Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 5045–5071
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# The immersed interface method for simulating prescribed motion of rigid objects in an incompressible viscous flow

Sheng Xu *

*Department of Mathematics, Southern Methodist University, 3200 Dyer Street, P.O. Box 750156, Dallas, TX 75275-0156, USA*

Received 9 July 2007; received in revised form 26 December 2007; accepted 20 January 2008
Available online 9 February 2008

**Abstract**

In the immersed interface method, a boundary immersed in a fluid is represented as a singular force in the Navier–Stokes equations. This paper presents an explicit approach for computing the singular force to enforce prescribed motion of a rigid boundary in an incompressible viscous flow. The tangential component of the singular force is related to the surface vorticity and is calculated from the normal derivative of the velocity. The normal component of the singular force is determined from a predictor and a corrector. The predictor uses the normal derivative of the vorticity. The corrector superposes a homogeneous solution to the pressure Poisson equation to achieve the desired normal derivative of the pressure. In the current immersed interface method, the velocity and the pressure are solved using the MAC scheme with the incorporation of jump conditions induced by the singular force and a discontinuous finite body force. The body force is applied to obtain the rigid motion of the fluid enclosed by the boundary. Circular Couette flow, flow past a cylinder, and flow around flappers are simulated to test the accuracy, stability, and efficiency of the method as well as the effect of the corrector. With no stiff springs to model rigid boundaries, the method is stable at relatively high Reynolds numbers.


*Keywords:* The immersed interface method; The immersed boundary method; Complex moving geometries; Flow around multiple objects; Singular forces; Jump conditions; Poisson solvers

## 1. Introduction

The immersed interface method was first proposed by LeVeque and Li [17,18] with the motivation to improve the accuracy of Peskin's immersed boundary method [25,26]. Both methods have been developed in various aspects and applied in various problems since their first appearance, as summarized in Peskin's overview paper [27] and a recent book by Li and Ito [20]. The current author's previous papers on the immersed interface method [36–38] provide some detailed comparisons between the two methods along with extensive citations.

* Tel.: +1 214 768 2985.
*E-mail address:* sxu@smu.edu.
*URL:* http://faculty.smu.edu/sxu.

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.01.053

When applied to flow simulation, the immersed interface method shares the same mathematical formulation as the immersed boundary method. In particular, the boundary of an immersed object is formulated as a singular force in the incompressible Navier–Stokes equations. For the 2D case shown in Fig. 1, the mathematical formulation reads

$$\frac{\partial \vec{v}}{\partial t} + \nabla \cdot (\vec{v}\vec{v}) = -\nabla p + \frac{1}{Re}\Delta \vec{v} + \int_{\Gamma} \vec{f}(\alpha, t)\delta(\vec{x} - \vec{X}(\alpha, t))\mathrm{d}\alpha + \vec{b}, \tag{1}$$

$$\Delta p = s_p + \nabla \cdot \left( \int_{\Gamma} \vec{f}(\alpha, t)\delta(\vec{x} - \vec{X}(\alpha, t))\mathrm{d}\alpha + \vec{b} \right), \tag{2}$$

where $\vec{v} = (u, v)$ is the velocity, $p$ is the pressure, $Re$ is the Reynolds number, $\Gamma$ is the immersed boundary parametrized by the Lagrangian parameter $\alpha$, $\vec{f} = (f_x, f_y)$ is the density of the singular force, $\delta(\cdot)$ is the 2D Dirac $\delta$ function, $\vec{x} = (x, y)$ is Cartesian coordinates, $\vec{X} = (X, Y)$ is the Cartesian coordinates of the boundary, and $\vec{b} = (b_x, b_y)$ is a finite body force. The term $s_p$ is

$$s_p = -\left( \frac{\partial D}{\partial t} + \nabla \cdot (2\vec{v}D) - \frac{1}{Re}\Delta D \right) + 2\left( \frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{\partial u}{\partial y}\frac{\partial v}{\partial x} \right), \tag{3}$$

where $D = \nabla \cdot \vec{v}$ is the divergence of the velocity. Terms with the divergence $D$ are zero in theory, but they may be kept in numerics to better enforce the divergence-free condition.

Eqs. (1) and (2) are defined on the entire region $\Omega$ which is composed by the subdomains $\Omega^+$ and $\Omega^-$ in Fig. 1. The subdomains $\Omega^+$ and $\Omega^-$ are filled with the same fluid and are separated by the immersed boundary $\Gamma$. Without loss of generality, the formulation given by Eqs. (1) and (2) along with Fig. 1 will be used hereafter for the presentation. In Fig. 1, the tangent $\vec{\tau}$ and the normal $\vec{n}$ to $\Gamma$ are calculated as

$$\vec{\tau} = (\tau_x, \tau_y) = \frac{1}{J}\frac{\partial \vec{X}}{\partial \alpha}, \tag{4}$$

$$\vec{n} = (n_x, n_y) = (\tau_y, -\tau_x), \tag{5}$$

where $J$ is

$$J = \left\| \frac{\partial \vec{X}}{\partial \alpha} \right\|_2. \tag{6}$$

With this formulation, both the immersed boundary method and the immersed interface method allow for fixed grids and fast flow solvers on the entire region $\Omega$. Therefore they can efficiently simulate the interaction of a fluid with multiple moving boundaries. Regarding this formulation, there are two crucial questions. One is

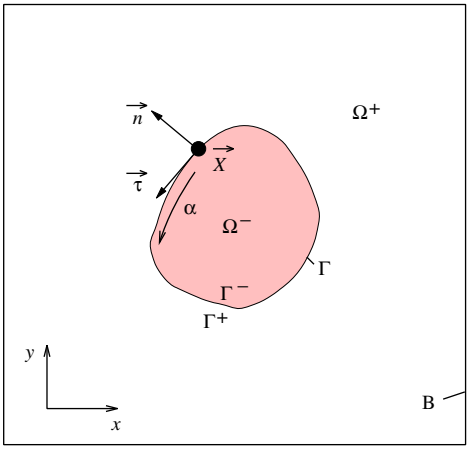



Fig. 1. Geometric description of an immersed boundary.

how to calculate the singular force (the force density $\vec{f}$). The other is how to numerically treat the force singularity (the Dirac $\delta$ function).

The immersed boundary method and the immersed interface method differ in the numerical treatment of the force singularity at $\Gamma$. The immersed boundary method uses discretized smooth functions to approximate the Dirac $\delta$ function. The immersed interface method directly incorporates singularity-induced jump conditions across $\Gamma$ into numerical schemes, so it can achieve second-order accuracy and sharp fluid–solid interfaces. The required jump conditions in the immersed interface method can be derived systematically [36]. They can be incorporated into finite difference and interpolation schemes according to a generalized Taylor expansion [36]

$$g(z_{m+1}^-) = \sum_{n=0}^{\infty} \frac{g^{(n)}(z_0^+)}{n!}(z_{m+1}-z_0)^n + \sum_{l=1}^{m}\sum_{n=0}^{\infty} \frac{[g^{(n)}(z_l)]}{n!}(z_{m+1}-z_l)^n, \tag{7}$$

where $g(z)$ is a nonsmooth function shown in Fig. 2a, and $[g^{(n)}(z_l)] = g^{(n)}(z_l^+) - g^{(n)}(z_l^-)$ denotes jump conditions along the $z$ direction. Below are examples of modified second-order central finite difference schemes which have discontinuities at $z = \xi(z_{i-1} \leqslant \xi \leqslant z_i)$ and $z = \eta(z_i \leqslant \eta \leqslant z_{i+1})$ on its stencil as shown in Fig. 2b.

$$\frac{\mathrm{d}g(z_i^-)}{\mathrm{d}z} = \frac{g(z_{i+1}^-) - g(z_{i-1}^+)}{2h} + \mathrm{O}(h^2) + \frac{1}{2h}\left(\sum_{n=0}^{2}\frac{-[g^{(n)}(\xi)]}{n!}(z_{i-1}-\xi)^n - \sum_{n=0}^{2}\frac{[g^{(n)}(\eta)]}{n!}(z_{i+1}-\eta)^n\right), \tag{8}$$

$$\frac{\mathrm{d}^2 g(z_i^-)}{\mathrm{d}z^2} = \frac{g(z_{i+1}^-) - 2g(z_i) + g(z_{i-1}^+)}{h^2} + \mathrm{O}(h^2) - \frac{1}{h^2}\left(\sum_{n=0}^{3}\frac{-[g^{(n)}(\xi)]}{n!}(z_{i-1}-\xi)^n + \sum_{n=0}^{3}\frac{[g^{(n)}(\eta)]}{n!}(z_{i+1}-\eta)^n\right). \tag{9}$$

An interpolation scheme also needs to account for the jump conditions if its stencil contains discontinuities. The following second-order interpolation scheme applies to the case shown in Fig. 2b:

$$g(z_i^-) = \frac{g(z_{i-1}^+) + g(z_{i+1}^-)}{2} + \mathcal{O}(h^2) + \frac{1}{2}\left[\frac{\partial g(\xi)}{\partial z}\right](z_{i-1}-\xi) - \frac{1}{2}\left[\frac{\partial g(\eta)}{\partial z}\right](z_{i+1}-\eta). \tag{10}$$

Using modified finite difference and interpolation schemes, the mathematical formulation, Eqs. (1) and (2), can be discretized on a fixed Cartesian grid, and fast flow solvers based on a Cartesian grid can be adopted. An overview of the immersed interface method is given in Section 2.

This paper focuses on the first question: how to calculate the singular force (the force density $\vec{f}$) in the immersed interface method. In the discrete form, the mapping between the singular force and the boundary motion can be written as

$$\mathcal{F}(\mathbf{f}) = \mathbf{V}, \tag{11}$$

where the force array $\mathbf{f}$ is composed of the values of $f_x$ and $f_y$ at all discrete Lagrangian points that are used to represent the boundary, the velocity array $\mathbf{V}$ is composed of the values of $u$ and $v$ at these points, and the operator $\mathcal{F}$ depends on the spatial discretization, interpolation, and temporal integration in the immersed interface method.

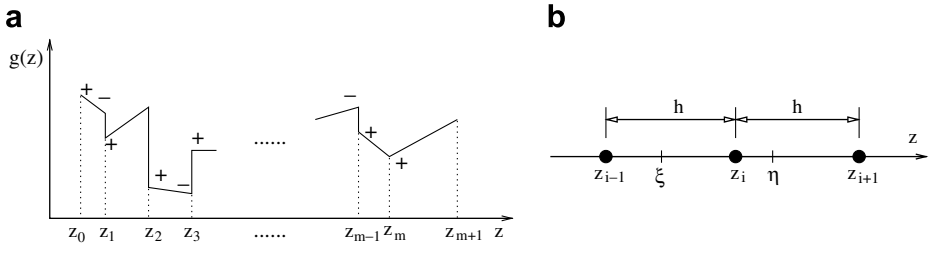



Fig. 2. (a) A nonsmooth function for generalized Taylor expansion, and (b) a stencil for generalized finite difference schemes.

Most papers on the immersed boundary method and the immersed interface method solve the usual free-boundary problem. In the free-boundary problem, the boundary is updated by

$$\mathbf{X} = \mathcal{I}(\mathbf{V}), \tag{12}$$

where the coordinate array $\mathbf{X}$ is composed of the values of $X$ and $Y$ at the Lagrangian points, and the operator $\mathcal{I}$ depends on the temporal integration for the boundary motion. The singular force is computed from a constitutive law, which relates the force density to the boundary configuration,

$$\mathbf{f} = \mathcal{G}(\mathbf{X}), \tag{13}$$

where the function $\mathcal{G}$ is given by a boundary model, for example, an elastic structure model. The computation of the singular force is therefore a modeling process. The boundary motion results from the coupling between the boundary model and fluid flow, and is unknown in advance. From the current time step $n$ to the next $n+1$, an explicit scheme gives

$$\mathbf{V}^{n+1} = \mathcal{F}(\mathcal{G}(\mathbf{X}^n)), \tag{14}$$

$$\mathbf{X}^{n+1} = \mathcal{I}(\mathbf{V}^n). \tag{15}$$

Numerical instability can occur if the boundary model, Eq. (13), is stiff. Tu and Peskin [34], Stockie and Wetton [32], and Cortez et al. [4] have conducted stability analysis for the immersed boundary method. To improve on the numerical stability, an implicit treatment was implemented in the immersed interface method [18,16], which is

$$\mathbf{V}^{n+1} = \mathcal{F}(\mathcal{G}(\mathbf{X}^{n+1})), \tag{16}$$

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \frac{1}{2}\delta t(\mathbf{V}^n + \mathbf{V}^{n+1}), \tag{17}$$

where $\delta t$ denotes the time step. Nonlinear solvers, such as BFGS or SR1, are needed in the implicit treatment [18,16].

