Journal of Computational Physics 521 (2025) 113524

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Simplified inverse distance weighting-immersed boundary method for simulation of fluid-structure interaction

Buchen Wu [a,1], Yinjie Du [a,b,1], Chang Shu [a,*]

[a] Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore, 119260, Singapore
[b] Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China

ARTICLE INFO

Keywords:
Immersed boundary method
Inverse distance weighting
Non-uniform mesh
Second-order approximation

ABSTRACT

When simulating fluid-structure interaction (FSI) problems involving moving objects, the implicit inverse distance weighting-immersed boundary method (IDW-IBM) developed by Du et al. [1] has to construct a large square correlation matrix and solve its inversion at each time step. In this work, a simplified inverse distance weighting-immersed boundary method (SIDW-IBM) is proposed to eliminate the intrinsic limitations in the original implicit IDW-IBM. Through error analysis using Taylor series expansion, a second order approximation can be derived, which allows us to approximate the large square correlation matrix into a diagonal matrix; thereby, we proposed the SIDW-IBM based on this second order approximation to explicitly evaluate the velocity corrections, where the needs to assemble the large correlation matrix and inverse it are circumvented. Owing to the fact that the inverse distance weighting interpolation removes the limitations in the Dirac delta function, the proposed SIDW-IBM has been successfully implemented on the non-uniform meshes to further improve the computational efficiency. The proposed SIDW-IBM is integrated with the reconstructed lattice Boltzmann flux solver (RLBFS) [2] to simulate some classic incompressible viscous flows, including flow past an in-line oscillating cylinder, flow past a heaving airfoil, and flow past a three-dimensional flexible plate. The good agreement between the present results and reference data demonstrates the capability and feasibility of the SIDW-IBM for simulating FSI problems with moving boundaries and large deformations.

1. Introduction

Fluid-structure interaction (FSI) phenomena exist in many science and engineering problems, which are attractive yet challenging. Many numerical approaches have been developed to tackle FSI problems for predicting the nonlinear dynamic characteristics behind complex fluid-solid coupled motions. Immersed boundary method (IBM) is one of the most popular numerical approaches, because it is easy to implement and employs a simple yet robust methodology to model the fluid-solid interaction. Differing from the conventional numerical approaches using body-fitted meshes, IBM modifies the local solution or adds a source term in governing equations to reflect the solid boundary effects. Consequently, IBM circumvents the needs of mesh regeneration for tackling FSI problems with moving boundaries and promotes the computational efficiency. IBM is pioneered by Peskin in 1970s [3]. Subsequently, many variants of IBM

---

[*] Corresponding author.
E-mail address: mpeshuc@nus.edu.sg (C. Shu).

[1] These authors have contributed equally to this work.

https://doi.org/10.1016/j.jcp.2024.113524
Received 21 February 2024; Received in revised form 6 October 2024; Accepted 20 October 2024

have been proposed to evaluate the restoring forcing term through different physical models, such as the penalty forcing scheme [3–5], which treats the solid boundary as an elastic boundary with high stiffness; the direct forcing scheme [6–8], which evaluates the restoring force through the velocity difference; the multi-direct forcing scheme [9,10], which eliminates the boundary velocity errors by iteratively applying the direct-forcing scheme; the momentum exchange scheme [11,12], which calculates the restoring force by the bounce-back condition; and the velocity correction scheme [13–16], which directly calculates the velocity corrections to satisfy the no-slip boundary condition.

The penalty forcing scheme is the first IBM proposed by Peskin [3] for modeling the complex interaction between heart valves and blood flows, where the restoring forcing is evaluated through Hooke's law with user-defined spring constant. Due to the pre-calculation of the restoring force, the numerical solutions corrected by the penalty IBM may not satisfy the no-slip boundary condition on the solid interface. Consequently, there are some streamlines penetrating the immersed boundary, inducing some non-physical phenomena and inaccurate numerical results. To circumvent the artificial parameters in the penalty scheme, Fadlun et al. [6] originally proposed the direct forcing scheme to evaluate the restoring forcing term through the difference between the desired boundary velocity and the flow velocity at Lagrangian points. Subsequently, Feng and Michaelides [7] extended the application of the direct forcing IBM into the lattice Boltzmann method (LBM) to simulate particle-laden flows. Later, Uhlmann [8] integrated the direct forcing IBM with the regularized delta function, which greatly simplifies the computational procedures and suppresses the numerical oscillations. Due to the straightforward methodology of the direct forcing IBM, it has been applied to simulate various complex FSI problems [17–19]. However, this one-step explicit direct forcing scheme does not take the mutual interactions between the surrounding flows and the immersed objects into consideration, which triggers that the no-slip boundary condition may not be accurately enforced on the solid interface. To alleviate the boundary velocity errors, Luo et al. [9] and Wang et al. [10] developed the multi-direct forcing IBM by multiple implementing the direct forcing IBM to fulfill the no-slip boundary condition. However, the iteration procedure in the multi-direct forcing IBM requires much higher computational cost when simulating FSI problems with moving boundaries. Shu et al. [13] demonstrated that the pre-calculated restoring force in these IBMs induces that the no-slip boundary condition cannot be accurately enforced. Therefore, Shu et al. [13] proposed a velocity correction IBM to accurately satisfy the no-slip boundary condition and suppress the streamline penetration, where the velocity corrections are taken as unknown. However, there are some numerical oscillations in the predicted forces, which might be induced by the linear relationships utilized on the Cartesian meshes. In their subsequent work, Wu and Shu [14,20] extended the velocity correction IBM with the smooth delta function interpolation, and the velocity corrections are implicitly resolved through solving a linear system. This implicit velocity correction IBM has been integrated with various flow solvers, such as the LBM [14,20], the Navier-Stokes solver [21], the lattice Boltzmann flux solver [15], and the gas kinetic flux solver [22] to simulate many challenging FSI problems. Although the implicit velocity correction IBM can accurately impose the no-slip boundary condition on the immersed objects, the smooth delta function requires that the distribution of Lagrangian points and the background fluid meshes should be uniform to construct proper correlations between the Lagrangian points and their surrounding Eulerian points. In practical FSI problems, if the complex geometries are described with the uniform Lagrangian points, the amount of the Lagrangian points would be very large, and it would require high computational cost to calculate the velocity corrections. To address this issue, Du et al. [1] combined the implicit velocity correction IBM with the inverse distance weighting (IDW) interpolation, and they successfully applied this implicit inverse distance weighting-immersed boundary method (IDW-IBM) on the non-uniform background Eulerian meshes and non-uniform Lagrangian point distribution, demonstrating that the computational efficiency can be greatly improved without accuracy degradation. Meanwhile, the implicit IDW-IBM also has to construct a correlation square matrix and solve its inversion at each time step when simulating FSI problems with moving boundaries. Consequently, with the increase of Lagrangian points, the memory space and computational cost would grow exponentially.

In this work, to remove the computational constraints induced by the memory space and matrix inversion of the square correlation matrix in the implicit IDW-IBM, we propose an approximation of the implicit IDW-IBM. Based on the Taylor series expansion, a second-order approximation can be derived through error analysis, which simplifies the direct inversion of the Large correlation matrix in the implicit IDW-IBM. Consequently, the velocity corrections can be explicitly predicted with second-order accuracy without the need to construct and inverse the square correlation matrix. With this second-order approximation, we develop a simplified inverse distance weighting-immersed boundary method (SIDW-IBM) that not only saves memory space but also promotes computational efficiency, all while maintaining computational accuracy.

The paper is organized as follows: the governing equations for FSI problems, the numerical approach for fluid dynamics, and the original implicit IDW-IBM are introduced in Section 2. Section 3.1 presents the detailed derivation of the second-order approximation in the proposed SIDW-IBM. Subsequently, Section 3.2 shows the comparisons of the memory consumption and computational time between the implicit IDW-IBM and the SIDW-IBM. In Section 4, the overall numerical accuracy test and numerical validations of the proposed SIDW-IBM are conducted. Conclusions are summarized in Section 5.

2. The mathematical model and numerical approach

In this section, the governing equations for describing the fluid-structure interaction behaviors are introduced, in which the solution procedure of the Navier-Stokes (N-S) equations is splitted into the prediction and correction steps through implementing a fractional step method. The reconstructed lattice Boltzmann flux solver (RLBFS) [2] is employed to predict the intermediate flow variables in the prediction step. Thereafter, IBM is applied to evaluate the velocity corrections for accurately satisfying the no-slip boundary conditions on the immersed boundary. In this section, we only introduce the original implicit IDW-IBM.

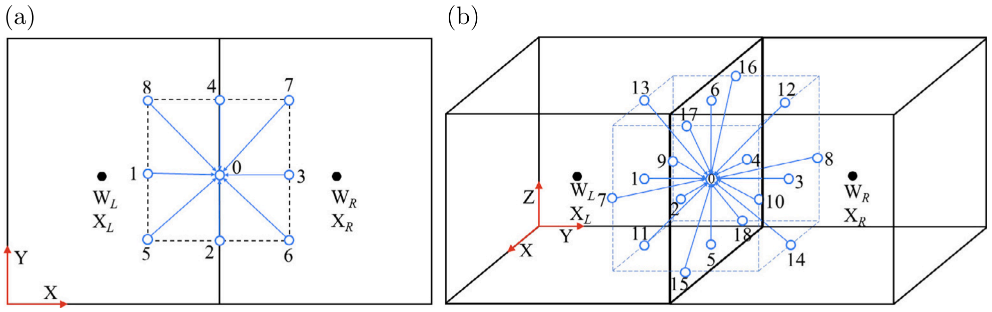

Fig. 1. Local reconstructed unit lattice at the cell interface in the RLBFS; (a) D2Q9 model for two-dimensional space and (b) D3Q19 model for three-dimensional space.

2.1. Governing equations

Based on the mass and momentum conservation laws, the macroscopic governing equations for modeling FSI problems can be given as:

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

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

where ρ, $\mathbf{u}$, ν, p, and $\mathbf{f}$ are the density of fluid, the flow velocity, the kinematic viscosity, the pressure, and the restoring force term in IBM, respectively. Note that the above weakly compressible N-S equations can simulate incompressible viscous flows when the low Mach number limit and small density variation are simultaneously satisfied.

A fractional step method is employed to decompose the solution process of N-S equations into two serial steps, namely, the prediction and correction steps. In the prediction step, the restoring forcing term is assumed to be $\mathbf{f}=0$, and the intermediate flow velocity $\mathbf{u}^*$ and the density at next time step ρ^{n+1} are predicted through solving the N-S equations using the finite volume method, where the inviscid and viscous fluxes at cell interface are simultaneously calculated by the RLBFS [2]. Then, in the correction step, IBM evaluates the velocity corrections at Lagrangian points and spreads them to the surrounding fluid points; consequently, the no-slip boundary condition is accurately satisfied in the corrected flow field. The solution procedure can be expressed as follow:

$$\frac{\partial\mathbf{W}}{\partial t}+\nabla\cdot\mathbf{F}=0,\quad \text{prediction step}, \tag{2a}$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t}=\mathbf{f},\quad \text{correction step}, \tag{2b}$$

where

$$\mathbf{W}=\begin{Bmatrix}\rho\\ \rho\mathbf{u}\end{Bmatrix},\mathbf{F}=\begin{Bmatrix}\mathbf{P}\\ \mathbf{\Pi}\end{Bmatrix}=\begin{Bmatrix}\rho\mathbf{u}\\ \rho\mathbf{uu}+p\boldsymbol{I}-\nu\left(\nabla\rho\mathbf{u}+(\nabla\rho\mathbf{u})^T\right)\end{Bmatrix}. \tag{3}$$

2.2. The RLBFS for predicting the intermediate flow variables

In this subsection, the governing equation Eq. (2a) in the prediction step is solved by the RLBFS, where the IBM forcing terms are excluded. In the framework of the RLBFS, a unit lattice (see Fig. 1) at the cell interface between two adjacent cells is required to construct for evaluating the inviscid and viscous fluxes.

The flow variables at lattice points can be obtained by the following interpolation scheme as

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

where $\mathbf{W}_L$ and $\mathbf{W}_R$ denote the flow variables at the center positions $\mathbf{X}_L$ and $\mathbf{X}_R$ of two adjacent cells, respectively. Note that in this work, the current, the sub-predicted, the predicted and the next time steps are represented by the superscripts "n", "m", "$*$" and "$n+1$", respectively. Once the flow variables at lattice points are evaluated, the sub-predicted flow variables at the cell interface can be calculated through solving the macroscopic equations recovered from the local Boltzmann equation as

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

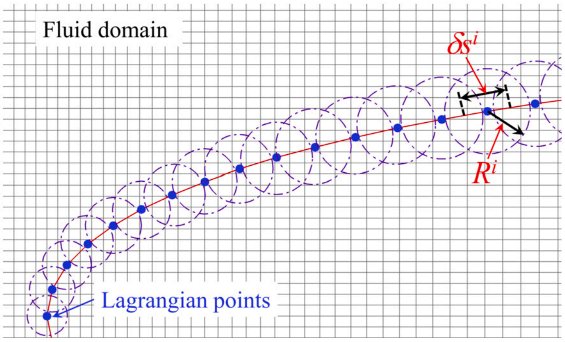

Fig. 2. Illustration of the implicit IDW-IBM around the immersed boundary, where δs^i and R^i denote the local Lagrangian point spacing and the local interpolation radius, respectively.

