the uniformity of the lattice and the relationship between the time interval and mesh spacing [41]. To circumvent these drawbacks of the N-S and LBM solvers and retain their distinctive advantages, Shu et al. [41] proposed the lattice Boltzmann flux solver (LBFS), which directly calculates macroscopic variables through a finite volume method with the interface fluxes by reconstructing the local solution of the lattice Boltzmann equation (LBE). Based on the LBFS [41] and the implicit boundary condition-enforced IBM [30], a series of FSI solvers [42–45] has been proposed. Note that these solvers focus on the rigid structure-fluid interaction, where the flexible structure is not taken into account.

Recently, based on the LBFS, a reconstructed lattice Boltzmann flux solver (RLBFS) [46] has been proposed for simulating both viscous and inviscid flows, which has higher computational efficiency than the conventional LBFS while retaining the same accuracy. The RLBFS calculates interface fluxes with macroscopic variables rather than reconstructing the LBE at the cell interface, which is crucial for reducing computational time. For instance, Ref. [46] shows that the computational time can be reduced by approximately 30% when RLBFS is employed instead of LBFS for 3D lid-driven cavity flow, simulating on identical grid size of 40^3. Similarly, the RLBFS is a finite volume solver and also preserves the advantages of LBFS, including restraining pressure oscillation and numerical stability.

Several typical FSI problems, such as natural swimmer propulsion and falling particles, require a large computational domain to obtain some statistical results. To effectively simulate these FSI problems, the arbitrary-Lagrangian-Eulerian (ALE) approach [44,45,47] has been extensively used to simulate the moving solid objects in an unbounded flow domain with less grid points, which can effectively reduce the computational cost. Therefore, the ALE approach can be integrated with the FSI numerical approaches to tackle these FSI problems.

1.1. Innovations

With these advanced features of RLBFS and the explicit boundary condition-enforced immersed boundary method (EIB) [31], it warrants further investigation into its capability for studying complex FSI problems. However, there are currently no scheme to combine these two methods to solve isothermal FSI problems. There are two main innovations of the present work. Firstly, the present study proposed a novel coupling approach: the explicit immersed boundary-reconstructed lattice Boltzmann flux solver (EIB-RLBFS) for efficiently solving complex FSI problems, which inherits the advantages of RLBFS and EIB. Secondly, the arbitrary-Lagrangian-Eulerian (ALE) approach is integrated with the EIB-RLBFS, which generates another novel coupling approach ALE-EIB-RLBFS to track moving solid objects in a moving frame with high computational efficiency.

In the present study, a 2D unconstrained flapping foil/undulatory foil and a 3D streamwise rotating sphere are considered to verify the ability of EIB-RLBFS for simulating moving boundary problems. A moving 2D filament and 3D flag are considered to examine the ability of EIB-RLBFS for simulating the dynamic behavior of flexible structure-fluid interaction.

Furthermore, a 2D unconstrained undulatory foil is considered to verify the feasibility of ALE-EIB-RLBFS for efficiently simulating moving objects.

1.2. Organization

The paper is organized as follows: the governing equations of the fluid dynamic and the numerical approaches are introduced in Section 2. The numerical accuracy test and numerical validations are presented in Section 3. Conclusions will be drawn in Section 4.

2. The mathematical model and numerical approach

The governing equations describing the dynamic behavior of the fluid and the numerical approach are introduced in this section. Briefly, the Navier-Stokes equations are discretized using a fractional step method, which involves a sequence of two steps(predictor and corrector). The RLBFS [46] predicts the flow field in the predictor step without considering the solid system. Subsequently, in the corrector step, the velocity correction around the solid system is obtained by implementing the no-slip boundary condition with EIB [31].

2.1. Navier-Stokes (N-S) equations

In a stationary frame, the mass and momentum conservation equations for fluid flows can be written as:

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot \left(\rho_f \mathbf{u}\right) = 0, \tag{1a}$$

$$\frac{\partial \left(\rho_f \mathbf{u}\right)}{\partial t} + \nabla \cdot \left(\rho_f \mathbf{u}\mathbf{u}\right) = -\nabla p + \nabla \cdot \left[\mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)\right] + \mathbf{f}, \tag{1b}$$

where ρ_f denotes the density of fluid, $\mathbf{u}$ is the flow velocity, μ is the dynamic viscosity, p is the pressure, and $\mathbf{f}$ is the external forcing term exerted by the solid system onto the fluid. If the density variation is small and Mach number is low, the above N-S equations can be applied to simulate incompressible flows.

For a moving frame, Eqs. (1a) and (1b) can be rewritten in the following form based on the ALE method as:

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot \left[\rho_f \left(\mathbf{u} - \boldsymbol{U}_{\text{grid}}\right)\right] = 0, \tag{2a}$$

$$\frac{\partial \left(\rho_f \mathbf{u}\right)}{\partial t} + \nabla \cdot \left[\rho_f \mathbf{u} \otimes \left(\mathbf{u} - \boldsymbol{U}_{grid}\right)\right] = -\nabla p + \nabla \cdot \left[\mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)\right] + \mathbf{f}, \tag{2b}$$

where $\boldsymbol{U}_{grid}$ denotes the local velocity of the computational domain.

2.2. The RLBFS for predicting the flow field

In the predictor step, without the inclusion of solid objects, the ρ^{n+1} and the intermediate $\mathbf{u}^m$ are obtained through RLBFS:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \boldsymbol{P} = 0, \tag{3a}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\Pi} = 0, \tag{3b}$$

$$\boldsymbol{P} = \rho \mathbf{u}, \boldsymbol{\Pi} = \rho \mathbf{u}\mathbf{u} + p\boldsymbol{I} - \mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right). \tag{3c}$$

First, RLBFS reconstructs a unit lattice at the cell interface with D2Q9 (for 2D) or D3Q19 (for 3D), as shown in Fig. 1.

Then, the macroscopic variables ρ and $\mathbf{u}$ at the lattice nodes can be obtained by interpolation, and the interpolation scheme is given as:

$$W = \begin{cases} W_L + \nabla W_L \cdot (\mathbf{X}_N - \mathbf{X}_L), \mathbf{X}_N \text{ locates at the left cell;} \\ W_R + \nabla W_R \cdot (\mathbf{X}_N - \mathbf{X}_R), \mathbf{X}_N \text{ locates at the right cell;} \\ 0.5[W_L + \nabla W_L \cdot (\mathbf{X}_N - \mathbf{X}_L) + W_R + \nabla W_R \cdot (\mathbf{X}_N - \mathbf{X}_R)], \\ \qquad\qquad \mathbf{X}_N \text{ locates at the interface;} \end{cases} \tag{4}$$

where $W = \rho$ or $\mathbf{u}$. $\mathbf{X}_L$ and $\mathbf{X}_R$ denote the cell center position of the left and right cells, respectively.

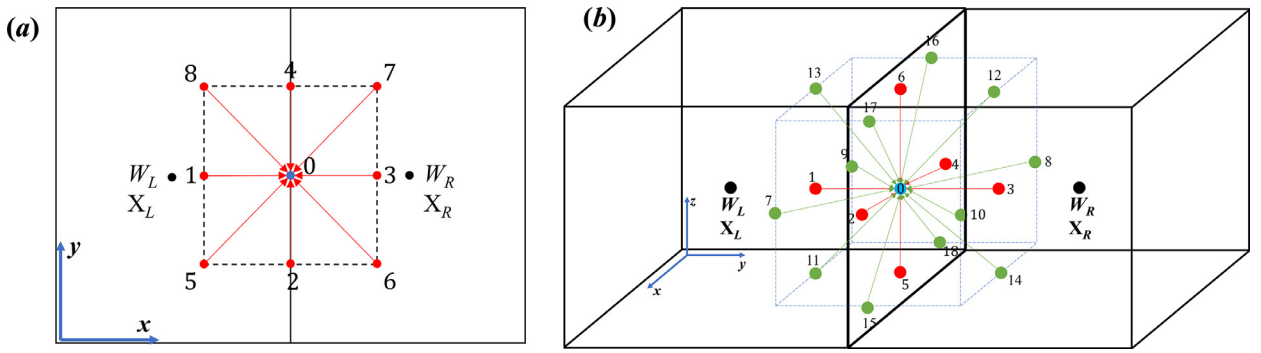

Fig. 1. Local reconstructed unit lattice at the cell interface, (*a*) D2Q9 and (*b*) D3Q19.

Subsequently, the sub-intermediate ρ^* and $(\rho u_\alpha)^*$ at the cell interface can be evaluated by solving the macroscopic equations recovered from the local LBE with the finite difference method, which can be written as follow:

$$\rho^* = \rho^n - \partial_\alpha (\rho u_\alpha)^n \delta t + \tfrac{1}{2}\delta t^2 \partial_\alpha \partial_\beta \left(\rho u_\alpha u_\beta + \rho c_s^2 \delta_{\alpha\beta}\right)^n + O\left(\delta t^3\right), \tag{5a}$$

$$\begin{aligned}(\rho u_\alpha)^* = {} & (\rho u_\alpha)^n - \delta t \partial_\beta \left(\rho u_\alpha u_\beta + \rho c_s^2 \delta_{\alpha\beta}\right)^n \\ & + 0.5 c_s^2 \delta t^2 \partial_\beta \left[\partial_\beta (\rho u_\alpha) + \partial_\alpha \left(\rho u_\beta\right) + \partial_\gamma \left(\rho u_\gamma\right)\delta_{\alpha\beta}\right]^n + O\left(\delta t^3\right),\end{aligned} \tag{5b}$$

where subscripts α, β and γ are coordinate components. c_s denotes the sound speed and δt is the time interval. In this work, the partial derivatives shown in Eq. (5) are discetized on the unit lattice using the second-order central difference scheme. Readers are referred to our previous work [48] for detailed information on the partial derivatives.

The relationships between the macroscopic variables at the interface and the fluxes in Eq. (3c) can be obtained through the second order Taylor Series Expansion [46] and they are given below for stationary and moving frames as:

$$P_\alpha = (\rho u_\alpha)^* - \rho^* \Psi, \tag{6a}$$

$$\begin{aligned}\Pi_{\alpha\beta} = {} & \left(\rho u_\alpha u_\beta + \rho c_s^2 \delta_{\alpha\beta}\right)^* - \nu\left[\partial_\beta (\rho u_\alpha) + \partial_\alpha \left(\rho u_\beta\right) + \partial_\gamma \left(\rho u_\gamma\right)\delta_{\alpha\beta}\right] \\ & - (\tau - 0.5)\left[\left(\rho u_\alpha u_\beta + \rho c_s^2 \delta_{\alpha\beta}\right)^* - \left(\rho u_\alpha u_\beta + \rho c_s^2 \delta_{\alpha\beta}\right)\right] - (\rho u_\alpha)^* \Psi + O\left(\delta t^2\right),\end{aligned} \tag{6b}$$

where

$$\Psi = \begin{cases} 0, & \textit{in the stationary frame}, \\ U_{\text{grid}}, & \textit{in the moving frame}. \end{cases} \tag{7}$$

Finally, the ρ^{n+1} and the intermediate $\mathbf{u}^m$ at the cell center are updated using the flux terms computed at the cell interface as follow:

$$\rho^{n+1} = \rho^n - \frac{\Delta t}{\Delta V} \sum_k P_\alpha \Delta S_{k\alpha}, \tag{8a}$$

$$(\rho u_\alpha)^m = (\rho u_\alpha)^n + \frac{\Delta t}{\Delta V} \sum_k \Pi_{\alpha\beta} \Delta S_{k\beta}, \tag{8b}$$

where ΔV is the volume of the control cell and Δt denotes the macroscopic time interval. ΔS_k denotes the area of the kth interface of the control cell.

