Journal of Computational Physics 228 (2009) 1504–1520

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Immersed boundary method for the simulation of 2D viscous flow based on vorticity–velocity formulations

Zeli Wang, Jianren Fan *, Kefa Cen

State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, PR China

ARTICLE INFO

Article history:
Received 30 May 2008
Received in revised form 2 September 2008
Accepted 28 October 2008
Available online 12 November 2008

Keywords:
Immersed boundary method
Multi-direct forcing
Vorticity–velocity formulations
Incompressible viscous flow

ABSTRACT

A new immersed boundary method based on vorticity–velocity formulations for the simulation of 2D incompressible viscous flow is proposed in present paper. The velocity and vorticity are respectively divided into two parts: one is the velocity and vorticity without the influence of the immersed boundary, and the other is the corrected velocity and the corrected vorticity derived from the influence of the immersed boundary. The corrected velocity is obtained from the multi-direct forcing to ensure the well satisfaction of the no-slip boundary condition at the immersed boundary. The corrected vorticity is derived from the vorticity transport equation. The third-order Runge–Kutta for time stepping, the fourth-order finite difference scheme for spatial derivatives and the fourth-order discretized Poisson for solving velocity are applied in present flow solver. Three cases including decaying vortices, flow past a stationary circular cylinder and an in-line oscillating cylinder in a fluid at rest are conducted to validate the method proposed in this paper. And the results of the simulations show good agreements with previous numerical and experimental results. This indicates the validity and the accuracy of present immersed boundary method based on vorticity–velocity formulations.

1. Introduction

Recent years, numerous numerical techniques are developed for the simulation of the interaction between fluid and an embedded body [1–5]. Especially, the immersed boundary method (IB), originally developed by Peskin [6] to investigate the fluid dynamics of blood flow in human heart has attracted considerable interests in the last few years [7]. In Peskin's cases, the singular force on the Lagrangian coordinates of the boundary is known and the effect of the force is spread to the flow field with the Eulerian coordinates via a regularized Dirac delta function. The primary advantage of the immersed boundary method based on Cartesian mesh is that the task of grid generation is greatly simplified.

Following the lead of Peskin [6], many variants of IB method have been developed. To calculate the interactions between solid boundary and fluid, Goldstein et al. [8] proposed a feedback scheme to iteratively determine the magnitude of the force required to obtain a desired velocity on the immersed boundary. Saiki and Biringen [9] implemented this feedback scheme with the virtual boundary method to compute the flow past a stationary, rotating and oscillating circular cylinder, but the feedback forcing induces oscillations and the computational time step is restricted for numerical stability. Mohd-Yusof [10] and Fadlun et al. [11] proposed an approach to calculate the interaction force between the immersed boundary and fluid called direct forcing scheme. The velocity at the points which are close to the immersed boundary is simply set to the desired

* Corresponding author. Tel.: +86 571 87951764.
E-mail addresses: zjuzeliwang@gmail.com (Z. Wang), fanjr@zju.edu.cn (J. Fan).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.10.038

velocity at every time-step. It seems like applying an equivalent forcing term to the Navier–Stokes equations. Compared to the feedback forcing, the direct forcing is more general because there is no unknown constant that should be determined in the formulations of solving the direct forcing. In order to allow for a smooth transfer between Eulerian and Lagrangian representations while at the same time avoiding strong restrictions of the time step, Uhlmann [12] proposed an improved method to incorporate the regularized delta function approach into a direct formulation of the fluid–solid interaction force. Wang et al. [13] and Luo et al. [14] proposed an explicit multi-direct forcing approach based on the Peskin's original IB method [6] to obtain a better satisfaction of the no-slip boundary condition at the immersed boundary of rigid body than that of the original direct forcing approach. Feng and Michaelides [15] introduced a numerical method called *Proteus* combined the immersed boundary method, the direct forcing method and lattice Boltzmann method for the simulation of particulate flows. Lee et al. [16] proposed the immersed finite element method for solving fluid–structure interaction problems. Kim et al. [17] present the IB method for simulating flows over complex geometries by introducing a mass source/sink as well as a momentum forcing based on a finite volume approach.

For computing the moving object, one challenge is calculating the trajectory of the objects smoothly or continuously. This problem can be handled by applying a smooth regularized delta function (or interpolation/extrapolation function) as demonstrated by Uhlmann [12]. However, in pressure-velocity formulations, the pressure is an implicit variable in the governing equations and the specification of the boundary conditions of pressure is necessary. The un-convergence of the coupling of the pressure field at the early stage of the computation can lead to the oscillation of the forcing exerting on the moving object. The oscillation of the trajectory of the object at the early stage of the computation occurs.

For vorticity formulations, the pressure field is eliminated by taking the curl of the Navier–Stokes equations [18,20]. And the oscillation of the implicit pressure at the early stage of the computation can be avoided. We expect a smooth solution of the forcing exerting on the moving object by applying the immersed boundary method based on vorticity formulations, despite needing the solutions of one more and two more Poisson equations respectively for 2D and 3D computation than that for velocity–pressure formulations.

The IB method is mainly based on the velocity–pressure formulations and seldom on the vorticity formulations. Calhoun [29] described a method for solving the two-dimensional Navier–Stokes equations in irregular physical domain based on uniform Cartesian gird, finite-difference/finite-volume discretizations of the streamfunction–vorticity equations. At the embedded boundary, Calhoun [29] determined the discontinuity conditions of the vorticity and streamfunction needed to impose the no-slip flow conditions. The correction terms of the vorticity and streamfunction at the embedded boundary are coupled in the Poisson-like equations and solved by applying the immersed interface method introduced by LeVeque and Li [41]. The calculation of the discontinuities needs solving a rather large linear system and the linear system must be regenerated at every time step for moving object. This is time-consuming for computing moving object.

Russell and Wang [19] presented a method for solving 2D incompressible viscous flows around multiple moving objects based on the streamfunction–vorticity formulations and with discontinuities representing the embedded objects. Instead of using a linear system to couple all the variables involved as Calhoun [29], Russell and Wang [19] calculates the values of the discontinuities in separate steps. The no-penetration condition for the moving geometry is satisfied by superposing a homogenous solution to the Poisson's equation for the streamfunction. The no-slip condition is satisfied by generating vorticity on the surfaces of the objects with the Thom's formula. And the immersed interface method introduced by LeVeque and Li [41] also is applied for solving Poisson-like equations.

In present paper, we develop a new immersed boundary method based on vorticity–velocity formulations for the simulation of 2D incompressible viscous flow. The spirit of the original immersed boundary proposed by Peskin [6] is applied. It means that the Lagrangian points are used to represent the boundary of the embedded object and the regularized delta function is applied for the interpolation/extrapolation process between Lagrangian points and Eulerian grids. This is different from the methods of Calhoun [29] and Russell and Wang [19]. The multi-direct forcing scheme proposed by Wang et al. [13] and Luo et al. [14] is applied to ensure the satisfaction of the no-slip boundary condition at the immersed boundary. And the vorticity at the immersed boundary is corrected from the conservation equations. To verify present method, the test cases including decaying vortices, flow past a stationary circular cylinder and an in-line oscillating cylinder in a fluid at rest are conducted in this paper.

The paper is organized as follows: Section 2 introduces the IB method based on vorticity–velocity formulations including the basic governing equations (Section 2.1), the IB method with direct forcing based on vorticity–velocity formulations (Section 2.2), the multi-direct forcing process (Section 2.3) and the flow solving process (Section 2.4). Section 3 is the numerical validation of present method including decaying vortices (Section 3.1), flow past a stationary circular cylinder (Section 3.2) and an in-line oscillating cylinder in a fluid at rest (Section 3.3). And Section 4 is the summary of conclusions.