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

where the subscripts α, β and γ denote the coordinate components. c_s is the sound speed and δt is the time interval of the lattice Boltzmann method. Note that following the previous work [23], the partial derivatives in Eq. (5) are discretized on the unit lattice with the second-order central difference scheme.

Subsequently, through implementing the second order Taylor Series Expansion [2], the numerical fluxes in Eq. (3) can be evaluated as

$$P_\alpha = \left(\rho u_\alpha\right)^m, \tag{6a}$$

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

where τ denotes the relaxation parameter and is defined as $\tau = 0.5 + \nu/(c_s^2 \delta t)$.

To be consistent with Lu et al. [2], the first-order explicit scheme is employed for the time discretization. Therefore, the macroscopic flow variables ρ^{n+1} and $\mathbf{u}^*$ at the cell center are updated using the flux terms evaluated at the cell interface as

$$\rho^{n+1} = \rho^n - \frac{\Delta t}{\Delta V} \sum_k P_\alpha \Delta S_{k\alpha}, \tag{7a}$$

$$\left(\rho u_\alpha\right)^* = \left(\rho u_\alpha\right)^n - \frac{\Delta t}{\Delta V} \sum_k \Pi_{\alpha\beta} \Delta S_{k\beta}, \tag{7b}$$

where ΔV, Δt and ΔS_k are the volume of the control cell, the macroscopic time interval and the area of the kth interface of the control cell, respectively. Since the flow solver is not the focus of this work, the details of RLBFS can be found in Ref. [2].

2.3. *The implicit inverse distance weighting-immersed boundary method*

In this subsection, for simplicity, the original implicit IDW-IBM for 2D FSI problems (see Fig. 2) is presented. The discretization form of Eq. (2b) can be expressed as

$$\rho^{n+1} \frac{\left(\mathbf{u}^{n+1} - \mathbf{u}^*\right)}{\Delta t} = \mathbf{f}. \tag{8}$$

Previous study [13] demonstrates that the IBM forcing term in the momentum equation is equivalent to adding velocity corrections to the velocity field. Therefore, the IBM volume force in Eq. (1b) is convert to the flow velocity corrections in the present work. In the correction step, the IBM is employed to evaluate the difference between the desired boundary condition and the predicted flow field. Subsequently, the corrected flow velocity for the next time step can be calculated by adding the velocity corrections to the predicted velocity as

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \Delta \mathbf{u}, \tag{9}$$

where $\Delta \mathbf{u}$ denotes the velocity corrections on the Eulerian points.

The no-slip boundary condition requires that the interpolated velocities at Lagrangian points should be identical to the physical velocities on the solid boundary, which can be written as:

$$\mathbf{u}_b = \boldsymbol{U}_{body}. \tag{10}$$

The intermediate velocity on the solid boundary can be interpolated through:

$$\mathbf{u}_b^*\left(\boldsymbol{X}_i\right)=\sum_j w_{ij}\mathbf{u}^*\left(\boldsymbol{x}_j\right), \qquad i=1,2,\ldots,N_L,\quad j=1,2,\ldots,N_E, \tag{11a}$$

$$w_{ij}=\begin{cases} w_{ij}, & d_{ij}\le R^i,\\ 0, & d_{ij}>R^i,\end{cases}\quad w_{ij}=\frac{\gamma_{ij}}{\sum_j\gamma_{ij}},d_{ij}\le R^i,\gamma_{ij}=\left(\frac{R^i-d_{ij}}{R^i d_{ij}}\right)^e, \tag{11b}$$

where the subscript "b" denotes the flow variables at the Lagrangian point. i is the ith Lagrangian point, and j represents the jth Eulerian point inside the interpolation region. N_L and N_E represent the numbers of Lagrangian points and affected Eulerian points, respectively. $\boldsymbol{R}^i$ is the local interpolation radius. w_{ij} denotes the correlation coefficient in IDW interpolation and d_{ij} denotes the distance between the ith Lagrangian point and the jth Eulerian point. e is the power index. According to the previous study [1], $\boldsymbol{R}^i=2.5h$ and $e=0.5$ are employed for the uniform mesh in the present work. For the Non-uniform mesh, $\boldsymbol{R}^i=0.8\delta s^i$ and $e=0.5$ are adopted.

Similarly, the velocity corrections on the solid boundary can be calculated through the following interpolation relationships:

$$\delta\mathbf{u}_b^i\left(\boldsymbol{X}_i\right)\delta s^i=\sum_j w_{ij}\Delta\mathbf{u}\left(\boldsymbol{x}_j\right), \tag{12}$$

where $\delta\mathbf{u}_b^i$ and δs^i denote the velocity correction flux and the arc-length of the ith immersed segment, respectively.

Then, the velocity corrections at the background Eulerian points are distributed from those at Lagrangian points, which can be expressed as:

$$\Delta\mathbf{u}\left(\boldsymbol{x}_j\right)=\sum_i w_{ji}\delta\mathbf{u}_b^i\left(\boldsymbol{X}_i\right)\delta s^i. \tag{13}$$

Substituting Eqs. (9), (11), and (12) into Eq. (10) yields the following relationship:

$$\mathbf{u}_b^*+\delta\mathbf{u}_b\delta s=\boldsymbol{U}_{body}. \tag{14}$$

Substituting Eq. (13) into Eq. (14) yields the following relationship:

$$\boldsymbol{U}_{body}-\mathbf{u}_b^*=\sum_j w_{ij}\Delta\mathbf{u}\left(\boldsymbol{x}_j\right)=\sum_j w_{ij}\left(\sum_m w_{jm}\delta\mathbf{u}_b^m\left(\boldsymbol{X}_m\right)\delta s^m\right), \tag{15}$$

which can be rewritten in the matrix form as follow:

$$\boldsymbol{CY}=\boldsymbol{D}, \tag{16a}$$

$$C_{im}=\sum_j w_{ij}w_{jm}, \tag{16b}$$

$$\boldsymbol{D}=\boldsymbol{U}_{body}-\mathbf{u}_b^*, \tag{16c}$$

$$\boldsymbol{Y}=\left[\delta\mathbf{u}_b^1\delta s^1,\delta\mathbf{u}_b^2\delta s^2,\ldots,\delta\mathbf{u}_b^{N_L}\delta s^{N_L}\right]^t. \tag{16d}$$

The calculation procedure of the original implicit IDW-IBM is summarized in **Algorithm 1**. The implicit IDW-IBM algorithm requires a significant amount of memory and time to construct the correlation matrix $\boldsymbol{C}$ and its inversion $\boldsymbol{C}^{-1}$ when a large number of Lagrangian points is introduced to describe the complex geometries. In this work, the Gaussian elimination method is employed to compute the matrix inversion.

Algorithm 1 The original IDW-IBM.

```
Step 1: Construct the correlation matrix $\boldsymbol{C}$
Step 2: Solve the matrix inversion $\boldsymbol{C}^{-1}=\frac{1}{|\boldsymbol{C}|}\begin{bmatrix} C_{11} & C_{21} & \ldots & C_{N_L1}\\ C_{12} & C_{22} & \ldots & C_{N_L2}\\ \ldots & \ldots & \ldots & \ldots\\ C_{1N_L} & C_{2N_L} & \ldots & C_{N_LN_L}\end{bmatrix}$
Step 3: $\delta\mathbf{u}_b\delta s=\boldsymbol{C}^{-1}\left(\boldsymbol{U}_{body}-\mathbf{u}_b^*\right)$
Step 4: Update the intermediate flow field by Eqs. (13) and (9).
```

3. Simplified inverse distance weighting-immersed boundary method

In this section, a novel technique for simplifying the calculation procedure of the original implicit IDW-IBM is proposed and the detailed derivation is presented. This novel technique is developed through the error analysis on the velocity corrections, where a second order approximation is made. Consequently, the construction and inversion of the large correlation matrix in the original implicit IDW-IBM are circumvented, saving enormous computational cost and memory space. Combining this novel technique and the original implicit IDW-IBM, a simplified inverse distance weighting-immersed boundary method is proposed to efficiently tackle FSI problems with moving boundaries. In addition, the comparisons of memory space and computational cost between the SIDW-IBM and the original implicit IDW-IBM are presented to exhibit the advantages of the proposed SIDW-IBM.

3.1. The second-order approximation in the novel technique

Recalling Eq. (15) from previous section, the matrix form can be rewritten as:

$$\sum_{m=1}^{N_L} C_{im}\delta\mathbf{u}_b^m\delta s^m = \boldsymbol{D}_i. \tag{17}$$

The unknown term $\delta\mathbf{u}_b^m\delta s^m$ is related to the $\boldsymbol{D}_i$ when the coefficient $C_{im}\neq 0$. Hence, Eq. (17) can be rewritten as:

$$\sum_{m\in\{C_{im}\neq 0\}} C_{im}\delta\mathbf{u}_b^m\delta s^m = \boldsymbol{D}_i. \tag{18}$$

The relationship $C_{im}\neq 0$ requires:

$$w_{ij}\neq 0,\quad w_{jm}\neq 0. \tag{19}$$

Assuming there are two sets of Eulerian points, $j\in S(I)$ and $j\in S(M)$, to satisfy the relationships $w_{ij}\neq 0$ and $w_{jm}\neq 0$, respectively, $C_{im}\neq 0$ can be achieved when the ith and mth Lagrangian points share common Eulerian points $j\in S(I)\cap S(M)$. Based on the IDW interpolation scheme, if $j\in S(I)$, $\left\|\boldsymbol{x}_j-\boldsymbol{X}_i\right\|\leq\|R\|=O(\delta s)=O(h)$. Similarly, if $j\in S(M)$, $\left\|\boldsymbol{x}_j-\boldsymbol{X}_m\right\|\leq\|R\|=O(\delta s)=O(h)$.

Thereby, $C_{im}\neq 0$ requires that the distance between two Lagrangian points should be:

$$\begin{aligned}\left\|d\boldsymbol{X}_{im}\right\|&=\left\|\boldsymbol{X}_i-\boldsymbol{X}_m\right\|\\&=\left\|\left(\boldsymbol{x}_j-\boldsymbol{X}_i\right)-\left(\boldsymbol{x}_j-\boldsymbol{X}_m\right)\right\|\\&\leq\left\|\boldsymbol{x}_j-\boldsymbol{X}_i\right\|+\left\|\boldsymbol{x}_j-\boldsymbol{X}_m\right\|\\&\leq\|2R\|=O(h).\end{aligned} \tag{20}$$

The unknown terms $\delta\mathbf{u}_b^m$ and δs^m can be expanded through using the second-order Taylor series expansion as:

$$\delta\mathbf{u}_b^m=\delta\mathbf{u}_b^i+\frac{\partial\delta\mathbf{u}_b^m}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}+O\left(d\boldsymbol{X}_{im}{}^2\right), \tag{21}$$

$$\delta s^m=\delta s^i+\frac{\partial\delta s}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}+O\left(d\boldsymbol{X}_{im}{}^2\right). \tag{22}$$

Multiplying Eqs. (21) and (22), the following relationship can be obtained as:

$$\begin{aligned}\delta\mathbf{u}_b^m\delta s^m=&\delta\mathbf{u}_b^i\delta s^i+\frac{\partial\delta\mathbf{u}_b}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}\delta s^i+\frac{\partial\delta s}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}\delta\mathbf{u}_b^i+\frac{\partial\delta\mathbf{u}_b}{\partial\boldsymbol{X}}\frac{\partial\delta s}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}d\boldsymbol{X}_{im}\\&+\delta\mathbf{u}_b^iO\left(d\boldsymbol{X}_{im}{}^2\right)+\delta s^iO\left(d\boldsymbol{X}_{im}{}^2\right)+O\left(d\boldsymbol{X}_{im}{}^4\right)\\=&\delta\mathbf{u}_b^i\delta s^i+\frac{\partial\delta\mathbf{u}_b}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}\delta s^i+\frac{\partial\delta s}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}\delta\mathbf{u}_b^i+O\left(d\boldsymbol{X}_{im}{}^2\right).\end{aligned} \tag{23}$$

To establish a proper correlation between Lagrangian points and Eulerian grids to prevent fluid leakage, δs^i should be:

$$\delta s^i=O(h). \tag{24}$$

In addition, $\delta\mathbf{u}_b\sim O\left(\boldsymbol{U}_{body}-\mathbf{u}_b^*\right)$, which is related to the accuracy order of the IDW interpolation; thus:

$$\delta\mathbf{u}_b\sim O(h). \tag{25}$$

Substituting Eqs. (24) and (25) into Eq. (23), the second and third terms in Eq. (23) can be simplified as:

$$\begin{aligned}\left\|\frac{\partial\delta\mathbf{u}_b^i}{\partial\boldsymbol{X}}d\boldsymbol{X}_{im}\Delta s^i\right\|&\leq\left\|\frac{\partial\delta\mathbf{u}_b^i}{\partial\boldsymbol{X}}\right\|\left\|d\boldsymbol{X}_{im}\right\|\left\|\delta s^i\right\|\\&=\left\|\frac{\partial\delta\mathbf{u}_b^i}{\partial\boldsymbol{X}}\right\|\cdot O(h)\cdot O(h)\\&=O\left(h^2\right),\end{aligned} \tag{26}$$

Table 1
Memory consumption for assembling the correlation matrix of the implicit IDW-IBM and the SIDW-IBM. The implicit IDW-IBM requires $N_L \times N_L$ memory space to construct $\boldsymbol{C}_{N_L\times N_L}$; the SIDW-IBM requires N_L memory space to assemble $\boldsymbol{A}_{N_L}$.