2.3. The EIB for correcting the flow field

The explicit immersed boundary method is applied, right after the intermediate velocity $\mathbf{u}^m$ is computed in the predictor step described above, to include the fluid-solid structure interaction through the enforcement of the no-slip boundary condition on the solid surface. The updated velocity can be calculated by the following relationship:

$$\mathbf{u}^{n+1} = \mathbf{u}^m + \Delta \mathbf{u}. \tag{9}$$

Hence, the forcing term in the momentum equation is given as:

$$\mathbf{f} = \rho^{n+1} \frac{\Delta \mathbf{u}}{\Delta t}. \tag{10}$$

The updated velocity should satisfy the no-slip boundary condition, which can be expressed as:

$$\boldsymbol{U}^{n+1}\left(\boldsymbol{X}_B^l\right) = \begin{cases} \sum_j \mathbf{u}^{n+1}\left(\boldsymbol{r}_j\right) \cdot D\left(\boldsymbol{r}_j - \boldsymbol{X}_B^l\right)\Delta h^2, & \text{For 2D} \\ \sum_j \mathbf{u}^{n+1}\left(\boldsymbol{r}_j\right) \cdot D\left(\boldsymbol{r}_j - \boldsymbol{X}_B^l\right)\Delta h^3, & \text{For 3D} \end{cases} \quad l = 1, 2, \ldots, N,\ j = 1, 2, \ldots, M \tag{11}$$

where Δh denotes the mesh spacing of the background fluid grid, N and M are the numbers of Lagrangian points and Eulerian points, respectively. $\boldsymbol{r}_j$ is defined as the physical position of Eulerian points. The kernel distribution function can be expressed as:

$$D(\boldsymbol{r}) = \begin{cases} D_x(r_x)D_y(r_y), & \text{For 2D} \\ D_x(r_x)D_y(r_y)D_z(r_z), & \text{For 3D} \end{cases} \tag{12}$$

Here, the commonly 4-point cosine function is adopted, which can be written as:

$$D_x = D_y = D_z = \begin{cases} \frac{1}{4}(1 + \cos(\pi |r|/2)) & |r| \le 2 \\ 0 & |r| > 2 \end{cases} \tag{13}$$

The correction velocity $\Delta\mathbf{u}(\boldsymbol{r}_j)$ at Eulerian points in Eq. (9) can be interpolated by the correction velocity $\delta\boldsymbol{u}_B^l$ at Lagrangian points as:

$$\Delta\mathbf{u}\left(\boldsymbol{r}_j\right) = \sum_l \delta\boldsymbol{u}_B^l \cdot D\left(\boldsymbol{r}_j - \boldsymbol{X}_B^l\right) \cdot \Delta s^l,\ l = 1, 2, \ldots, N, \tag{14}$$

where Δs^l is the arc length of the solid boundary element for the 2D case, and Δs^l is the area of the solid boundary element for the 3D case. Substituting Eqs. (9) and (14) into Eq. (11), the following formula is obtained as:

$$\begin{aligned} \boldsymbol{U}^{n+1}\left(\boldsymbol{X}_B^l\right) = & \sum_j \mathbf{u}^m\left(\boldsymbol{r}_j\right) \cdot D\left(\boldsymbol{r}_j - \boldsymbol{X}_B^l\right)\Delta h^n \\ & + \sum_j \sum_k \delta\boldsymbol{u}_B^k \Delta s^k \cdot D\left(\boldsymbol{r}_j - \boldsymbol{X}_B^l\right) \cdot D\left(\boldsymbol{r}_j - \boldsymbol{X}_B^k\right)\Delta h^n. \end{aligned} \tag{15}$$

Eq. (15) can be rewritten in the matrix form as follow:

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{B} \tag{16a}$$

$$\boldsymbol{X} = \left[\delta\boldsymbol{u}_B^1\Delta s^1, \quad \delta\boldsymbol{u}_B^2\Delta s^2, \quad \cdots, \quad \delta\boldsymbol{u}_B^N\Delta s^N\right]^T, \tag{16b}$$

$$\boldsymbol{A} = \begin{pmatrix} D_{11} & D_{12} & \cdots & D_{1M} \\ D_{21} & D_{22} & \cdots & D_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ D_{N1} & D_{N2} & \cdots & D_{NM} \end{pmatrix} \begin{pmatrix} D_{11} & D_{12} & \cdots & D_{1N} \\ D_{21} & D_{22} & \cdots & D_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ D_{M1} & D_{M2} & \cdots & D_{MN} \end{pmatrix} \Delta h^n \tag{16c}$$

$$\boldsymbol{B} = \begin{pmatrix} U_B^1 \\ U_B^2 \\ \vdots \\ U_B^N \end{pmatrix}^{n+1} - \begin{pmatrix} D_{11} & D_{11} & \cdots & D_{1M} \\ D_{21} & D_{22} & \cdots & D_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ D_{N1} & D_{N2} & \cdots & D_{NM} \end{pmatrix} \begin{pmatrix} \mathrm{u}_1^{\mathrm{m}} \\ \mathrm{u}_2^{\mathrm{m}} \\ \vdots \\ \mathrm{u}_M^{\mathrm{m}} \end{pmatrix} \tag{16d}$$

Eq. (16) shows that the matrix term $\boldsymbol{A}$ is associated with the kernel function D, and Eq. (13) explicitly states that $D = 0$ when the relative distance between a Lagrangian point and an Eulerian point exceeds $2\Delta h$. The nonzero elements of the kth row of the matrix $\boldsymbol{A}$ are taken into account. If there exist q nonzero elements in the kth row of the matrix $\boldsymbol{A}$, the kth row of the matrix $\boldsymbol{A}$ can be simplified as

$$a_\omega = A_{k,\omega} = \sum_{j=1}^{M} D\left(x_j - x_B^k\right)D\left(x_j - x_B^\omega\right)\Delta h^n, \tag{17a}$$

$$\vdots$$

$$a_{\omega+q} = A_{k,\omega+q} = \sum_{j=1}^{M} D\left(x_j - x_B^k\right)D\left(x_j - x_B^{\omega+q}\right)\Delta h^n. \tag{17b}$$

Thus, Eq. (16) can be expressed as

$$a_\omega \delta \boldsymbol{u}_B^\omega \Delta s^\omega + a_{\omega+1} \delta \boldsymbol{u}_B^{\omega+1} \Delta s^{\omega+1} + \cdots + a_{\omega+q} \delta \boldsymbol{u}_B^{\omega+q} \Delta s^{\omega+q} = B_k. \tag{18}$$

Eq. (18) can be further simplified by invoking the relation $\delta \boldsymbol{u}_B^i \Delta s^i = \delta \boldsymbol{u}_B^j \Delta s^j + O(\Delta h^2)$ proposed by Zhao et al. [31]. Thus, Eq. (18) can be rewritten as

$$\delta \boldsymbol{u}_B^k \Delta s^k = \frac{B_k}{\sum_\omega^{\omega+q} a_n} \tag{19}$$

Once the velocity correction is obtained, the hydrodynamic forces exerted on the solid surface by the ambient fluid can be calculated as follow:

$$\boldsymbol{F}_H = (F_x, F_y, F_z) = -\sum_l \frac{\rho_f \delta \boldsymbol{u}_B^l \Delta s^l}{\Delta t} + \rho_f V_B \frac{d\boldsymbol{U}_B}{dt}, \tag{20}$$

where V_B and $\boldsymbol{U}_B$ denote the volume and the translational velocity of the rigid object, respectively.

3. Numerical results and validations

In this section, the numerical accuracy test is conducted at first. Subsequently, several validation cases, including a 2D flapping foil/undulatory foil self-propulsion in the absence of free stream, 3D streamwise rotating sphere and 2D filament/3D flag flapping in a Newtonian free stream, are used to validate the proposed methods.

3.1. Numerical accuracy test

Two dimensional Taylor Green vortex is used in this study to evaluate the order of accuracy of EIB-RLBFS. The analytical solutions of this benchmark case can be expressed as

$$u(x, y, t) = -u_0 \cos\left(\frac{\pi x}{L}\right) \sin\left(\frac{\pi y}{L}\right) e^{-\frac{2\pi^2 u_0 t}{\mathrm{Re} L}}, \tag{21a}$$

$$v(x, y, t) = u_0 \sin\left(\frac{\pi x}{L}\right) \cos\left(\frac{\pi y}{L}\right) e^{-\frac{2\pi^2 u_0 t}{\mathrm{Re} L}}, \tag{21b}$$

$$p(x, y, t) = p_0 - \frac{\rho_0 u_0^2}{4}\left[\cos\left(\frac{2\pi x}{L}\right) + \cos\left(\frac{2\pi y}{L}\right)\right] e^{-\frac{4\pi^2 u_0 t}{\mathrm{Re} L}}, \tag{21c}$$

where ρ_0, u_0 and p_0 denote the fluid density, the reference velocity and the reference pressure, respectively. The Reynolds number is defined as $Re = \frac{u_0 L}{\nu}$ and is set as $Re = 20$. The computational domain is set as [-L, L] × [-L, L], and periodic boundary condition is imposed on all boundaries. the numerical errors introduced by different mesh sizes and δt are analyzed to investigate the convergence orders of EIB-RLBFS. The widely employed parameters $\Delta t = 0.5$, $\Delta x = 1$, $c_s = \frac{1}{\sqrt{3}}$ and $\nu = 0.05$ are adopted. A circular cylinder of radius $0.5L$ is placed at the center of the computational domain. The boundary velocity of the immersed body is provided by the analytical values. The computational domain is discretized uniformly in space using 5 different sets of mesh, namely, 40^2, 60^2, 80^2, 100^2, and 120^2, and simulated at six different δt of $0.5\Delta t$, $0.6\Delta t$, $0.7\Delta t$, $0.8\Delta t$, $0.9\Delta t$, and Δt, yielding a total of 30 cases to investigate the order of accuracy of the proposed method. The initial condition at $t^* = \frac{u_0 t}{L} = 0$ is provided by the analytical solutions and the relative errors Eu_2 and Ev_2 are calculated at $t^* = 1$ through the following expressions:

$$Eu_2 = \sqrt{\frac{\sum_{i=1}^{N\times N}\left[\frac{(u_{\mathrm{numerical}} - u_{\mathrm{exact}})}{u_0}\right]^2}{N \times N}}, \tag{22a}$$

$$Ev_2 = \sqrt{\frac{\sum_{i=1}^{N\times N}\left[\frac{(v_{\mathrm{numerical}} - v_{\mathrm{exact}})}{u_0}\right]^2}{N \times N}}, \tag{22b}$$

where u and v represent the velocities in x and y directions, respectively. The subscripts, numerical and exact, found in u or v denote the numerical results and analytical solutions, respectively.

The relative errors are plotted against the mesh spacing at different δt in Fig. 2, showing that the EIB-RLBFS is of second order accurate in space. In addition, the results indicate that the EIB and the ratio of $\delta t/\Delta t$ have negligible effects on the global order of accuracy.

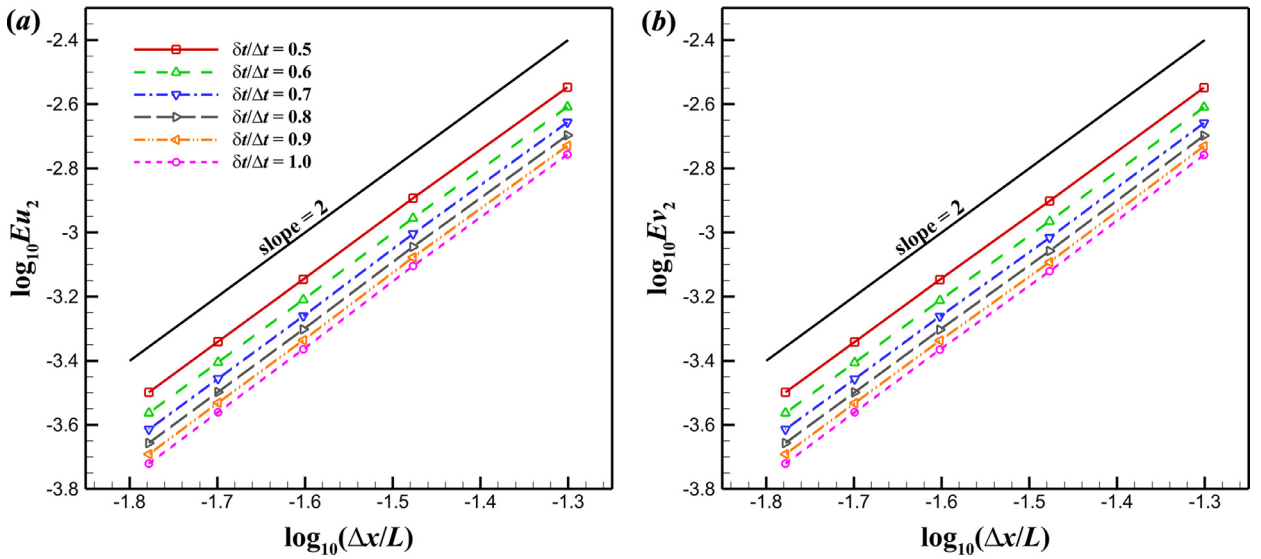

Fig. 2. Numerical errors of EIB-RLBFS at different mesh sizes and δt: (a) L_2 norm of error for u; (b) L_2 norm of error for v.