2. Numerical schemes

2.1. Governing equations

The two-dimensional dimensionless Navier–Stokes equations for incompressible viscous flow in the whole computational domain Ω with immersed boundary method are

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+f_x \tag{1}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+f_y \tag{2}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{3}$$

where $\boldsymbol{u} = (u, v)$ is the velocity of fluid, p is the pressure, Re is the Reynolds number defined as $Re=\frac{\rho \cdot U \cdot L}{\mu}$. Here ρ is the density of fluid, Re is the characteristic velocity of flow field, Re is the characteristic length of flow field and μ is the viscosity of fluid. $\boldsymbol{f} = (f_x, f_y)$ in Eqs. (1) and (2) is the external force exerted on the flow field which represents the mutual interaction force between fluid and immersed boundary expressed as following:

$$\boldsymbol{f}(\boldsymbol{x})=\int_{\Omega}\boldsymbol{F}_k(\boldsymbol{x}_k)\cdot\delta(\boldsymbol{x}-\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k \tag{4}$$

where $\delta(\boldsymbol{x}-\boldsymbol{x}_k)$ is the Dirac delta function, $\boldsymbol{x}_k$ is the position of the Lagrangian points set at the immersed boundary, $\boldsymbol{x}$ is the position of the computational Eulerian mesh and $\boldsymbol{F}_k(\boldsymbol{x}_k)$ is the force exerted on the Lagrangian point $\boldsymbol{x}_k$.

The vorticity is defined as

$$\omega=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y} \tag{5}$$

and the dimensionless vorticity–velocity formulations of the Navier–Stokes equations for incompressible viscous flow in the entire computational domain Ω including the immersed boundary are:

$$\frac{\partial \omega}{\partial t}+u\frac{\partial \omega}{\partial x}+v\frac{\partial \omega}{\partial y}=\frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2}+\frac{\partial^2 \omega}{\partial y^2}\right)+\frac{\partial f_y}{\partial x}-\frac{\partial f_x}{\partial y} \tag{6}$$

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=-\frac{\partial \omega}{\partial y} \tag{7}$$

$$\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}=\frac{\partial \omega}{\partial x} \tag{8}$$

The Poisson equations for solving velocity in Eqs. (7) and (8) are obtained from the continuity equation Eq. (3) and vorticity definition equation Eq. (5).

2.2. Immersed boundary method with direct forcing based on vorticity–velocity formulations

The direct forcing technique based on velocity–pressure formulations or primitive variables (Eqs. (1) and (2)) has been studied extensively [10–12,15]. In order to let the velocity on the Lagrangian points at the immersed boundary satisfy the no-slip boundary condition, a forcing $\boldsymbol{F}_k$ $(\boldsymbol{x}_k)$ is imposed on the Lagrangian point to modify its velocity equal the desired velocity $\boldsymbol{u}_L$. The forcing $\boldsymbol{F}_k(\boldsymbol{x}_k)$ is determined as follows.

From the momentum equation (Eqs. (1) and (2)) of the flow field, one can get

$$\boldsymbol{f}=\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\nabla P-\frac{1}{Re}\nabla^2\boldsymbol{u}=\frac{\partial \boldsymbol{u}}{\partial t}-\boldsymbol{rhs}=\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\Delta t}-\boldsymbol{rhs} \tag{9}$$

where n and $n + 1$ represent two different time and $\boldsymbol{rhs}=-(\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\nabla P-\frac{1}{Re}\nabla^2\boldsymbol{u})$.

And for the Lagrangian point $\boldsymbol{x}_k$ at the immersed boundary, one can get

$$\boldsymbol{F}_k(\boldsymbol{x}_k)=\frac{\boldsymbol{u}_k^{n+1}-\boldsymbol{u}_k^n}{\Delta t}-\boldsymbol{rhs}=\frac{\boldsymbol{u}_k^{n+1}-\hat{\boldsymbol{u}}_k}{\Delta t}+\frac{\hat{\boldsymbol{u}}_k-\boldsymbol{u}_k^n}{\Delta t}-\boldsymbol{rhs} \tag{10}$$

where $\hat{\boldsymbol{u}}_k$ is a temporary parameter which satisfies the common momentum equation at the immersed boundary, that is

$$\frac{\hat{\boldsymbol{u}}_k-\boldsymbol{u}_k^n}{\Delta t}-\boldsymbol{rhs}=0 \tag{11}$$

Therefore, the forcing exerted on the Lagrangian points at the immersed boundary is

$$\boldsymbol{F}_k(\boldsymbol{x}_k)=\frac{\boldsymbol{u}_k^{n+1}-\hat{\boldsymbol{u}}_k}{\Delta t}=\frac{\boldsymbol{u}_L-\hat{\boldsymbol{u}}_k}{\Delta t} \tag{12}$$

Under the effect of the forcing, the velocity on the Lagrangian point $\boldsymbol{x}_k$ at $n + 1$ time $\boldsymbol{u}_k^{n+1}$ can be modified to the desired velocity $\boldsymbol{u}_L$. And the Dirac delta function is applied in the two-way coupling between Eulerian girds and Lagrangian points. The temporary velocity on the Lagrangian point at the immersed boundary $\boldsymbol{x}_k$ is obtained from its surrounding Eulerian grids $\boldsymbol{x}$ as following:

$$\hat{\boldsymbol{u}}_k = \sum_{\boldsymbol{x}\in\Omega} \hat{\boldsymbol{u}} \cdot \delta_h(\boldsymbol{x}_k - \boldsymbol{x}) \cdot h^2 \tag{13}$$

where $\hat{\boldsymbol{u}}$ is also the temporary parameter on the Eulerian grids which satisfies the common momentum equation. The effect of the forcing is spread from the Lagrangian points to the Eulerian grids through the following equation,

$$\boldsymbol{f}(\boldsymbol{x}) = \int_\Omega \boldsymbol{F}_k(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k = \sum_{k=1}^{N} \boldsymbol{F}_k(\boldsymbol{x}_k) \cdot \delta_h(\boldsymbol{x} - \boldsymbol{x}_k) \cdot \Delta V_k \tag{14}$$

where N is the number of Lagrangian points, and ΔV_k is the discrete volume for each Lagrangian point.

For the vorticity–velocity formulations (Eqs. (6)–(8)), the velocity of the flow field is calculated through Poisson equations (Eqs. (7) and (8)). Based on the concept of the immersed boundary method in velocity–pressure formulations, we propose the following scheme with direct forcing and vorticity–velocity formulations.

The vorticity ω is divided into $\omega = \hat{\omega} + \omega'$. The variable $\hat{\omega}$ is the vorticity without the influence of the immersed boundary. And the variable ω' is the corrected vorticity due to the effect of the immersed boundary. The velocity is also divided into $\boldsymbol{u} = \hat{\boldsymbol{u}} + \boldsymbol{u}'$, the variable $\hat{\boldsymbol{u}} = (\hat{u}, \hat{v})$ is the velocity without the influence of the immersed boundary. And the variable $\boldsymbol{u}' = (u', v')$ is the corrected velocity due to the effect of the immersed boundary.

And the vorticity transport equations (Eq. (6)) can be written as

$$\frac{\partial(\hat{\omega}+\omega')}{\partial t} + (\hat{u}+u')\frac{\partial(\hat{\omega}+\omega')}{\partial x} + (\hat{v}+v')\frac{\partial(\hat{\omega}+\omega')}{\partial y} = \frac{1}{Re}\left[\frac{\partial^2(\hat{\omega}+\omega')}{\partial x^2} + \frac{\partial^2(\hat{\omega}+\omega')}{\partial y^2}\right] + \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \tag{15}$$

or

$$\frac{\partial\hat{\omega}}{\partial t} + \hat{u}\frac{\partial\hat{\omega}}{\partial x} + \hat{v}\frac{\partial\hat{\omega}}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2\hat{\omega}}{\partial x^2} + \frac{\partial^2\hat{\omega}}{\partial y^2}\right) \tag{16}$$

$$\frac{\partial\omega'}{\partial t} + \left[(\hat{u}+u')\frac{\partial\omega'}{\partial x} + (\hat{v}+v')\frac{\partial\omega'}{\partial y}\right] + \left(u'\frac{\partial\hat{\omega}}{\partial x} + v'\frac{\partial\hat{\omega}}{\partial y}\right) = \frac{1}{Re}\left(\frac{\partial^2\omega'}{\partial x^2} + \frac{\partial^2\omega'}{\partial y^2}\right) + \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \tag{17}$$

And the Poisson equations for solving the velocity in Eqs. (7) and (8) can be divided into the following equations.

$$\frac{\partial^2\hat{u}}{\partial x^2} + \frac{\partial^2\hat{u}}{\partial y^2} = -\frac{\partial\hat{\omega}}{\partial y} \tag{18}$$

$$\frac{\partial^2\hat{v}}{\partial x^2} + \frac{\partial^2\hat{v}}{\partial y^2} = \frac{\partial\hat{\omega}}{\partial x} \tag{19}$$

$$\frac{\partial^2 u'}{\partial x^2} + \frac{\partial^2 u'}{\partial y^2} = -\frac{\partial\omega'}{\partial y} \tag{20}$$

$$\frac{\partial^2 v'}{\partial x^2} + \frac{\partial^2 v'}{\partial y^2} = \frac{\partial\omega'}{\partial x} \tag{21}$$

If there is not any immersed boundary in the flow field, the vorticity and velocity of the whole flow field can be calculated from Eqs. (16) (18) and (19). And if there is immersed boundary in the flow field, the variables ω' and $\boldsymbol{u}' = (u', v')$ are not zero due to the effect of immersed boundary on the flow field.