In this paper, an inverse problem is pursued, in which the boundary motion is prescribed in advance, and the singular force is sought to enforce the known boundary motion. There are two ways to solve this inverse problem. One is the use of ad hoc penalty approaches, such as stiff spring models [14,37] and feedback controls [9,37]. They have the same algorithmic nature as constitutive laws. A widely-used feedback control is the one by Goldstein et al. [9]

$$\vec{f} = K_s(\vec{X}_e - \vec{X}) + K_d(\vec{V}_e - \vec{V}), \tag{18}$$

where $K_s$ and $K_d$ are two positive constants, $\vec{X}_e$ and $\vec{X}$ are the prescribed and simulated boundary coordinates, respectively, and $\vec{V}_e$ and $\vec{V}$ are the prescribed and simulated boundary velocity, respectively. If $K_d$ is zero, this feedback control is equivalent to a spring model [14,37]. The response times of spring models or feedback controls have to be much shorter than the characteristic time scales of the flow [14,37], which requires large $K_s$ or $K_d$ and causes numerical instability at high Reynolds numbers. The disadvantages of ad hoc penalty approaches include adjustment of free parameters, spurious oscillations, and numerical instability.

The other way to find the singular force for the prescribed boundary motion is to directly solve the mapping between the singular force and the boundary motion. If the nonlinear terms in the Navier–Stokes equations are discretized explicitly from their convective form, the mapping given by Eq. (11) is linear and can be reduced to the matrix-vector form

$$F\mathbf{f} + \mathbf{V}_0 = \mathbf{V}, \tag{19}$$

where $\mathbf{V}_0$ corresponds to $\mathbf{f} = 0$. If the boundary velocity is prescribed as $\mathbf{V} = \mathbf{V}_e$, the above linear system can be solved to obtain $\mathbf{f}$. Depending on algorithms, the linear system is formed and solved directly or indirectly. The idea to form and solve a linear system to enforce given boundary conditions was introduced by Calhoun [3]. Calhoun considered the immersed interface method in a streamfunction–vorticity formulation. The no-slip boundary condition is imposed through a distribution of vorticity sources along the boundary. Calhoun presented in detail how to set up and solve a linear system for the vorticity sources along a stationary boundary,

and proposed to use GMRES to handle a moving boundary. Li et al. [21] has implemented a similar idea to impose Neumann pressure boundary conditions in Stokes flows.

Le et al. [15] employed the same idea in their immersed interface method to enforce the prescribed velocity. Their linear system can be written as

$$F\mathbf{f}^{n+\frac{1}{2}} + \mathbf{V}_0 = \mathbf{V}_e^{n+1}, \tag{20}$$

where the time level of the force density $\mathbf{f}$ is denoted as $n+\frac{1}{2}$ since mixed explicit and implicit schemes were used in their flow solver. By setting $\mathbf{f}$ be each column of an identity matrix, they integrated the discretized governing equations to obtain $\mathbf{V}$. The matrix $F$ has the corresponding column equal to $\mathbf{V} - \mathbf{V}_0$. If all boundaries are stationary and the time step of the temporal integration is constant, the matrix $F$ needs to be formed and inverted only once. Otherwise, the matrix $F$ changes in each time step and is not formed explicitly, and GMRES is used to solve the linear system iteratively.

In this paper, an explicit approach is proposed to compute the singular force for a rigid boundary with prescribed motion. The basic process is to explicitly compute the singular force from available jump conditions. The available jump conditions are numerically calculated from the currently known flow field. This approach does not need to form the above-mentioned matrix-vector system; it does not require iterative solvers; and it does not have numerical instability at relatively high Reynolds numbers. It has advantages in stability, efficiency, simplicity, and extensibility. Based on Eq. (11), the explicit form in this approach can be written as

$$\mathbf{f}^n = \mathcal{F}^{-1}(\mathbf{V}_e^n). \tag{21}$$

So the singular force is computed from the known flow information at the current time step $n$. After the singular force $\mathbf{f}^n$ is computed, the flow field is updated to the next time step $n+1$. This current approach may be better interpreted as a way to explicitly implement velocity and pressure boundary conditions on moving rigid objects that are embedded in a fixed Cartesian grid. Essentially, its algorithmic treatment of the boundary is very similar to that of a body-fitted grid method which couples the velocity conditions explicitly into one-sided finite difference schemes and applies the Neumann boundary condition for the pressure Poisson equation.

The current approach shares the ideas from the methods by Russell and Wang [31] and Linnick and Fasel [22]. In Russell and Wang's method, the non-penetration boundary condition is satisfied by superposing a homogeneous solution to the Poisson equation for the streamfunction. In Linnick and Fasel's method, one-sided finite differences are used to obtain jump conditions for the vorticity and the streamfunction. The current approach is built upon the primitive-variable formulation, Eqs. (1) and (2). The tangential component of a singular force is determined from the normal derivative of the velocity using one-sided finite difference. The normal component is determined from a predictor and a corrector. The predictor uses the normal derivative of the vorticity. The corrector solves a Neumann–Dirichlet map and superposes a homogeneous solution to the Poisson equation for the pressure to achieve the desired normal derivative of the pressure. In addition, a discontinuous finite body force is applied such that the fluid enclosed by the boundary is in rigid motion.

## 2. Overview of the immersed interface method

In the current implementation of the immersed interface method, the momentum equation, Eq. (1), and the pressure Poisson equation, Eq. (2), are solved using the MAC scheme [11] along with a fourth-order RK time integration scheme and an FFT-based Poisson solver. More details on the implementation are referred to [37]. A MAC grid is a staggered Cartesian grid, on which the pressure $p$ and the velocity components $u$ and $v$ are arranged as in Fig. 3. As pointed out by Johnston and Liu [12,13] and E and Liu [7], high order explicit time-marching schemes are appropriate for flow of moderate to high Reynolds numbers, where the viscous time step constraint is less restrictive than the convective one. However, explicit time-marching schemes are not a requirement. If low Reynolds numbers are of interest, implicit treatment of the viscous terms can be adopted instead. Since the MAC grid is uniform in the current implementation, an FFT-based Poisson solver can be used, which solves the pressure Poisson equation in $\mathcal{O}(N \ln N)$ time, where $N$ is the total number of MAC grid

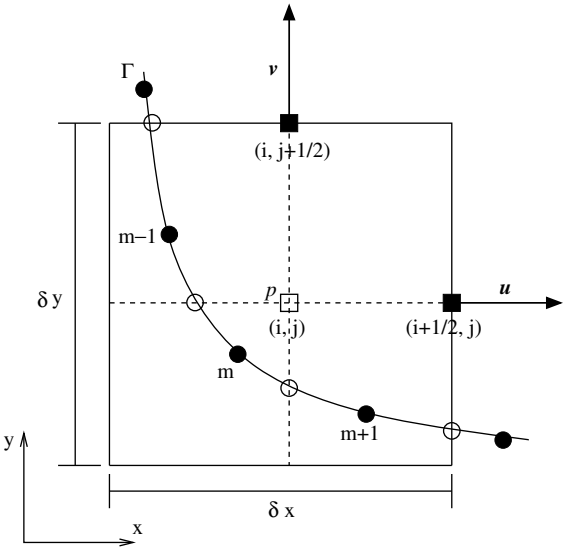



Fig. 3. Arrangement of the velocity components and the pressure on a MAC grid.

nodes for the pressure. FFT-based Poisson solvers can handle periodic, Dirichlet, Neumann, or mixed boundary conditions at the far-field boundary $B$ in Fig. 1 by using FFT, sine, cosine, or quarter-wave transformations [28].

A flow quantity at a fixed point in space may have a jump with respect to time when a boundary passes that point, and a temporal jump condition can be related to a corresponding spatial jump condition [36,37]. In simulating a viscous flow, the incorporation of temporal jump conditions in temporal discretization has negligible effect on simulation results [37,38]. In the current implementation, temporal jump conditions are not included.

The boundary $\Gamma$ is rigid in the current consideration. It is represented by periodic cubic splines formed from Lagrangian points. In Fig. 3, Lagrangian points are denoted as solid circles. As indicated by Eqs. (8)–(10), the jump contributions in the finite difference and interpolation schemes are non-zero only if their stencils cross the boundary. In order to distinguish these stencils, the intersections between MAC grid lines and the boundary are identified. The unknown coordinates of the intersections and the necessary jump conditions at the intersections are interpolated from the Lagrangian points. In Fig. 3, the intersections are denoted as open circles.

Define the central finite difference operators $\delta_x$, $\delta_y$, $\delta_{xx}$, and $\delta_{yy}$ as

$$\delta_x(\cdot)_{i,j} = \frac{(\cdot)_{i+\frac{1}{2},j} - (\cdot)_{i-\frac{1}{2},j}}{\delta x} + c_x(\cdot)_{i,j}, \tag{22}$$

$$\delta_y(\cdot)_{i,j} = \frac{(\cdot)_{i,j+\frac{1}{2}} - (\cdot)_{i,j-\frac{1}{2}}}{\delta y} + c_y(\cdot)_{i,j}, \tag{23}$$

$$\delta_{xx}(\cdot)_{i,j} = \frac{(\cdot)_{i+1,j} - 2(\cdot)_{i,j} + (\cdot)_{i-1,j}}{\delta x^2} + c_{xx}(\cdot)_{i,j}, \tag{24}$$

$$\delta_{yy}(\cdot)_{i,j} = \frac{(\cdot)_{i,j+1} - 2(\cdot)_{i,j} + (\cdot)_{i,j-1}}{\delta y^2} + c_{yy}(\cdot)_{i,j}, \tag{25}$$

where $\delta x$ and $\delta y$ are the spatial steps as shown in Fig. 3, and $c_x, c_y, c_{xx}$, and $c_{yy}$ denote jump contributions. If the stencils of the above finite difference operators cross $\Gamma$, the jump contributions are non-zero, and they can be calculated according to Eqs. (8) and (9). Otherwise, the jump contributions are zero, and usual central finite difference schemes are recovered. With these central finite difference operators, the spatially discretized momentum equation for the velocity component $u$ at $(i+\frac{1}{2}, j)$ can be written as

$$\frac{\partial u}{\partial t} = -\delta_x(uu) - \delta_y(vu) - \delta_x p + \frac{1}{Re}(\delta_{xx} + \delta_{yy})u, \tag{26}$$

where the subscript $(i+\frac{1}{2}, j)$ is neglected in the operators. A similar equation for $v$ at $(i, j+\frac{1}{2})$ can be obtained. The discretized Poisson equation for the pressure $p$ at $(i, j)$ can be written as

$$(\delta_{xx} + \delta_{yy})p = -\frac{\partial D}{\partial t} - 2(\delta_x(uD) + \delta_y(vD)) + \frac{1}{Re}(\delta_{xx} + \delta_{yy})D + 2(\delta_x u \delta_y v - \delta_y u \delta_x v), \tag{27}$$

where $\frac{\partial D}{\partial t}$ is discretized by assuming $D = 0$ at the next time level.

The values of $u_{i,j}, u_{i,j+\frac{1}{2}}, v_{i,j}$, and $v_{i+\frac{1}{2},j}$ are needed in the discretized equations. They can be interpolated from $u_{i+\frac{1}{2},j}$ and $v_{i,j+\frac{1}{2}}$. Define the interpolation operators $\varepsilon_i$ and $\varepsilon_j$ as

$$\varepsilon_i(\cdot)_{i,j} = \frac{(\cdot)_{i+\frac{1}{2},j} + (\cdot)_{i-\frac{1}{2},j}}{2} + c_i(\cdot)_{i,j}, \tag{28}$$

$$\varepsilon_j(\cdot)_{i,j} = \frac{(\cdot)_{i,j+\frac{1}{2}} + (\cdot)_{i,j-\frac{1}{2}}}{2} + c_j(\cdot)_{i,j}, \tag{29}$$

where $c_i$ and $c_j$ denote jump contributions. If the stencils of the above interpolation operators cross $\Gamma$, the jump contributions are non-zero, and they can be calculated according to Eq. (10). Otherwise, the jump contributions are zero, and usual interpolation schemes are recovered. With these interpolation operators, the interpolation for $u_{i,j}$ and $u_{i,j+\frac{1}{2}}$ can be written as follows:

$$u_{i,j} = \varepsilon_i u_{i,j}, \tag{30}$$

$$u_{i,j+\frac{1}{2}} = \varepsilon_j u_{i,j+\frac{1}{2}}. \tag{31}$$

The similar interpolation for $v_{i,j}$ and $v_{i+\frac{1}{2},j}$ can be obtained.

As indicated by Eqs. (8)–(10), necessary jump conditions are needed to obtain the jump contributions in the above finite difference and interpolation schemes. The jump conditions are related to the singular force (the force density $\vec{f}$). Section 4 lists these relations. The singular force is calculated to enforce the prescribed motion of the boundary $\Gamma$ using the explicit approach, which is mentioned in Section 1 and described in detail in Sections 5–7. In this explicit approach, a discontinuous finite body force is applied such that the fluid enclosed by the boundary is in rigid motion. The body force is found in Section 3.