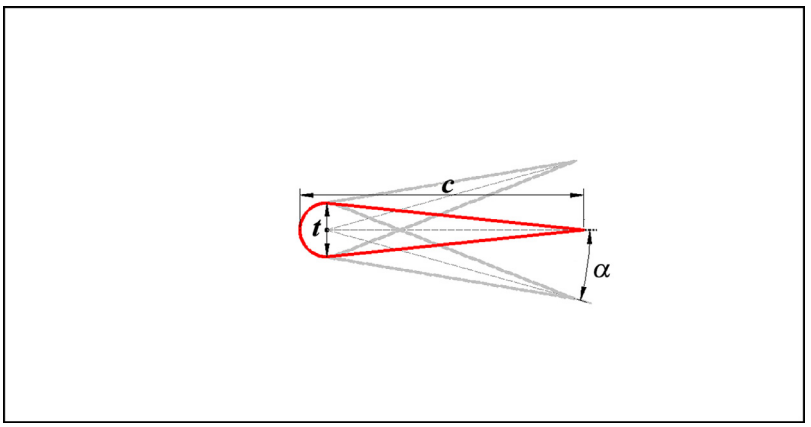

Fig. 3. Sketch view of an unconstrained self-propelled pitching foil, where c denotes the chord length, t is the thickness and α is the instantaneous pitching angle.

3.2. The 2D unconstrained flapping foil

The dynamic behavior of an unconstrained self-propelled flapping foil in the absence of free stream is considered here. The physical problem is described in Fig. 3, where the 2D unconstrained flapping foil pitches around the semicircular center with a simple harmonic pitching motion as follows:

$$\alpha(t) = \alpha_m \sin(2\pi f_p t), \tag{23}$$

where $\alpha(t)$ is the instantaneous pitching angle, α_m is the pitching amplitude, and f_p is the pitching frequency.

The motion of the 2D flapping foil is solved by the Euler scheme as follows

$$M_{foil}\frac{\boldsymbol{U}_{foil}^{n+1} - \boldsymbol{U}_{foil}^{n}}{\Delta t} = \boldsymbol{F}_{sum}^{n}, \tag{24}$$

where $\boldsymbol{F}_{sum}^{n}$ is the force acting on the foil and $\boldsymbol{U}_{foil}$ denotes the foil velocity. Thus, the position of the flapping foil is obtained by adopting a trapezoidal rule, written as

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^{n} + \frac{\Delta t}{2}\left(\boldsymbol{U}_{foil}^{n+1} + \boldsymbol{U}_{foil}^{n}\right). \tag{25}$$

The Neumann boundary condition ($\partial \mathbf{u}/\partial n = 0$) is implemented on four computational domain boundaries. A grid independence study is conducted to determine the appropriate grid spacing. Fig. 4 shows the hydrodynamic performance of the 2D self-propelled flapping foil at different grid spacings, including the horizontal and vertical propulsive speeds. It is

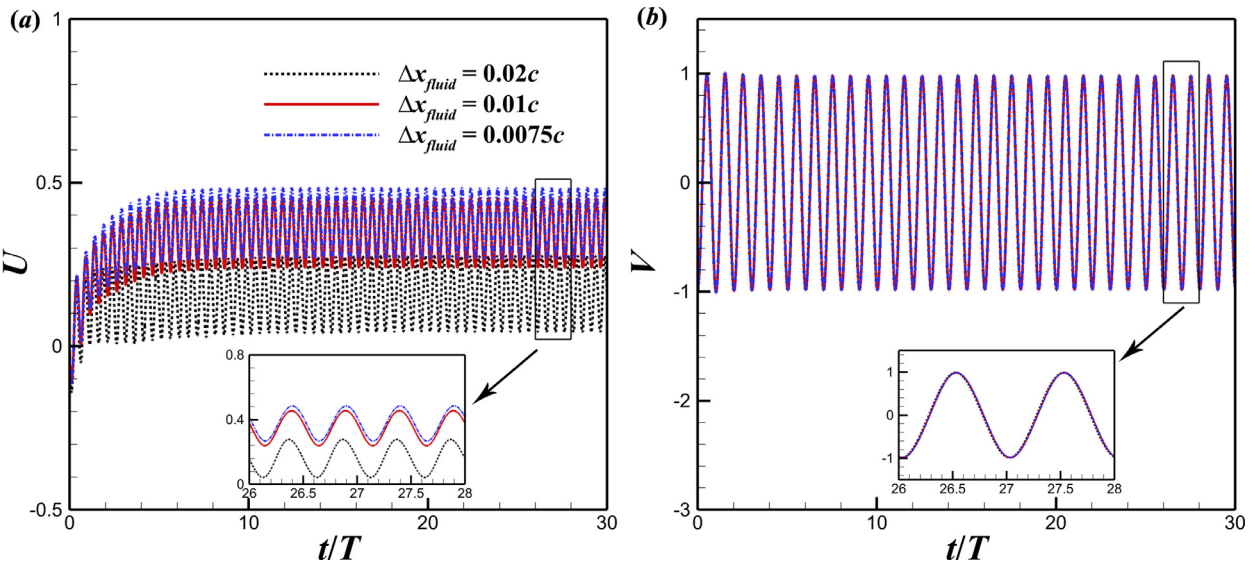

Fig. 4. The grid independence in the case of $Re = 200$, $f = 1.0$ and $\alpha_m = 20^o$. Time histories of (a) horizontal propulsive speed and (b) vertical propulsive speed of the self-propelled flapping foil.

evident from Fig. 4 that the spatial resolution of $\Delta x_{fluid} = 0.01c$ is sufficient to investigate the hydrodynamic performance of the swimmer with appropriate computational cost and accuracy, which is also consistent with the results of Lin et al. [8]. The computational domain is [-15c, 10c] × [-5c, 5c], where the region of [-14c, 2c] × [-2c, 2c] is discretized by the uniform grid with a mesh spacing of Δx_{fluid}=0.01c. The thickness-chord ratio $t/c = 0.1$ is applied and the flapping foil is discretized by uniform Lagrangian points with the spacing of Δx_{solid}=0.015c. Here, case 1: f_p=1.0, $\alpha_m = 20^o$ and case 2: f_p=2.0, $\alpha_m = 10^o$ are adopted to validate the present solver. The Reynolds number based on the reference velocity U_{ref} ($Re = \rho_f U_{ref} c/\mu$) is 200.

The hydrodynamic performance of the flapping foil is obtained when steady propulsion is achieved, and is used to compare with the reference data. As shown in Fig. 4, the self-propelled flapping foil reaches steady propulsive state after 20 cycles. It should be noted that the numerical test is implemented with using the OpenMP parallel strategy with 40 cores (2.4 GHz). For reference, in our numerical test for the case 1 on the mesh size of 1800 × 470, the computational time is around 16 min to reach 20 cycles. Fig. 5(a,b) show the time evolution of the propulsive speed along the horizontal and lateral directions in one pitching period of case 1 and case 2, respectively. The time histories of thrust force and lateral force in case 1 are shown in Fig. 5(c). The dynamic and hydrodynamic force responses obtained from this work agree very well with previous numerical results in Lin et al. [8]. Moreover, the present solver is able to capture accurately the vortical evolution and the wake street type, as observed in Fig. 5(d).

3.3. The 2D self-propelled undulatory foil

A self-propelled undulatory foil with deformable boundaries is considered in this section to test the capability of the present solver with a moving frame. In the present study, the shape of the undulatory foil is taken to be that of the NACA0012 foil. In order to mimic fish swimming motion, the movement of the undulating body in the y direction is given as:

$$y^*(x^*, t) = A(x^*)cos(2\pi\,(x^* - f_u t)), \tag{26a}$$

$$A(x^*) = 0.02 - 0.0825x^* + 0.1625(x^*)^2, \tag{26b}$$

where f_u is the undulatory frequency. x^* and y^* denote the local displacement in x and y direction of the backbone, respectively. Similarly, the locomotion of a carangiform swimmer can also be governed through the Euler scheme shown in Eq. (24) for unconstrained flapping foil.

In the absence of free stream, the undulatory swimmer propels itself through the hydrodynamic force generated by the ambient fluid. Here, two computational domains, namely, Domain 1 and 2 are set up to compare the performance of EIB-RLBFS and ALE-EIB-RLBFS solvers, respectively. The Reynolds number ($Re = \rho_f Uc/\mu$) of 5000 is adopted and the mass is set as $0.12\rho_f c^2$, where U denotes the reference velocity. Domain 1: the computational domain is [-12c, 4c] × [-3c, 3c], where the region of [-10c, 2c] × [-0.2c, 0.2c] is discretized by the uniform grid with a mesh spacing of Δx_{fluid}=0.003c. Domain 2: the computational domain is [-4c, 6c] × [-3c, 3c], where the region of [-0.5c, 2c] × [-0.2c, 0.2c] is discretized with the same

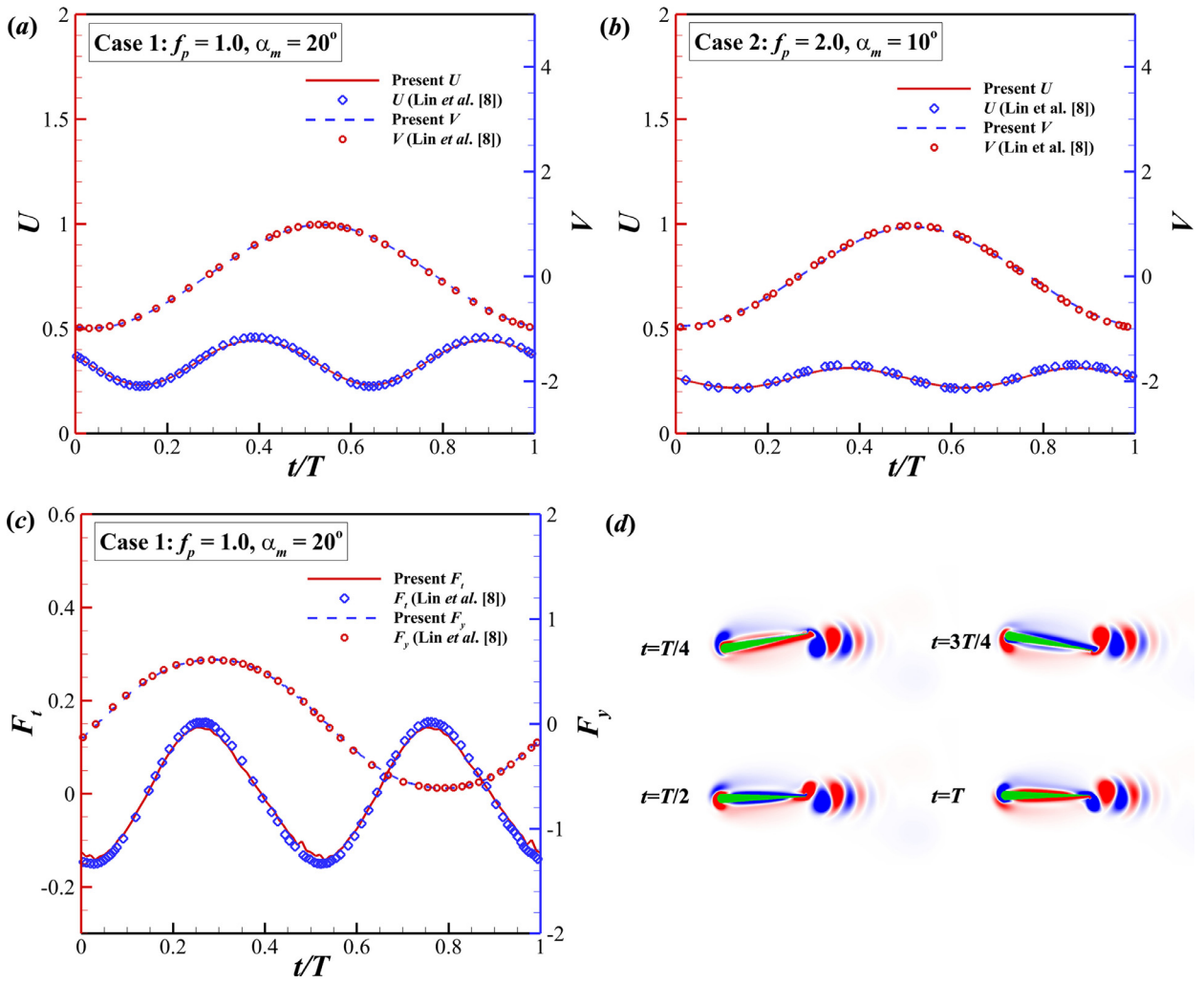

Fig. 5. Comparisons of time histories of (*a*) propulsive speeds in case 1, (*b*) propulsive speeds in case 2, and (*c*) thrust and lift force in case 1 of an isolated ping foil. (*d*) shows the vorticity contours of the flapping foil in case 2 over one flapping period. The results obtained by Lin et al. [8] are adopted as the reference data.