Number of Lagrangian points	Algorithm		Ratio
	Implicit IDW-IBM	SIDW-IBM	
1000	7.6294 MB	7.8125 KB	1000
5000	190.7349 MB	39.0625 KB	5000
10000	762.9395 MB	78.125 KB	10000

Table 2
Computational time cost of the implicit IDW-IBM and the SIDW-IBM for one time step with single AMD processor (3.4 GHz).

Number of Lagrangian points	Algorithm		Ratio
	Implicit IDW-IBM	SIDW-IBM	
1000	1.4531 s	0.4844 s	3.000
5000	591.6875 s	12.7656 s	46.350
10000	6239.9688 s	50.7031 s	123.069

$$\begin{aligned}\left\|\frac{\partial \delta s}{\partial \boldsymbol{X}} d\boldsymbol{X}_{im}\delta \mathbf{u}_b^i\right\| &\le \left\|\frac{\partial \delta s}{\partial \boldsymbol{X}}\right\| \left\|d\boldsymbol{X}_{im}\right\| \left\|\delta \mathbf{u}_b^i\right\| \\ &= \left\|\frac{\partial \delta s}{\partial \boldsymbol{X}}\right\| \cdot O(h)\cdot O(h) \\ &= O\left(h^2\right).\end{aligned} \tag{27}$$

Substituting Eqs. (26) and (27) into Eq. (23), Eq. (23) can be simplified as:

$$\delta \mathbf{u}_b^m \delta s^m = \delta \mathbf{u}_b^i \delta s^i + O(h^2). \tag{28}$$

Substituting Eq. (28) into Eq. (18), the unknown terms $\delta \mathbf{u}_b \delta s$ can be explicitly resolved as:

$$\delta \mathbf{u}_b^i \delta s^i = \frac{\boldsymbol{D}_i}{\sum_{m\in\{C_{im}\neq 0\}} C_{im}}. \tag{29}$$

Based on the second order approximation, the SIDW-IBM is developed with high computational efficiency and a small memory space requirement. The procedure of the proposed SIDW-IBM is presented in **Algorithm 2**. In Section 4, the overall accuracy test of the proposed SIDW-IBM integrated with the RLBFS will be conducted to check whether the second order approximation affects the global accuracy.

Algorithm 2 The proposed SIDW-IBM.

```
Step 1: Construct the correlation matrix $A_i = \sum_{m\in\{C_{im}\neq 0\}} C_{im}$
Step 2: $\delta \mathbf{u}_b \delta s = \left(\boldsymbol{U}_{body} - \mathbf{u}_b^*\right)/\boldsymbol{A}$
Step 3: Update the intermediate flow field by Eqs. (13) and (9).
```

3.2. Comparisons of the memory space and computational efficiency

To elucidate the advantages of the proposed SIDW-IBM in solving complex FSI problems with a large amount of Lagrangian points, comparisons of the required memory space and computational cost between the original implicit IDW-IBM and the proposed SIDW-IBM are presented with different number of Lagragian points. In this subsection, a vertical rotating sphere in a concentric cubical enclosure is considered, and all numerical tests are performed on the same Eulerian grids of 240^3, in which the amount of Lagrangian points varies from 10^3 to 10^4.

According to the **Algorithm 1** and **Algorithm 2**, the dimension of the correlation matrix $\boldsymbol{C}_{N_L\times N_L}$ is N_L times larger than the correlation matrix $\boldsymbol{A}_{N_L}$, indicating that the memory consumption of the implicit IDW-IBW is N_L times than that of the proposed SIDW-IBM (see Table 1). To further evaluate the computational efficiency of the proposed SIDW-IBM, Table 2 tabulates the computational time of the implicit IDW-IBM and the SIDW-IBM, which demonstrates that the proposed SIDW-IBM is more efficient than the original implicit IDW-IBM. In conclusion, these comparisons present that the proposed SIDW-IBM is suitable for modeling FSI problems with flexible bodies and complex geometries.

To illustrate the advantages of the proposed SIDW-IBM, comparisons with existing IBMs for incompressible flows are presented in Appendix A. Compared with the discussed IBMs in Appendix A for FSI problems, the advantages of the proposed SIDW-IBM can be concluded as

Table 3
Numerical error of the velocity component u for decaying vortex flow when using different mesh spacing, where N, h and N_L denote the number of Eulerian points, mesh spacing and the number of Lagrangian points, respectively.

N	h	N_L $\left(\frac{\delta s}{h} \approx 2.618\right)$	$L_2 error$	N_L $\left(\frac{\delta s}{h} \approx 1.309\right)$	$L_2 error$
10^2	0.2	6	2.511×10^{-2}	12	2.980×10^{-2}
20^2	0.1	12	7.281×10^{-3}	24	7.637×10^{-3}
40^2	0.5	24	1.860×10^{-3}	48	1.917×10^{-3}
80^2	0.025	48	4.768×10^{-4}	96	4.775×10^{-4}
160^2	0.0125	96	1.206×10^{-4}	192	1.267×10^{-4}

- With respect to the direct forcing IBM and the MLS-based direct forcing IBM, the proposed SIDW-IBM can accurately enforce the no-slip boundary condition by eliminating the errors induced by the inequality between the interpolation and spreading processes.
- With respect to the penalty forcing IBM and the feedback forcing IBM, the SIDW-IBM does not require artificial parameters between the restoring force and the velocity difference.
- With respect to the multi-direct forcing IBM, the proposed SIDW-IBM can achieve certain convergence rate and maintain the boundary velocity errors within $O(h^2)$ without internal iterations.
- With respect to the implicit and explicit velocity correction IBMs, the proposed SIDW-IBM removes the limitations in the Dirac delta function and can be implemented on non-uniform meshes to improve the calculation efficiency. Additionally, compared to the explicit velocity correction IBM, the SIDW-IBM approximates the large correlation matrix into a diagonal matrix through error analysis using Taylor series expansion, which has rigorous mathematical proof.
- With respect to the original IDW-IBM, the proposed SIDW-IBM eliminates the intrinsic limitations in IDW-IBM, such as the construction and inversion of the large correlation matrix, while maintaining the accuracy.

4. Results and discussion

In this section, the accuracy test on the proposed SIDW-IBM integrated with the RLBFS is conducted to analyze the effects induced by the second order approximation on the global accuracy. Subsequently, several classic benchmark problems are numerically investigated to validate the capability and robustness of the proposed SIDW-IBM, and the numerical results demonstrate that the SIDW-IBM can accurately simulate complex FSI problems with moving boundaries and large deformations.

4.1. Overall accuracy test

In this subsection, the decaying vortex problem [24] is adopted to examine the overall accuracy of the proposed SIDW-IBM integrated with the flow solver. The analytic solution of this problem can be expressed as follows:

$$u(x, y, t) = -U \cos(\pi x/L) \sin(\pi y/L)\, e^{-\frac{2\pi^2 U t}{\mathrm{Re}\, L}}, \tag{30a}$$

$$v(x, y, t) = U \sin(\pi x/L) \cos(\pi y/L)\, e^{-\frac{2\pi^2 U t}{\mathrm{Re}\, L}}, \tag{30b}$$

$$\rho(x, y, t) = \rho_0 - \frac{\rho_0 U^2}{4C_s^2}\left[\cos(2\pi x/L) + \cos(2\pi y/L)\right] e^{-\frac{4\pi^2 U t}{\mathrm{Re}\, L}}, \tag{30c}$$

where $\rho_0 = 1$ and $C_s = 1/3$ are adopted. U and L denote the characteristic velocity and the characteristic length, respectively. The computational domain is $[-L, L] \times [-L, L]$ with $L = 1$. As shown in Table 3, this numerical accuracy test is conducted on five different uniform meshes, namely, 11×11, 21×21, 41×41, 81×81, and 161×161, to evaluate the convergence order. To test the impact of the proposed SIDW-IBM on the numerical accuracy, a circular cylinder with radius $0.5L$ is immersed in the fluid domain and its center is located at $(0, 0)$, where the exact solution is imposed on the immersed boundary. In addition, the effects of spacing ratio between the Lagrangian and Eulerian grid spacings on the global accuracy are also investigated, where two spacing ratios are adopted in Table 3. The numerical error of velocity component u is calculated at the simulation time $Ut/L = 1$. Note that the numerical error is quantified by L_2 norm, which is defined as:

$$L_2 error = \sqrt{\frac{\sum_{i=1}^{N}\left[\frac{(u_{numerical} - u_{exact})}{U}\right]^2}{N}}, \tag{31}$$

where the subscripts, "*numerical*" and "*exact*" denote the numerical solutions and analytical results, respectively. N represents the number of Eulerian points. The obtained numerical errors of five meshes are also listed in Table 3. And the L_2 error versus mesh spacing in the log scale is displayed in Fig. 3. The numerical results indicate that the numerical accuracy of the SIDW-IBM integrated with the RLBFS is slightly lower than second order and similar to that of the RLBFS. Meanwhile, the numerical results also demonstrate that the second order approximation in the SIDW-IBM has negligible effects on the global accuracy.

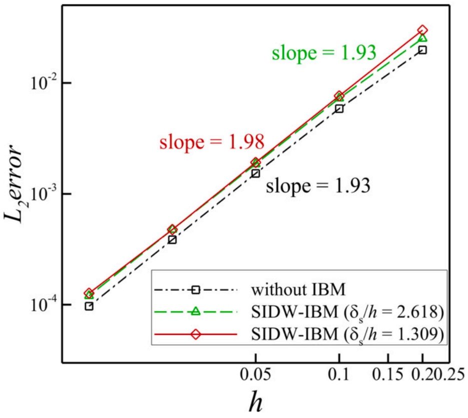

Fig. 3. Grid convergence of the RLBFS, the SIDW-IBM with the RLBFS $(\delta_s/h = 2.618)$, and the SIDW-IBM with the RLBFS $(\delta_s/h = 1.309)$.

Table 4
Comparison of drag and lift coefficients for the in-line oscillating cylinder in a free stream at $Re = 100$.

$Re = 100$	f_c/f_0	$C_{d,mean}$	$C_{l,max}$
Luo et al. [25]	2	1.68	0.95
Örley et al. [26]	2	1.73	0.93
Su et al. [27]	2	1.7	0.97
Liao et al. [28]	2	1.71	0.95
IDW-IBM	2	1.72	0.97
Present SIDW-IBM	2	**1.72**	**0.97**

4.2. Flow past an in-line oscillating cylinder

In this subsection, flow past an in-line oscillating cylinder is numerically investigated to evaluate the capability of the proposed SIDW-IBM in simulating FSI problems with moving boundaries. This FSI problem is governed by the following dimensionless parameters: the Reynolds number $Re = U_0 D/\nu$, the dimensionless oscillating amplitude $A = A_d/D$, and the dimensionless oscillating frequency $f = f_c/f_0$, where U_0, D, A_d, f_c, f_0 denote the free stream velocity, the cylinder's diameter, the dimensional oscillating amplitude, the dimensional oscillating frequency, and the vortex shedding frequency behind a stationary cylinder at the same Reynolds number, respectively. Besides, the dimensionless oscillating period is defined as $T = 1/f$. The governing equation of the cylinder's oscillating motion can be written as:

$$X_c(t) = A\sin(2\pi f t)\,. \tag{32}$$

The parameters of $Re = 100$, $A = 0.14$ and $f = 2$ are employed in this numerical test. The computational size of this test is set as $[-14D, 20D] \times [-15D, 15D]$, which is discretized by a non-uniform Eulerian grid of 700×900. Note that the flow region around the cylinder of $[-1D, 4D] \times [-4D, 4D]$ is uniformly discretized with grid spacing of $h = D/100$, where the interaction behaviors between surrounding fluids and the immersed cylinder can be precisely captured. As mentioned in Section 3.1, the Lagrangian point spacing should be $O(h)$; thereby, 125 Lagrangian points are introduced to discretize the immersed boundary to construct a proper correlation between Lagrangian and Eulerian points.

As shown in Fig. 4, the instantaneous vorticity contours generated by the proposed SIDW-IBM agree well with those in Luo et al. [25] at different time instants, qualitatively demonstrating that the proposed SIDW-IBM can accurately predict the nonlinear dynamic behaviors behind complex FSI problems with moving boundaries.

Furthermore, to quantitatively validate the accuracy of the aerodynamic forces predicted by the proposed SIDW-IBM, the time evolutions of the drag and lift coefficients are illustrated in Fig. 5. It is evident from Fig. 5 that the time histories of force coefficients generated by the proposed SIDW-IBM are in a good agreement with the reference data [25], indicating that the complex dynamic interaction between the immersed object and the surrounding fluid is accurately captured by the proposed SIDW-IBM. Moreover, the aerodynamic forces predicted by the SIDW-IBM are almost identical to those of the IDW-IBM, demonstrating that the second-order approximation employed in the SIDW-IBM has negligible effects on the quantitative results. In addition, the time-averaged drag force and the maximum lift force are compared with the reference data tabulated in Table 4. Table 4 further confirms that the quantitative statistical results predicted by the proposed SIDW-IBM are consistent with the reference solutions, indicating that this novel SIDW-IBM can provide reasonable results for complex FSI problems with moving boundaries.