Based on the concept of direct forcing, the velocity of the whole flow field will be modified by the forcing $\boldsymbol{f} = (f_x, f_y)$ to ensure the satisfaction of no-slip boundary at the immersed boundary.

$$\frac{\boldsymbol{u}_k^{n+1} - \hat{\boldsymbol{u}}_k}{\Delta t} = \boldsymbol{F}_k(\boldsymbol{x}_k) = \frac{\boldsymbol{u}_L - \hat{\boldsymbol{u}}_k}{\Delta t} \tag{22}$$

$$\frac{\boldsymbol{u}^{n+1} - \hat{\boldsymbol{u}}}{\Delta t} = \boldsymbol{f}(\boldsymbol{x}) = \int_\Omega \boldsymbol{F}_k(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k \tag{23}$$

where $\hat{\boldsymbol{u}} = (\hat{u}, \hat{v})$ is the velocity without the influence of the immersed boundary and can be obtained from Eqs. (18) and (19). Then the corrected velocity $\boldsymbol{u}' = (u', v')$ can be calculated as

$$\frac{\boldsymbol{u}'}{\Delta t} = \frac{\boldsymbol{u}^{n+1} - \hat{\boldsymbol{u}}}{\Delta t} = \boldsymbol{f}(\boldsymbol{x}) = \int_\Omega \boldsymbol{F}_k(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k \tag{24}$$

or

$$\boldsymbol{u}' = \boldsymbol{f}(\boldsymbol{x}) \cdot \Delta t = \int_\Omega \boldsymbol{F}_k(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k \cdot \Delta t \tag{25}$$

And for the corrected vorticity ω', we can get following equations from Eqs. (17), (20), (21) and (25).

$$A = \frac{\partial \omega'}{\Delta t \cdot \partial x} = \left(\frac{\partial^2 f_y}{\partial x^2} + \frac{\partial^2 f_y}{\partial y^2} \right) \tag{26}$$

$$B = \frac{\partial \omega'}{\Delta t \cdot \partial y} = -\left(\frac{\partial^2 f_x}{\partial x^2} + \frac{\partial^2 f_x}{\partial y^2} \right) \tag{27}$$

$$\frac{\partial \omega'}{\partial t} = \left(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) + \left[-(u^{n+1} \cdot A + v^{n+1} \cdot B) - \left(f_x \frac{\partial \hat{\omega}}{\partial x} + f_y \frac{\partial \hat{\omega}}{\partial y} \right) + \frac{1}{Re} \left(\frac{\partial A}{\partial x} + \frac{\partial B}{\partial y} \right) \right] \cdot \Delta t \tag{28}$$

or

$$\omega'^{n+1} = \omega'^{n} + \left(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) \cdot \Delta t + \left[-(u^{n+1} \cdot A + v^{n+1} \cdot B) - \left(f_x \frac{\partial \hat{\omega}}{\partial x} + f_y \frac{\partial \hat{\omega}}{\partial y} \right) + \frac{1}{Re} \left(\frac{\partial A}{\partial x} + \frac{\partial B}{\partial y} \right) \right] \cdot \Delta t^2 \tag{29}$$

Here $u^{n+1} = \hat{u} + u'$ and $v^{n+1} = \hat{v} + v'$ in Eqs. (28) and (29). The corrected vorticity at n time level is zero $\omega'^{n} = 0$, and then the corrected vorticity at $n + 1$ time level is

$$\omega'^{n+1} = \left(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) \cdot \Delta t + \left[-(u^{n+1} \cdot A + v^{n+1} \cdot B) - \left(f_x \frac{\partial \hat{\omega}}{\partial x} + f_y \frac{\partial \hat{\omega}}{\partial y} \right) + \frac{1}{Re} \left(\frac{\partial A}{\partial x} + \frac{\partial B}{\partial y} \right) \right] \cdot \Delta t^2 \tag{30}$$

The vorticity at $n + 1$ time level is $\omega^{n+1} = \hat{\omega} + \omega'^{n+1}$. The presence of the immersed body is considered in both the corrected velocity equation (Eq. (25)) and the corrected vorticity equation (Eq. (30)).

2.3. Multi-direct forcing process

The multi-direct forcing has been proposed by Wang et al. [13] and Luo et al. [14] to ensure a better satisfaction of the no-slip boundary condition of the velocity at the immersed boundary than that of the original direct forcing scheme proposed by Mohd-Yusof [10] and Fadlun et al. [11]. Here we just rewrite the process of multi-direct forcing process.

By solving Eq. (23), the velocity of the whole flow field $\boldsymbol{u}^{n+1,l}$ is obtained where $n + 1$ is the time level and the upper subscript l represents exerting the direct forcing for the lth time. Then the velocity on the Lagrangian point is

$$\hat{\boldsymbol{u}}_k^l = \sum_{x \in \Omega} \boldsymbol{u}^{n+1,l} \cdot \delta(\boldsymbol{x}_k - \boldsymbol{x}) \cdot h^2 \tag{31}$$

The best result is $\hat{\boldsymbol{u}}_k^l = \boldsymbol{u}_L$, but always the result is $\hat{\boldsymbol{u}}_k^l \neq \boldsymbol{u}_L$. Though the velocity at the immersed boundary can get close to the desired velocity after a long period of time, the no-slip boundary condition is still not satisfied very well. For the sake of getting the velocity on the Lagrangian point much close to the desired velocity, the direct forcing is exerted for the $(l + 1)$th time which makes

$$\boldsymbol{F}_k^{l+1}(\boldsymbol{x}_k) = \frac{\boldsymbol{u}_L - \hat{\boldsymbol{u}}_k^l}{\Delta t} \tag{32}$$

Then the forcing is spread from the Lagrangian points to the Eulerian grids through the Dirac-delta function

$$\boldsymbol{f}^{\,l+1}(\boldsymbol{x}) = \sum_{k=1}^{N} \boldsymbol{F}_k^{l+1}(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \cdot \Delta V_k \tag{33}$$

After exerting the direct forcing for the second time, the velocity of the whole flow field becomes

$$\boldsymbol{u}^{n+1,l+1} = \boldsymbol{u}^{n+1,l} + \boldsymbol{f}^{\,l+1}(\boldsymbol{x}) \cdot \Delta t \tag{34}$$

Thus the velocity on the Lagrangian point at the immersed boundary becomes

$$\hat{\boldsymbol{u}}_k^{l+1} = \sum_{x \in \Omega} \boldsymbol{u}^{n+1,l+1} \cdot \delta(\boldsymbol{x}_k - \boldsymbol{x}) \cdot h^2 \tag{35}$$

The value of $\hat{\boldsymbol{u}}_k^{l+1}$ is expected to be closer to the desired velocity $\boldsymbol{u}_L$ than that of $\hat{\boldsymbol{u}}_k^l$. After NF times of this procedure during one time step, the velocity at the immersed boundary can get much close to the desired velocity. The total forcing exerting on each Lagrangian point $\boldsymbol{F}_k(\boldsymbol{x}_k)$ is the sum of the forcing exerting on each Lagrangian point for the whole NF times, that is:

$$\boldsymbol{F}_k(\boldsymbol{x}_k) = \sum_{i=1}^{NF} \boldsymbol{F}_k^i(\boldsymbol{x}_k) \tag{36}$$

The multi-direct forcing scheme envisages the fact that the formulation of direct forcing is based on a single Lagrangian point and when applying the direct forcing on a group of interactional Lagrangian points and spreading the effect of forcing to Eulerian grids through the interpolation/extrapolation scheme (or the Dirac delta function), the direct forcing will not be so effectively. And the multi-direct forcing scheme can handle this problem.