Table 1
Comparison of grid size with and without ALE method.

	Grid size	Grid ratio
EIB-RLBFS	4091 × 231	434.75%
ALE-EIB-RLBFS	941 × 231	1

mesh spacing of Δx_{fluid}=0.003c. These two domains for different approaches are shown in Fig. 7, which clearly indicates that ALE-EIB-RLBFS uses a relatively small domain to track moving solid objects. Table 1 tabulates the grid sizes of the two domains adopted for this test.

The instantaneous vorticity contours of the undulatory swimmer at different time instants are shown in Fig. 6. After a transitional stage, the dynamic balance of the thrust and drag force is achieved, inducing that the time-averaged horizontal speed is constant. Fig. 8(a) shows the evolution of the propulsive speed of undulatory foil computed with Domain 1, which indicates that although the initial results obtained by different numerical approaches have some discrepancy, the terminal time-averaged propulsive speed generated by EIB-RLBFS is close to that of previous studies [31,49] at steady propulsive stage. It can be seen that the propulsive speed obtained by the present solver is consistent with the terminal velocity -1.23 [49].

In the above simulations, the original EIB-RLBFS in fixed Eulerian coordinates requires a large computational domain to obtain the statistical results of propulsive performance. To efficiently simulate the hydrodynamic behavior of the undulatory swimmer, the ALE-EIB-RLBFS is applied to simulate the self-propelled undulatory swimmer in an infinitely large domain

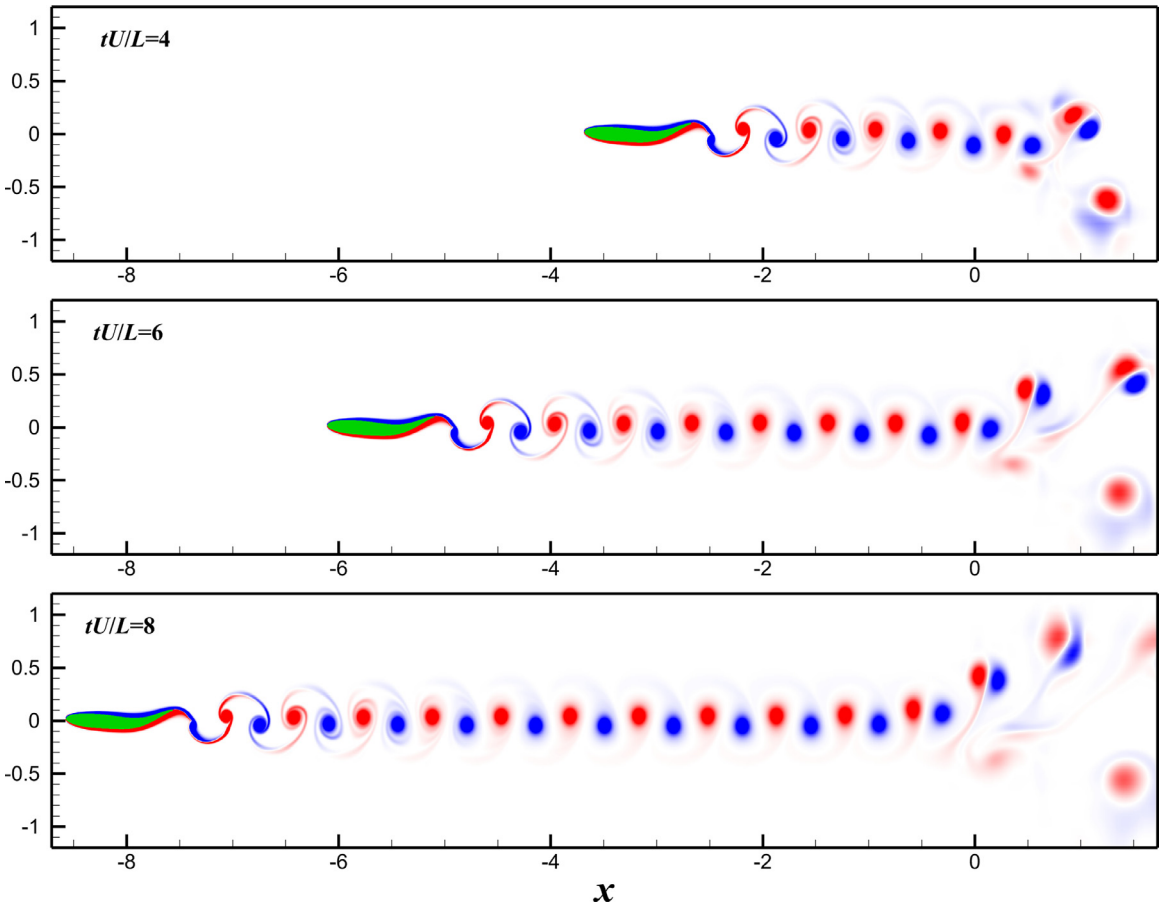

Fig. 6. The instantaneous vorticity contours of the undulatory swimmer at different time instants computed with Domain 1.

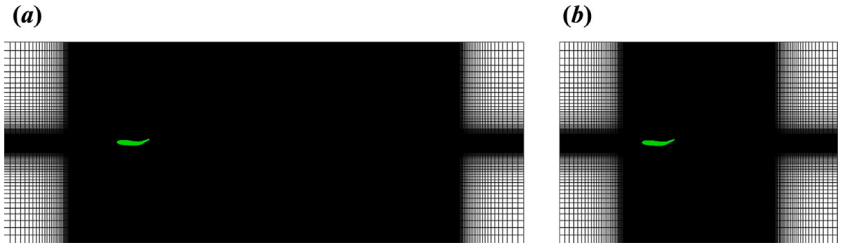

Fig. 7. The grids of (*a*) Domain 1 for EIB-RLBFS and (*b*) Domain 2 for ALE-EIB-RLBFS.

with much fewer grid points. Fig. 8(b) shows that the results obtained from the ALE-EIB-RLBFS in a moving frame with Domain 2 are identical to the original results generated by the EIB-RLBFS solver, indicating that the ALE-EIB-RLBFS is suitable for investigating the propulsive behavior in a small moving computational domain. Most importantly, Table 1 confirms that the ALE-EIB-RLBFS solver can track the freely moving object in a small moving domain with relatively fewer grid points.

3.4. The 2D flexible filament

A flexible filament in a free stream is simulated in this section to test the ability of the present solver to capture the dynamic behavior of a flexible structure accurately. The governing equations for a 2D flexible filament can be written in a Lagrangian form as [9]:

$$\rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} = \frac{\partial}{\partial s}\left(T\frac{\partial \mathbf{X}}{\partial s}\right) - \frac{\partial^2}{\partial s^2}\left(k_b \frac{\partial^2 \mathbf{X}}{\partial s^2}\right) + \mathbf{F}_H, \tag{27}$$

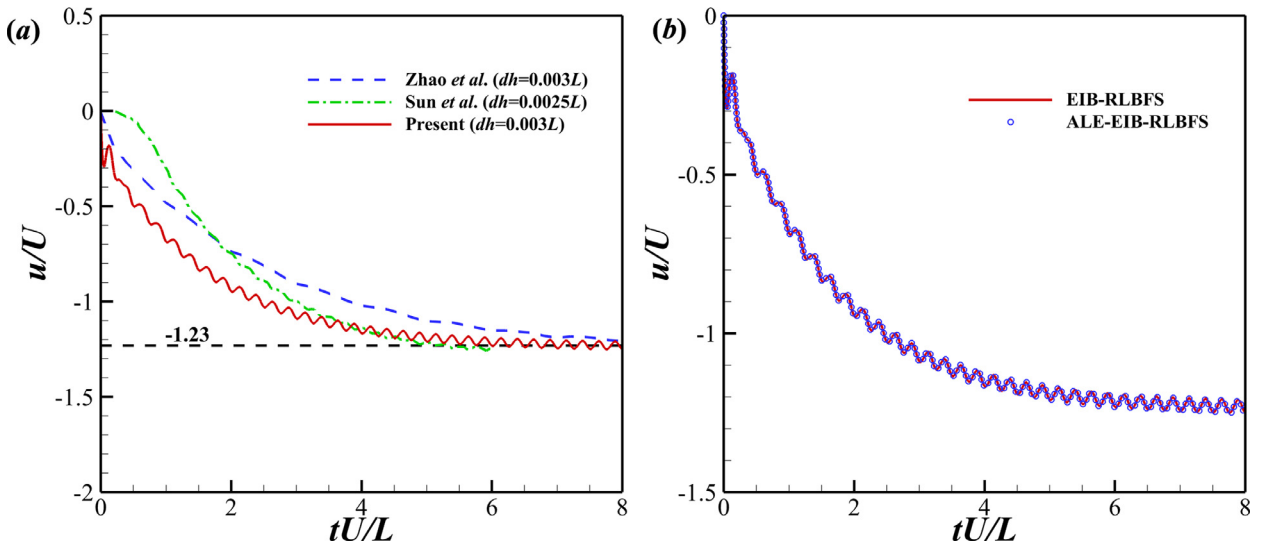

Fig. 8. Comparison of the time histories of (*a*) horizontal propulsive speeds and (*b*) horizontal propulsive speeds generated by EIB-RLBFS and ALE-EIB-RLBFS.

where ρ_s denotes the line density of the filament, $\boldsymbol{F}_H$ is the hydrodynamic force applied on the flexible filament by the ambient fluid, $\mathbf{X}$ is the position of the Lagrangian points on the filament, T is the tension force along the filament, k_b is the bending stiffness, and s denotes the arc length.