To evaluate whether the no-slip boundary condition is accurately imposed on the immersed boundary by the proposed SIDW-IBM, the instantaneous boundary velocity errors are shown in Fig. 6, where the selected time instants correspond to those in Fig. 4. It is

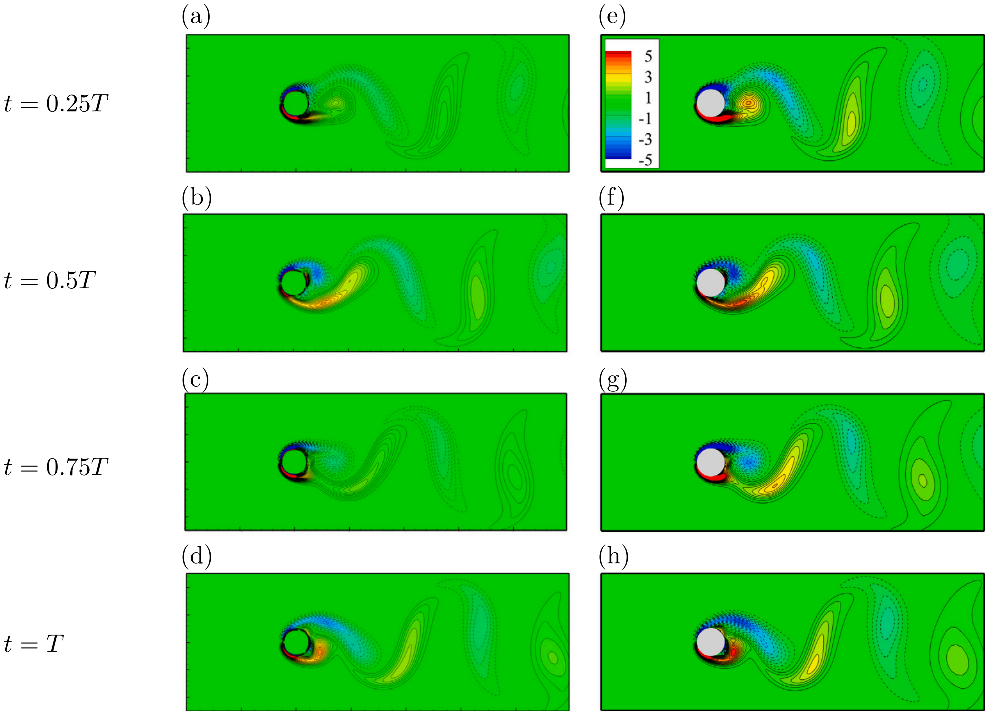

Fig. 4. Instantaneous vorticity contours around the in-line oscillating cylinder. Left column: reference vorticity contours of Luo et al. [25]; right column: present results generated by the proposed SIDW-IBM.

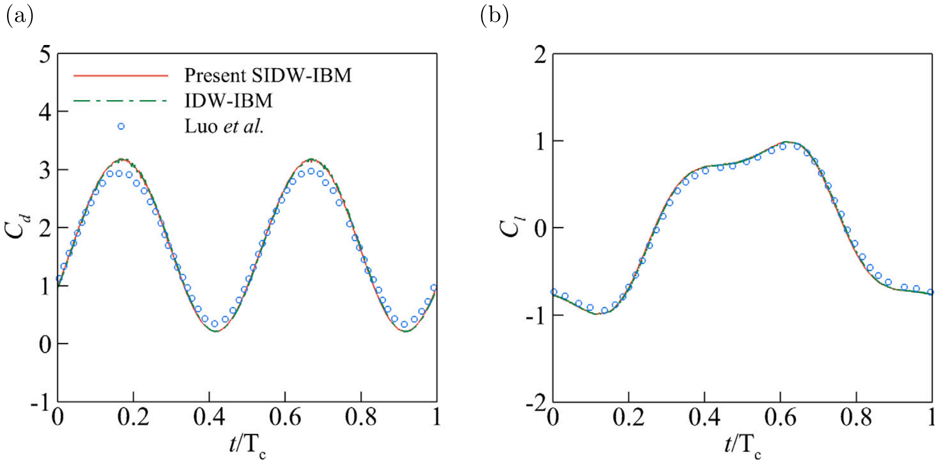

Fig. 5. Time evolutions of the (a) drag and (b) lift coefficients for the in-line oscillating cylinder in a uniform flow at $Re = 100$.

evident from Fig. 6 that the boundary velocity errors within the proposed SIDW-IBM are negligible, indicating that the flow field corrected by the proposed SIDW-IBM can accurately satisfy the no-slip boundary condition on the solid boundary.

4.3. Flow past a heaving NACA0012 airfoil

In this subsection, the flow past a stationary NACA0012 airfoil is firstly adopted to evaluate the ability of the proposed SIDW-IBM to preserve the conservation of force and momentum. Subsequently, the flow past a heaving NACA0012 airfoil is employed to examine the accuracy and effectiveness of the proposed SIDW-IBM for simulating FSI problems with irregular moving boundaries. In addition, the capability of the proposed SIDW-IBM for simulating FSI problems on non-uniform meshes is also investigated. In this numerical examination, the Reynolds number is defined as $Re = U_\infty c/\nu$, in which U_∞ and c denote the free stream velocity and the airfoil chord length, respectively.

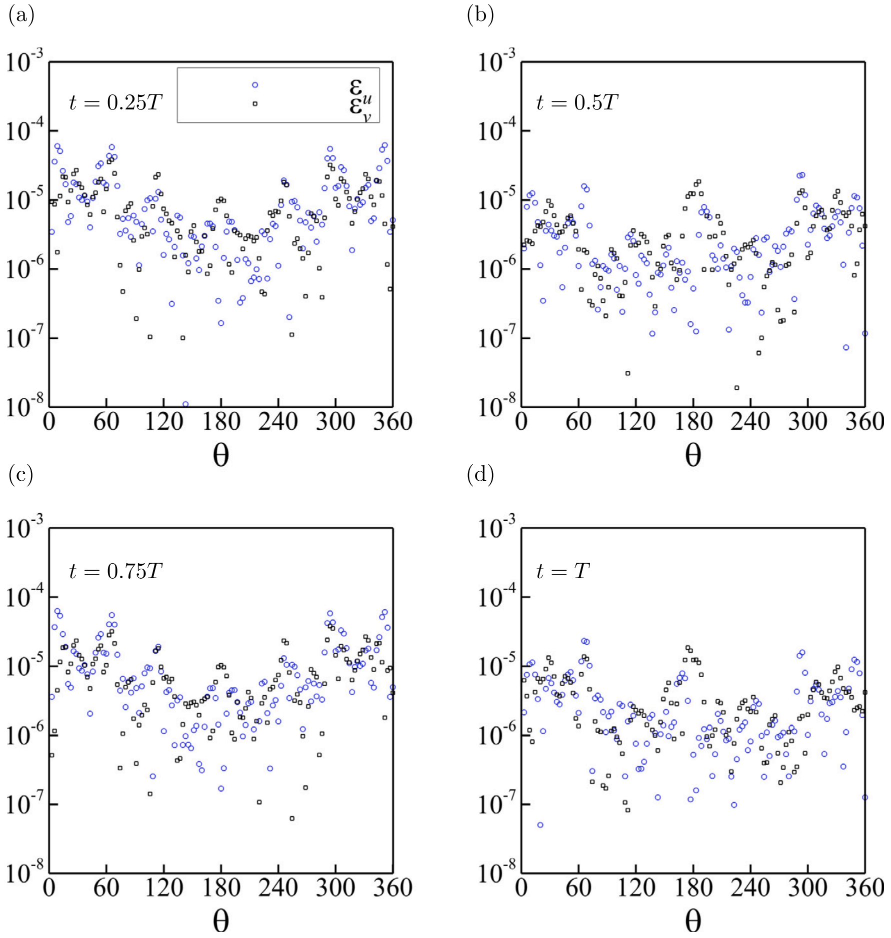

Fig. 6. The instantaneous boundary velocity errors of the proposed SIDW-IBM. (a) $t = 0.25T$, (b) $t = 0.5T$, (c) $t = 0.75T$, and (d) $t = T$. The boundary velocity errors are defined as: $\varepsilon_u = |U_{body} - u_b|$ and $\varepsilon_v = |V_{body} - v_b|$.

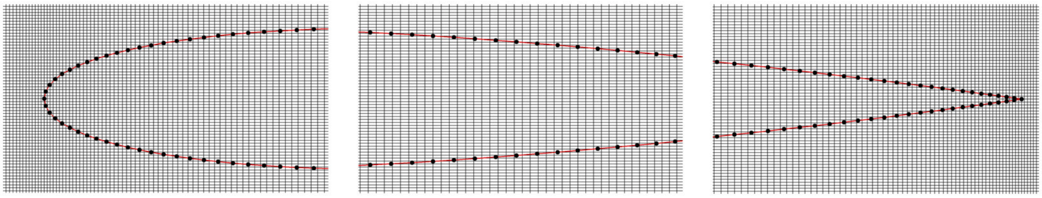

Fig. 7. Distribution of the Lagrangian points and the Eulerian mesh around the heaving NACA0012 airfoil on Mesh II.

The computational domain of $[-20c, 30c] \times [-15c, 15c]$ is discretized by two different meshes (Mesh I and Mesh II), where the flow region $[-0.05c, 1.05c] \times [-0.15c, 1.15c]$ around the NACA0012 airfoil is discretized by a uniform mesh (Mesh I) and a non-uniform mesh (Mesh II) to evaluate the suitability of the proposed SIDW-IBM for tackling FSI problems on non-uniform meshes. On the Mesh I, the background Eulerian mesh spacing is $h = 0.0025$ and the Lagrangian point spacing is $\delta s = 0.00636$; thereby, the total mesh size of the computational domain is 681×691 and the number of Lagrangian points is 320. On the Mesh II, to improve the numerical accuracy at the leading and trailing edges of the airfoil, the distribution of Lagrangian points is denser near these two edges than in other regions. Correspondingly, the Eulerian mesh is also denser close to the leading and trailing edges as shown in Fig. 7. The minimum Lagrangian point spacing and the Eulerian mesh spacing near the leading and trailing edges are $\delta s = 0.00625$ and $h = 0.0025$, respectively. Consequently, the whole Eulerian mesh size and the number of Lagrangian points decrease to 501×691 and 160, respectively, which greatly promotes the computational efficiency and saves the memory space.

Firstly, to evaluate whether the proposed SIDW-IBM can preserve the conservation of force and torque on both uniform and non-uniform meshes, the numerical experiments about the conservation ability are conducted. In this numerical test, the simulation of the flow around a stationary NACA0012 airfoil at $Re = 500$ is performed on the Mesh I and Mesh II. Uhlmann [8] and Pinelli

Table 5
Conservation errors of force and torque on the uniform and non-uniform meshes.

Error	$\varepsilon_{\mathrm{F}_x}$	$\varepsilon_{\mathrm{F}_y}$	ε_t
Uniform mesh	6.724×10^{-7}	4.071×10^{-17}	3.268×10^{-17}
Non-uniform mesh	1.151×10^{-6}	1.278×10^{-16}	7.609×10^{-17}

et al. [29] demonstrate that the ability to maintain the conservation of force and torque is a fundamental property of IBM, which can be expressed as

$$\sum_j \mathbf{f}\left(\mathbf{x}_j\right) \Delta \mathrm{v}_j = \sum_i \mathbf{F}_b\left(\mathbf{X}_i\right) \Delta V_i, \tag{33a}$$

$$\sum_j \mathbf{x}_j \times \mathbf{f}\left(\mathbf{x}_j\right) \Delta \mathrm{v}_j = \sum_i \mathbf{X}_i \times \mathbf{F}_b\left(\mathbf{X}_i\right) \Delta V_i, \tag{33b}$$

where $\Delta \mathrm{v}_j$, ΔV_i represent the forcing volume of the j-th Eulerian point and the defined forcing volume of the i-th Lagrangian point, respectively. In this work, the Lagrangian forcing volume is defined as $\Delta V_i = \delta s^i h^i$, where h^i denotes the local thickness and is defined as $h^i = \frac{1}{2}\sum_{j=1}^{N_E} w_{ij}\left(\Delta x_j + \Delta y_j\right)$. The errors of the conservation of force and torque are defined as $\varepsilon_{\mathrm{F}_x} = |\sum_j \mathrm{f}_x\left(\mathbf{x}_j\right)\Delta v_j - \sum_i \mathrm{F}_{b,x}\left(\mathbf{X}_i\right)\Delta V_i|$, $\varepsilon_{\mathrm{F}_y} = |\sum_j \mathrm{f}_y\left(\mathbf{x}_j\right)\Delta v_j - \sum_i \mathrm{F}_{b,y}\left(\mathbf{X}_i\right)\Delta V_i|$, $\varepsilon_t = |\sum_j \mathbf{x}_j \times \mathbf{f}\left(\mathbf{x}_j\right)\Delta v_j - \sum_i \mathbf{X}_i \times \mathbf{F}_b\left(\mathbf{X}_i\right)\Delta V_i|$, respectively.

Table 5 indicates that the conservation errors on the uniform and non-uniform meshes are very small and within $O(h^2)$, which demonstrates the ability of the present SIDW-IBM to preserve the conservation of force and torque. Note that the x-direction is the streamwise direction, so the error of the x-direction force is in the 10^{-6} order on the non-uniform mesh, which is slightly larger than those of the y-direction force and torque.

Subsequently, the flow past a heaving NACA0012 airfoil is conducted at $Re = 1000$ and $c = 1$. The governing equation of the airfoil's heaving motion can be written as

$$y = 0.25t^2\left(3 - t\right), \tag{34}$$

where y denotes the vertical displacement of the airfoil's center and $t \in [0, 2]$ is the time instant. At $t = 0$, the heaving airfoil is initiated from the initial position $y = 0$; at $t = 2$, the heaving airfoil moves to its destination $y = 1$.