## 3. Body force

The motion of the rigid boundary $\Gamma$ can be prescribed through $\vec{x}_c(t)$ and $\theta(t)$, where $\vec{x}_c = (x_c, y_c)$ denotes the Cartesian coordinates of a point $c$ which is fixed with respect to the boundary, and $\theta$ denotes the rotational angle of the boundary in the Cartesian coordinate system. When the fluid in the region $\Omega^-$, which is enclosed by $\Gamma$, is in the rigid motion, the velocity $\vec{v} = (u, v)$ at a fixed point in $\Omega^-$ is

$$u = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\mathrm{d}x_c}{\mathrm{d}t} - \frac{\mathrm{d}\theta}{\mathrm{d}t}(y - y_c), \tag{32}$$

$$v = \frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}y_c}{\mathrm{d}t} + \frac{\mathrm{d}\theta}{\mathrm{d}t}(x - x_c), \tag{33}$$

which satisfies

$$s_p = 2\left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2, \tag{34}$$

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -\nabla\left(-\frac{\mathrm{d}^2 x_c}{\mathrm{d}t^2}x - \frac{\mathrm{d}^2 y_c}{\mathrm{d}t^2}y + \frac{1}{2}\left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2((x - x_c)^2 + (y - y_c)^2)\right) + \frac{1}{Re}\Delta\vec{v} + \vec{b}, \tag{35}$$

where the expression for $s_p$ is given by Eq. (3), $\Delta\vec{v} = 0$, and $\vec{b} = (b_x, b_y)$ is

$$b_x = -\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}(y - y_c), \tag{36}$$

$$b_y = \frac{d^2\theta}{dt^2}(x - x_c). \tag{37}$$

So a discontinuous body force $\vec{b}$ can be applied to achieve the rigid motion of the fluid in $\Omega^-$. In $\Omega^-$, it is given by Eqs. (36) and (37). Since the flow of interest is in $\Omega^+$, it must be zero in $\Omega^+$. The body force is discontinuous and has the following finite jumps across $\Gamma$:

$$[b_\tau] = -\frac{d^2\theta}{dt^2}(-(Y - y_c)\tau_x + (X - x_c)\tau_y), \tag{38}$$

$$[b_n] = -\frac{d^2\theta}{dt^2}(-(Y - y_c)n_x + (X - x_c)n_y), \tag{39}$$

where $[b_\tau] = [\vec{b}] \cdot \vec{\tau}$ and $[b_n] = [\vec{b}] \cdot \vec{n}$. Hereafter $[\cdot] = (\cdot)_{\Gamma^+} - (\cdot)_{\Gamma^-}$ denotes a jump across $\Gamma$ along the normal $\vec{n}$, where $\Gamma^+$ and $\Gamma^-$ denote the outer and the inner sides of $\Gamma$, respectively, as shown in Fig. 1.

As indicated by Eq. (35), the pressure in $\Omega^-$, subject to a constant, is

$$p = -\frac{d^2x_c}{dt^2}x - \frac{d^2y_c}{dt^2}y + \frac{1}{2}\left(\frac{d\theta}{dt}\right)^2((x - x_c)^2 + (y - y_c)^2). \tag{40}$$

Thus, the pressure at $\Gamma^-$ is

$$p|_{\Gamma^-} = -\frac{d^2x_c}{dt^2}X - \frac{d^2y_c}{dt^2}Y + \frac{1}{2}\left(\frac{d\theta}{dt}\right)^2((X - x_c)^2 + (Y - y_c)^2), \tag{41}$$

and the normal derivative of the pressure at $\Gamma^-$ is

$$\left.\frac{\partial p}{\partial n}\right|_{\Gamma^-} = -\frac{d^2x_c}{dt^2}n_x - \frac{d^2y_c}{dt^2}n_y + \left(\frac{d\theta}{dt}\right)^2((X - x_c)n_x + (Y - y_c)n_y), \tag{42}$$

where $\frac{\partial(\cdot)}{\partial n} = \nabla(\cdot) \cdot \vec{n}$ denotes a normal derivative.

In the numerical implementation, the momentum equation, Eq. (1), is solved with $\vec{b} = 0$ in the entire region $\Omega$, and the discontinuous body force $\vec{b}$ is equivalently applied by directly setting the velocity in the region $\Omega^-$ to the desired values and incorporating the known jump conditions $[\vec{b}]$, $[b_\tau]$, and $[b_n]$ into numerical schemes.

## 4. Jump conditions

The necessary jump conditions induced by the singular force have been derived in [36]. They have been used in the 2D immersed interface method in [37] and the 3D immersed interface method in [38]. The jump conditions listed in this section are functions of both the singular force and the discontinuous body force, and they are modified from those in [37] to take into account the discontinuous body force. The singular force appears in the expressions of the jump conditions in the form of the tangential component $f_\tau$ and the normal component $f_n$, which are defined as

$$f_\tau = \frac{1}{J}(f_x\tau_x + f_y\tau_y), \tag{43}$$

$$f_n = \frac{1}{J}(f_xn_x + f_yn_y). \tag{44}$$

The jump condition for the velocity is

$$[\vec{v}] = 0. \tag{45}$$

The jump conditions for the first derivatives of the velocity are

$$\left[\frac{\partial\vec{v}}{\partial x}\right] = -Ref_\tau\tau_y\vec{\tau}, \tag{46}$$

$$\left[\frac{\partial\vec{v}}{\partial y}\right] = Ref_\tau\tau_x\vec{\tau}. \tag{47}$$

The jump conditions for the second derivatives of the velocity are

$$\left[\frac{\partial^2 \vec{v}}{\partial x^2}\right] = \vec{r}_{u1}(\tau_x^2 - \tau_y^2) + \vec{r}_{u2}(2\tau_x\tau_y) + \vec{r}_{u3}(\tau_y^2), \tag{48}$$

$$\left[\frac{\partial^2 \vec{v}}{\partial y^2}\right] = \vec{r}_{u1}(\tau_y^2 - \tau_x^2) - \vec{r}_{u2}(2\tau_x\tau_y) + \vec{r}_{u3}(\tau_x^2), \tag{49}$$

$$\left[\frac{\partial^2 \vec{v}}{\partial x \partial y}\right] = \vec{r}_{u1}(2\tau_x\tau_y) + \vec{r}_{u2}(\tau_y^2 - \tau_x^2) - \vec{r}_{u3}(\tau_x\tau_y), \tag{50}$$

where $\vec{r}_{u1}, \vec{r}_{u2}$ and $\vec{r}_{u3}$ are

$$\vec{r}_{u1} = -\frac{1}{J^2}\left(\frac{\partial J\tau_x}{\partial\alpha}\left[\frac{\partial \vec{v}}{\partial x}\right] + \frac{\partial J\tau_y}{\partial\alpha}\left[\frac{\partial \vec{v}}{\partial y}\right]\right), \tag{51}$$

$$\vec{r}_{u2} = -\frac{1}{J}\left(Re\frac{\partial f_\tau \vec{\tau}}{\partial\alpha} + \frac{\partial n_x}{\partial\alpha}\left[\frac{\partial \vec{v}}{\partial x}\right] + \frac{\partial n_y}{\partial\alpha}\left[\frac{\partial \vec{v}}{\partial y}\right]\right), \tag{52}$$

$$\vec{r}_{u3} = Re([\nabla p] - [\vec{b}]). \tag{53}$$

The jump conditions for the first and second derivatives of the pressure $p$ can be expressed in terms of the jump conditions $[p], [\frac{\partial p}{\partial n}]$ and $[\Delta p]$. The jump conditions for the first derivatives of the pressure are

$$\left[\frac{\partial p}{\partial x}\right] = \frac{\tau_x}{J}\frac{\partial [p]}{\partial\alpha} + \tau_y\left[\frac{\partial p}{\partial n}\right], \tag{54}$$

$$\left[\frac{\partial p}{\partial y}\right] = \frac{\tau_y}{J}\frac{\partial [p]}{\partial\alpha} - \tau_x\left[\frac{\partial p}{\partial n}\right]. \tag{55}$$

The jump conditions for the second derivatives of the pressure are

$$\left[\frac{\partial^2 p}{\partial x^2}\right] = r_{p1}(\tau_x^2 - \tau_y^2) + r_{p2}(2\tau_x\tau_y) + r_{p3}(\tau_y^2), \tag{56}$$

$$\left[\frac{\partial^2 p}{\partial y^2}\right] = r_{p1}(\tau_y^2 - \tau_x^2) - r_{p2}(2\tau_x\tau_y) + r_{p3}(\tau_x^2), \tag{57}$$

$$\left[\frac{\partial^2 p}{\partial x \partial y}\right] = r_{p1}(2\tau_x\tau_y) + r_{p2}(\tau_y^2 - \tau_x^2) - r_{p3}(\tau_x\tau_y), \tag{58}$$

where $r_{p1}, r_{p2}$ and $r_{p3}$ are

$$r_{p1} = \frac{1}{J^2}\left(\frac{\partial^2 [p]}{\partial\alpha^2} - \frac{\partial J\tau_x}{\partial\alpha}\left[\frac{\partial p}{\partial x}\right] - \frac{\partial J\tau_y}{\partial\alpha}\left[\frac{\partial p}{\partial y}\right]\right), \tag{59}$$

$$r_{p2} = \frac{1}{J}\left(\frac{\partial}{\partial\alpha}\left[\frac{\partial p}{\partial n}\right] - \frac{\partial n_x}{\partial\alpha}\left[\frac{\partial p}{\partial x}\right] - \frac{\partial n_y}{\partial\alpha}\left[\frac{\partial p}{\partial y}\right]\right), \tag{60}$$

$$r_{p3} = [\Delta p]. \tag{61}$$

In general, if the function $\phi$ satisfies a Poisson equation on the region $\Omega$ with the given jump conditions $[\phi], [\frac{\partial \phi}{\partial n}]$ and $[\Delta\phi]$ across the immersed boundary $\Gamma$, the immersed interface method can be used to solve for the function $\phi$. The required jump conditions of the first and second derivatives of the function $\phi$ can be obtained by replacing $p$ with $\phi$ in the above expressions. For the pressure $p$, the jump conditions $[p], [\frac{\partial p}{\partial n}]$ and $[\Delta p]$ are given by

$$[p] = f_n, \tag{62}$$

$$\left[\frac{\partial p}{\partial n}\right] = \frac{1}{J}\frac{\partial f_\tau}{\partial\alpha} + [b_n] = \frac{\partial f_\tau}{\partial\tau} + [b_n], \tag{63}$$

$$[\Delta p] = 2\left[\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right] - 2\left[\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right], \tag{64}$$

where $\frac{\partial(\cdot)}{\partial\tau} = \nabla(\cdot)\cdot\vec{\tau}$ denotes a tangential derivative.

Surface derivatives with respect to the Lagrangian parameter $\alpha$ are involved in computing the above jump conditions. They are calculated using cubic splines or Fourier transformations.

## 5. Tangential singular force

As indicated by Eqs. (46) and (47), the tangential singular force $f_\tau$ is related to the normal derivative of the velocity and the surface vorticity as follows:

$$f_\tau = -\frac{1}{Re}\left[\vec{\tau}\cdot\frac{\partial\vec{v}}{\partial n}\right] = -\frac{1}{Re}[\omega], \tag{65}$$

where $\omega$ is the vorticity in 2D. The fluid in the region $\Omega^-$ is in the rigid motion, so

$$\vec{\tau}\cdot\left.\frac{\partial\vec{v}}{\partial n}\right|_{\Gamma^-} = \frac{\mathrm{d}\theta}{\mathrm{d}t}, \tag{66}$$

$$\omega|_{\Gamma^-} = 2\frac{\mathrm{d}\theta}{\mathrm{d}t}, \tag{67}$$

and the above relations can be written as

$$f_\tau = -\frac{1}{Re}\left(\vec{\tau}\cdot\left.\frac{\partial\vec{v}}{\partial n}\right|_{\Gamma^+} - \frac{\mathrm{d}\theta}{\mathrm{d}t}\right) = -\frac{1}{Re}\left(\omega|_{\Gamma^+} - 2\frac{\mathrm{d}\theta}{\mathrm{d}t}\right). \tag{68}$$

In the current approach, $f_\tau$ is determined from $\frac{\partial\vec{v}}{\partial n}|_{\Gamma^+}$ explicitly.