With the inextensibility condition ($\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial \mathbf{X}}{\partial s}=1$), the Poisson equation for the tension force T [9] can be written as

$$\begin{aligned}&\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial^2}{\partial s^2}\left(T\frac{\partial \mathbf{X}}{\partial s}\right)\\&\quad=\frac{1}{2}\frac{\partial^2}{\partial t^2}\left(\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial \mathbf{X}}{\partial s}\right)-\frac{\partial^2 \mathbf{X}}{\partial t\partial s}\cdot\frac{\partial^2 \mathbf{X}}{\partial t\partial s}-\frac{\partial \mathbf{X}}{\partial s}\cdot\frac{\partial}{\partial s}(\boldsymbol{F}_b+\boldsymbol{F}_H),\end{aligned}\tag{28}$$

where $\boldsymbol{F}_b=-\frac{\partial^2}{\partial s^2}\left(k_b\frac{\partial^2 \mathbf{X}}{\partial s^2}\right)$ is the bending force term. The corresponding discretized form of the above equations can be expressed as

$$(\boldsymbol{F}_b)_i=-\left[D_s^+D_s^-(k_bD_{ss}\mathbf{X}^*)\right]_i,\tag{29}$$

$$\begin{aligned}&\left(D_s^0\mathbf{X}^*\right)_{i+0.5}\bullet\left[D_s^0\left(D_s\left(T^{n+0.5}D_s\mathbf{X}^*\right)\right)\right]_{i+0.5}\\&\quad=\frac{1}{2}D_t^+D_t^-\left(D_s^0\mathbf{X}^n\cdot D_s^0\mathbf{X}^n\right)_{i+0.5}\\&\qquad-\left(D_s^0\boldsymbol{U}^n\cdot D_s^0\boldsymbol{U}^n\right)_{i+0.5}-\left(D_s^0\mathbf{X}^*\right)_{i+0.5}\cdot\left[D_s^0\left(\boldsymbol{F}_b^*+\boldsymbol{F}_H\right)\right]_{i+0.5},\end{aligned}\tag{30}$$

where

$$\begin{aligned}&\mathbf{X}^*=2\mathbf{X}^n-\mathbf{X}^{n-1},\\&\frac{1}{2}D_t^+D_t^-\left(D_s^0\mathbf{X}^n\cdot D_s^0\mathbf{X}^n\right)=\frac{1-2\left(D_s^0\mathbf{X}\cdot D_s^0\mathbf{X}\right)^n+\left(D_s^0\mathbf{X}\cdot D_s^0\mathbf{X}\right)^{n-1}}{2\Delta t^2}.\end{aligned}\tag{31}$$

D_s and D_{ss} (see Appendix A) denote the first order and seconder order derivatives of the arc length s, respectively.

Once the tension force T is obtained, the implicit second-order finite difference method is adopted to discretize the governing equation Eq. (27), which can be written as

$$\frac{\mathbf{X}^{n+1}-2\mathbf{X}^n+\mathbf{X}^{n-1}}{\Delta t^2}=\frac{\partial}{\partial s}\left(T^{n+\frac{1}{2}}\frac{\partial \mathbf{X}^{n+1}}{\partial s}\right)-\frac{\partial^2}{\partial s^2}\left(k_b\frac{\partial^2 \mathbf{X}^*}{\partial s^2}\right)+\mathbf{F}_H.\tag{32}$$

The new position $\mathbf{X}^{n+1}$ can be obtained by solving Eqn 32.

Fig. 9 shows the sketch view of a flexible filament flapping in a free stream and the leading edge of the flexible filament is fixed at the origin, where the boundary conditions are specified. The length of the filament is L and the leading edge is fixed. The computational domain is set as $[-20L, 40L]\times[-23.5L, 23.5L]$ to eliminate the boundary effects, where the region

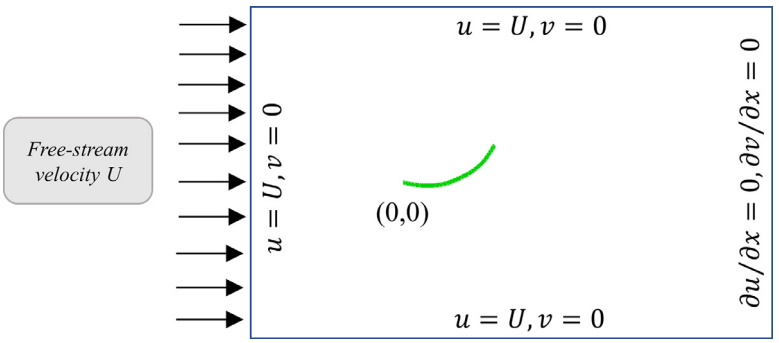

Fig. 9. Scheme of the computational setup for a flexible filament flapping in a free stream, where the green line denotes the flexible filament.

Table 2
Comparison of quantitative results at different mass densities.

	Mass ratio β	Amplitude A/L	Drag coefficient C_d
Present	0.3	0.235	0.412
Yuan et al. [11]	0.3	0.215	0.425
Present	0.6	0.347	0.736
Yuan et al.	0.6	0.360	0.732

of [-0.5L, 1.5L] × [-1.5L, 1.5L] is discretized with Δx_{fluid}=0.01L uniform grid. The filament is discretized uniformly with 75 Lagrangian points. Mass ratios ($\beta = \rho_s/\rho_f L$) of 0.3 and 0.6, Reynolds number ($Re = \rho_f U_\infty L/\mu$) of 165, and dimensionless bending stiffness k_b of 1×10^{-3} are chosen in this section.

Quantitative results (flapping amplitude A/L and drag force coefficient $F_d/(0.5\rho_f U_\infty^2 L)$) obtained from present solver are compared in Table 2 with numerical results obtained by Yuan et al. [11], showing that they are extremely close. Fig. 10(a,b) show the comparison of the predicted flapping profile of the filament predicted with the present solver and previous work performed by Yuan et al. [11] for $\beta = 0.3$ and 0.6, respectively. Basically, the trajectories of the free end of the flexible filament (see Fig. 10(a,b)) predicted by the proposed approach agree with that of Yuan et al. [11]. The small discrepancy between our results and that of Yuan et al. [11] may be induced by the error introduced by the constant filament stretching coefficient which is applied in the work of Yuan et al. [11], while the inextensibility condition is adopted in the present work. Huang et al. [9] demonstrated that a constant stretching coefficient induces a larger length error than that of the inextensibility condition. The instantaneous vorticity contours are shown in Fig. 10(c) at two instants in one flapping cycle, and the vortices are alternatively released at the trailing edge of filament.

3.5. 3D rotating sphere

To test the capability of the present solver to capture the dynamic response in 3D moving boundary FSI problems, a streamwise rotating sphere is considered in this section. The sphere rotates in the streamwise direction (z axis) with different rotating speeds. The two dimensionless parameters widely used for flow past a rotating sphere are the dimensionless rotation speed $\Omega = \omega D/2U_0$ and the Reynolds number $Re = \rho_f U_0 D/\mu$. The ω denotes the rotational speed, D is the diameter of the sphere, and U_0 is the free-stream flow velocity. Here, the Reynolds number is fixed at $Re = 300$.

Fig. 11 shows the instantaneous vortex structures visualized with the Q-criterion [52], showing a double-twisted vortex pair around the z axis in the wake region. Fig. 12 shows the time histories of the hydrodynamic forces exerted on the sphere surface obtained at different rotational speeds. When $\Omega = 0.1, 0.3,$ and 1.0, the unsteady asymmetric flow causes the hydrodynamic forces to change periodically with time. In contrast, at $\Omega = 0.5$, the unsteady "frozen" flow causes the drag and lift forces to remain constant. To further demonstrate the feasibility of present solver, comparison of the time-averaged drag and lift forces at $\Omega = 0.3$ and 0.5 with those in previous studies are conducted and tabulated in Table 3. The present results are in excellent agreement with results obtained using different approaches.

3.6. 3D flexible flag

In this section, the case of a 3D flapping flag in a free stream is adopted to examine the ability of present solver for simulating the dynamic behavior of 3D deformable body. The governing equations for a 3D rectangular flexible flag can be written in a curvilinear coordinate form as [10]:

$$\rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} = \sum_{i,j=1}^{2} \left[\frac{\partial}{\partial s_i} \left(\sigma_{ij} \frac{\partial \mathbf{X}}{\partial s_j} \right) - \frac{\partial^2}{\partial s_i \partial s_j} \left(k_{ij}^b \frac{\partial^2 \mathbf{X}}{\partial s_i \partial s_j} \right) \right] + \boldsymbol{F}_H, \tag{33}$$

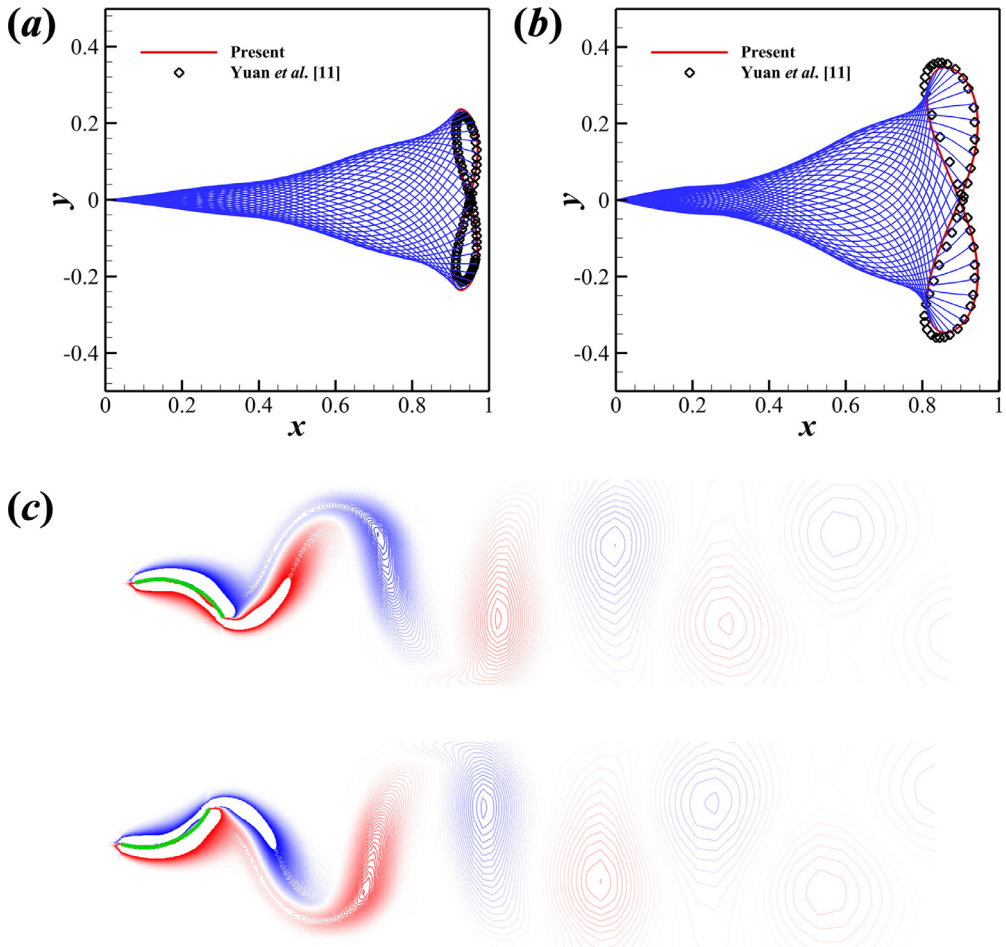

Fig. 10. In (*a*) and (*b*), blue solid lines present the trajectory of the filament flapping in the fluid and red bold line denotes the trajectory of the free end of the filament. (*a*) $\beta = 0.3$, (*b*) $\beta = 0.6$. (*c*) The instantaneous vorticity contours of the flapping filament with $\beta = 0.6$ at two time instants.

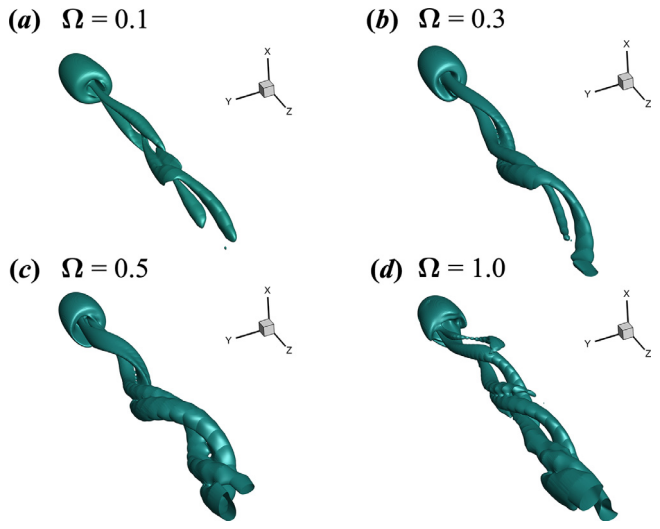

Fig. 11. The instantaneous vortical structures for flow past a streamwise rotating sphere at $Re = 300$ with various rotational speeds.