As shown in Fig. 8, during the upward movement of the airfoil, the process of vortex generation and shedding from the lower surface and trailing edge of the airfoil can be clearly observed. It can be obviously seen that the instantaneous vorticity contours around the heaving airfoil predicted by the proposed SIDW-IBM are in a good agreement with the reference vorticity contours [30], which qualitatively demonstrates that the no-slip boundary condition is accurately enforced by the proposed SIDW-IBM on irregular moving boundaries. Furthermore, the instantaneous distributions of the pressure coefficient C_p along the upper and lower surfaces of the heaving airfoil are compared with the reference data [30] in Fig. 8. It is evident from Fig. 8 that the present results agree well with the reference data [30], quantitatively indicating that the proposed SIDW-IBM can accurately predict the nonlinear dynamic responses in complex FSI systems. Moreover, the pressure distributions predicted by the proposed SIDW-IBM are identical to those generated by the IDW-IBM, indicating that the second-order approximation in the proposed SIDW-IBM has negligible effects on the global accuracy. Meanwhile, the results obtained with the non-uniform mesh are in a good agreement with those generated by the uniform mesh and the reference data [30], which demonstrates the effectiveness and suitability of the SIDW-IBM for resolving FSI problems on non-uniform meshes. On the other hand, it confirms that the proposed SIDW-IBM circumvents the limitations of the uniform Cartesian mesh requirement within the traditional diffuse interface IBM. In addition, the time evolutions of the drag and lift coefficients are presented in Fig. 9, and the results predicted by the SIDW-IBM agree well with the reference data [30,31], which further confirms the accuracy of this novel SIDW-IBM in predicting the nonlinear characteristics of a FSI system with irregular moving boundaries. Note that the pressure distributions and aerodynamic forces generated by the SIDW-IBM are consistent with those provided by the IDW-IBM (see Figs. 8 and 9), indicating that the second-order approximation in the proposed SIDW-IBM has a negligible impact on the accuracy of force prediction.

To further evaluate the computational cost saving performance of the proposed SIDW-IBM, the original implicit IDW-IBM, the multi-direct forcing IBM, and the SIDW-IBM are compared in terms of computational time for one heaving period, where the convergence criterion for the multi-direct forcing IBM is set as $\varsigma_c = 10^{-5}$. As shown in Fig. 10, the computational efficiency of the proposed SIDW-IBM is superior than the original implicit IDW-IBM. On the uniform mesh, the computational efficiency the proposed SIDW-IBM is similar to that of the multi-direct forcing method. When this novel SIDW-IBM is implemented on the non-uniform mesh, the computational time is saved by 50.98% compared to the original implicit IDW-IBM and 36.09% compared to the multi-direct forcing IBM. These results demonstrate that this novel SIDW-IBM significantly promotes the computational efficiency than the original implicit IDW-IBM and is suitable to be applied on the non-uniform mesh.

4.4. Flow past a 3D torus

In this subsection, to further evaluate the accuracy and efficiency of the proposed SIDW-IBM for 3D flows around complex geometries, the flow past a 3D torus is conducted. The geometry of this 3D torus is presented in Fig. 11, where D and d denote the mean

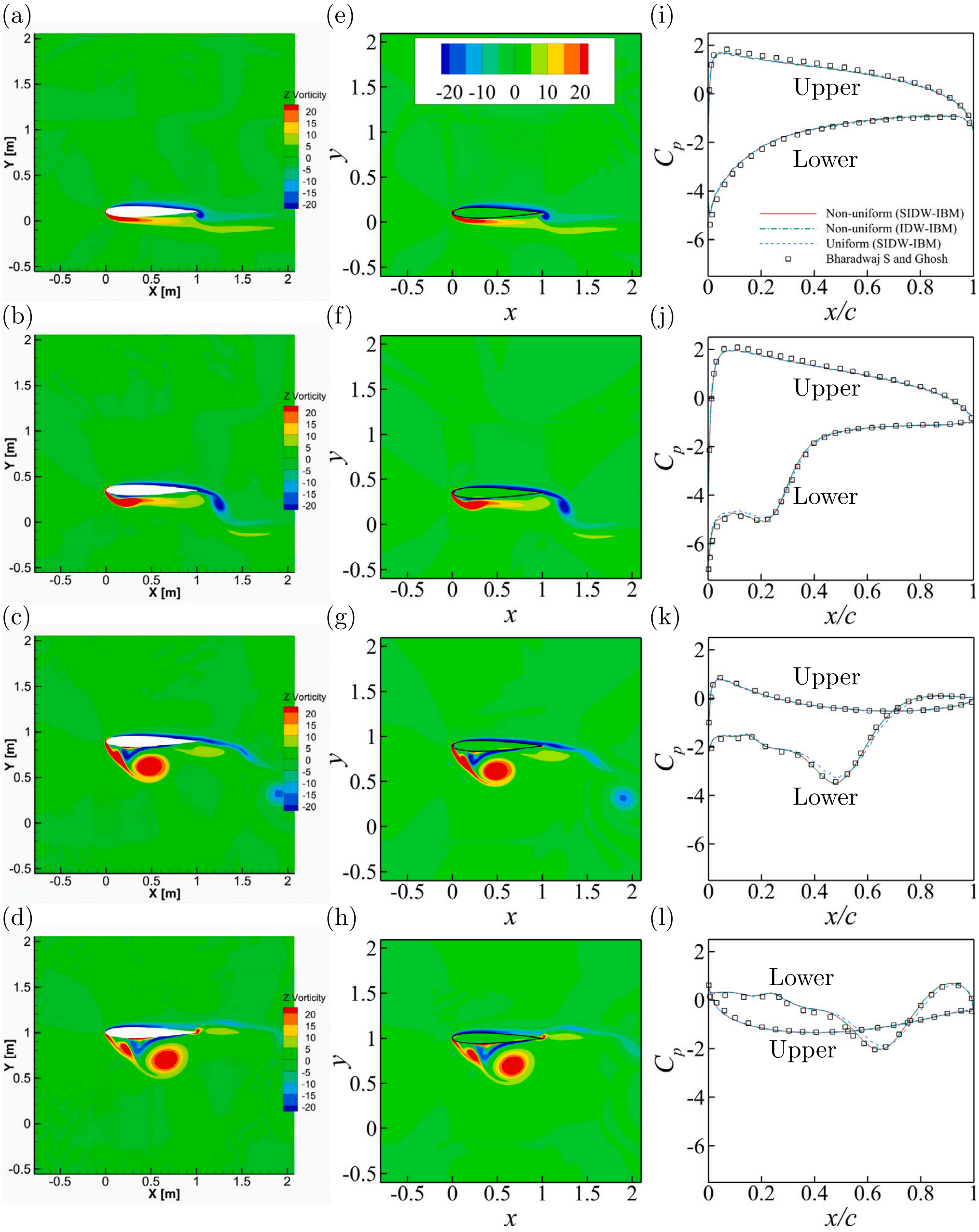

Fig. 8. Instantaneous vorticity contours (the left column is the reference [30] and the middle column is the present result generated by the SIDW-IBM) and the distributions of pressure coefficient C_p (right column) at different time instants for flow past a heaving NACA0012 airfoil at $Re = 1000$. $t = 0.4$ (first row), $t = 0.8$ (second row), $t = 1.6$ (third row), $t = 2.0$ (fourth row).

diameter and the cross-section diameter of the 3D torus, respectively. The geometry of the 3D torus is determined by the aspect ratio $Ar = D/d$, which is set to $Ar = 2$ in this numerical simulation. The Reynolds number is defined as $Re = U_0 d/\nu$, where U_0 denotes the free stream velocity. The flow direction is along with the x-axis.

Firstly, to evaluate the accuracy of the proposed SIDW-IBM for 3D FSI problems, the flow past a stationary torus is conducted at $Re = 40$ using the presented SIDW-IBM, the IDW-IBM, and the multi-direct forcing IBM. The 3D torus interface is evenly discretized by 2016 Lagrangian points. The computational domain size is set to $[-10d, 15d] \times [-10d, 10d] \times [-10d, 10d]$ with a mesh size of $101 \times 121 \times 121$, where the center of the 3D torus is located at the origin $(0, 0, 0)$ and the rectangular region $(1.3d \times 3.6d \times 3.6d)$ around the 3D torus is discretized by a uniform mesh of $27 \times 73 \times 73$. The drag and y-direction force coefficients are defined as

$$C_d = \frac{F_x}{\frac{1}{2}\rho U_0^2 \pi D d}, \quad C_y = \frac{F_y}{\frac{1}{2}\rho U_0^2 \pi D d}. \tag{35}$$

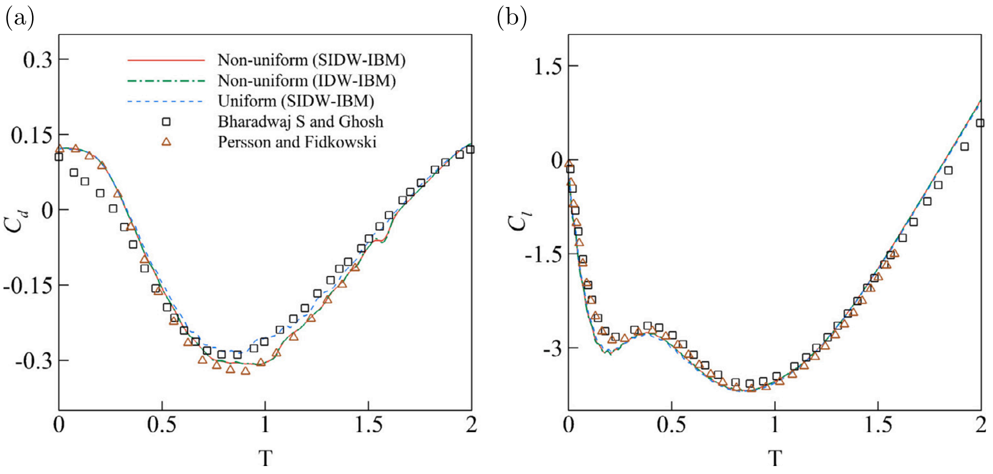

Fig. 9. Time evolutions of the (a) drag and (b) lift coefficients for the heaving NACA0012 airfoil in a uniform flow at $Re = 1000$.

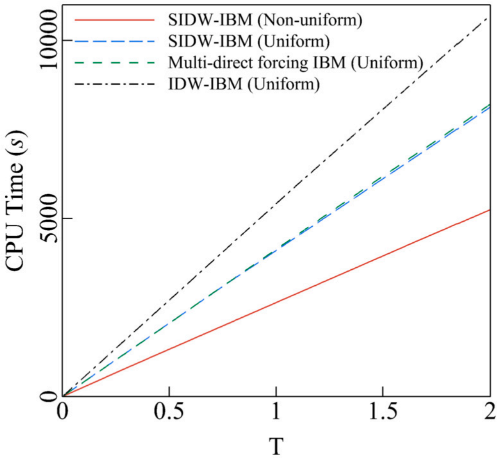

Fig. 10. Computational time consumption of the original implicit IDW-IBM, the multi-direct forcing IBM and the proposed SIDW-IBM with non-uniform mesh and uniform mesh for the heaving NACA0012 airfoil.

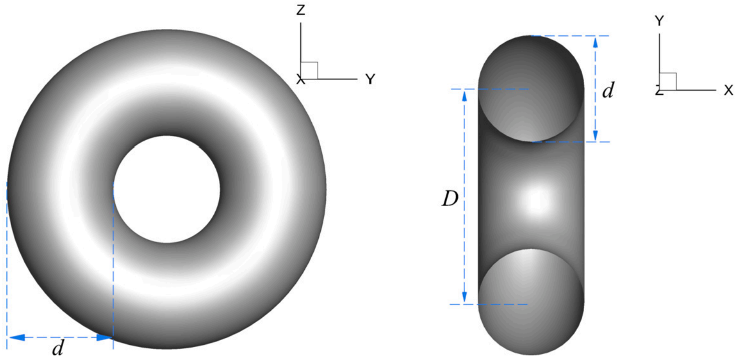

Fig. 11. Schematic view of the 3D torus. Left: front view, right: side view.

Table 6 presents the comparison of the drag coefficient C_d between the proposed SIDW-IBM and previous IBMs. It is evident from Table 6 that the drag coefficient predicted by the SIDW-IBM is consistent with reference data [14,32,33], quantitatively demonstrating the accuracy and capability of the proposed SIDW-IBM in addressing 3D FSI problems with complex geometries. In addition, the SIDW-IBM can provide reasonable force prediction, comparable to the performance of the IDW-IBM and the multi-direct forcing IBM.

To further evaluate the computational efficiency of the proposed SIDW-IBM for complex moving boundaries, the flow past a 3D rotating torus is simulated, where the torus rotates around the z-axis. To accurately capture the fluid dynamics around the rotating

Table 6
Comparison of drag coefficient for the flow past 3D stationary torus at $Re = 40$.