To calculate $\frac{\partial\vec{v}}{\partial n}|_{\Gamma^+}$, a one-sided finite difference scheme along the normal $\vec{n}$ is used. The one-sided finite difference scheme with the three-point stencil shown in Fig. 4 is

$$\frac{\partial\vec{v}(S_0)}{\partial n} = \frac{-3\vec{v}(S_0) + 4\vec{v}(S_1) - \vec{v}(S_2)}{2\delta n} + \mathrm{O}(\delta n^2), \tag{69}$$

where $\delta n$ is the distance between two adjacent points on the stencil, and $\delta n \geqslant \sqrt{\delta x^2 + \delta y^2}$ to make sure that two adjacent points are in different Cartesian-grid cells. This stencil can also be used for one-sided extrapolation. More points can be added in the stencil for higher accuracy. The velocity at the boundary point $S_0$ is prescribed. The velocity at the points $S_1$ and $S_2$ are interpolated from four surrounding Cartesian-grid nodes. For example, the velocity at the point $S_2$ can be interpolated from the nodes $I, II, III$, and $IV$ in Fig. 4. Similarly, more nodes can be used for higher accuracy. When there are multiple boundaries, they

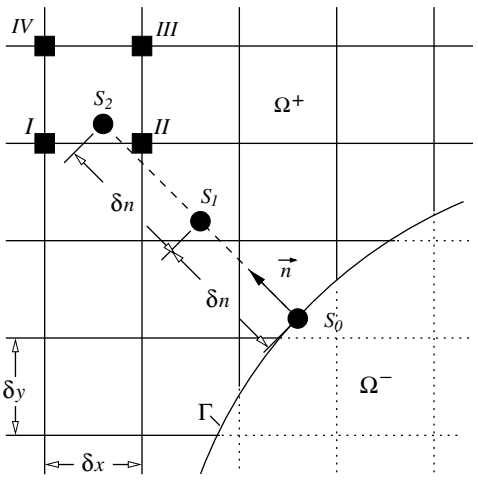



Fig. 4. Stencils for one-sided finite difference, one-sided extrapolation, and bilinear interpolation.

are separated with each other by enough Cartesian-grid nodes so that no boundaries cross the interpolation cell. No discontinuities are therefore involved in the interpolation, and the standard bilinear interpolation scheme can be adopted for the case in Fig. 4. Because of the staggered arrangement of the velocity components, the interpolation cells shown in Fig. 4 are different for the different velocity components.

## 6. Predictor for the normal singular force

The following relation can be obtained from Eqs. (54), (55) and (62):

$$\left[\frac{\partial p}{\partial \tau}\right] = \frac{\partial f_n}{\partial \tau}. \tag{70}$$

In general, derived from the momentum equation with the use of the identity $\Delta\vec{v} = -\nabla\times\vec{\omega} + \nabla D$, where $\vec{\omega} = \nabla\times\vec{v}$ is the vorticity vector, the jump condition for the pressure gradient satisfies

$$[\nabla p] = -\frac{1}{Re}[\nabla\times\vec{\omega}] + [\vec{b}]. \tag{71}$$

For the 2D case shown in Fig. 1, Eq. (71) gives

$$\left[\frac{\partial p}{\partial \tau}\right] = \frac{1}{Re}\left[\frac{\partial \omega}{\partial n}\right] + [b_\tau]. \tag{72}$$

Eqs. (70) and (72) indicate

$$\frac{1}{J}\frac{\partial f_n}{\partial \alpha} = \frac{1}{Re}\left[\frac{\partial \omega}{\partial n}\right] + [b_\tau], \tag{73}$$

which can be integrated to give an expression for the normal singular force $f_n$

$$f_n = \int\left(\frac{1}{Re}\left[\frac{\partial \omega}{\partial n}\right] + [b_\tau]\right)J\mathrm{d}\alpha, \tag{74}$$

where $\left[\frac{\partial\omega}{\partial n}\right] = \frac{\partial\omega}{\partial n}|_{\Gamma^+}$ as $\omega = 2\frac{\mathrm{d}\theta}{\mathrm{d}t}$ is uniform in the region $\Omega^-$, and $\frac{\partial\omega}{\partial n}|_{\Gamma^+}$ can be calculated using a one-sided finite difference scheme similar to Eq. (69).

The pressure in the region $\Omega^+$ satisfies the Poisson equation

$$\Delta p = s_p, \tag{75}$$

along with the Dirichlet and Neumann boundary conditions at $\Gamma^+$

$$p|_{\Gamma^+} = [p] + p|_{\Gamma^-}, \tag{76}$$

$$\left.\frac{\partial p}{\partial n}\right|_{\Gamma^+} = \left[\frac{\partial p}{\partial n}\right] + \left.\frac{\partial p}{\partial n}\right|_{\Gamma^-}, \tag{77}$$

where $p|_{\Gamma^-}$ and $\frac{\partial p}{\partial n}|_{\Gamma^-}$ are given by Eqs. (41) and (42), respectively. The Dirichlet and Neumann boundary conditions must be consistent, which indicates that $[p]$ must be consistent with $\left[\frac{\partial p}{\partial n}\right]$. Eqs. (62) and (63) therefore imply the necessary consistency between $f_n$ and $\frac{\partial f_\tau}{\partial \tau}$. The consistency can be checked by the pressure in $\Omega^-$, which should satisfy Eqs. (40)–(42). Because of numerical errors, the consistency can be violated. Plotted in Fig. 5a are the contours of the computed pressure for flow past a rotating cylinder. In the computation of the pressure, $\frac{\partial f_\tau}{\partial \tau}$ was calculated using Eq. (68) and $f_n$ by Eq. (74). The violation of the consistency in this example is indicated by the non-axisymmetric pressure inside the cylinder. To enforce the consistency, Eq. (68) is used for calculating $\frac{\partial f_\tau}{\partial \tau}$, but Eq. (74) is used only as a predictor for $f_n$, and a corrector for $f_n$ as described in Section 7 is employed. In other words, $[p]$ is corrected to be consistent with $\left[\frac{\partial p}{\partial n}\right]$. The axisymmetric pressure inside the cylinder in Fig. 5b is obtained with the corrector.

The reason to correct $f_n$ instead of $\frac{\partial f_\tau}{\partial \tau}$ is as follows. It is expected that a thin shear layer may form around the boundary $\Gamma$, and the vorticity in the shear layer has larger gradient in the normal direction than the tangential direction. When a uniform MAC gird is used, it is likely that the vorticity field is better resolved in the tangential direction. Eq. (74) indicates that $f_n$ is calculated from $\frac{\partial\omega}{\partial n}|_{\Gamma^+}$, and Eq. (68) indicates that $\frac{\partial f_\tau}{\partial \tau}$ is

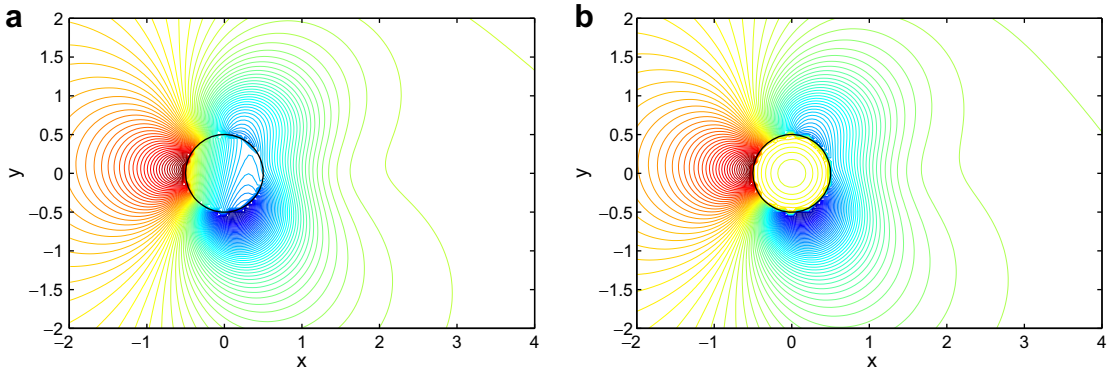



Fig. 5. Computed pressure field of flow past a rotating cylinder with $f_n$ obtained from (a) a predictor only and (b) a predictor and a corrector.

calculated from $\frac{\partial\omega}{\partial\tau}|_{\Gamma^+}$. So it is expected that $\frac{\partial f_\tau}{\partial\tau}$ is calculated more accurately than $f_n$. In addition, $\frac{\partial\omega}{\partial n}|_{\Gamma^+}$ is currently approximated by a one-sided finite difference scheme and $\frac{\partial\omega}{\partial\tau}|_{\Gamma^+}$ by Fourier transformation, and the approximation error associated with the former is larger than the latter.

## 7. Corrector for the normal singular force

Eq. (74) is used only as a predictor for $f_n$. A correction $\delta f_n$ is added to the right hand side of Eq. (74) to enforce the consistency between $f_n$ and $\frac{\partial f_\tau}{\partial\tau}$. Hereafter, the right hand side of Eq. (74) is denoted as $f_{n0}$, and $f_n$ is written as $f_n = f_{n0} + \delta f_n$. The correction $\delta f_n$ is computed such that in $\Omega^+$ the pressure $p$ satisfies the Poisson equation, Eq. (75), and the Neumann boundary condition

$$\left.\frac{\partial p}{\partial n}\right|_{\Gamma^+} = \left[\frac{\partial p}{\partial n}\right] + \left.\frac{\partial p}{\partial n}\right|_{\Gamma^-}, \tag{78}$$

where $[\frac{\partial p}{\partial n}]$ is given by Eq. (63) and $\frac{\partial p}{\partial n}|_{\Gamma^-}$ is given by Eq. (42). In $\Omega^-$, the pressure $p$ thus satisfies Eq. (42). The pressure $p$ in $\Omega^-$ satisfies a poisson equation with $s_p$ given by Eq. (34), so Eqs. (40) and (41) are also satisfied.

### 7.1. Computing the correction

Using the linearity of the pressure Poisson equation, Eq. (27), The pressure $p$ in $\Omega$ can be split into the Poisson part $p_0$ and Laplace part $q$ as $p = p_0 + q$. The Poisson part $p_0$ satisfies

$$\Delta p_0 = s_p, \tag{79}$$

$$\left[\frac{\partial p_0}{\partial n}\right] = \left[\frac{\partial p}{\partial n}\right] = \frac{\partial f_\tau}{\partial\tau} + [b_n], \tag{80}$$

$$[p_0] = f_{n0}, \tag{81}$$

$$\left.\frac{\partial p_0}{\partial n}\right|_B = g_B, \tag{82}$$

where the Neumann data $g_B$ are assigned at the far-field boundary $B$ for the pressure $p = p_0 + q$. Dirichlet or mixed boundary conditions at $B$ can be considered similarly. The Laplace part $q$ thus satisfies

$$\Delta q = 0, \tag{83}$$

$$\left[\frac{\partial q}{\partial n}\right] = 0, \tag{84}$$

$$[q] = \delta f_n, \tag{85}$$

$$\left.\frac{\partial q}{\partial n}\right|_B = 0. \tag{86}$$

As mentioned in Section 4, the immersed interface method with cosine transformations can be used to solve for $p_0$. After $p_0$ is known, $p_0|_{\Gamma^-}$ and $\frac{\partial p_0}{\partial n}|_{\Gamma^-}$ can be obtained by one-sided extrapolation and finite difference schemes as in Fig. 4. To solve for $q$, the correction $\delta f_n$ needs to be known first.

With the right correction $\delta f_n, p|_{\Gamma^-}$ satisfies Eq. (41). So $q|_{\Gamma^-}$ can be known from $q|_{\Gamma^-} = p|_{\Gamma^-} - p_0|_{\Gamma^-}$. If $q|_{\Gamma^+}$ is also known, then $\delta f_n = q|_{\Gamma^+} - q|_{\Gamma^-}$ is known. The following system defined on $\Omega^+$ can be used to determine $q|_{\Gamma^+}$

$$\Delta q = 0, \tag{87}$$

$$\left.\frac{\partial q}{\partial n}\right|_{\Gamma^+} = \left.\frac{\partial p}{\partial n}\right|_{\Gamma^-} + \left[\frac{\partial p}{\partial n}\right] - \left.\frac{\partial p_0}{\partial n}\right|_{\Gamma^-} - \left[\frac{\partial p_0}{\partial n}\right] = \left.\frac{\partial p}{\partial n}\right|_{\Gamma^-} - \left.\frac{\partial p_0}{\partial n}\right|_{\Gamma^-}, \tag{88}$$

$$\left.\frac{\partial q}{\partial n}\right|_B = 0, \tag{89}$$

where $\frac{\partial p}{\partial n}|_{\Gamma^-}$ is given by Eq. (42). By solving the Neumann–Dirichlet map [10] based on this system, $q|_{\Gamma^+}$ can be obtained.

It can be verified that the computed pressure $p = p_0 + q$ satisfies Eq. (75) in $\Omega^+$ and the Neumann boundary condition given by Eq. (78) on $\Gamma^+$, and this Neumann boundary condition is equivalent to

$$\left.\frac{\partial p}{\partial n}\right|_{\Gamma^+} = -\left(\left.\frac{\nabla \times \vec{\omega}}{Re}\right|_{\Gamma^+} + \vec{a}\right) \cdot \vec{n}, \tag{90}$$

where $\vec{a}$ is the acceleration of the boundary. As discussed in [29], rigorous pressure boundary conditions are related to the global velocity field to achieve zero divergence, and they can be obtained via the influence matrix method. Because the cost of the influence matrix method for an irregular domain is generally prohibitive, the local pressure boundary condition given by Eq. (90) is commonly used in practice for moving walls in a body-fitted grid method. The corrector in the current approach can be regarded as the superposition of a homogeneous solution to the pressure Poisson equation to achieve this desired normal derivative of the pressure, Eq. (90).