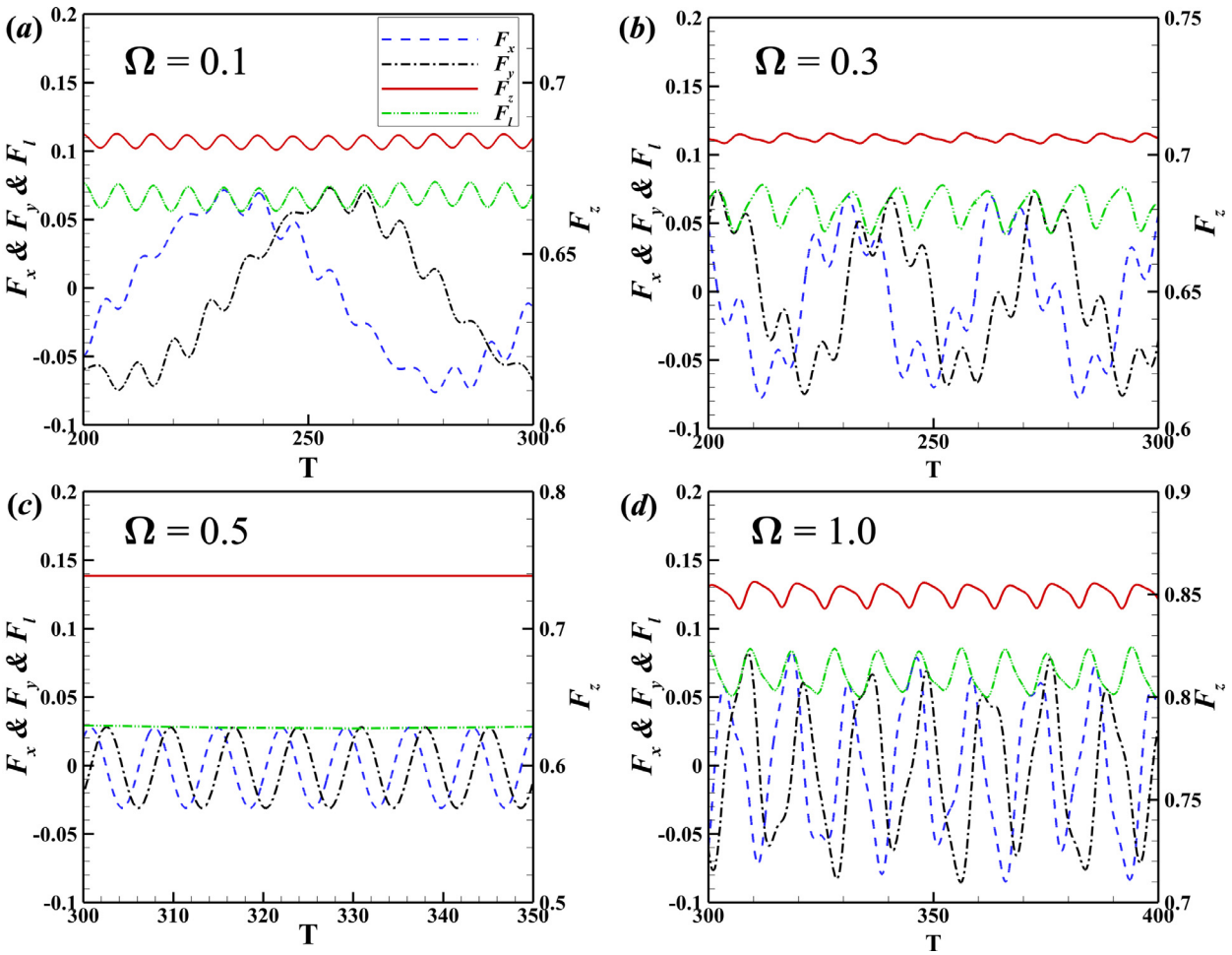

Fig. 12. The time histories of the hydrodynamic forces for flow past a streamwise rotating sphere at $Re = 300$ with various rotational speeds. F_x, F_y and F_z are the hydrodynamic force components along the x, y and z axis, respectively. F_z denotes the drag force and $F_l = \sqrt{F_x^2 + F_y^2}$ is the lift force.

Table 3
Comparison of time-averaged dimensionless force for flow past a streamwise rotating sphere at $Re = 300$ with $\Omega = 0.1$ and 0.5.

Ω	Refs.	$\overline{F_d}$	$\overline{F_l}$
0.1	Present	0.683 ± 0.002	0.067 ± 0.008
	Wu and Shu [50]	0.670 ± 0.002	0.068 ± 0.009
	Wang et al. [43]	0.683 ± 0.002	0.069 ± 0.008
	Yang et al. [51]	0.676 ± 0.002	0.067 ± 0.007
0.5	Present	0.738	0.0295
	Wu and Shu [50]	0.728	0.0325
	Wang et al. [43]	0.738	0.0275
	Yang et al. [51]	0.730	0.0275

where ρ_s is the area density of the 3D flexible flag, $\sigma_{ij} = \varphi_{ij}\left(\frac{\partial \mathbf{X}}{\partial s_i} \cdot \frac{\partial \mathbf{X}}{\partial s_j} - T_{ij}^0\right)$. The initial unstretched condition T_{ij}^0 is defined as

$$T_{ij}^0 = \begin{cases} 1, \text{if } i = j, \\ 0, \text{if } i \neq j. \end{cases} \tag{34}$$

φ_{ij} and k_{ij}^b represent either tension and bending coefficients or shearing and twisting coefficients depending on $i = j$ or $i \neq j$, respectively.

In the present work, following the previous work [15], the 3D flexible flag is regarded as a series of parallel flexible filaments along the streamwise direction, and these filaments are connected along the spanwise direction. A predicted position

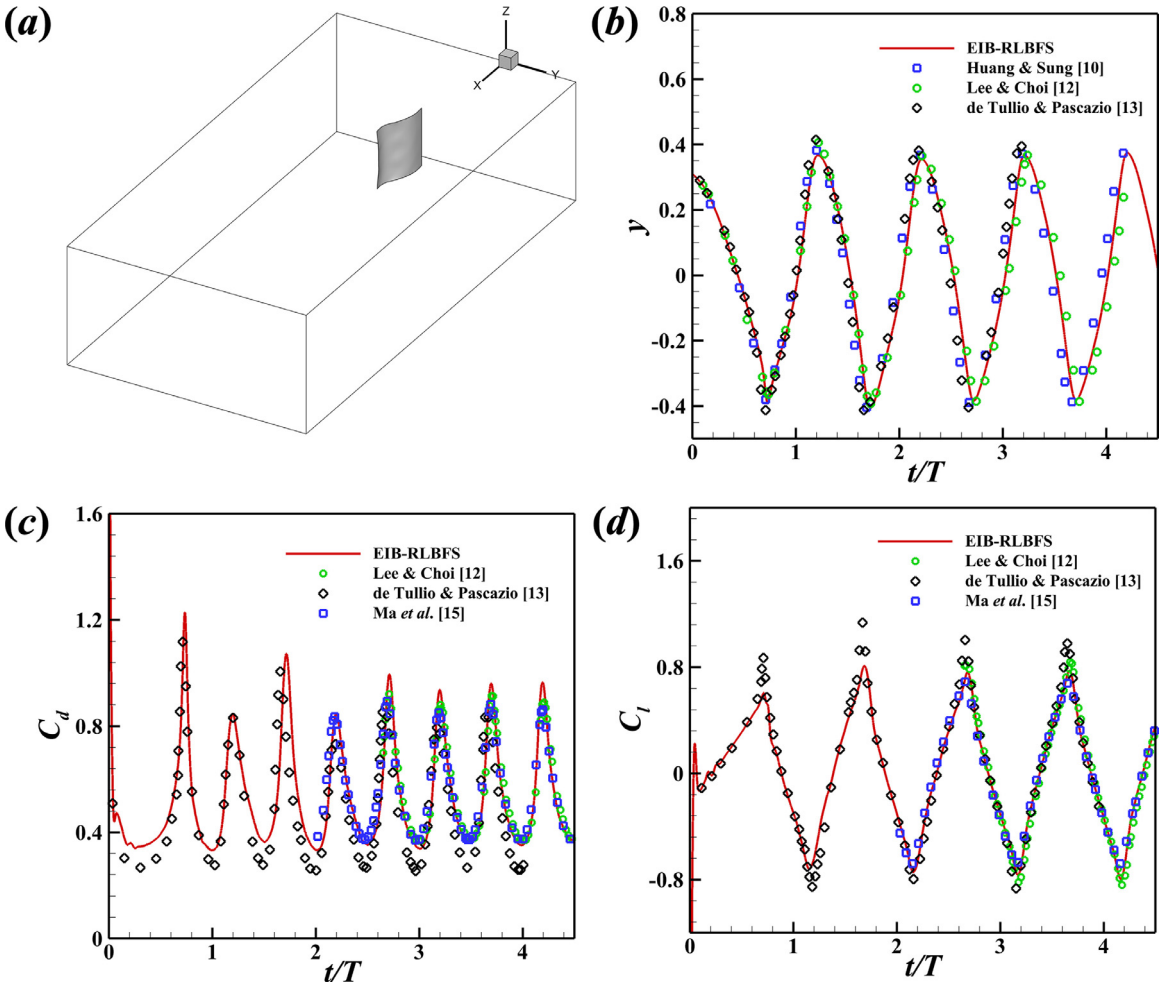

Fig. 13. (*a*) Sketch view of the physical problem. Time histories of (*b*) the transverse displacement of the midpoint on the trailing edge, (*c*) drag coefficient $C_d = 2F_d/\rho_f U^2 L^2$ and (*d*) lift coefficient $C_l = 2F_l/\rho_f U^2 L^2$.

$\mathbf{X}^* = 2\mathbf{X}^n - \mathbf{X}^{n-1}$ is introduced to simplify the process of updating the position. For the flexible filaments, when $i = j$=1, the bending force term $\frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^b \frac{\partial^2 \mathbf{X}}{\partial s_1^2}\right)$, and stretching force term $\frac{\partial}{\partial s_1}\left(\sigma_{11}\frac{\partial \mathbf{X}}{\partial s_1}\right) = \frac{\partial}{\partial s_1}\left(T\frac{\partial \mathbf{X}}{\partial s_1}\right)$ are the internal force terms; the other force terms are regarded as external force terms. Therefore, Eq. (33) can be simplified as

$$\rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} = \frac{\partial}{\partial s_1}\left(T\frac{\partial \mathbf{X}}{\partial s_1}\right) - \frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^b \frac{\partial^2 \mathbf{X}}{\partial s_1^2}\right) + \mathbf{F}_H + \boldsymbol{F}_{external}(\mathbf{X}^*), \tag{35}$$

where $\boldsymbol{F}_{external}(\mathbf{X}^*)$ is the summation force of the external force terms, which can be found in Appendix B. These parallel filaments satisfy the inextensibility condition along s_1 direction. Thus, the tensile stress T can be obtained by the Poisson equation, which is shown as follows:

$$\begin{aligned}
&\frac{\partial \mathbf{X}}{\partial s_1}\cdot\frac{\partial^2}{\partial s_1^2}\left(T\frac{\partial \mathbf{X}}{\partial s_1}\right)\\
&= \frac{1}{2}\frac{\partial^2}{\partial t^2}\left(\frac{\partial \mathbf{X}}{\partial s_1}\cdot\frac{\partial \mathbf{X}}{\partial s_1}\right)\\
&\quad - \frac{\partial^2 \mathbf{X}}{\partial t \partial s_1}\cdot\frac{\partial^2 \mathbf{X}}{\partial t \partial s_1} - \frac{\partial \mathbf{X}}{\partial s_1}\cdot\frac{\partial}{\partial s_1}\left[\frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^b\frac{\partial^2 \mathbf{X}}{\partial s_1^2}\right) + \mathbf{F}_H + \mathbf{F}_{external}\right].
\end{aligned} \tag{36}$$

Here, the same method is adopted to solve the Poisson equation as in Section 3.4. After the tension force T is obtained, Eq. (35) can be discretized by the implicit second-order finite-difference scheme. The discretized form of Eq. (35) can be

Table 4
Comparison of grid size with different numerical methods.