The procedure of the multi-direct forcing is performed in $O(NF \cdot N)$ operations for an object. And N_P objects need $O(N_P \cdot NF \cdot N)$ operations. When the whole computational domain is full of the objects with the distance between two Lagrangian points lightly smaller than the mesh size, we can get $(N_P \cdot N) \sim N_E$, where N_E is the number of the Eulerian grids. This means the maximum operations is $O(N_E \cdot NF)$. Uhlmann [12] and Le et al. [40] applied an implicit forcing to couple the fluid and the immersed boundary. The coupling is performed through solving the Helmholtz equations which need $O(N_E \ln(N_E))$ operations [19]. The computational cost for present multi-direct forcing scheme is much cheaper than the implicit forcing proposed by Uhlmann [12] and Le et al. [40] especially when the number of Eulerian grids is large.

2.4. Flow solving process

The flow solving process can be summarized as follows.

At time level n, set $\hat{\omega}^{(0)} = \omega^n$, $\hat{\boldsymbol{u}} = \boldsymbol{u}^n$

do i = 1, 3

$$\hat{\omega}^{(i)} = \alpha(i) \cdot \hat{\omega}^n + \beta(i) \cdot \hat{\omega}^{(i-1)} + \left[\frac{1}{Re}\left(\frac{\partial^2 \hat{\omega}}{\partial x^2} + \frac{\partial^2 \hat{\omega}}{\partial y^2}\right) - \hat{u}\frac{\partial \hat{\omega}}{\partial x} - \hat{v}\frac{\partial \hat{\omega}}{\partial y}\right]^{(i-1)} \cdot \gamma(i) \cdot \Delta t \tag{37}$$

$$\frac{\partial^2 \hat{u}}{\partial x^2} + \frac{\partial^2 \hat{u}}{\partial y^2} = -\frac{\partial \hat{\omega}^{(i)}}{\partial y} \tag{38}$$

$$\frac{\partial^2 \hat{v}}{\partial x^2} + \frac{\partial^2 \hat{v}}{\partial y^2} = \frac{\partial \hat{\omega}^{(i)}}{\partial x} \tag{39}$$

enddo

Set $\boldsymbol{u}^{(1)} = \hat{\boldsymbol{u}}$, $\hat{\omega} = \hat{\omega}^{(3)}$.

do j = 1, NF

$$\boldsymbol{u}_k^{(j)} = \sum_{x \in \Omega} \boldsymbol{u}^{(j)} \cdot \delta(\boldsymbol{x}_k - \boldsymbol{x}) \cdot h^2 \tag{40}$$

$$\boldsymbol{F}_k^{(j)}(\boldsymbol{x}_k) = \frac{\boldsymbol{u}_k^{n+1} - \boldsymbol{u}_k^{(j)}}{\Delta t} = \frac{\boldsymbol{u}_L - \boldsymbol{u}_k^{(j)}}{\Delta t} \tag{41}$$

$$\boldsymbol{f}^{(j)}(\boldsymbol{x}) = \sum_{k=1}^{N} \boldsymbol{F}_k^{(j)}(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \cdot \Delta V_k \tag{42}$$

$$\boldsymbol{u}^{(j+1)} = \boldsymbol{u}^{(j)} + \boldsymbol{f}^{\ (j)}(\boldsymbol{x}) \cdot \Delta t \tag{43}$$

enddo

Set $\boldsymbol{u}^{n+1} = \boldsymbol{u}^{(NF+1)}$ and $\boldsymbol{f}(\boldsymbol{x}) = \sum_{k=1}^{N}\left[\sum_{j=1}^{NF} \boldsymbol{F}_k^{(j)}(\boldsymbol{x}_k)\right] \cdot \delta_h(\boldsymbol{x} - \boldsymbol{x}_k) \cdot \Delta V_k = \sum_{j=1}^{NF} \boldsymbol{f}^{\ (j)}(\boldsymbol{x})$.

$$A = \left(\frac{\partial^2 f_y}{\partial x^2} + \frac{\partial^2 f_y}{\partial y^2}\right) \tag{44}$$

$$B = -\left(\frac{\partial^2 f_x}{\partial x^2} + \frac{\partial^2 f_x}{\partial y^2}\right) \tag{45}$$

$$\omega' = \left(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y}\right) \cdot \Delta t + \left[-(u^{n+1} \cdot A + v^{n+1} \cdot B) - \left(f_x \frac{\partial \hat{\omega}}{\partial x} + f_y \frac{\partial \hat{\omega}}{\partial y}\right) + \frac{1}{Re}\left(\frac{\partial A}{\partial x} + \frac{\partial B}{\partial y}\right)\right] \cdot \Delta t^2 \tag{46}$$

$$\omega^{n+1} = \hat{\omega} + \omega' \tag{47}$$

The third-order Runge–Kutta method for the time integration of the vorticity transport equation is applied. The parameters for time integration in Eq. (37) are α = [1,3/4,1/3], β = [0,1/4,2/3] and γ = [1,1/4,2/3] [21]. Eqs. (40)–(43) are the multi-direct forcing process with NF times.

The discrete delta function is chosen as that of Griffith and Peskin [22]

$$\delta_h(\boldsymbol{x} - \boldsymbol{x}_k) = \frac{1}{h^2} d_h\left(\frac{x - x_k}{h}\right) \cdot d_h\left(\frac{y - y_k}{h}\right) \tag{48}$$

where $\boldsymbol{x} = (x,y)$, $\boldsymbol{x}_k = (x_k,y_k)$, h is the Eulerian mesh size, and

$$d_h(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right) & 0 \leqslant |r| < 1 \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}) & 1 \leqslant |r| < 2 \\ 0 & 2 \leqslant |r| \end{cases} \tag{49}$$

The spatial derivatives are discretized using the fourth-order compact finite difference scheme [23] based on non-staggered grid. The Poisson equations for velocity in Eqs. (38) and (39) are discretized with forth-order accuracy proposed by Zhuang and Sun [24], and the Gauss–Seidel iterative method is applied to solve the discrete Poisson equations with the infinite-norm error during two iterative steps smaller than 10^{-5}.

To validate present immersed boundary method based on vorticity–velocity formulations, three test cases including decaying vortices, flow past a stationary circular cylinder and an in-line oscillating cylinder in a fluid at rest are conducted. In the cases of flow past a stationary circular cylinder and flow past an in-line oscillating cylinder, the total hydrodynamic force exerted on the cylinder is

$$\begin{aligned}\boldsymbol{F} &= \oint_{\partial\Omega_P} \tau\cdot\boldsymbol{n}\,\mathrm{d}\sigma = \int_{\Omega_P}(-\nabla p+\mu\nabla^2\boldsymbol{u})\mathrm{d}V = \rho\tfrac{\mathrm{d}\boldsymbol{u}_P}{\mathrm{d}t}V_P-\rho\int_{\Omega_P}\boldsymbol{f}\mathrm{d}V \\ &= \rho\tfrac{\mathrm{d}\boldsymbol{u}_P}{\mathrm{d}t}V_P-\rho\int_1^N\boldsymbol{F}_k(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k = \rho\tfrac{\mathrm{d}\boldsymbol{u}_P}{\mathrm{d}t}V_P-\rho\sum_{k=1}^{N}\left[\sum_{j=1}^{NF}\boldsymbol{F}_k^{(j)}(\boldsymbol{x}_k)\right]\cdot\Delta V_k\end{aligned} \tag{50}$$

where $\partial\Omega_P$ is the surface of the circular cylinder region Ω_P, V_P is the area of the cylinder and $\boldsymbol{u}_P$ is the velocity of the circular cylinder. The term $\rho_f\frac{\mathrm{d}\boldsymbol{u}_P}{\mathrm{d}t}V_P$ is derived from the relation of rigid circular cylinder motion (Eq. (51)).

$$\rho_f\frac{\mathrm{d}\boldsymbol{u}_P}{\mathrm{d}t}V_P=\rho_f\int_{\Omega_P}\left(\frac{\partial\boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right)\mathrm{d}V \quad \text{in } \Omega_P \tag{51}$$

For the case of flow past a stationary circular cylinder, this term is zero $\left(\rho_f\frac{\mathrm{d}\boldsymbol{u}_P}{\mathrm{d}t}V_P=0\right)$.