The underlying boundary integral equation for the Neumann–Dirichlet map is

$$\beta q(\vec{\xi}_0) = \int_C \ln \|\vec{\xi} - \vec{\xi}_0\|_2 \frac{\partial q(\vec{\xi})}{\partial n} \mathrm{d}s - \int_C q(\vec{\xi}) \frac{(\vec{\xi} - \vec{\xi}_0) \cdot \vec{n}}{\|\vec{\xi} - \vec{\xi}_0\|_2^2} \mathrm{d}s, \tag{91}$$

where, as shown in Fig. 6, $C$ is the composite boundary that is composed of $\Gamma$ and $B$ but excludes the point $\vec{\xi}_0$, $\vec{\xi}_0$ and $\vec{\xi}$ are coordinates of two different points on the composite boundary composed of $\Gamma$ and $B$, $\beta$ is equal to $\pi$ if $C$ is smooth at $\vec{\xi}_0$ and is equal to the angle of a corner if $\vec{\xi}_0$ is at the corner, and the normal $\vec{n}$ to $C$ points toward $\Omega^+$. In the current implementation, the above boundary integral equation is converted to a linear system using the boundary element method. The linear system is solved using LU decomposition and backward substitution in $\mathcal{O}(L^3)$ time, where $L$ represents the total number of boundary elements for $C$. If $\Gamma$ is static, the LU decomposition is done once at the beginning and stored for later use. It is possible to use a multipole method to solve the Neumann–Dirichlet map in $\mathcal{O}(L)$ time.

### 7.2. Numerical test of a corrector-based Poisson solver

The corrector described in Section 7.1 can be used as a Cartesian-grid Poisson solver for a Poisson equation with given boundary conditions on embedded irregular boundaries. This Poisson solver is similar to the one proposed by Mayo [24].

As a test of the corrector-based Poisson solver, the Poisson equation $\Delta\phi = s_\phi$ is solved, where $s_\phi = -2\sin(x)\cos(y)$ is generated by taking $\phi = \sin(x)\cos(y)$. The domain for the equation is the region $\Omega^+$ between the far-field boundary $B$ (at $x = \pm 1$ and $y = \pm 1$) and the circle $\Gamma$ (at $(0,0)$) with radius equal to 0.5. The Neumann boundary conditions $\frac{\partial \phi}{\partial n}|_B$ and $\frac{\partial \phi}{\partial n}|_{\Gamma^+}$ are given according to $\phi = \sin(x)\cos(y)$.

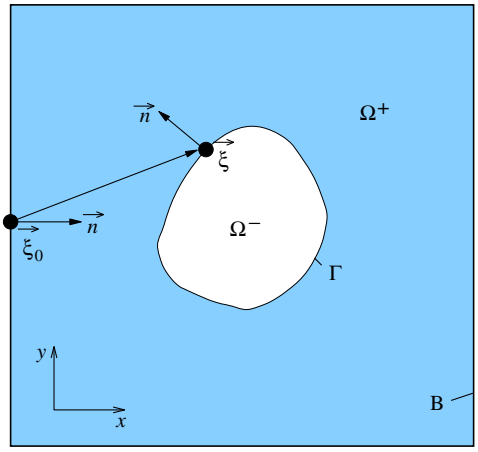



Fig. 6. Boundaries and notations for a Neumann–Dirichlet map.

Let $\phi$ be a known function in the region $\Omega^-$. For simplicity, let $\phi$ be an arbitrary constant $c$ here. Then $[\frac{\partial\phi}{\partial n}]$ is known, and $[\phi]$ is computed such that the given $\frac{\partial\phi}{\partial n}|_{\Gamma^+}$ is enforced. Split $\phi$ as $\phi = \phi_0 + \psi$. Solve for the Poisson part $\phi_0$ with $[\phi_0] = 0$ (0 is chosen arbitrarily) in the entire rectangular region $\Omega$ enclosed by $B$ from

$$\Delta\phi_0 = s_{\phi_0}, \tag{92}$$

$$\left[\frac{\partial\phi_0}{\partial n}\right] = \left.\frac{\partial\phi}{\partial n}\right|_{\Gamma^+}, \tag{93}$$

$$[\phi_0] = 0, \tag{94}$$

$$\left.\frac{\partial\phi_0}{\partial n}\right|_B = \left.\frac{\partial\phi}{\partial n}\right|_B, \tag{95}$$

where $s_{\phi_0} = s_\phi$ in $\Omega^+$ and $s_{\phi_0} = 0$ in $\Omega^-$. After $\phi_0$ is known, $\phi_0|_{\Gamma^-}$ and $\frac{\partial\phi_0}{\partial n}|_{\Gamma^-}$ are calculated. The Laplace part $\psi$ in $\Omega$ satisfies

$$\Delta\psi = 0, \tag{96}$$

$$\left[\frac{\partial\psi}{\partial n}\right] = 0, \tag{97}$$

$$[\psi] = [\phi], \tag{98}$$

$$\left.\frac{\partial\psi}{\partial n}\right|_B = 0, \tag{99}$$

where $[\phi] = \phi_0|_{\Gamma^+} + \psi|_{\Gamma^+} - c = \phi_0|_{\Gamma^-} + \psi|_{\Gamma^+} - c$, and $\psi_{\Gamma^+}$ is determined from a Neumann–Dirichlet map corresponding to the following system defined on $\Omega^+$:

$$\Delta\psi = 0, \tag{100}$$

$$\left.\frac{\partial\psi}{\partial n}\right|_{\Gamma^+} = \left.\frac{\partial\phi}{\partial n}\right|_{\Gamma^+} - \left.\frac{\partial\phi_0}{\partial n}\right|_{\Gamma^+} = \left[\frac{\partial\phi_0}{\partial n}\right]_\Gamma - \left.\frac{\partial\phi_0}{\partial n}\right|_{\Gamma^+} = -\left.\frac{\partial\phi_0}{\partial n}\right|_{\Gamma^-}, \tag{101}$$

$$\left.\frac{\partial\psi}{\partial n}\right|_B = 0. \tag{102}$$

The computation is done with $N_x \times N_y \times M_\alpha = n \times n \times n$, where $N_x$ and $N_y$ are the numbers of MAC grid cells along the $x$ and $y$ axes, respectively, and $M_\alpha$ is the number of Lagrangian points for $\Gamma$. The Neumann–Dirichlet map is solved using $L_B = 4n$ boundary elements on $B$ and $L_\Gamma = \frac{n}{2}$ on $\Gamma$. Table 1 summarizes the results of the

Table 1
Convergence analysis for the corrector-based Poisson solver

| $n$ | 30 | 60 | 120 | 240 | 480 |
|---|---|---|---|---|---|
| $\|e_\phi\|_\infty$ | $2.84 \times 10^{-3}$ | $9.19 \times 10^{-4}$ | $2.57 \times 10^{-4}$ | $5.92 \times 10^{-5}$ | $1.41 \times 10^{-5}$ |
| Order | – | 1.63 | 1.84 | 2.12 | 2.07 |

convergence analysis, indicating second-order convergence rate in the infinity norm. Because the computed solution $\phi$ is subject to a constant, the infinity norm $\|e_\phi\|_\infty$ in Table 1 is calculated by

$$\|e_\phi\|_\infty = \frac{\max(e_\phi) - \min(e_\phi)}{2}. \tag{103}$$

The order in Table 1 is calculated by

$$\text{order} = \frac{\ln(\|e_{\text{previous}}\|_\infty / \|e_{\text{current}}\|_\infty)}{\ln(n_{\text{current}}/n_{\text{previous}})}, \tag{104}$$

where $e_{\text{current}}$ and $e_{\text{previous}}$ denote the errors at the current and the previous columns in Table 1, respectively.

## 8. Numerical results

Numerical examples are provided in this section to test the accuracy, efficiency, and stability of the proposed immersed interface method. The effect of the corrector is also investigated in these examples.

### 8.1. Circular Couette flow

In the first example, steady circular Couette flow is simulated. The domain of the simulation and the geometry of the two rotating concentric cylinders are shown in Fig. 7. The domain is the rectangle of the size $l_x \times l_y$. The angular velocity of the inner and outer cylinders is denoted as $\Pi_1$ and $\Pi_2$, respectively. In the simulation, $r_1 = 0.5, r_2 = 2.0, l_x = l_y = 2, \Pi_1 = 1$ and $\Pi_2 = -1$. The temporal resolution of the simulation is controlled by the convective and viscous CFL numbers

$$\text{CFL}_c = \delta t\left(\frac{u_{\max}}{\delta x} + \frac{v_{\max}}{\delta y}\right), \tag{105}$$

$$\text{CFL}_\mu = \frac{\delta t}{Re}\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right). \tag{106}$$

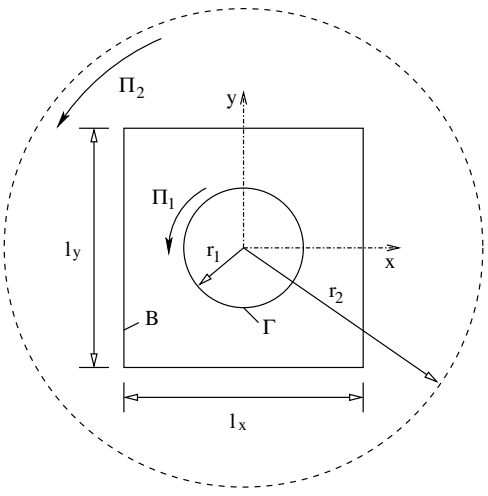



Fig. 7. Geometry and domain for the simulation of circular Couette flow.

The analytical solution of the steady flow between the two cylinders is given by

$$u = -\left(A_1 + \frac{A_2}{r^2}\right)y, \tag{107}$$

$$v = \left(A_1 + \frac{A_2}{r^2}\right)x, \tag{108}$$

$$p = \frac{A_1^2 r^2}{2} - \frac{A_2^2}{2r^2} + A_1 A_2 \cdot \ln(r^2) + p_0, \tag{109}$$

where $r^2 = x^2 + y^2, p_0$ is an arbitrary constant, and $A_1$ and $A_2$ are

$$A_1 = \frac{\Pi_2 r_2^2 - \Pi_1 r_1^2}{r_2^2 - r_1^2}, \tag{110}$$

$$A_2 = \frac{(\Pi_1 - \Pi_2) r_1^2 r_2^2}{r_2^2 - r_1^2}. \tag{111}$$

Dirichlet boundary conditions for the velocity and Neumann boundary conditions for the pressure are applied at the far-field boundary $B$ in Fig. 7, and they are obtained from the analytical solution. In the current numerical setup, only the inner cylinder $\Gamma$ is contained in the simulation domain, and its motion is enforced by the current explicit approach.

With $N_x \times N_y \times M_\alpha = n \times n \times n, L_B = \frac{4n}{3}$, and $L_\Gamma = \frac{n}{2}$, spatial convergence analysis is conducted at $Re = 10$ by altering $n$, where $N_x, N_y, M_\alpha, L_B$, and $L_\Gamma$ are defined in Section 7.2. The results are provided in Table 2, indicating near second-order accuracy in the infinity norm for the velocity and the pressure.

To test the stability of the method, the flow at a relatively high Reynolds number $Re = 2000$ is simulated with $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 64 \times 64 \times 64 \times 128 \times 32$, and $\mathrm{CFL}_c = \mathrm{CFL}_\mu = 0.8$. The simulation starts from a zero velocity field. After a relatively longer transient process, a steady flow state is reached, as shown in Fig. 8. The method is thus stable at this relatively high Reynolds number with the relatively large CFL numbers.

Table 2
Spatial convergence analysis for steady circular Couette flow

| $n$ | $\|e_u\|_\infty$ | Order | $\|e_v\|_\infty$ | Order | $\|e_p\|_\infty$ | Order |
|---|---|---|---|---|---|---|
| 30 | $1.56 \times 10^{-2}$ | – | $1.33 \times 10^{-2}$ | – | $2.00 \times 10^{-2}$ | – |
| 60 | $3.18 \times 10^{-3}$ | 2.29 | $3.19 \times 10^{-3}$ | 2.06 | $5.92 \times 10^{-3}$ | 1.76 |
| 120 | $1.05 \times 10^{-3}$ | 1.60 | $1.08 \times 10^{-3}$ | 1.56 | $1.78 \times 10^{-3}$ | 1.73 |
| 240 | $2.67 \times 10^{-4}$ | 1.98 | $2.65 \times 10^{-4}$ | 2.03 | $1.15 \times 10^{-3}$ | 0.63 |

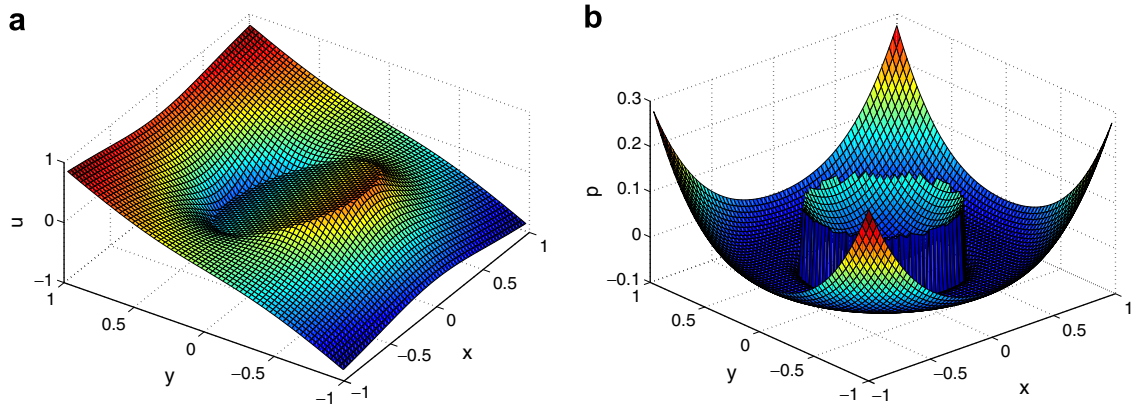



Fig. 8. Computed circular Couette flow at $Re = 2000$: (a) $u$ and (b) $p$.