Numerical method	Grid size	Grid ratio
IB-NS [12]	353 × 353 × 193	425.50%
IB-LBM [15]	512 × 256 × 128	296.84%
EIB-RLBFS	314 × 150 × 120	1

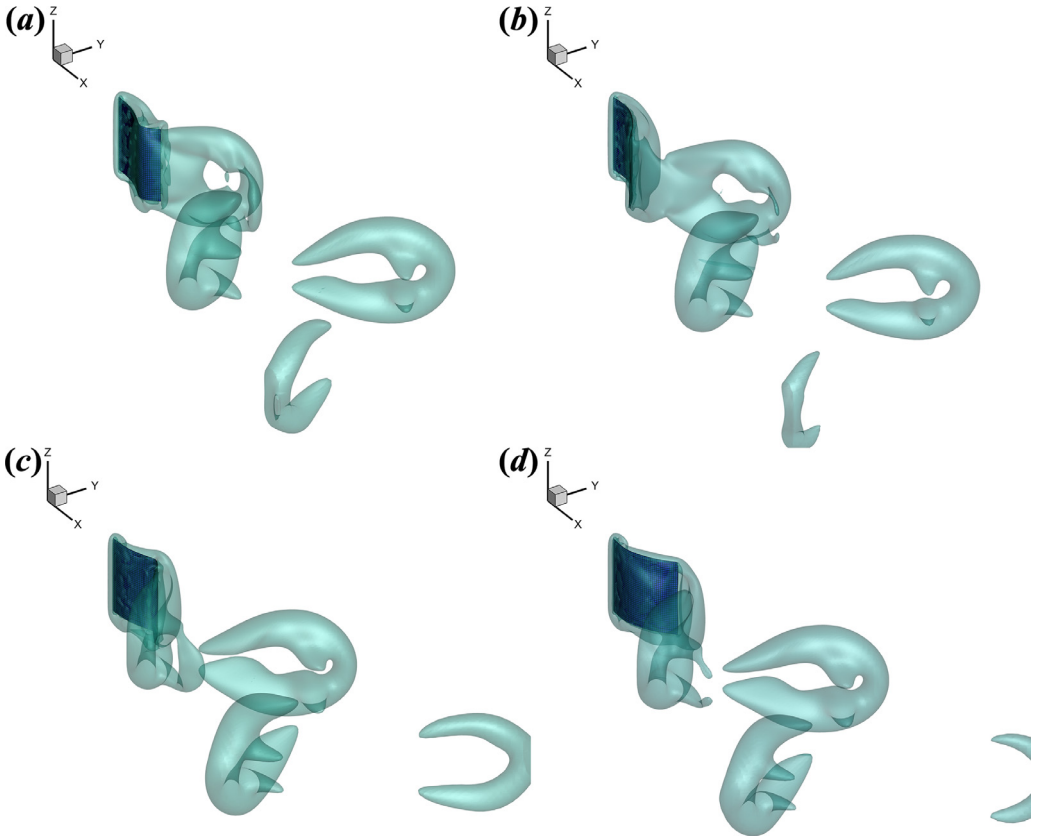

Fig. 14. The vortical structures around the 3D flag: (*a*) $t = 87$; (*b*) $t = 87.6$; (*c*) $t = 88.8$; (*d*) $t = 89.4$. The vortical structures are visualized by the iso-surface of Q = 0.2.

written as

$$\frac{\mathbf{X}^{n+1} - 2\mathbf{X}^{n} + \mathbf{X}^{n-1}}{\Delta t^2} = \frac{\partial}{\partial s_1}\left(T^{n+\frac{1}{2}}\frac{\partial \mathbf{X}^{n+1}}{\partial s_1}\right) - \frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^{b}\frac{\partial^2 \mathbf{X}^{*}}{\partial s_1^2}\right) + \mathbf{F}_{H} + \mathbf{F}_{external}(\mathbf{X}^{*}). \tag{37}$$

The updated position can be obtained by solving Eq. (37).

The flexible square flag has length of L and an inclination of $\theta_0 = 0.1\pi$ from the xz plane to accelerate the oscillation motion, which is a flat plate at the initial instant with the midpoint of the leading edge fixing at the origin. It should be noted that the leading edge is pinned and the free moving boundary condition is implemented at the other three edges. The overview of the problem is illustrated in Fig. 13(a), where the computational domain is [-L, 7L] × [-2L, 2L] × [-L, L] along x (streamwise direction), y and z axis, respectively. The computational domain is discretized by a non-uniform grid of 314 × 150 × 120, where a uniformly discretized fine mesh of $\Delta x_{fluid} = L/75$ is applied at [-0.2L, 1.3L] × [-0.5L, 0.5L] × [-0.6L, 0.6L], encapsulating the flag and the wake region. Due to the robustness of the present solver, the mesh outside the fine region can be stretched without affecting the solver's accuracy, resulting in a relatively smaller grid size than other numerical methods (see Table 4). The flag is discretized uniformly with Lagrangian points ($\Delta x_{solid} = L/50$) in both streamwise and lateral directions. Dirichlet boundary conditions ($u = U$, $v = w = 0$) are applied at the inlet boundary and the Neumann

Table 5
Comparison of quantitative results, including the peak-to-peak oscillation amplitude A/L and the Strouhal number $St = fL/U$, f being the oscillation frequency.

	Amplitude A/L	Strouhal number St
Present	0.754	0.266
Lee & Choi [12]	0.752	0.265
de Tullio & Pascazio [13]	0.795	0.265
Tian et al. - Flag 2 [53]	0.806	0.266
Huang & Sung [10]	0.780	0.260

boundary condition ($\partial \mathbf{u}/\partial n = 0$) is implemented on the remaining boundaries. The parameters: mass ratio $\beta = \rho_s/\rho_f L = 1$, $\varphi_{11} = \varphi_{22} = 10^3$, $\varphi_{12} = 10$, $k^b_{11} = k^b_{22} = k^b_{12} = 10^{-4}$, $Re = \rho_f UL/\mu = 200$, are adopted.

The time histories of the transverse displacement, drag coefficient, and the lift coefficient evaluated at the flag's trailing center point are shown in Fig. 13(b), 13(c), 13(d), respectively. Overall, our results are in good agreement with the results obtained from previous studies [10,12,13,15]. However, when the flag moves through the origin with a significant drag force, small discrepancies between the results generated by different numerical methods are observed. These differences are induced by the errors in predicting the friction force with different IBMs. Ma et al. [15] indicated that first-order IBMs overpredict friction. Since the EIB introduced in the present work is a first-order IBM, this explains why a higher drag is predicted by the present solver when the flag passes through the origin. Quantitative comparisons of peak to peak oscillation amplitude and Strouhal number are tabulated in Table 5, and good agreement between present and previous studies [10,12,13,53] are observed. Hence, the present solver not only dramatically reduces the computational cost, but also attains the required accuracy. Finally, Fig. 14 shows the instantaneous vortical structures visualized with the Q-criterion [52] at Q = 0.2, where a hairpin-like structure is accurately captured in the wake region [10].

4. Conclusion

An explicit immersed boundary-reconstructed lattice Boltzmann flux solver (EIB-RLBFS) is proposed in the present work for a wide range of fluid-structure interaction problems [8,10,11,40,54]. The present solver exploits the fractional step method to decouple the fluid field solver into predictor and corrector steps. Briefly, in the predictor step, the RLBFS predicts the intermediate flow field without the inclusion of any solid boundaries. Subsequently, in the corrector step, the explicit IBM is employed to implement the no-slip boundary conditions around the solid surface. Note that RLBFS not only retains the advantages of LBFS, but also further simplifies the flux evaluation method at the cell interface by using the macroscopic variables rather than reconstructing the solution of the LBE. On the other hand, the explicit boundary condition-enforced IBM accelerates the velocity correction step by avoiding the assembling and inversion of large matrices, with the Dirichlet boundary conditions accurately satisfied on the surface of the solid domain. Hence, the EIB-RLBFS proposed in this study is a highly efficient FSI solver that inherits the advantages mentioned above.

The EIB-RLBFS is first validated by simulating the hydrodynamic performance of a unconstrained flapping foil/undulatory foil. The present solver is not only able to accurately capture the passive oscillation of the solid body, but also the hydrodynamic force exerted on the solid surface by the ambient fluid. In addition, the present solver is tested for a 3D moving boundary problem, where the present results are in excellent agreement with previous results. Furthermore, this study demonstrates the feasibility of coupling EIB-RLBFS with ALE to further reduce the required number of grid points without sacrificing any accuracy for predicting moving structure-fluid interaction in a moving frame. For a slender elastic body (2D filament and 3D flag), the EIB-RLBFS is coupled with a finite difference method to simulate the flexible structure-fluid interaction. Excellent agreement with other numerical approaches is also observed for the slender elastic body problems. Therefore, the present study demonstrates that the proposed method is very suited for FSI problems with large deformations and moving boundaries. It has been shown to be more efficient than other numerical methods while maintaining similar accuracy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Buchen Wu: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Jinhua Lu:** Methodology, Writing – original draft. **HsuChew Lee:** Writing – original draft. **Chang Shu:** Formal analysis, Supervision, Writing – original draft, Writing – review & editing. **Minping Wan:** Formal analysis, Supervision, Writing – original draft, Writing – review & editing.

Acknowledgments

This work has been supported by the Key-Area Research and Development Program of Guangdong Province (Grant No.2021B0101190003), the Department of Science and Technology of Guangdong Province (Grant No. 2020B1212030001), and the Shenzhen Science & Technology Program (Grant No. KQTD20180411143441009). Numerical simulations have been supported by the Center for Computational Science and Engineering of Southern University of Science and Technology. Buchen Wu would like to acknowledge and thank Dr. Fangbao Tian for his kind assistance.

For 2D flexible filament, the first order and seconder order derivatives of the arc length s in Eqs. (29) and (30) are defined as

$$\begin{aligned} D_s^0\phi &= (\phi(s+\Delta s/2) - \phi(s-\Delta s/2))/\Delta s, \\ D_s^+\phi &= (\phi(s+\Delta s) - \phi(s))/\Delta s, \\ D_s^-\phi &= (\phi(s) - \phi(s-\Delta s))/\Delta s, \\ D_{ss}\phi &= D_s^+ D_s^-\phi = (\phi(s+\Delta s) - 2\phi(s) + \phi(s-\Delta s))/\Delta s^2. \end{aligned} \tag{A.1}$$

The external force terms in 3D flexible flag governing equation Eq. (35) can be obtained by [10] for $i = j$=2,

$$\frac{\partial}{\partial s_2}\left(\sigma_{22}\frac{\partial \mathbf{X}^*}{\partial s_2}\right) = D_2^-\left(\sigma_{22}D_2^+\mathbf{X}^*\right), \tag{B.1}$$

$$\frac{\partial^2}{\partial s_2^2}\left(\kappa_{22}^b\frac{\partial^2 \mathbf{X}^*}{\partial s_2^2}\right) = D_{22}^0\left(\kappa_{22}^b D_{22}^0\mathbf{X}^*\right); \tag{B.2}$$

for i=1,j=2 or i=2,j=1,

$$\frac{\partial}{\partial s_1}\left(\sigma_{12}\frac{\partial \mathbf{X}^*}{\partial s_2}\right) = \frac{1}{2}\left[D_1^-\left(\sigma_{12}^+D_2^+\mathbf{X}^*\right) + D_1^-\left(\sigma_{12}^-D_2^-\mathbf{X}^*\right)\right], \tag{B.3}$$

$$\frac{\partial}{\partial s_2}\left(\sigma_{21}\frac{\partial \mathbf{X}^*}{\partial s_1}\right) = \frac{1}{2}\left[D_2^-\left(\sigma_{21}^+D_1^+\mathbf{X}^*\right) + D_2^+\left(\sigma_{21}^-D_1^+\mathbf{X}^*\right)\right], \tag{B.4}$$

$$\frac{\partial^2}{\partial s_1 s_2}\left(\kappa_{12}^b\frac{\partial^2 \mathbf{X}^*}{\partial s_1 s_2}\right) = \frac{\partial^2}{\partial s_2 s_1}\left(\kappa_{21}^b\frac{\partial^2 \mathbf{X}^*}{\partial s_2 s_1}\right) = D_{12}^-\left(\kappa_{12}^b D_{12}^+\mathbf{X}^*\right). \tag{B.5}$$