3. Results

3.1. Accuracy of flow solving

The case of Taylor–Green decaying vortices is calculated to analyze the spacial accuracy of present immersed boundary method based on vorticity–velocity formulations. The analytical solutions of the two-dimensional decaying vortices are

$$u(x,y,t)=-\cos(\pi\cdot x)\cdot\sin(\pi\cdot y)\cdot\exp(-2\pi^2t/Re) \tag{52}$$

$$v(x,y,t)=\sin(\pi\cdot x)\cdot\cos(\pi\cdot y)\cdot\exp(-2\pi^2t/Re) \tag{53}$$

$$\omega(x,y,t)=2\pi\cdot\cos(\pi\cdot x)\cdot\cos(\pi\cdot y)\cdot\exp(-2\pi^2t/Re) \tag{54}$$

Kim et al. [17] calculated this problem in a quadrilateral embedded domain and Uhlmann [12] solved this case in an embedded circle domain. And here, we solve this problem in an embedded circle domain with radius unity and centered at the entire computational domain $\Omega=[-1.5,1.5]\times[-1.5,1.5]$ with $Re=100$. The boundary condition for whole computational domain and the desired velocity at the immersed boundary are given by Eqs. (52)–(54).

The one-norm and two-norm for variable $\phi=(u,\omega)$ are defined as

$$l_{1-\phi}=\frac{1}{N_XN_Y}\sum_{k=1}^{N_XN_Y}|\phi_k^{numerical}-\phi_k^{exact}| \tag{55}$$

$$l_{2-\phi}=\sqrt{\frac{1}{N_XN_Y}\sum_{k=1}^{N_XN_Y}(\phi_k^{numerical}-\phi_k^{exact})^2} \tag{56}$$

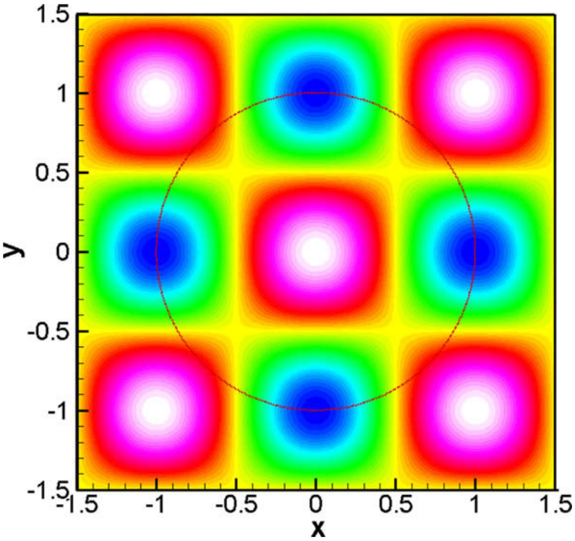

Fig. 1. Position of the embedded circle and the counter of vorticity for time step $\Delta t=0.001$ and the results at $t=0.3$ with mesh size $h=1/32$.

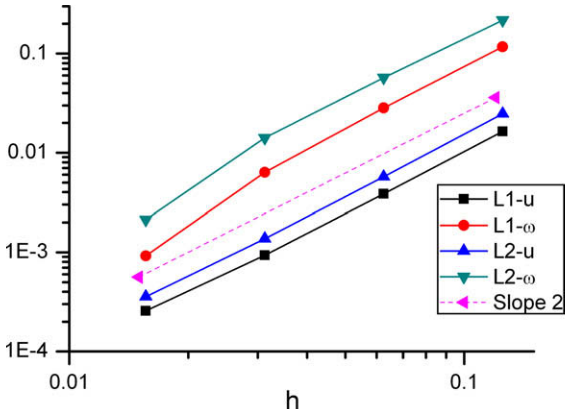

Fig. 2. Convergence of spatial accuracy for $\Delta t = 0.001$, $NF = 20$ (Results at $t = 0.3$ with mesh size $h = 1/8$, 1/16 and 1/32 and 65, 110 and 200 Lagrangian points are used at the immersed boundary of the circle, respectively).

where N_X and N_Y are the total mesh points in x and y direction, respectively. The upper subscripts '*numerical*' and '*exact*' represent the value obtained from present flow solving scheme (Eqs. (37)–(47)) and from Eqs. (52)–(54), respectively. Fig. 1 shows the position of the embedded circle and the counter of vorticity for $\Delta t = 0.001$ and the results at $t = 0.3$ with mesh size $h = 1/32$. Fig. 2 shows the convergence of spatial accuracy for $\Delta t = 0.001$, $NF = 20$ and the results at $t = 0.3$ with mesh size $h = 1/8$, 1/16, 1/32 and 1/64 and 55, 110, 220 and 440 Lagrangian points are used at the immersed boundary of the circle, respectively. The errors of one-norm and two-norm are decreased in slope 2 as the decrement of mesh size. This confirms that the spacial second-order accuracy is obtained in present method. The effect of multi-direct forcing NF will be analyzed by computing the flow past a stationary circular cylinder in Section 3.2.

3.2. Flow past a stationary circular cylinder

A rectangular domain is used to simulate the flow past a stationary circular cylinder using the immersed boundary method and multi-direct forcing based on the vorticity–velocity formulations proposed in this paper. The schematic of the computational domain is shown in Fig. 3 with dimension $20L \times 15L$. The characteristic length is the diameter of the circular cylinder. The non-dimensional mesh size is $h = \frac{1}{32}$ and the total computational grids are 641×481. A constant velocity profile U_∞ is specified at the inflow boundary and a non-reflecting boundary condition [25] is applied at the outflow boundary. The Neumann boundary conditions are imposed on the other boundaries. And the boundary conditions are summarized as follows:

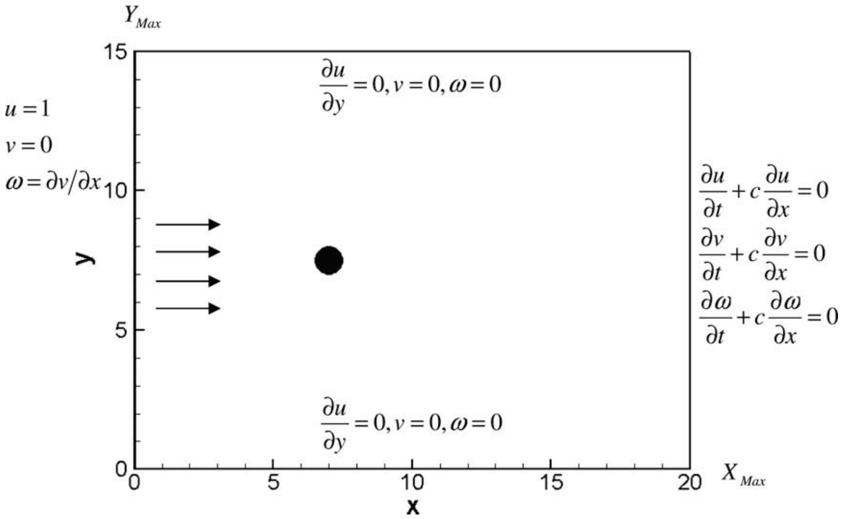
Y_{Max}
15
$\frac{\partial u}{\partial y} = 0, v = 0, \omega = 0$
$u = 1$
$v = 0$
$\omega = \partial v / \partial x$
10
y
5
$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$
$\frac{\partial v}{\partial t} + c\frac{\partial v}{\partial x} = 0$
$\frac{\partial \omega}{\partial t} + c\frac{\partial \omega}{\partial x} = 0$
$\frac{\partial u}{\partial y} = 0, v = 0, \omega = 0$
0
0
5
10
15
20
X_{Max}
x

Fig. 3. Schematic of the computational domain for the case of flow past a stationary circular cylinder.

$$\text{Inflow boundary (at } x = 0) : u = 1, \ v = 0, \ \omega = \partial v / \partial x \tag{57}$$

$$\text{Outflow boundary (at } x = X_{Max}) : \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{58}$$

$$\frac{\partial v}{\partial t} + c\frac{\partial v}{\partial x} = 0 \tag{59}$$

$$\frac{\partial \omega}{\partial t} + c\frac{\partial \omega}{\partial x} = 0 \tag{60}$$

$$\text{Neumann boundary (at } y = 0 \text{ and } y = Y_{Max}) : \frac{\partial u}{\partial y} = 0, \ v = 0, \ \omega = 0 \tag{61}$$

where the parameter c in Eqs. (58)–(60) is the mean velocity at the outflow boundary [17,26].