Table 3
Spatial convergence analysis for steady circular Couette flow computed without the corrector

| $n$ | $\|e_u\|_\infty$ | Order | $\|e_v\|_\infty$ | Order | $\|e_p\|_\infty$ | Order |
|---|---|---|---|---|---|---|
| 30 | $1.54 \times 10^{-2}$ | – | $1.33 \times 10^{-2}$ | – | $3.52 \times 10^{-2}$ | – |
| 60 | $3.63 \times 10^{-3}$ | 2.08 | $3.62 \times 10^{-3}$ | 1.88 | $2.57 \times 10^{-2}$ | 0.45 |
| 120 | $1.10 \times 10^{-3}$ | 1.72 | $1.10 \times 10^{-3}$ | 1.72 | $2.48 \times 10^{-2}$ | 0.05 |
| 240 | $2.82 \times 10^{-4}$ | 1.96 | $3.05 \times 10^{-4}$ | 1.85 | $6.94 \times 10^{-2}$ | −1.48 |

The effect of the corrector on the simulation accuracy is investigated by turning it off. Similar to Table 2, the results of the convergence analysis without the corrector are shown in Table 3. The numerical errors for the velocity in Table 3 are about the same as those in Table 2, and the accuracy for the velocity is still second-order. However, the numerical errors for the pressure are not reduced by increasing the grid resolution.

### 8.2. Flow past a stationary cylinder

In the second example, flow past a stationary cylinder is simulated at $Re = 20, 40, 50, 100$, and 200. Shown in Fig. 9 are the geometry and domain of the base simulation. The base simulation is referred as Case $(n, l)$ and is used for plotting figures. The spatial resolution of the base computation for $Re = 20, 40$ and 50 is given by $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 960 \times 480 \times 256 \times 256 \times 128$, and for $Re = 100$ and 200 by $1600 \times 800 \times 256 \times 256 \times 128$. The effect of the spatial resolution and the domain size is investigated by reducing the discretization density to $\frac{N_x}{2} \times \frac{N_y}{2} \times \frac{M_\alpha}{2} \times \frac{L_B}{2} \times \frac{L_\Gamma}{2}$ in Case $(\frac{n}{2}, l)$ and by extending the domain size to $\frac{3l_x}{2} \times \frac{3l_y}{2}$ in Case $(n, \frac{3l}{2})$. The effect of the corrector is also investigated by turning it off in Case $\delta f_n = 0$. The time step of all the computation is controlled by $\mathrm{CFL}_c = \mathrm{CFL}_\mu = 0.5$.

A free stream with $u = 1$ enters the domain in the direction of the $x$ axis. The Neumann boundary condition $\frac{\partial p}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}$ is applied for the pressure at the inlet. At the two sides of the domain, symmetric boundary conditions are used. At the domain outlet, the boundary conditions $\frac{\partial \vec{v}}{\partial x} = 0$ and $\frac{\partial p}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}$ are used. The initial velocity field $\vec{v}_0$ is given by

$$\vec{v}_0 = \vec{U} - \nabla\chi, \tag{112}$$

where $-\nabla\chi$ is a divergence-free correction to the uniform velocity field $\vec{U} = (1, 0)$ to enforce the no-penetration condition on the cylinder, and $\chi$ satisfies

$$\Delta\chi = 0, \tag{113}$$

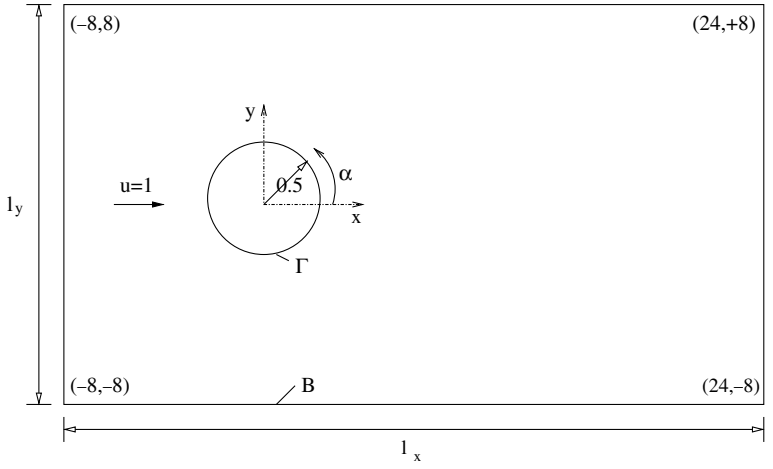



Fig. 9. Geometry and domain for the simulation of flow past a cylinder.

$$\left.\frac{\partial\chi}{\partial n}\right|_{\Gamma^+} = \vec{U}\cdot\vec{n}, \tag{114}$$

$$\left.\frac{\partial\chi}{\partial n}\right|_{B} = 0. \tag{115}$$

The corrector-based Poisson solver described in Section 7.2 is used to solve for $\chi$.

In general, the fluid force $\vec{F} = (F_x, F_y)$ applied by the fluid on $\Gamma^+$ can be calculated from

$$\vec{F} = -\int_\Gamma \vec{f}\,\mathrm{d}\alpha + S\frac{\mathrm{d}^2\vec{x}_c}{\mathrm{d}t^2} + S\left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2(\vec{x}_c - \vec{x}_{\mathrm{cm}}), \tag{116}$$

where $S$ is the area enclosed by $\Gamma$, and $\vec{x}_{\mathrm{cm}}$ is the coordinates of the geometric center (the center of mass) of $\Omega^-$. The drag and lift coefficients are defined as $C_x = 2F_x$ and $C_y = 2F_y$, respectively. The surface vorticity $\omega_s$ and the surface pressure $p_s$ on $\Gamma^+$ are calculated from

$$\omega_s = -Ref_\tau + \omega|_{\Gamma^-}, \tag{117}$$

$$p_s = f_n + p|_{\Gamma^-}, \tag{118}$$

where $\omega|_{\Gamma^-} = 2\frac{\mathrm{d}\theta}{\mathrm{d}t}$, and $p^-$ is given by Eq. (41). In the current example, the cylinder is stationary and $\vec{x}_c = \vec{x}_{\mathrm{cm}}$, so $C_x = -2\int_\Gamma f_x\mathrm{d}\alpha$, $C_y = -2\int_\Gamma f_y\,\mathrm{d}\alpha, \omega_s = -Ref_\tau$ and $p_s = f_n$ (subject to a constant).

*8.2.1. $Re = 20, 40,$ and 50*

At $Re = 20$ and 40, the flow reaches a steady state. Two recirculating bubbles are formed behind the cylinder, as shown in Fig. 10. The characteristics of the recirculating bubbles, the separation angle $\Theta$, and the drag coefficient $C_x$ are compared with experimental and other numerical results in Table 4, where the length

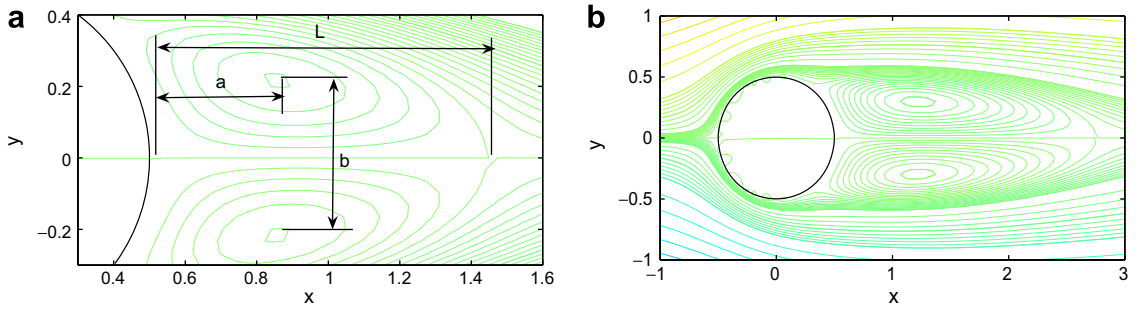



Fig. 10. Streamfunction contours for flow past a cylinder: (a) $Re = 20$ and (b) $Re = 40$.

Table 4
Summary of flow characteristics for flow past a cylinder at $Re = 20$ and $Re = 40$

| | $Re = 20$ | | | | | $Re = 40$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $L$ | $a$ | $b$ | $\Theta$ | $C_x$ | $L$ | $a$ | $b$ | $\Theta$ | $C_x$ |
| Ref. [33] | – | – | – | – | 2.22 | – | – | – | – | 1.48 |
| Ref. [5] | 0.93 | 0.33 | 0.46 | 45.0° | – | 2.13 | 0.76 | 0.59 | 53.8° | – |
| Ref. [6] | 0.94 | – | – | 43.7° | 2.05 | 2.35 | – | – | 53.8° | 1.52 |
| Ref. [8] | 0.91 | – | – | 45.7° | 2.00 | 2.24 | – | – | 55.6° | 1.50 |
| Ref. [37] | 0.92 | – | – | 44.2° | 2.23 | 2.21 | – | – | 53.5° | 1.66 |
| Ref. [22] | 0.93 | 0.36 | 0.43 | 43.9° | 2.16 | 2.23 | 0.71 | 0.59 | 53.4° | 1.61 |
| Case $(n, l)$ | 0.93 | 0.36 | 0.43 | 44.0° | 2.23 | 2.24 | 0.72 | 0.60 | 53.8° | 1.66 |
| Case $(\frac{n}{2}, l)$ | 0.90 | 0.38 | 0.42 | 43.2° | 2.24 | 2.21 | 0.73 | 0.56 | 52.9° | 1.70 |
| Case $(n, \frac{3l}{2})$ | 0.91 | 0.36 | 0.42 | 43.6° | 2.14 | 2.24 | 0.71 | 0.60 | 53.4° | 1.60 |
| Case $\delta f_n = 0$ | 0.93 | 0.36 | 0.43 | 44.0° | 2.14 | 2.24 | 0.72 | 0.60 | 53.8° | 1.60 |

$L$ and the location $(0.5 + a, \pm\frac{b}{2})$ of the recirculating bubbles are defined in Fig. 10. The current results lie well within the range of values given by the other data.

Between $Re = 40$ and 50, the flow becomes unstable. Round-off and other sources of numerical errors can destabilize the flow in the simulation. The time evolution of the drag and lift coefficients of the flow at $Re = 50$ is shown in Fig. 11, indicating that an unsteady state is reached after a considerably long time.

In Fig. 12, the surface vorticity and pressure on the cylinder are compared with the previous computational results by Braza et al. [2] for $Re = 20$ and 40. Very good agreement is obtained for the surface vorticity. The surface pressure behind the cylinder is slightly lower in the current simulation, which explains why the current drag coefficient is slightly higher.

The results from Case $\delta f_n = 0$, where the corrector is turned off, are also shown in Table 4. They agree with the others. Comparison of steady flow details near the cylinder computed with and without the corrector is shown in Fig. 13, indicating a close match.