In the above equations, based on the arc length s_1 and s_2, the downwind difference $\left[D_1^+\phi\right]_{i,j}$ and upwind difference $\left[D_1^-\phi\right]_{i,j}$, the second-order central difference $\left[D_{22}^0\phi\right]_{i,j}$, and the second-order cross difference $\left[D_{12}^+\phi\right]_{i,j}$, $\left[D_{12}^-\phi\right]_{i,j}$ are defined as

$$\begin{aligned} \left[D_1^+\phi\right]_{i,j} &= \left(\phi_{i+1,j} - \phi_{i,j}\right)/\Delta s_1, \\ \left[D_1^-\phi\right]_{i,j} &= \left(\phi_{i,j} - \phi_{i-1,j}\right)/\Delta s_1, \\ \left[D_{22}^0\phi\right]_{i,j} &= \left(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}\right)/\Delta s_2^2, \\ \left[D_{12}^+\phi\right]_{i,j} &= \left(\phi_{i+1,j+1} - \phi_{i+1,j} - \phi_{i,j+1} + \phi_{i,j}\right)/(\Delta s_1 \Delta s_2), \\ \left[D_{12}^-\phi\right]_{i,j} &= \left(\phi_{i,j} - \phi_{i,j-1} - \phi_{i-1,j} + \phi_{i-1,j-1}\right)/(\Delta s_1 \Delta s_2). \end{aligned} \tag{B.6}$$

References

[1] M. Gazzola, M. Argentina, L. Mahadevan, Scaling macroscopic aquatic locomotion, Nat. Phys. 10 (10) (2014) 758–761.
[2] M.S. Triantafyllou, G. Triantafyllou, D. Yue, Hydrodynamics of fishlike swimming, Annu. Rev. Fluid Mech. 32 (1) (2000) 33–53.
[3] I. Akhtar, R. Mittal, G.V. Lauder, E. Drucker, Hydrodynamics of a biologically inspired tandem flapping foil configuration, Theor. Comput. Fluid Dyn. 21 (3) (2007) 155–170.
[4] G.V. Lauder, Fish locomotion: recent advances and new directions, Ann. Rev. Mar. Sci. 7 (2015) 521–545.
[5] G. Liu, Y. Ren, H. Dong, O. Akanyeti, J.C. Liao, G.V. Lauder, Computational analysis of vortex dynamics and performance enhancement due to body–fin and fin–fin interactions in fish-like locomotion, J. Fluid Mech. 829 (2017) 65–88.
[6] M.S.U. Khalid, J. Wang, I. Akhtar, H. Dong, M. Liu, A. Hemmati, Why do anguilliform swimmers perform undulation with wavelengths shorter than their bodylengths? Phys. Fluids 33 (3) (2021) 031911.
[7] M.S.U. Khalid, J. Wang, H. Dong, M. Liu, Flow transitions and mapping for undulating swimmers, Phys. Rev. Fluids 5 (6) (2020) 063104.
[8] X. Lin, J. Wu, T. Zhang, Self-directed propulsion of an unconstrained flapping swimmer at low reynolds number: hydrodynamic behaviour and scaling laws, J. Fluid Mech. 907 (2021).
[9] W.-X. Huang, S.J. Shin, H.J. Sung, Simulation of flexible filaments in a uniform flow by the immersed boundary method, J. Comput. Phys. 226 (2) (2007) 2206–2228.
[10] W.-X. Huang, H.J. Sung, Three-dimensional simulation of a flapping flag in a uniform flow, J. Fluid Mech. 653 (2010) 301.
[11] H.-Z. Yuan, X.-D. Niu, S. Shu, M. Li, H. Yamaguchi, A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating a flexible filament in an incompressible flow, Comput. Math. Appl. 67 (5) (2014) 1039–1056.
[12] I. Lee, H. Choi, A discrete-forcing immersed boundary method for the fluid–structure interaction of an elastic slender body, J. Comput. Phys. 280 (2015) 529–546.

[13] M.D. de Tullio, G. Pascazio, A moving-least-squares immersed boundary method for simulating the fluid–structure interaction of elastic bodies with arbitrary thickness, J. Comput. Phys. 325 (2016) 201–225.
[14] F.-B. Tian, H. Luo, L. Zhu, J.C. Liao, X.-Y. Lu, An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments, J. Comput. Phys. 230 (19) (2011) 7266–7283.
[15] J. Ma, Z. Wang, J. Young, J.C. Lai, Y. Sui, F.-B. Tian, An immersed boundary-lattice Boltzmann method for fluid-structure interaction problems involving viscoelastic fluids and complex geometries, J. Comput. Phys. 415 (2020) 109487.
[16] A. Gilmanov, F. Sotiropoulos, A hybrid cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies, J. Comput. Phys. 207 (2) (2005) 457–492.
[17] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[18] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.
[19] Z.-G. Feng, E.E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems, J. Comput. Phys. 195 (2) (2004) 602–628.
[20] Z.-G. Feng, E.E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (1) (2005) 20–51.
[21] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (2) (1993) 354–366.
[22] L. Zhu, G. He, S. Wang, L. Miller, X. Zhang, Q. You, S. Fang, An immersed boundary method based on the lattice Boltzmann approach in three dimensions, with application, Comput. Math. Appl. 61 (12) (2011) 3506–3518.
[23] A. De Rosis, S. Ubertini, F. Ubertini, A partitioned approach for two-dimensional fluid–structure interaction problems by a coupled lattice Boltzmann-finite element method with immersed boundary, J. Fluids Struct. 45 (2014) 202–215.
[24] A.E. Giannenas, S. Laizet, A simple and scalable immersed boundary method for high-fidelity simulations of fixed and moving objects on a cartesian mesh, Appl. Math. Model. 99 (2021) 606–627.
[25] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.
[26] A. De Rosis, S. Ubertini, F. Ubertini, A comparison between the interpolated bounce-back scheme and the immersed boundary method to treat solid boundary conditions for laminar flows in the lattice Boltzmann framework, J. Sci. Comput. 61 (3) (2014) 477–489.
[27] K. Suzuki, T. Inamuro, Effect of internal mass in the simulation of a moving body by the immersed boundary method, Comput. Fluids 49 (1) (2011) 173–187.
[28] X. Niu, C. Shu, Y. Chew, Y. Peng, A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows, Phys. Lett. A 354 (3) (2006) 173–182.
[29] Y. Chen, Q. Cai, Z. Xia, M. Wang, S. Chen, Momentum-exchange method in lattice Boltzmann simulations of particle-fluid interactions, Phys. Rev. E 88 (1) (2013) 013303.
[30] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (6) (2009) 1963–1979.
[31] X. Zhao, Z. Chen, L. Yang, N. Liu, C. Shu, Efficient boundary condition-enforced immersed boundary method for incompressible flows with moving boundaries, J. Comput. Phys. (2021) 110425.
[32] Z. Guo, C. Shu, Lattice Boltzmann Method and its Application in Engineering, 3, World Scientific, 2013.
[33] Y.-H. Qian, D. d'Humières, P. Lallemand, Lattice BGK models for Navier-Stokes equation, EPL (Europhys. Lett.) 17 (6) (1992) 479.
[34] H. Lai, C. Ma, Numerical study of the nonlinear combined sine-cosine-gordon equation with the lattice Boltzmann method, J. Sci. Comput. 53 (3) (2012) 569–585.
[35] J. Wu, C. Shu, Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme, Commun. Comput. Phys. 7 (4) (2010) 793.
[36] P. Wang, L. Zhu, Z. Guo, K. Xu, A comparative study of LBE and dugks methods for nearly incompressible flows, Commun. Comput. Phys. 17 (3) (2015) 657–681.
[37] Z. Guo, H. Liu, L.-S. Luo, K. Xu, A comparative study of the LBE and GKS methods for 2d near incompressible laminar flows, J. Comput. Phys. 227 (10) (2008) 4955–4976.
[38] Y. Sui, Y.-T. Chew, P. Roy, H.-T. Low, A hybrid immersed-boundary and multi-block lattice Boltzmann method for simulating fluid and moving-boundaries interactions, Int. J. Numer. Methods Fluids 53 (11) (2007) 1727–1754.
[39] A. Dupuis, P. Chatelain, P. Koumoutsakos, An immersed boundary–lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder, J. Comput. Phys. 227 (9) (2008) 4486–4498.
[40] J. Wu, C. Shu, An improved immersed boundary-lattice Boltzmann method for simulating three-dimensional incompressible flows, J. Comput. Phys. 229 (13) (2010) 5022–5042.
[41] C. Shu, Y. Wang, C.J. Teo, J. Wu, Development of lattice Boltzmann flux solver for simulation of incompressible flows, Adv. Appl. Math. Mech. 6 (4) (2014) 436–460.
[42] Y. Wang, C. Shu, C. Teo, J. Wu, An immersed boundary-lattice Boltzmann flux solver and its applications to fluid–structure interaction problems, J. Fluids Struct. 54 (2015) 440–465.
[43] Y. Wang, C. Shu, C. Teo, L. Yang, An efficient immersed boundary-lattice Boltzmann flux solver for simulation of 3d incompressible flows with complex geometry, Comput. Fluids 124 (2016) 54–66.
[44] Y. Wang, C. Shu, L. Yang, Y. Sun, On the immersed boundary-lattice Boltzmann simulations of incompressible flows with freely moving objects, Int. J. Numer. Methods Fluids 83 (4) (2017) 331–350.
[45] Y. Wang, C. Shu, L. Yang, C. Teo, An immersed boundary-lattice Boltzmann flux solver in a moving frame to study three-dimensional freely falling rigid bodies, J. Fluids Struct. 68 (2017) 444–465.
[46] J. Lu, H. Lei, C. Dai, L. Yang, C. Shu, Analyses and reconstruction of the lattice Boltzmann flux solver, J. Comput. Phys. (2022) 110923.
[47] A. Andersen, U. Pesavento, Z.J. Wang, Unsteady aerodynamics of fluttering and tumbling plates, J. Fluid Mech. 541 (2005) 65–90.
[48] J. Lu, H. Lei, C. Shu, C. Dai, The more actual macroscopic equations recovered from lattice Boltzmann equation and their applications, J. Comput. Phys. 415 (2020) 109546.
[49] P.-N. Sun, A. Colagrossi, A.-M. Zhang, Numerical simulation of the self-propulsive motion of a fishlike swimming foil using the δ+-sph model, Theor. Appl. Mech. Lett. 8 (2) (2018) 115–125.
[50] J. Wu, C. Shu, Simulation of three-dimensional flows over moving objects by an improved immersed boundary–lattice Boltzmann method, Int. J. Numer. Methods Fluids 68 (8) (2012) 977–1004.
[51] L. Yang, C. Shu, W. Yang, Y. Wang, J. Wu, An immersed boundary-simplified sphere function-based gas kinetic scheme for simulation of 3d incompressible flows, Phys. Fluids 29 (8) (2017) 083605.
[52] J.C. Hunt, A.A. Wray, P. Moin, Eddies, streams, and convergence zones in turbulent flows, in: Studying Turbulence Using Numerical Simulation Databases, 2. Proceedings of the 1988 Summer Program, 1988.
[53] F.-B. Tian, H. Dai, H. Luo, J.F. Doyle, B. Rousseau, Fluid–structure interaction involving large deformations: 3d simulations and applications to biological systems, J. Comput. Phys. 258 (2014) 451–469.
[54] L. Tian, Z. Zhao, W. Wang, N. Liu, Length and stiffness effects of the attached flexible plate on the flow over a traveling wavy foil, Acta Mech. Sin. (2021) 1–12.