The simulations of flow past a stationary circular cylinder at $Re = 100$ with different times of multi-direct forcing NF are conducted to show the efficiency of the multi-direct forcing on reducing the error between the calculated velocity and desired velocity at the immersed boundary. The dimensionless mesh size is $h = \frac{1}{32}$, the time step is $\Delta t = 1.25 \times 10^{-3}$ and 110 Lagrangian points are used at the immersed boundary ($N = 110$). The two-norm error of the velocity on the Lagrangian point with respect to no-slip boundary condition is defined as

$$l_2 = \sqrt{\frac{\sum_{k=1}^{N}[(u_k - u_L)^2 + (v_k - v_L)^2]}{N}} \tag{62}$$

where the dimensionless desired velocity at the immersed boundary is $\boldsymbol{u}_L = (u_L, v_L) = (0,0)$.

Fig. 4 shows the correlation of l_2-norm and NF. Here $NF = 1, 2, 4, 8$ and 16 are the times of the multi-direct forcing. As the number of NF increases, the l_2-norm decreases to zero which indicates the calculated velocity at the immersed boundary get close to the desired velocity under the effect of the multi-direct forcing. When $NF = 1$, the l_2-norm is 1.468×10^{-2}, which means that the direct forcing has effect on modifying the velocity at the immersed boundary to the desired velocity. When NF increases to 16, the value of l_2-norm decreases to 2.56×10^{-4}. The multi-direct forcing can make the velocity at the immersed boundary get more close to the desired velocity and the no-slip boundary condition at the immersed boundary is satisfied better for $NF = 16$ than that for $NF = 1$. In this case, the value of NF is chosen as 16, and the l_2-norm of the velocity at the immersed boundary is smaller than 3×10^{-4} which can be treated as the well satisfaction of the no-slip boundary condition at the immersed boundary.

The flow past the stationary circular cylinder at low Reynolds number ($Re = 20$ and 40) and at moderate Reynolds number ($Re = 80$, 100, 200 and 300) are conducted to validate present immersed boundary method based on vorticity–velocity formulations. The dimensionless mesh size is $h = \frac{1}{32}$, the time step is $\Delta t = 1.25 \times 10^{-3}$ and 110 Lagrangian points are used at the immersed boundary.

A steady state with a pair of symmetric weak behind the circular cylinder is obtained at Reynolds numbers 20 and 40 which consists with previous studies [9,29,31]. Fig. 5 shows the streamline and vorticity counter for $Re = 20$(Fig. 5(a)) and $Re = 40$ (Fig. 5(b)). The drag coefficient and wake length including present results and previous numerical and experimental results at $Re = 20$ and $Re = 40$ are summarized in Table 1. The results of the drag coefficient and wake length at $Re = 20$ and $Re = 40$ in present work agree well with the experimental result of Tritton [28] and the numerical results of Russell and Wang [19], Xu and Wang [27], Calhoun [29] and Silva et al. [30].

When the Reynolds number is greater than 47, unsteady vortex shedding occurs [35,36]. Fig. 6(a)–(d) shows the streamline and vorticity counter near the circular cylinder at $t = 200$ for $Re = 80$, 100, 200 and 300, respectively. The results of the

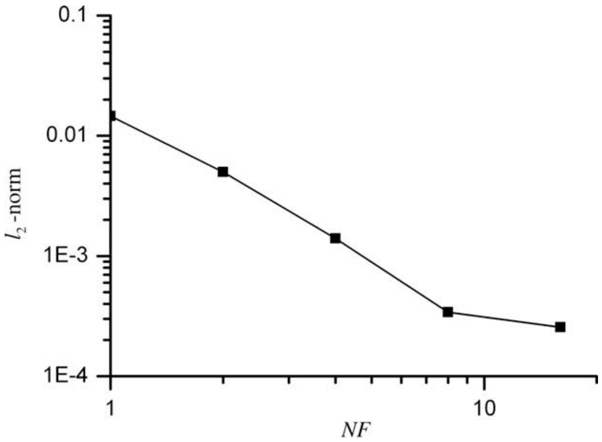

Fig. 4. Correlation of l_2-norm and NF. ($NF = 1, 2, 4, 8$ and 16 are the times of the multi-direct forcing.)

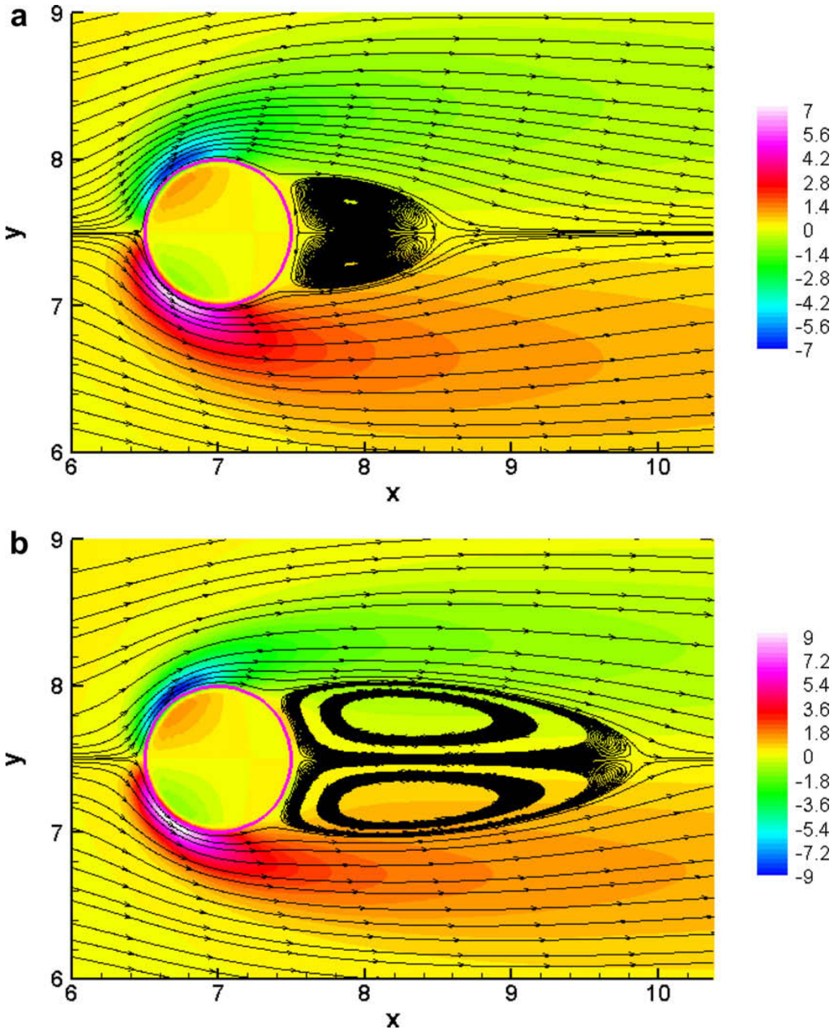

Fig. 5. Streamline and vorticity counter for *Re* = 20(a) and *Re* = 40(b). The dimensionless mesh size is $h = \frac{1}{32}$, the time step is $\Delta t = 1.25 \times 10^{-3}$ and 110 Lagrangian points are used at the immersed boundary.