*8.2.2. Re = 100 and 200*

At $Re = 100$ and 200, the flow is unsteady, and alternatively shedded vortices form the well-known Karman vortex street behind the cylinder. The time evolution of the drag and lift coefficients is shown in Fig. 14. Table 5 compares with previous numerical results the drag coefficient $C_x$, the lift coefficient $C_y$, and the Strouhal

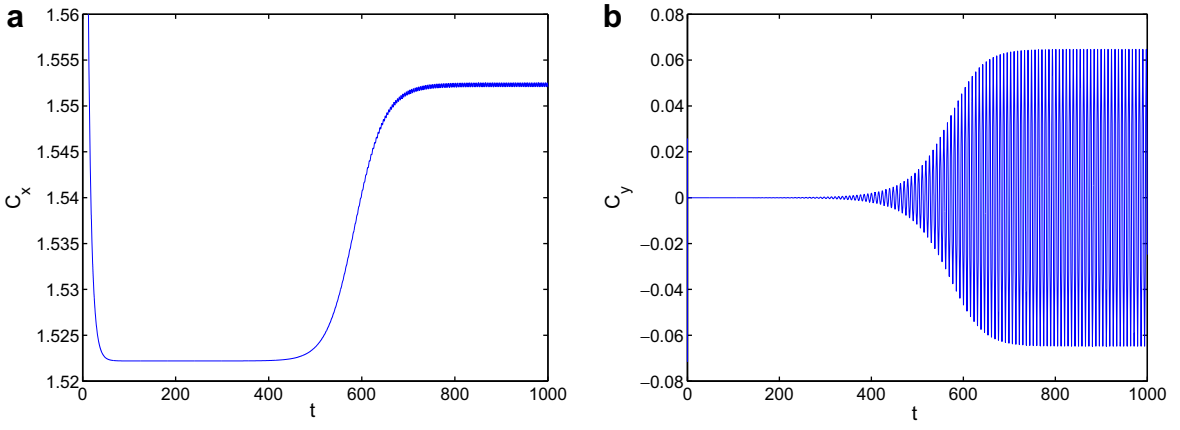



Fig. 11. (a) Drag and (b) lift coefficients versus time for flow past a cylinder at $Re = 50$.

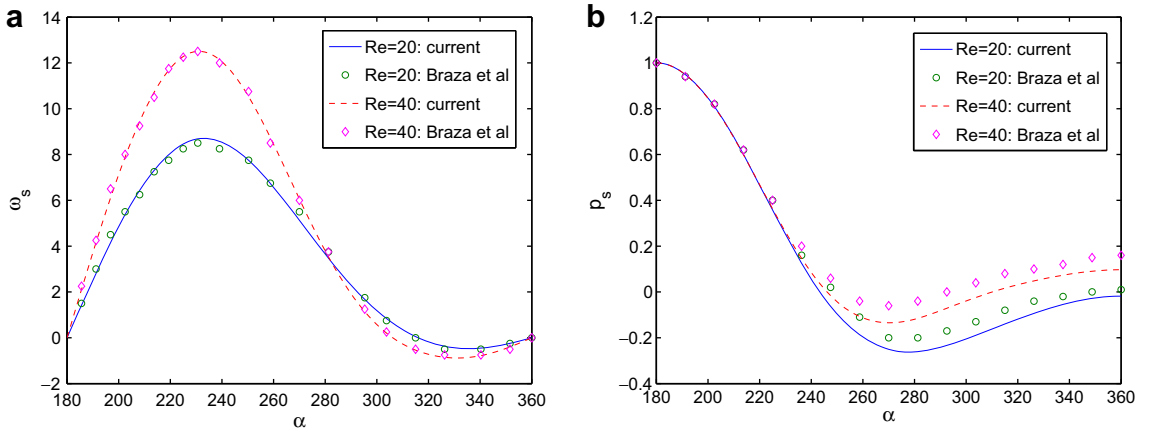
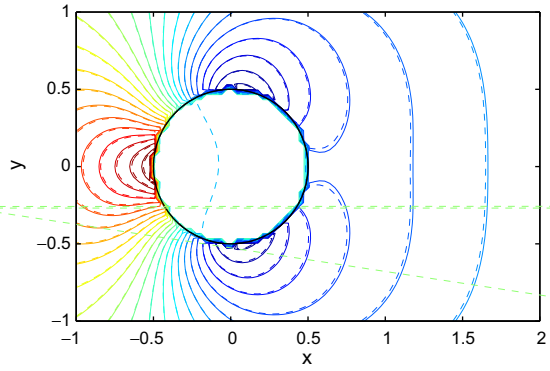



Fig. 12. (a) Surface vorticity and (b) pressure on a cylinder in flow at $Re = 20$ and $Re = 40$.




(a) $p$, contourlevels $=-1{:}0.05{:}1$

number $S_t$ (the nondimensional vortex shedding frequency). Good agreement is found again. The results from Case $\delta f_n = 0$ are shown in Table 5. They also agree with the others.

### *8.3. Flow around a hovering flapper*

In the third example, flow around a hovering rounded–rectangular flapper is simulated. The motion of the flapper is prescribed by

$$x_c = 1.25(\cos(0.8t) + 1)\cos\left(\frac{\pi}{3}\right), \tag{119}$$

$$y_c = 1.25(\cos(0.8t) + 1)\sin\left(\frac{\pi}{3}\right), \tag{120}$$

$$\theta = \frac{3\pi}{4} + \frac{\pi}{4}\sin(0.8t)(1 - \exp(-t)), \tag{121}$$

where a ramping process is applied in $\theta$ to avoid the impulsive rotation of the flapper. The geometry and domain for the simulation are shown in Fig. 15. The spatial resolution of the base simulation, Case $(n, \delta f_n \neq 0)$, is given by $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 512 \times 512 \times 256 \times 256 \times 128$. The effect of the spatial resolution is

Table 5
Summary of flow characteristics for flow past a cylinder at $Re = 100$ and $Re = 200$

| | $Re = 100$ | | | $Re = 200$ | | |
|---|---|---|---|---|---|---|
| | $C_x$ | $C_y$ | $S_t$ | $C_x$ | $C_y$ | $S_t$ |
| Ref. [2] | $1.36 \pm 0.015$ | $\pm 0.250$ | – | $1.40 \pm 0.050$ | $\pm 0.75$ | – |
| Ref. [31] | $1.43 \pm 0.009$ | $\pm 0.322$ | 0.172 | $1.45 \pm 0.036$ | $\pm 0.63$ | 0.201 |
| Ref. [37] | $1.42 \pm 0.013$ | $\pm 0.340$ | 0.171 | $1.42 \pm 0.040$ | $\pm 0.66$ | 0.202 |
| Ref. [22] | $1.38 \pm 0.010$ | $\pm 0.337$ | 0.169 | $1.37 \pm 0.046$ | $\pm 0.70$ | 0.199 |
| Ref. [15] | $1.37 \pm 0.009$ | $\pm 0.323$ | 0.160 | $1.34 \pm 0.030$ | $\pm 0.43$ | 0.200 |
| Case $(n, l)$ | $1.42 \pm 0.010$ | $\pm 0.353$ | 0.172 | $1.43 \pm 0.050$ | $\pm 0.71$ | 0.202 |
| Case $(\frac{n}{2}, l)$ | $1.47 \pm 0.010$ | $\pm 0.344$ | 0.172 | $1.57 \pm 0.044$ | $\pm 0.66$ | 0.201 |
| Case $(n, \frac{3l}{2})$ | $1.40 \pm 0.011$ | $\pm 0.328$ | 0.167 | $1.45 \pm 0.044$ | $\pm 0.67$ | 0.199 |
| Case $\delta f_n = 0$ | $1.38 \pm 0.012$ | $\pm 0.341$ | 0.172 | $1.46 \pm 0.054$ | $\pm 0.71$ | 0.202 |

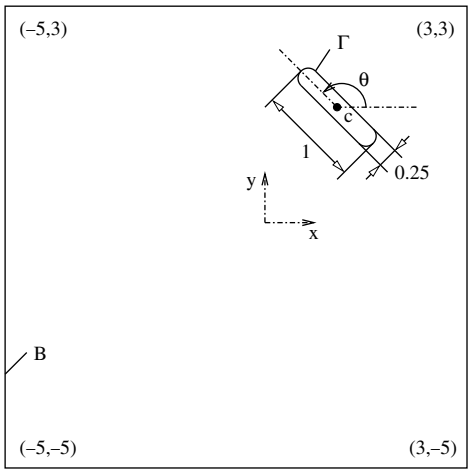



Fig. 15. Geometry and domain for the simulation of flow around a flapper.

investigated by reducing the discretization density to $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 256 \times 256 \times 128 \times 128 \times 64$ in Case $(\frac{n}{2}, \delta f_n \neq 0)$. The effect of the corrector is also investigated by turning it off in Case $(n, \delta f_n = 0)$ and Case $(\frac{n}{2}, \delta f_n = 0)$ correspondingly. Except where otherwise stated, a fixed time step $\delta t = 3.927 \times 10^{-3} \approx \frac{T_f}{2000}$ is used in the computation, where $T_f = \frac{2\pi}{0.8}$ is the flapping period of the flapper.

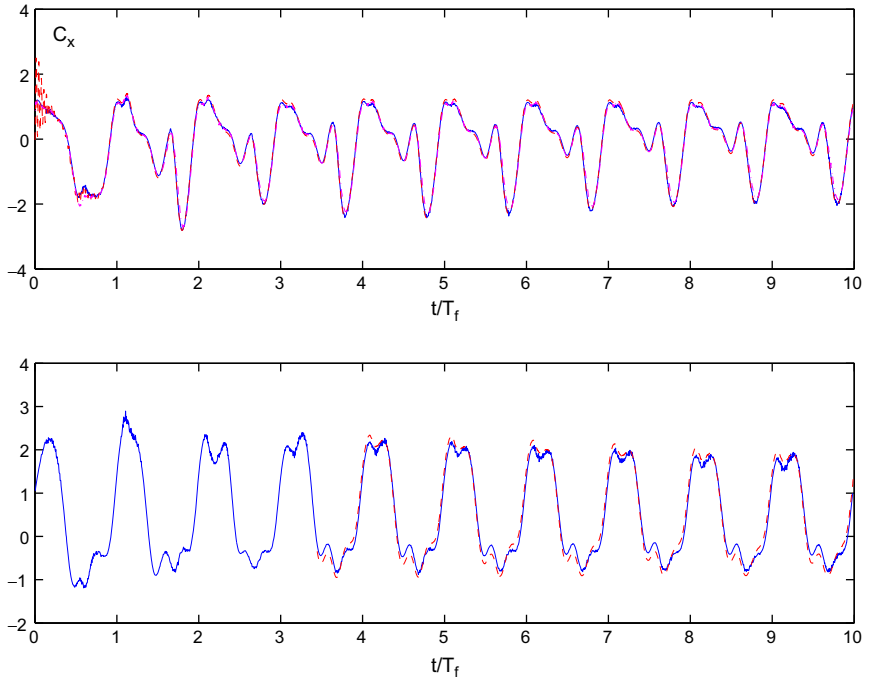
An alternative method described in [37] is also used to simulate the same flow with the same numerical setup. The simulation cases are called Case $(n, s)$ and Case $(\frac{n}{2}, s)$, corresponding to the above two spatial resolutions. The simulation results are compared below with those from the current immersed interface method. The alternative method is the immersed interface method with a spring model for the singular force. In Case $(n, s)$ and Case $(\frac{n}{2}, s)$, the spring model is

$$\vec{f} = K_s(\vec{X}_e - \vec{X}), \tag{122}$$

where the spring stiffness $K_s = 160$. Extensive testing has been done in [37,1] on this alternative method. It is found that an $N_x \times N_y \times M_\alpha = 256 \times 256 \times 128$ grid sufficiently resolves the current flow.

*8.3.1. Re = 157*

Flow around a hovering elliptic flapper of the same aspect ratio with the same kinematics and the same Reynolds number has been simulated in [37,35], where simulation results are compared between the alternative method and a coordinate transformation method. Here, comparison of results is made between the current base simulation (Case $(n, \delta f_n \neq 0)$), the simulation by the alternative method (Case $(n, s)$), and the simulation without the corrector (Case $(n, \delta f_n = 0)$).




In Fig. 16, the time evolution of the drag and lift coefficients is compared. Very good agreement is achieved. The high-frequency oscillations at the beginning of the force curves from the alternative method are caused by the spring model. Comparison of flow fields at $t = 10T_f$ is given in Fig. 17, and comparison of flow details very near the flapper at the same instant is given in Fig. 18. Agreement for the velocity and vorticity fields is very good. The pressure fields close to the flapper have some discrepancies. Overall, all the comparison shows good agreement. In Case $(n, \delta f_n \neq 0)$, the pressure inside the flapper is compatible with the enforced rigid motion of the fluid enclosed by the flapper. In Case $(n, \delta f_n = 0)$, the corrector is turned off, and the compatibility may be violated. In Case $(n, s)$, the motion of the enclosed fluid naturally results from the boundary motion of the flapper, and it is not artificially enforced to be in the rigid motion. Apparent differences in the pressure fields inside the flapper are therefore expected, as indicated in Fig. 18.

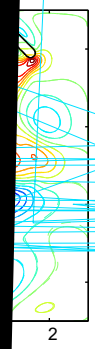
Fig. 19 shows the time evolution of the drag and lift coefficients computed with the corrector at different grid resolution. It is found that an $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 256 \times 256 \times 128 \times 128 \times 64$ grid can sufficiently resolve the flow, which is also confirmed by comparison of flow details at a specified instant (not shown here).

Shown in Fig. 20 are results of the grid refinement study without the corrector. The peak values of both the drag and lift curves are underpredicted at the lower grid resolution in Fig. 20. Nevertheless, the flow can be simulated with fidelity at the higher resolution, as demonstrated previously in Figs. 16 and 17.