Reference	C_d
Wu and Shu [20]	1.335
Wang et al. [32]	1.318
Yang et al. [33]	1.324
Multi-direct forcing IBM	1.315
IDW-IBM	1.315
SIDW-IBM	1.317

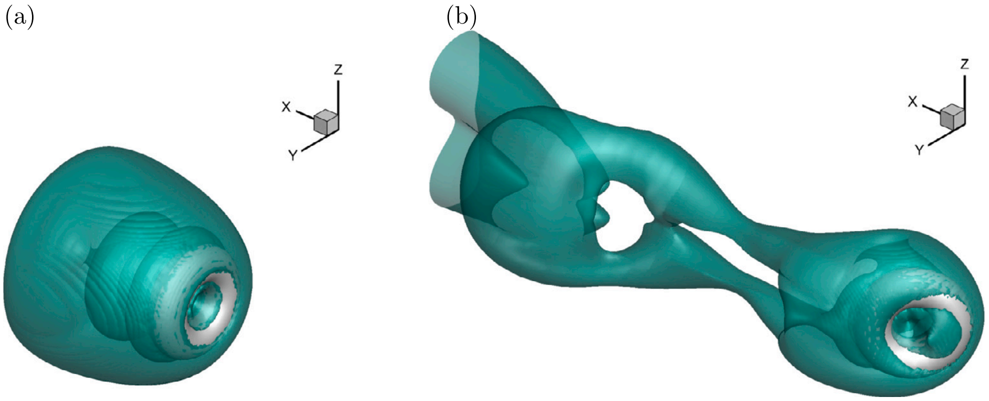

Fig. 12. The vortical structures around (a) the stationary torus and (b) the rotating torus at $Re = 40$.

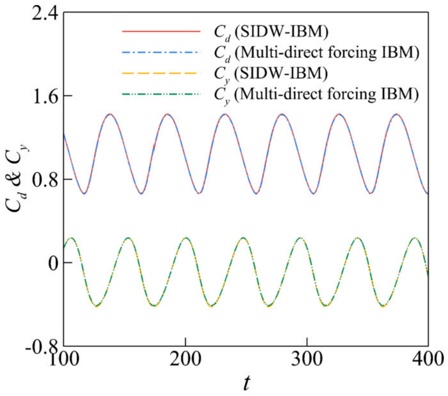

Fig. 13. The drag and y-direction force coefficients of the 3D rotating torus at $Re = 40$ and $\Omega = 0.1$.

Table 7
Comparison of time consumption of the IDW-IBM, the multi-direct forced IBM and the proposed SIDW-IBM for one time step.

Method	IDW-IBM	Multi-direct forcing IBM	SIDW-IBM
CPU time for IBM (s)	144.633	10.390	4.266

torus, the rectangular region ($3.6d \times 3.6d \times 3.6d$) is discretized by using a uniform mesh of $73 \times 73 \times 73$. The dimensionless rotating speed is defined as $\Omega = \omega R/U_0 = 0.1$, where $R = (D + d)/2$. Fig. 12 presents the vortex structures around the 3D stationary and rotating torus. It can be seen that the rotating behavior triggers vortex shedding behind the rotating torus. To assess the accuracy of the force prediction, Fig. 13 presents the comparison of force coefficients between the proposed SIDW-IBM and the multi-direct forcing IBM, where good agreement is observed. Table 7 compares the computational time of different numerical approaches for one time step. It is evident from Table 7 that the proposed SIDW-IBM is more efficient than both the IDW-IBM and the multi-direct forcing IBM, highlighting the efficiency of the proposed SIDW-IBM in simulating 3D FSI problems with moving boundaries.

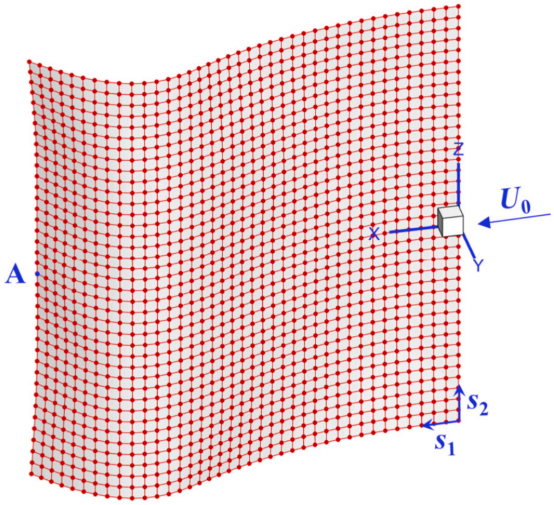

Fig. 14. Schematic view of the 3D flapping flexible plate, in which the leading edge's midpoint is attached at the origin of the computational domain and the blue point A denotes the midpoint on the trailing edge. The s_1 and s_2 directions denote the local Lagrangian coordinate system. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

4.5. A 3D flexible plate flapping in a free stream

In this subsection, to examine the accuracy and capability of the proposed SIDW-IBM in simulating challengeable three-dimensional (3D) FSI problems, flow past a 3D flexible plate is numerically investigated, where complex geometries, moving boundaries and large deformations are involved in this complex system. Fig. 14 shows the sketch of the 3D flexible square plate with length of L, where the 3D flexible plate is uniformly discretized by 1521 Lagrangian points (red points shown in Fig. 14) with Lagrangian point spacing of $L/38$. The flapping motion of the 3D flexible plate is governed by the following equation [34] as:

$$\rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} = \sum_{i,j=1}^{2} \left[\frac{\partial}{\partial s_i} \left(\sigma_{ij} \frac{\partial \mathbf{X}}{\partial s_j} \right) - \frac{\partial^2}{\partial s_i \partial s_j} \left(\kappa_{ij}^{b} \frac{\partial^2 \mathbf{X}}{\partial s_i \partial s_j} \right) \right] + \boldsymbol{F}_f, \tag{36}$$

where $\mathbf{X}$ is the location of Lagrangian points, ρ_s denotes the area density of the 3D flexible plate, and $\boldsymbol{F}_f$ represents the fluid force exerted by the ambient fluid. The definition of σ_{ij} can be expressed as:

$$\sigma_{ij} = \varphi_{ij} \left(\frac{\partial \mathbf{X}}{\partial s_i} \cdot \frac{\partial \mathbf{X}}{\partial s_j} - \Gamma_{ij}^{0} \right), \quad \Gamma_{ij}^{0} = \begin{cases} 1, \mathit{if}\ i = j, \\ 0, \mathit{if}\ i \neq j. \end{cases} \tag{37}$$

φ_{ij} represents the tension ($i = j$) and shearing ($i \neq j$) coefficients, and κ_{ij}^{b} denotes the bending ($i = j$) and twisting ($i \neq j$) coefficients. The solving procedure of this governing equation (36) follows the work of Ma et al. [35], and details on solving Eq. (36) can be found in Appendix B. The fluid structure coupling scheme is described in Appendix C.

In this numerical examination, the dynamic behavior of the 3D flexible plate is governed by the dimensionless group: Reynolds number $Re = U_\infty L/\nu$, the dimensionless stretching and shearing coefficients $\Phi_{ij} = \varphi_{ij}/(\rho_s U_\infty^2)$, the dimensionless bending and twisting coefficients $K_{ij} = \kappa_{ij}^{b}/(\rho_s U_\infty^2 L^2)$, and the mass ratio $\Theta = \rho_s/(\rho L)$, where U_∞ is the free stream velocity. The parameters of $Re = 200$, $\Theta = 1$, $\Phi_{11} = \Phi_{22} = 10^3$, $\Phi_{12} = 10$, and $K_{11} = K_{22} = K_{12} = 10^{-4}$ are employed in this test. The size of the rectangular computational domain is $[-2L, 7L]$, $[-4L, 4L]$, and $[-L, L]$ along the x, y, and z axis, respectively; and it is discretized by a non-uniform mesh. To ensure the numerical accuracy around the immersed object and resolve the dynamic interactions between the 3D flexible plate and the ambient fluids, the sub-domain ($[-0.2L, 1.3L] \times [-0.7L, 0.7L] \times [-0.7L, 0.7L]$) around the 3D flexible plate is discretized by a finer uniform mesh with mesh spacing of $h = L/75$. The leading edge of the 3D flexible plate is pinned and the free moving boundary condition is imposed on the other three edges. The boundary conditions of the computational domain are: the Dirichlet boundary conditions ($u_x = U_\infty$, $u_y = u_z = 0$) for the inflow boundary and the Neumann boundary conditions ($\partial \mathbf{u}/\partial n = 0$) for the outflow boundary and the four sidewalls.

The time evolutions of the transverse displacement of the midpoint at the trailing edge, the drag and lift coefficients are presented in Fig. 15, where the results predicted by the proposed SIDW-IBM are in a good agreement with those in the reference [34–37]. It demonstrates that the proposed SIDW-IBM is able to provide accurate solutions and capture nonlinear dynamic behaviors for this complex FSI system. In addition, the statistical results of flapping amplitude and Strouhal number are tabulated in Table 8, and the results predicted by the SIDW-IBM agree well with previous studies [34–38].

As shown in Fig. 16, the evolutions of the instantaneous vortical structures around the 3D flexible plate clearly show that the vortices are shedded from the trailing edge and subsequently merged with the vortices from the side edges. Consequently, a hairpin-like vortical structure is generated at the downstream of the 3D flexible flapping plate. These vortical evolutions are aligned with

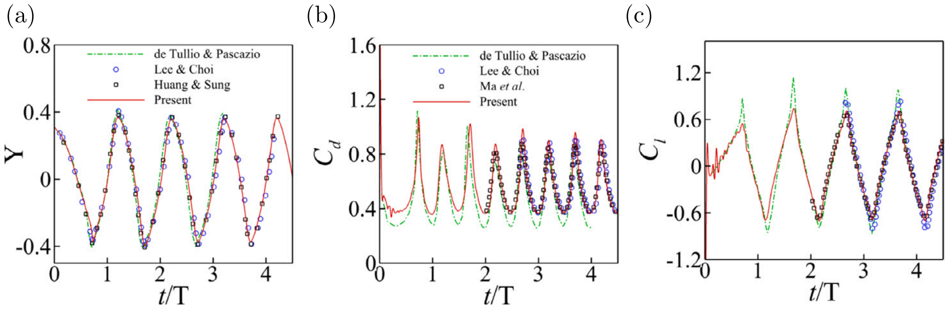

Fig. 15. (a) Comparisons of (a) the transverse displacement of the midpoint at the trailing edge (point A in Fig. 14), (b) the drag coefficient and (c) the lift coefficient, where Figs. 15(b) and 15(c) share the same legend.

Table 8
Statistical results of the flapping amplitude A/L and the Strouhal number $St = fL/U_0$, where f represents the flapping frequency.

$Re = 200$	Amplitude A/L	Strouhal number St
de Tullio and Pascazio [37]	0.795	0.265
Tian et al. [38] - Flag 2	0.806	0.266
Huang and Sung [34]	0.780	0.260
Lee and Choi [36]	0.752	0.265
Ma et al. [35]	0.736	/
Present	0.736	0.262

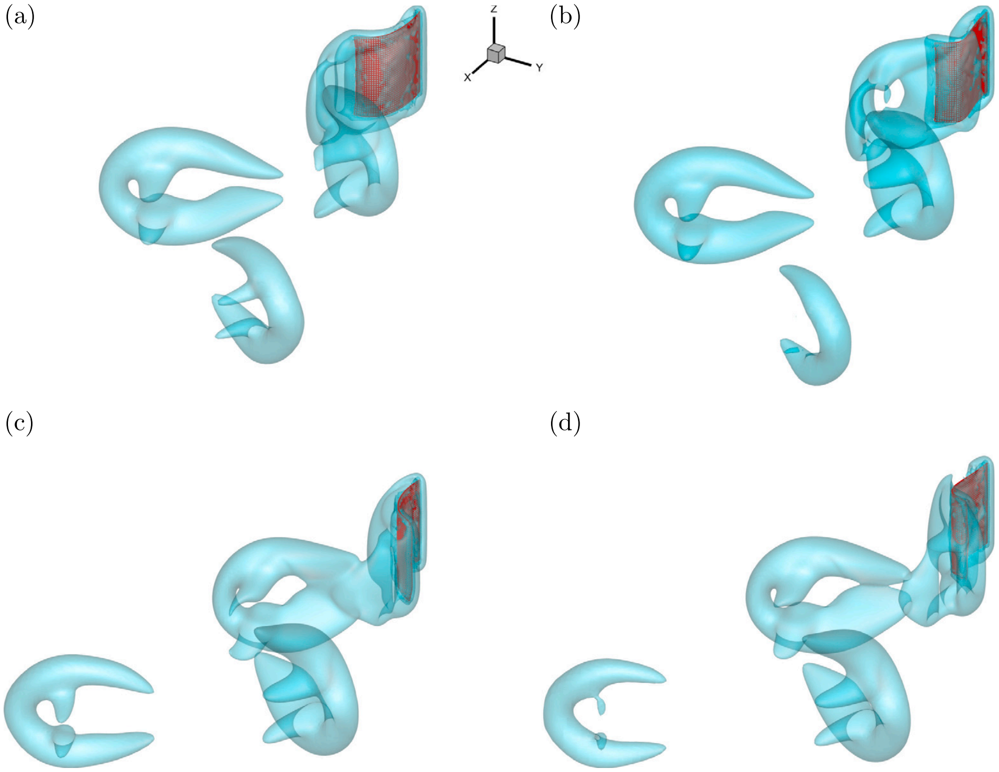

Fig. 16. The evolutions of the instantaneous vortical structures around the 3D flexible plate, where the vortical structures are generated by the Q-criterion of $Q = 0.2$.

the phenomena reported in the reference [34,36–38]. The numerical results qualitatively and quantitatively demonstrate that the proposed SIDW-IBM is capable of capturing the nonlinear dynamic behaviors in a fully coupled FSI system and providing accurate solutions to characterize the unsteady aerodynamic forces.