Table 1
Drag coefficient and wake length for flow past a stationary circular cylinder at *Re* = 20 and *Re* = 40.

Authors	*Re* = 20		*Re* = 40	
	Drag coefficient	Weak length	Drag coefficient	Weak length
Tritton [28][a]	2.22	–	1.48	–
Xu and Wang [27][b]	2.23	0.92	1.66	2.21
Calhoun [29][b]	2.19	0.91	1.62	2.18
Russell and Wang [19][b]	2.22	0.94	1.63	2.35
Silva et al. [30][b]	2.04	1.04	1.54	2.55
Present	2.25	0.978	1.66	2.35

[a] Experimental results.
[b] Numerical results.

drag coefficient at *Re* = 80, 100, 200 and 300 with previous studies are summarized in Table 2. For *Re* = 80, the drag coefficient is 1.428 in present simulation which is close to the numerical results of Silva et al. [30] and Ye et al. [31] and 11% larger than the experimental result of Clift et al. [32]. For *Re* = 100, the drag coefficient is 1.379 in present simulation which is close to the numerical results of Russell and Wang [19], Silva et al. [30], Xu and Wang [27], Calhoun [29] and 11% larger than the experimental result of Clift et al. [32]. For *Re* = 200, the drag coefficient is 1.262 in present simulation which is close to the result of Russell and Wang [19], 11% smaller than the numerical result of Xu et al. [27] and about 9% larger than the numer-

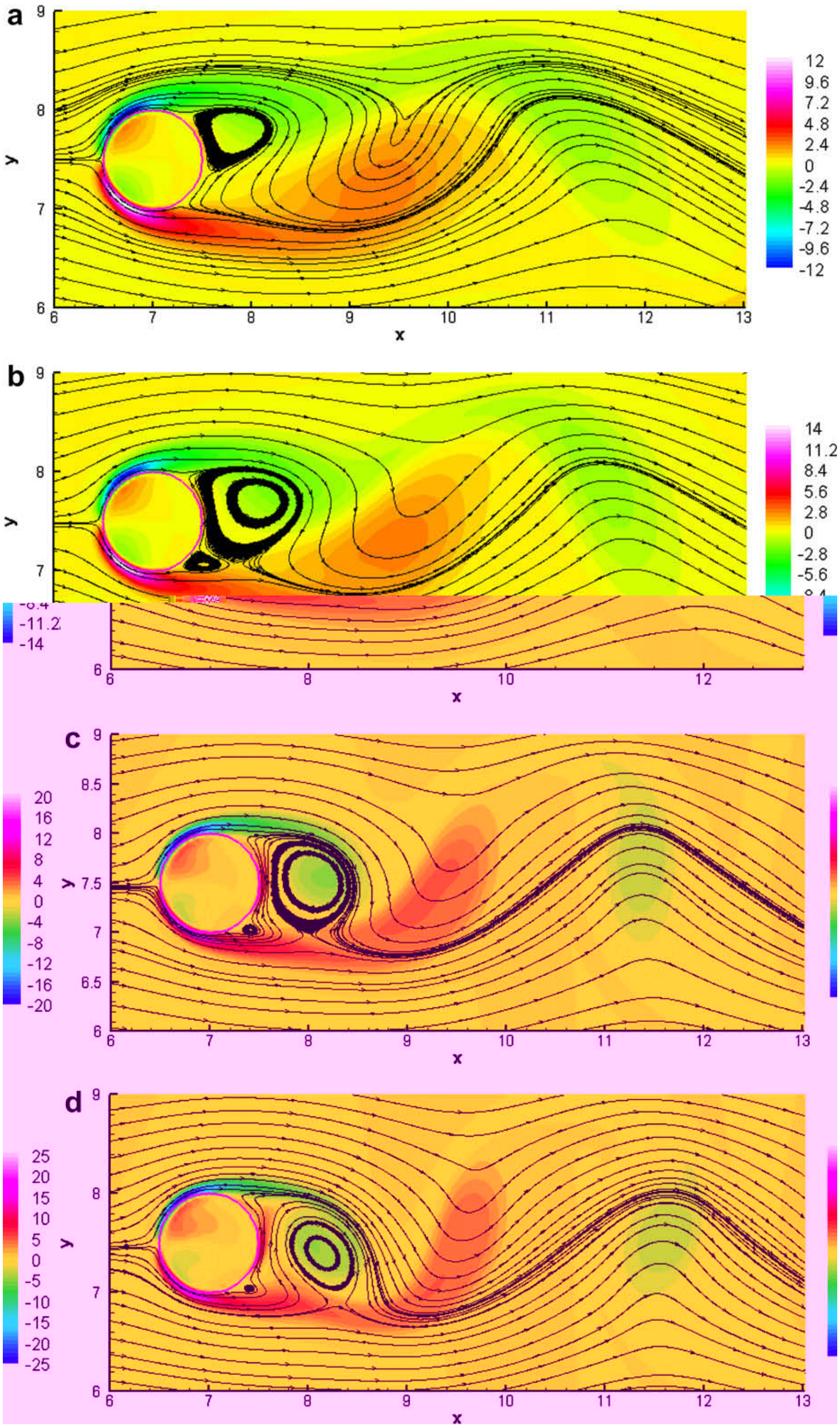

Fig. 6. Streamline and vorticity counter near the circular cylinder at $t = 200$ (a–d represent the cases for $Re = 80$, 100, 200 and 300, respectively).

ical result of Calhoun [29] and the experimental result of Clift et al. [32]. For $Re = 300$, the drag coefficient is 1.174 in present simulation which is close to the experimental result of Clift et al. [32], and smaller than the numerical results of Silva et al. [30] and Ye et al. [31].

Table 2
Drag coefficient for flow past a stationary circular cylinder at *Re* = 80, 100, 200 and 300.

Authors	*Re* = 80	*Re* = 100	*Re* = 200	*Re* = 300
Clift et al. [32][a]	1.28	1.24	1.16	1.13
Russell and Wang [19][b]	–	1.34	1.26	–
Silva et al. [30][b]	1.40	1.39	–	1.27
Xu and Wang [27][b]	–	1.423	1.42	–
Calhoun [29][b]	–	1.33	1.172	–
Ye et al. [31][b]	1.37	–	–	1.38
Present	1.428	1.379	1.262	1.174

[a] Experimental results.
[b] Numerical results.

Table 3 shows the lift coefficient obtained by present method and previous numerical studies at *Re* = 80, 100 and 200. For *Re* = 80, the lift coefficient is 0.263 in present simulation which is close to the result of Le et al. [40]. For *Re* = 100, the lift coefficient is 0.357 in present simulation which is close to the numerical results of Xu and Wang [27] and Le et al. [40] and larger than the numerical results of Braza et al. [33] and Calhoun [29]. For *Re* = 200, the lift coefficient is 0.708 in present simulation which is smaller than the numerical result of Braza et al. [33] and larger than the numerical results of Xu and

Table 3
Lift coefficient for flow past a stationary circular cylinder at *Re* = 80, 100 and 200.