*8.3.2. $Re = 1000$*

To test the stability of the current method, a relatively high Reynolds number $Re = 1000$ is considered with $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 1024 \times 1024 \times 256 \times 256 \times 128$ and $\mathrm{CFL}_c = \mathrm{CFL}_\mu = 0.6$. Stable simulation is achieved by the method. The alternative method [37] with a stiff spring model is unstable to simulate the flow at this high Reynolds number with the same spatial and temporal resolution.

Fig. 21 compares the time evolution of the drag and lift coefficients at $Re = 1000$ and $Re = 157$ in the first 5 flapping periods. Fig. 22 plots the flow fields at $Re = 1000$ and $t = 5T_f$.




1 : 0.1 : 1

current method to handle multiple moving objects is investigated by flappers. The simulation domain and the flapper geometry are the ppers, $n$, varies from 1 to 4. The motion of the flapper $l$ is given by

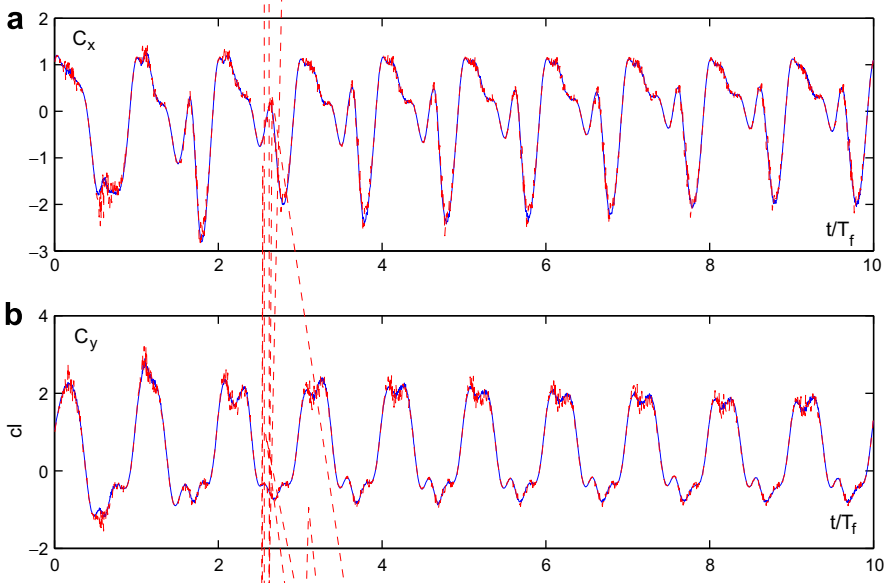
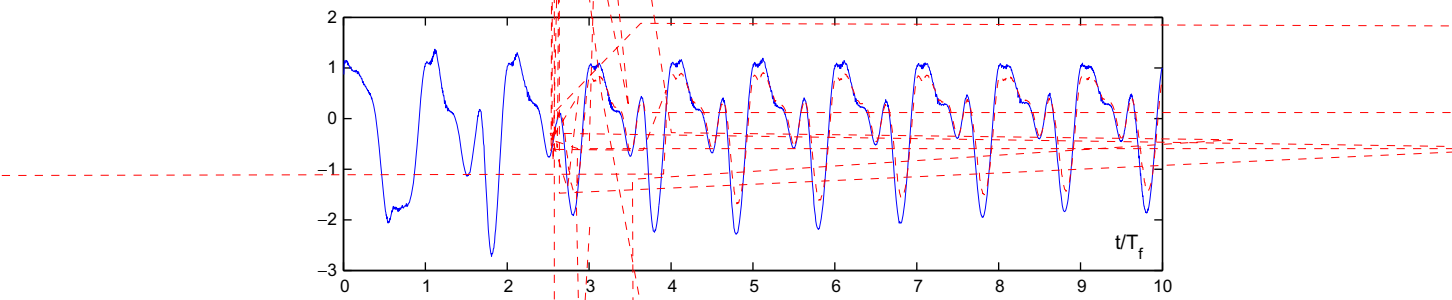



Fig. 19. (a) Drag and (b) lift coefficients versus time for flow around a flapper at $Re = 157$. solid lines: Case $(n, \delta f_n \neq 0)$, dashed lines: Case $(\frac{n}{2}, \delta f_n \neq 0)$.




$$x_c(l) = x_{c0}(l) + 1.25(\cos(0.8t) + 1)\cos\left(\frac{\pi}{3}\right), \tag{123}$$

$$y_c(l) = y_{c0}(l) + 1.25(\cos(0.8t) + 1)\sin\left(\frac{\pi}{3}\right), \tag{124}$$

$$\theta(l) = \frac{3\pi}{4} + \frac{\pi}{4}\sin(0.8t)(1 - \exp(-t)), \tag{125}$$

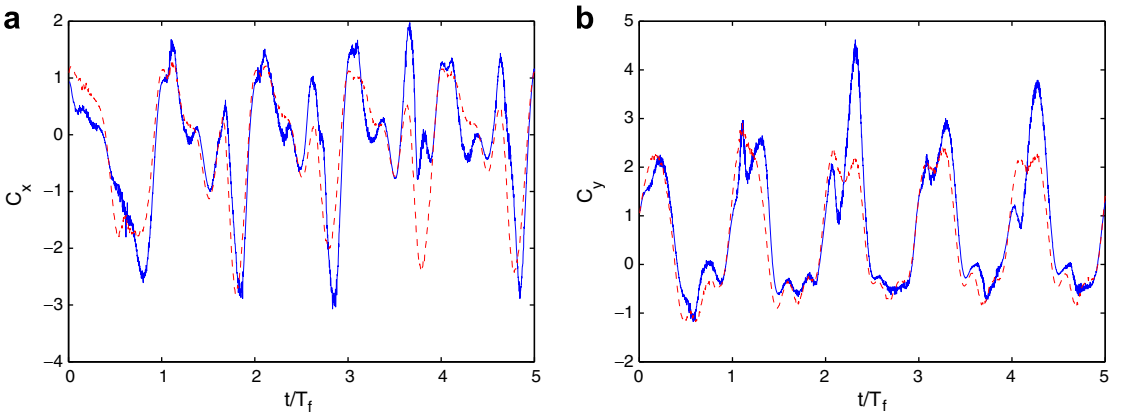
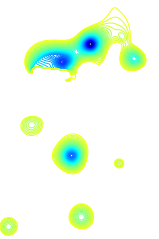



Fig. 21. (a) Drag and (b) lift coefficients versus time for flow around a flapper. solid lines: $Re = 1000$, dashed lines: $Re = 157$.



where $(x_{c0}(l), y_{c0}(l)) = (0,0), (0,-3), (-3,0)$, and $(-3, -3)$ for $l = 1,2,3,4$, respectively. The spatial resolution of the simulation is given by $N_x \times N_y \times M_\alpha \times L_B \times L_\Gamma = 512 \times 512 \times 256 \times 256 \times 128$, where $M_\alpha$ and $L_\Gamma$ are the numbers of Lagrangian markers and boundary elements for one flapper, respectively. A fixed time step $\delta t = 3.927 \times 10^{-3} \approx \frac{T_f}{2000}$ is used in the current simulation, where $T_f = \frac{2\pi}{0.8}$ is the flapping period.

Table 6 gives the relative computational time spent on 4000 time steps (two flapping periods) with and without the corrector for the different number of flappers. Without the corrector, each flapper is handled in $\mathcal{O}(M_\alpha)$ time, the computational time scales linearly with the number of the flappers, and multiple moving objects can be handled very efficiently. Extra computational cost associated with the corrector (the time difference in Table 6) involves solving a Neumann–Dirichlet map and a Laplace equation. In the current method, the Neumann–Dirichlet map is solved by the boundary element method in $\mathcal{O}(L^3)$, where $L = L_B + n \cdot L_\Gamma$ represents the total number of boundary elements. The computational time therefore increases significantly as the number of

flappers, $n$, increases. As pointed out in Section 7.1, if all objects in flow are stationary, the cost to solve the Neumann–Dirichlet map in every time step can be cut to $\mathcal{O}(L)$ just for backward substitution. It is also possible to use a multipole method to solve the Neumann–Dirichlet map in $\mathcal{O}(L)$ time for moving boundaries. In either case, the method can handle multiple objects efficiently.

## 9. Conclusions

In this paper, an explicit approach is proposed to enforce the prescribed motion of a rigid object in the immersed interface method. The motion of the object boundary is generated by a singular force in the Navier–Stokes equations. The tangential component of the singular force is related to the surface vorticity and is calculated from the normal derivative of the velocity. The normal component of the singular force is determined from a predictor and a corrector. The predictor uses the normal derivative of the vorticity. The corrector superposes a homogeneous solution to the pressure Poisson equation to achieve the desired normal derivative of the pressure. In general, the corrector can be used as a Poisson solver for Poisson equations with Neumann, Dirichlet or mixed data on embedded irregular boundaries.

It is simple to implement the current explicit approach in existing codes of the immersed interface method. Grid convergence study shows that the current immersed interface method with this approach has achieved near second-order accuracy in the infinity norm for the velocity and the pressure. The method is stable to simulate relatively high Reynolds number flow with large CFL numbers by eliminating the need of any ad hoc constitutive laws or feedback control and associated stiffness. Without iterative solvers, it is efficient for moving rigid boundaries, especially if the corrector is turned off when the spatial resolution is sufficient. There are no fundamental obstacles to extend the current method to 3D. A 3D version of the method is under development and will be reported in the future.

## Acknowledgments

The author thanks Professor Jane Wang at Cornell University for her support on this work. The author also thanks Professor Johannes Tausch at Southern Methodist University and the anonymous reviewers for their helpful comments.

## References

[1] Attila J. Bergou, Sheng Xu, Z. Jane Wang, Passive wing pitch reversal in insect flight, J. Fluid Mech. 591 (2007) 321–337.
[2] M. Braza, P. Chassaing, H. Ha Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79–130.
[3] Donna Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction–vorticity equations in irregular regions, J. Comput. Phys. 176 (2002) 231–275.
[4] R. Cortez, C.S. Peskin, J. Stockie, D. Varela, Parametric resonance in immersed elastic boundaries, SIAM J. Appl. Math. 65 (2004) 494–520.
[5] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (2) (1977) 231–256.
[6] S.C.R. Dennis, Gau-Zu Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, J. Fluid Mech. 42 (3) (1970) 471–489.
[7] E. Weinan, Jian-Guo Liu, Vorticity boundary condition and related issues for finite difference schemes, J. Comput. Phys. 124 (1996) 368–382.
[8] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (4) (1980) 819–855.
[9] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (1993) 354–366.
[10] A. Greenbaum, L. Greengard, G.B. McFadden, Laplace's equation and the Dirichlet–Neumann map in multiply connected domains, J. Comput. Phys. 105 (1993) 267–278.
[11] Francis H. Harlow, J. Eddie Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
[12] Hans Johnston, Jian-Guo Liu, Finite difference schemes for incompressible flow based on local pressure boundary conditions, J. Comput. Phys. 180 (2002) 120–154.
[13] Hans Johnston, Jian-Guo Liu, Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term, J. Comput. Phys. 199 (2004) 221–259.

[14] Ming-Chih Lai, Charles S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.
[15] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (2006) 109–138.
[16] Long Lee, Randall J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (3) (2003) 832–856.
[17] Randall J. LeVeque, Zhilin Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044.
[18] Randall J. LeVeque, Zhilin Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (3) (1997) 709–735.
[20] Zhilin Li, Kazufumi Ito, The immersed interface method – numerical solutions of PDEs involving interfaces and irregular domains, SIAM Frontiers Appl. Math. 33 (2006), ISBN: 0-89971-609-8.
[21] Z. Li, X. Wan, K. Ito, S.R. Lubkin, An augmented approach for the pressure boundary condition in a Stokes flow, Comm. Comp. Phys. 1 (2006) 874–885.
[22] Mark N. Linnick, Hermann F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.
[24] Anita Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (2) (1984) 285–299.
[25] Charles S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[26] Charles S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[27] Charles S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[28] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in Fortran 77: The Art of Scientific Computing, second ed., Cambridge University Press, 1999, pp. 848–852.
[29] Dietmar Rempfer, On boundary conditions for incompressible Navier–Stokes problems, Appl. Mech. Rev. 59 (3) (2006) 107–125.
[31] David Russell, Z. Jane Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (2003) 177–205.
[32] John M. Stockie, Brian R. Wetton, Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes, J. Comput. Phys. 154 (1999) 41–64.
[33] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, J. Fluid Mech. 6 (4) (1959) 547–567.
[34] C. Tu, C.S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries, SIAM J. Statist. Comput. 13 (1992) 70–83.
[35] Z. Jane Wang, Two dimensional mechanism for insect hovering, Phys. Rev. Lett. 85 (10) (2000) 2216–2219.
[36] Sheng Xu, Z. Jane Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, SIAM J. Sci. Comput. 27 (6) (2006) 1948–1980.
[37] Sheng Xu, Z. Jane Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2) (2006) 454–493.
[38] Sheng Xu, Z. Jane Wang, A 3D immersed interface method for fluid–solid interaction, Comput. Meth. Appl. Mech. Eng., in press.