5. Conclusions

In this work, the simplified inverse distance weighting-immersed boundary method is proposed to efficiently simulate FSI problems with moving boundaries and large deformations, which eliminates the large memory space requirement and high computational cost for constructing the correlation matrix and its inversion in the original implicit IDW-IBM. The SIDW-IBM is developed based on a second order approximation derived through error analysis using Taylor series expansion. Based on this second order approximation, the square correlation matrix $\boldsymbol{C}_{N_L \times N_L}$ in the original implicit IDW-IBM can be approximated into a diagonal matrix, and it relieves the computational cost for matrix inversion and the memory space for matrix storage. In addition, by incorporating the inverse distance weighting interpolation scheme, the proposed SIDW-IBM can be implemented on the non-uniform mesh to further promote the computational efficiency.

The SIDW-IBM integrated with the flow solver (RLBFS) has been employed to solve some classic but challengeable FSI problems with moving boundaries and large deformations. The overall accuracy test demonstrates that the global spatial accuracy is close to second-order and the proposed SIDW-IBM has negligible effects on the global accuracy. The numerical results generated by the SIDW-IBM are in a good agreement with the previous studies, showing the capability and feasibility of the proposed SIDW-IBM in solving complex FSI problems. Moreover, the boundary velocity errors within the proposed SIDW-IBM are negligible, indicating that the no-slip boundary condition is accurately enforced by the SIDW-IBM and the second order approximation has negligible effects on the numerical solutions. Most importantly, the numerical results calculated by the SIDW-IBM on the non-uniform mesh agree well with those generated on the uniform mesh and the reference data, and the computational efficiency is significantly improved by the SIDW-IBM on the non-uniform mesh. Finally, the proposed SIDW-IBM is employed to solve nonlinear dynamic behaviors of the 3D flexible flapping plate in a uniform flow, which demonstrates the suitability and feasibility of the SIDW-IBM for simulating 3D FSI problems with large deformations. In conclusion, the numerical results demonstrate that the proposed SIDW-IBM is much more efficient than the original implicit IDW-IBM and exhibits significant potentials for simulating complex multi-scale FSI systems on non-uniform meshes.

CRediT authorship contribution statement

Buchen Wu: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yinjie Du:** Writing – original draft, Methodology, Formal analysis, Conceptualization. **Chang Shu:** Supervision, Resources, Project administration, Funding acquisition, Formal analysis.

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.

Acknowledgements

This work has been supported by National University of Singapore (Grant A-0005235-01-00).

Appendix A. Comparisons with existing IBMs for incompressible flows

A.1. The penalty forcing IBM

The penalty forcing IBM is firstly developed by Peskin [3], where massive and massless material points are restrained to keep close enough through a stiff spring. Then, Huang et al. [5] proposed an improved penalty forcing IBM, in which massive material points are employed to represent an elastic boundary. Therefore, the Lagrangian body force exerted on massless material points can be obtained through

$$\mathbf{F}_b = -\kappa \left[\left(\mathbf{X}_{ib} - \mathbf{X} \right) + \Delta t \left(\mathbf{U}_{ib} - \mathbf{U} \right) \right], \tag{A.1}$$

where κ denotes a large constant. $\mathbf{X}_{ib}$ and $\mathbf{X}$ represent the position of the massless and massive material points, respectively. $\mathbf{U}_{ib}$ and $\mathbf{U}$ denote the velocity of the massless and massive material points, respectively. Once the Lagrangian restoring force is obtained, it can be transferred to the background Eulerian points through

$$\mathbf{f}(\mathbf{x}) = \int\limits_{\Omega_s} \mathbf{F}_b D(\mathbf{x} - \mathbf{X}) d s. \tag{A.2}$$

The penalty forcing IBM has to introduce an artificial parameter κ to pre-calculate the body force, which may lead to non-physical flow leakage at the immersed boundary because the no-slip boundary condition is not accurately satisfied.

A.2. The direct forcing IBM

The direct forcing IBM is originally proposed by Fadlun et al. [6]. Subsequently, Uhlmann [8] combined the direct forcing IBM with the regularized delta function. The intermediate velocity at Lagrangian points can be obtained through

$$\mathbf{u}_b^{*,i} = \sum_{j=1}^{N_E} \mathbf{u}^{*,j} D_{ij}\left(\mathbf{X}_i - \mathbf{x}_j\right) h^2, \tag{A.3}$$

where $D_{ij}\left(\mathbf{X}_i - \mathbf{x}_j\right)$ denotes the smooth function and is defined as

$$D_{ij}\left(\mathbf{X}_i - \mathbf{x}_j\right) = \frac{1}{h^2}\phi\left(\frac{X_i - x_j}{h}\right)\phi\left(\frac{Y_i - y_j}{h}\right). \tag{A.4}$$

The four-point delta function is widely employed in many IBMs, which can be written as

$$\phi(r) = \begin{cases} \left[3 - 2|r| + \sqrt{1 + 4|r| - 4|r|^2}\right]/8, & 0 \le |r| < 1, \\ \left[5 - 2|r| - \sqrt{-7 + 12|r| - 4|r|^2}\right]/8, & 1 \le |r| < 2, \\ 0, & |r| \ge 2. \end{cases} \tag{A.5}$$

The Lagrangian restoring forces are directly evaluated through the difference between the intermediate velocity $\mathbf{u}_b^*$ and the physical velocity $\boldsymbol{U}_{body}$ at the Lagrangian point, which can be expressed as

$$\mathbf{F}_b^i = \rho \frac{\boldsymbol{U}_{body}^i - \mathbf{u}_b^{*,i}}{\Delta t}. \tag{A.6}$$

Once the Lagrangian body forces are obtained, the Eulerian restoring forces can be calculated through the following spreading scheme as

$$\mathbf{f}^j = \sum_{i=1}^{N_L} D_{ji}\left(\mathbf{x}_j - \mathbf{X}_i\right)\mathbf{F}_b^i \delta s^i. \tag{A.7}$$

Based on the methodology of the direct forcing IBM, the errors generated by the inequality between the interpolation and spreading processes cannot be eliminated [39]. Consequently, the direct forcing IBM cannot accurately enforce the no-slip boundary condition on the immersed boundary [39].

A.3. The moving-least-squares based direct forcing IBM

To accurately interpolate the flow variables at Lagrangian points, Coclite et al. [40] combined the moving-least-squares (MLS) reconstruction technique with the original direct forcing IBM. The MLS reconstruction operator can be expressed as

$$\mathbf{u}_b^{*,i} = \mathbf{\Phi}^T \mathbf{u}^* = \sum_{j=1}^{N_E} \phi_i^j \mathbf{u}^{*,j}, \tag{A.8}$$

where $\mathbf{\Phi}^T$ denotes the transfer operator, which can describe the shape function values of the i-th Lagrangian point. The definition of the transfer operator can be expressed as

$$\mathbf{\Phi}^T = \boldsymbol{p}^T(\mathbf{X})\boldsymbol{A}^{-1}(\mathbf{X})\boldsymbol{B}(\mathbf{X}), \tag{A.9a}$$

$$\boldsymbol{A}(\mathbf{X}) = \sum_{j=1}^{N_E} W\left(\mathbf{X} - \mathbf{x}^j\right)\boldsymbol{p}\left(\mathbf{x}^j\right)\boldsymbol{p}^T\left(\mathbf{x}^j\right), \tag{A.9b}$$

$$\boldsymbol{B}(\mathbf{X}) = \left[W\left(\mathbf{X} - \mathbf{x}^1\right)\boldsymbol{p}\left(\mathbf{x}^1\right) \cdots W\left(\mathbf{X} - \mathbf{x}^{N_E}\right)\boldsymbol{p}\left(\mathbf{x}^{N_E}\right)\right], \tag{A.9c}$$

$$\boldsymbol{p}^T(\mathbf{X}) = [1, X, Y, Z], \quad W\left(\mathbf{X} - \mathbf{x}^j\right) = \begin{cases} e^{-\left(r_j/\varepsilon\right)^2}, & r_j \le 1; \\ 0, & r_j > 1. \end{cases} \tag{A.9d}$$

Once the flow variables at Lagrangian points are obtained, the Lagrangian body forces can be evaluated through Eq. (A.6). Subsequently, the Lagrangian restoring forces are distributed on the surrounding Eulerian grid points by the following spreading scheme as

$$\mathbf{f}^j = \sum_{i=1}^{N_L} c^i \phi_i^j \mathbf{F}_b^i, \tag{A.10}$$

where c^i denotes the scaling factor to ensure the conservation of force and torque. Although the MLS approach can promote the accuracy of reconstruction, this explicit direct forcing scheme also contains the errors induced by the inequality between the interpolation and spreading processes.

A.4. The multi-direct forcing IBM

To reduce the boundary errors within the direct forcing IBM, Luo et al. [9] proposed the multi-direct forcing IBM through multiple implementing the direct forcing IBM. The calculation procedure of the multi-direct forcing IBM can be summarized as follows

(1) Initialize the iteration number $k = 0$ and $\mathbf{u}^{*(0)} = \mathbf{u}^*$.

(2) Calculate the Lagrangian restoring forces on the immersed boundary by using $\mathbf{u}^{*(k)}$ as

$$\mathbf{F}_b^{(k)} = \rho \frac{\boldsymbol{U}_{body} - \mathbf{u}_b^{*(k)}}{\Delta t}. \tag{A.11}$$

(3) Spread the Lagrangian restoring forces to the surrounding Eulerian points as

$$\mathbf{f}^j = \sum_{i=1}^{N_L} D_{ji} \mathbf{F}_b^i \delta s^i. \tag{A.12}$$

(4) Distribute the Eulerian restoring forces to correct the flow field as

$$\mathbf{u}^{*(k+1)} = \mathbf{u}^{*(k)} + \frac{\mathbf{f}\Delta t}{\rho}. \tag{A.13}$$

(5) Repeat steps 2-4 until the numerical solutions achieving the convergence standard as

$$\varsigma = \frac{1}{U_{ref}} \sqrt{\frac{1}{N_L} \sum_{i=1}^{N_L} \left| \boldsymbol{U}_{body} - \mathbf{u}_b^{*(k)} \right|^2} < \varsigma_c, \tag{A.14}$$

where U_{ref} and ς_c denote the reference velocity and convergence criterion, respectively.

Although the multi-direct forcing IBM can accurately achieve the no-slip boundary condition through an iterative process, it consumes significantly more computational time for FSI problems with multiple objects and moving boundaries.

A.5. The feedback forcing IBM

The feedback forcing IBM is proposed by Goldstein et al. [41]. The Lagrangian restoring forces can be evaluated through

$$\mathbf{F}_b = \alpha \int_0^t \left(\boldsymbol{U}_{body} - \mathbf{u}_b^* \right) dt + \beta \left(\boldsymbol{U}_{body} - \mathbf{u}_b^* \right), \tag{A.15}$$

where α and β are artificial coefficients [42]. Subsequently, the Lagrangian restoring forces are distributed on the Eulerian grids through the same spreading scheme in Eq. (A.7). Similar to the penalty forcing IBM, the artificial parameters in the feedback forcing IBM may lead to boundary velocity errors.

A.6. The implicit velocity correction IBM

The implicit velocity correction IBM is proposed by Wu and Shu [14], where the velocity corrections are implicitly resolved to accurately enforce the no-slip boundary condition. The velocity corrections at Lagrangian points can be spread to the surrounding Eulerian grids through

$$\Delta \mathbf{u}^j = \sum_{i=1}^{N_L} D_{ji} \delta \mathbf{u}_b^i \delta s^i. \tag{A.16}$$

Substituting Eqs. (A.3) and (A.16) into (10), the following linear system can be constructed as

$$U_{body}^i - \sum_j \mathbf{u}^{*,j} D_{ij} h^2 = \sum_j D_{ij} \left(\sum_m D_{jm} \delta \mathbf{u}_b^m \delta s^m \right) h^2. \tag{A.17}$$

By solving the above linear system, the velocity corrections can be implicitly obtained.

A.7. The explicit velocity correction IBM

The explicit velocity correction IBM is proposed by Wu et al. [43], where the linear system in the implicit velocity correction IBM is assumed to be a diagonally dominant banded sparse matrix. The methodology of the explicit velocity correction IBM is that each Lagrangian point only affects its surrounding Lagrangian points within the delta function region. Therefore, the linear system of Eq. (A.17) can be rewritten as

$$U^{i}_{body} - \sum_{j} \mathbf{u}^{*,j} D_{ij} h^2 = \sum_{m=i-2}^{i+2} A_{im} \delta \mathbf{u}_b^m \delta s^m, \tag{A.18}$$

where $A_{im} = \sum_j D_{ij} D_{jm}$. Cai et al. [44] demonstrated through numerical experiments that the difference between each Lagrangian points and its surrounding Lagrangian points within the delta function region is extremely small. Consequently, Eq. (A.18) can be approximately rewritten as

$$U^{i}_{body} - \sum_{j} \mathbf{u}^{*,j} D_{ij} h^2 = \delta \mathbf{u}_b^i \delta s^i \sum_{m=i-2}^{i+2} A_{im}. \tag{A.19}$$

Appendix B. Solving procedure of 3D flexible plate

In the present work, the 3D flexible plate is decomposed into multiple parallel flexible filaments along the streamwise direction, with these filaments connected to adjacent filaments in the spanwise direction. Therefore, 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 $\frac{\partial}{\partial s_1}\left(\sigma_{11} \frac{\partial \mathbf{X}}{\partial s_1}\right)$ are regarded as the internal forces of each flexible filament; while the other force terms can be categorized into external forces. In addition, the predicted position $\mathbf{X}^* = 2\mathbf{X}^n - \mathbf{X}^{n-1}$ is employed to calculate the external forces. Consequently, the Eq. (36) can be rewritten as

$$\rho_s \frac{\partial^2 \mathbf{X}}{\partial t^2} = \frac{\partial}{\partial s_1}\left(T \frac{\partial \mathbf{X}}{\partial s_1}\right) - \frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^b \frac{\partial^2 \mathbf{X}}{\partial s_1^2}\right) + \mathbf{F}_f + \mathbf{F}_e\left(\mathbf{X}^*\right), \tag{B.1}$$

where $T = \sigma_{11}$ denotes the tension force along the filament. $\mathbf{F}_e(\mathbf{X}^*)$ represents the summation of external forces. Following the previous work [34,35], the external forces can be calculated through
for $i = j = 2$,

$$\frac{\partial}{\partial s_2}\left(\sigma_{22} \frac{\partial \mathbf{X}^*}{\partial s_2}\right) = D_2^-\left(\sigma_{22} D_2^+ \mathbf{X}^*\right), \tag{B.2}$$

$$\frac{\partial^2}{\partial s_2^2}\left(\kappa_{22}^b \frac{\partial^2 \mathbf{X}^*}{\partial s_2^2}\right) = D_{22}^0\left(\kappa_{22}^b D_{22}^0 \mathbf{X}^*\right); \tag{B.3}$$

for $i \neq j$,

$$\frac{\partial}{\partial s_1}\left(\sigma_{12} \frac{\partial \mathbf{X}^*}{\partial s_2}\right) = \frac{1}{2}\left[D_1^-\left(\sigma_{12}^+ D_2^+ \mathbf{X}^*\right) + D_1^-\left(\sigma_{12}^- D_2^- \mathbf{X}^*\right)\right], \tag{B.4}$$

$$\frac{\partial}{\partial s_2}\left(\sigma_{21} \frac{\partial \mathbf{X}^*}{\partial s_1}\right) = \frac{1}{2}\left[D_2^-\left(\sigma_{21}^+ D_1^+ \mathbf{X}^*\right) + D_2^+\left(\sigma_{21}^- D_1^+ \mathbf{X}^*\right)\right], \tag{B.5}$$

$$\frac{\partial^2}{\partial s_1 s_2}\left(\kappa_{12}^b \frac{\partial^2 \mathbf{X}^*}{\partial s_1 s_2}\right) = \frac{\partial^2}{\partial s_2 s_1}\left(\kappa_{21}^b \frac{\partial^2 \mathbf{X}^*}{\partial s_2 s_1}\right) = D_{12}^-\left(\kappa_{12}^b D_{12}^+ \mathbf{X}^*\right); \tag{B.6}$$

where

$$\begin{aligned}
\left[D_1^+ \phi\right]_{i,j} &= \left(\phi_{i+1,j} - \phi_{i,j}\right)/\Delta s_1,\\
\left[D_1^- \phi\right]_{i,j} &= \left(\phi_{i,j} - \phi_{i-1,j}\right)/\Delta s_1,\\
\left[D_2^+ \phi\right]_{i,j} &= \left(\phi_{i,j+1} - \phi_{i,j}\right)/\Delta s_2,\\
\left[D_2^- \phi\right]_{i,j} &= \left(\phi_{i,j} - \phi_{i,j-1}\right)/\Delta s_2,\\
\left[D_{22}^0 \phi\right]_{i,j} &= \left(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}\right)/\Delta s_2^2,\\
\left[D_{12}^+ \phi\right]_{i,j} &= \left(\phi_{i+1,j+1} - \phi_{i+1,j} - \phi_{i,j+1} + \phi_{i,j}\right)/\left(\Delta s_1 \Delta s_2\right),\\
\left[D_{12}^- \phi\right]_{i,j} &= \left(\phi_{i,j} - \phi_{i,j-1} - \phi_{i-1,j} + \phi_{i-1,j-1}\right)/\left(\Delta s_1 \Delta s_2\right).
\end{aligned} \tag{B.7}$$

In the present work, the flexible filaments are considered to be inextensible along s_1 axis, and thus the tension force can be calculated by solving the Poisson equation as

$$\begin{aligned}\frac{\partial \mathbf{X}}{\partial s_1}\cdot\frac{\partial^2}{\partial s_1^2}\left(T\frac{\partial \mathbf{X}}{\partial s_1}\right) &= \frac{1}{2}\frac{\partial^2}{\partial t^2}\left(\frac{\partial \mathbf{X}}{\partial s_1}\cdot\frac{\partial \mathbf{X}}{\partial s_1}\right)\\ &- \frac{\partial^2 \mathbf{X}}{\partial t\partial s_1}\cdot\frac{\partial^2 \mathbf{X}}{\partial t\partial s_1} - \frac{\partial \mathbf{X}}{\partial s_1}\cdot\frac{\partial}{\partial s_1}\left[\frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^{b}\frac{\partial^2 \mathbf{X}}{\partial s_1^2}\right)+\mathbf{F}_f+\mathbf{F}_e\right].\end{aligned} \tag{B.8}$$

Through employing the implicit second-order finite-difference scheme, the discretization form of Eq. (B.1) can be expressed as

$$\begin{aligned}&\frac{\mathbf{X}^{n+1}-2\mathbf{X}^{n}+\mathbf{X}^{n-1}}{\Delta t^2}\\ &= \frac{\partial}{\partial s_1}\left(T^{n+\frac{1}{2}}\frac{\partial \mathbf{X}^{n+1}}{\partial s_1}\right)-\frac{\partial^2}{\partial s_1^2}\left(\kappa_{11}^{b}\frac{\partial^2 \mathbf{X}^{*}}{\partial s_1^2}\right)+\mathbf{F}_f+\mathbf{F}_e\left(\mathbf{X}^{*}\right),\end{aligned} \tag{B.9}$$

where $T^{n+\frac{1}{2}}$ can be obtained through solving Eq. (B.8). Therefore, the position $\mathbf{X}^{n+1}$ of filaments can be updated by solving Eq. (B.9).

Appendix C. Fluid-plate interaction process

In the present work, a weak coupling approach is employed to solve the FSI coupling process, where the fluid flow and the structural dynamics of 3D flexible plate are solved sequentially in each time step [45]. The overall computational sequences can be concluded as follows

(1) Initialize the fluid flow and the 3D flexible plate, where the fluid force exerted on the 3D flexible structure is set as $\mathbf{F}_f = 0$.
(2) Solve Eq. (36) for the updated position $\mathbf{X}^{n+1}$ and the velocity $\boldsymbol{U}_{body}^{n+1}$ of each Lagrangian point.
(3) Predict the intermediate flow variables ρ^{n+1} and $\mathbf{u}^{*}$ by solving Eq. (2a) using the RLBFS.
(4) Evaluate the velocity corrections to update $\mathbf{u}^{n+1}$ and the fluid force $\mathbf{F}_b^{n+1} = \rho^{n+1}\delta\mathbf{u}_b/\Delta t$ by the proposed SIDW-IBM in Section 3.
(5) Calculate the fluid force exerted on the 3D flexible plate $\mathbf{F}_f = \mathbf{F}_b^{n+1}$.
(6) Repeat steps 2-5 for achieving final solutions.

Data availability

Data will be made available on request.

References

[1] Y. Du, L. Yang, C. Shu, Y. Xiao, Y. Song, Inverse distance weighting interpolation-based immersed boundary velocity correction method for incompressible flows, Phys. Fluids 35 (2023).
[2] J. Lu, H. Lei, C. Dai, L. Yang, C. Shu, Analyses and reconstruction of the lattice Boltzmann flux solver, J. Comput. Phys. (2022) 110923.
[3] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[4] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[5] W.-X. Huang, C.B. Chang, H.J. Sung, An improved penalty immersed boundary method for fluid–flexible body interaction, J. Comput. Phys. 230 (2011) 5061–5079.
[6] 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 (2000) 35–60.
[7] Z.-G. Feng, E.E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (2005) 20–51.
[8] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448–476.
[9] K. Luo, Z. Wang, J. Fan, K. Cen, Full-scale solutions to particle-laden flows: multidirect forcing and immersed boundary method, Phys. Rev. E 76 (2007) 066709.
[10] Z. Wang, J. Fan, K. Luo, Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles, Int. J. Multiph. Flow 34 (2008) 283–302.
[11] 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 (2006) 173–182.
[12] 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 (2014) 1039–1056.
[13] C. Shu, N. Liu, Y.-T. Chew, A novel immersed boundary velocity correction–lattice Boltzmann method and its application to simulate flow past a circular cylinder, J. Comput. Phys. 226 (2007) 1607–1622.
[14] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (2009) 1963–1979.
[15] 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.
[16] B. Wu, J. Lu, H. Lee, C. Shu, M. Wan, An efficient explicit immersed boundary-reconstructed lattice Boltzmann flux solver for isothermal fluid-structure interaction problems with large deformations and complex geometries, Appl. Math. Model. 114 (2023) 627–645.
[17] S. Tenneti, R. Garg, S. Subramaniam, Drag law for monodisperse gas–solid systems using particle-resolved direct numerical simulation of flow past fixed assemblies of spheres, Int. J. Multiph. Flow 37 (2011) 1072–1092.
[18] M. Vanella, T. Fitzgerald, S. Preidikman, E. Balaras, B. Balachandran, Influence of flexibility on the aerodynamic performance of a hovering wing, J. Exp. Biol. 212 (2009) 95–105.
[19] A.G. Kidanemariam, M. Uhlmann, Direct numerical simulation of pattern formation in subaqueous sediment, J. Fluid Mech. 750 (2014) R2.
[20] J. Wu, C. Shu, An improved immersed boundary-lattice Boltzmann method for simulating three-dimensional incompressible flows, J. Comput. Phys. 229 (2010) 5022–5042.

[21] W. Ren, C. Shu, J. Wu, W. Yang, Boundary condition-enforced immersed boundary method for thermal flow problems with Dirichlet temperature condition and its applications, Comput. Fluids 57 (2012) 40–51.

[22] Y. Sun, C. Shu, Y. Wang, C. Teo, Z. Chen, An immersed boundary-gas kinetic flux solver for simulation of incompressible flows, Comput. Fluids 142 (2017) 45–56.

[23] 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.

[24] D.-J. Chen, K.-H. Lin, C.-A. Lin, Immersed boundary method based lattice Boltzmann method to simulate 2d and 3d complex geometry flows, Int. J. Mod. Phys. C 18 (2007) 585–594.

[25] K. Luo, Z. Zhuang, J. Fan, N.E.L. Haugen, A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions, Int. J. Heat Mass Transf. 92 (2016) 708–717.

[26] F. Örley, V. Pasquariello, S. Hickel, N.A. Adams, Cut-element based immersed boundary method for moving geometries in compressible liquid flows with cavitation, J. Comput. Phys. 283 (2015) 1–22.

[27] S.-W. Su, M.-C. Lai, C.-A. Lin, An immersed boundary technique for simulating complex flows with rigid boundary, Comput. Fluids 36 (2007) 313–324.

[28] C.-C. Liao, Y.-W. Chang, C.-A. Lin, J. McDonough, Simulating flows with moving rigid boundary using immersed-boundary method, Comput. Fluids 39 (2010) 152–167.

[29] A. Pinelli, I. Naqavi, U. Piomelli, J. Favier, Immersed-boundary methods for general finite-difference and finite-volume Navier–Stokes solvers, J. Comput. Phys. 229 (2010) 9073–9091.

[30] A. Bharadwaj S, S. Ghosh, Data reconstruction at surface in immersed-boundary methods, Comput. Fluids 196 (2020) 104236.

[31] P.-O. Persson, C. Fidkowski, Test case cl1–heaving and pitching airfoil, in: 5th International Workshop on High-Order CFD Methods, 2018.

[32] 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.

[33] 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 (2017) 083605.

[34] W.-X. Huang, H.J. Sung, Three-dimensional simulation of a flapping flag in a uniform flow, J. Fluid Mech. 653 (2010) 301.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] M. Jiang, Z. Liu, A boundary thickening-based direct forcing immersed boundary method for fully resolved simulation of particle-laden flows, J. Comput. Phys. 390 (2019) 203–231.

[40] A. Coclite, S. Ranaldo, M. De Tullio, P. Decuzzi, G. Pascazio, Kinematic and dynamic forcing strategies for predicting the transport of inertial capsules via a combined lattice Boltzmann–immersed boundary method, Comput. Fluids 180 (2019) 41–53.

[41] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (1993) 354–366.

[42] L. Ménez, P. Parnaudeau, M. Beringhier, E.G. Da Silva, Assessment of volume penalization and immersed boundary methods for compressible flows with various thermal boundary conditions, J. Comput. Phys. 493 (2023) 112465.

[43] B. Wu, J. Lu, H. Lee, C. Shu, M. Wan, An explicit immersed boundary-reconstructed thermal lattice Boltzmann flux solver for thermal–fluid-structure interaction problems, Int. J. Mech. Sci. 235 (2022) 107704.

[44] Y. Cai, S. Li, J. Lu, An improved immersed boundary-lattice Boltzmann method based on force correction technique, Int. J. Numer. Methods Fluids 87 (2018) 109–133.

[45] Y. Wang, C. Shu, T. Wang, P. Valdivia y Alvarado, A generalized minimal residual method-based immersed boundary-lattice Boltzmann flux solver coupled with finite element method for non-linear fluid-structure interaction problems, Phys. Fluids 31 (2019).