	Re = 80	*Re* = 100	*Re* = 200
Braza et al. [33]	–	0.25	0.75
Xu and Wang [27]	–	0.34	0.66
Calhoun [29]	–	0.298	0.668
Le et al. [40]	0.261	0.346	0.676
Present	0.263	0.357	0.708

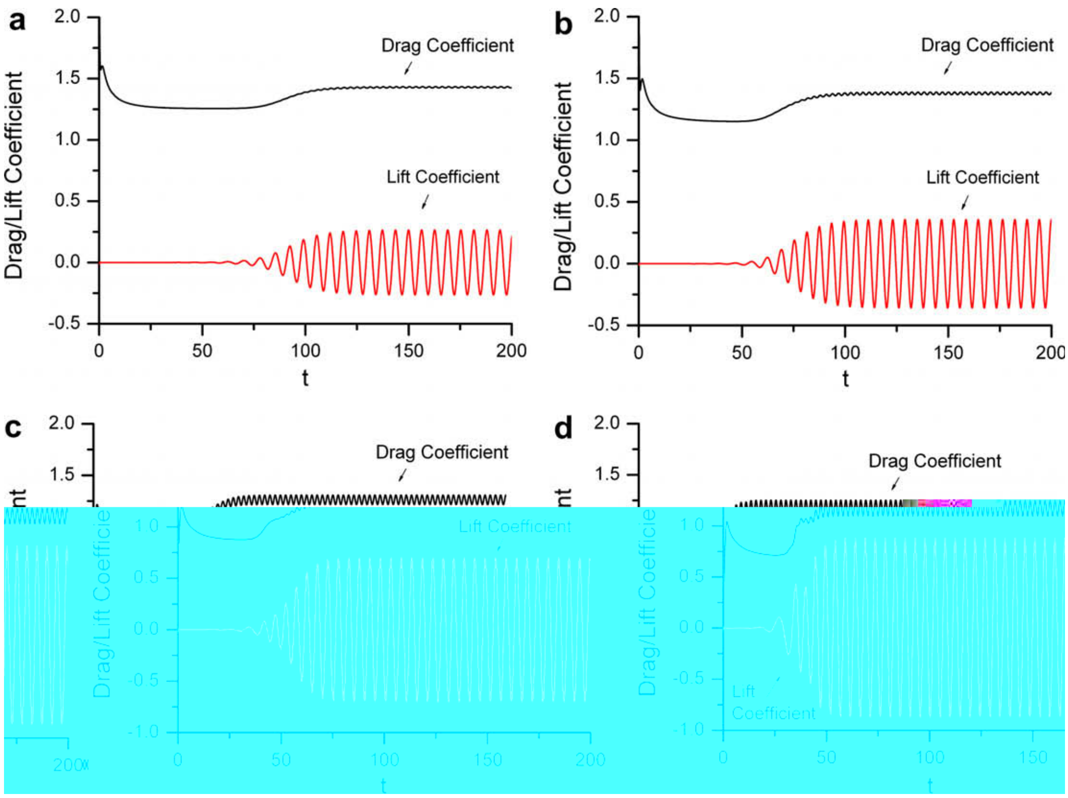

Fig. 7. Time history of the drag coefficient and the lift coefficient (a–d represent the cases for *Re* = 80, 100, 200 and 300, respectively).

Table 4
Strouhal number for flow past a stationary circular cylinder at $Re = 80$, 100, 200 and 300.

Authors	$Re = 80$	$Re = 100$	$Re = 200$	$Re = 300$
Williamson [34][a]	0.15	0.163	0.197	0.20
Xu and Wang [27][b]	–	0.171	0.202	–
Calhoun [29][b]	–	0.175	0.202	–
Ye et al. [31][b]	0.15	–	–	0.21
Present	0.158	0.170	0.195	0.206

[a] Experimental results.
[b] Numerical results.

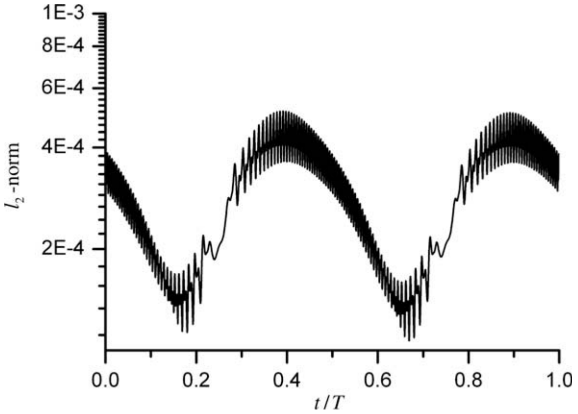

Fig. 8. Time history of the l_2-norm of the velocity for a circular cylinder in-line oscillation at $Re = 100$ and $KC = 5$ in a periodic time T of the oscillation.

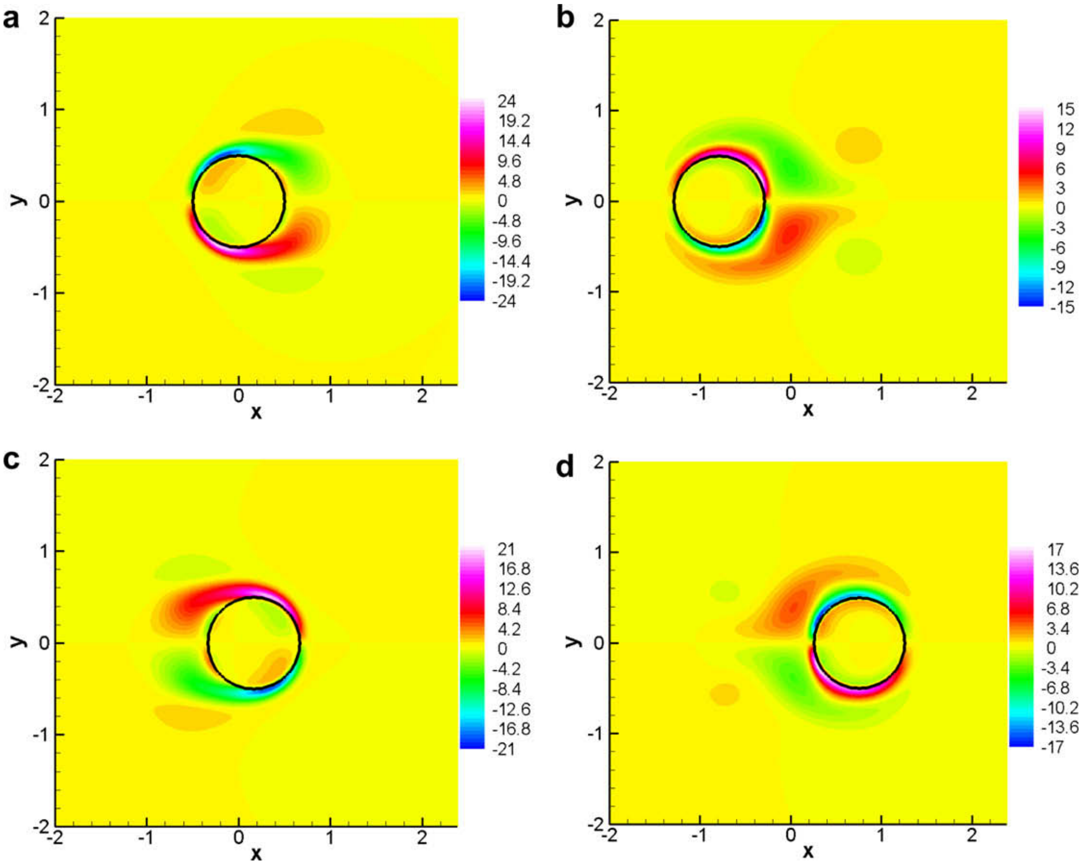

Fig. 9. Vorticity counter at four different phase angles at $Re = 100$ and $KC = 5$ (a–d represent the phase angles 0°, 96°, 192° and 288°, respectively).

Wang [27], Calhoun [29] and Le et al. [40]. The time history of the drag and lift coefficient for *Re* = 80, 100, 200 and 300 is shown in Fig. 7(a)–(d), respectively.

The Strouhal numbers obtained from the Fast Fourier Transform of the lift coefficient time distribution for flow past the stationary at *Re* = 80, 100, 200 and 300 are summarized in Table 4 including present results and previous experimental and numerical results. For *Re* = 80, the Strouhal number is 0.158 in present simulation which is 5% larger than the experimental result of Williamson [34] and the numerical result of Ye et al. [31]. For *Re* = 100, the Strouhal number is 0.17 in present simulation which is 4% larger than the experimental result of Williamson [34], 3% smaller than the numerical result of Calhoun [29] and close to the result of Xu and Wang [27]. For *Re* = 200, the Strouhal number is 0.195 in present simulation which is close to the experimental result of Williamson [34] and smaller than the numerical results of Xu and Wang [27] and Calhoun [29]. For *Re* = 300, the Strouhal number is 0.206 in present simulation which is larger than the experimental result of Williamson [34] and smaller than the numerical results of Ye et al. [31].

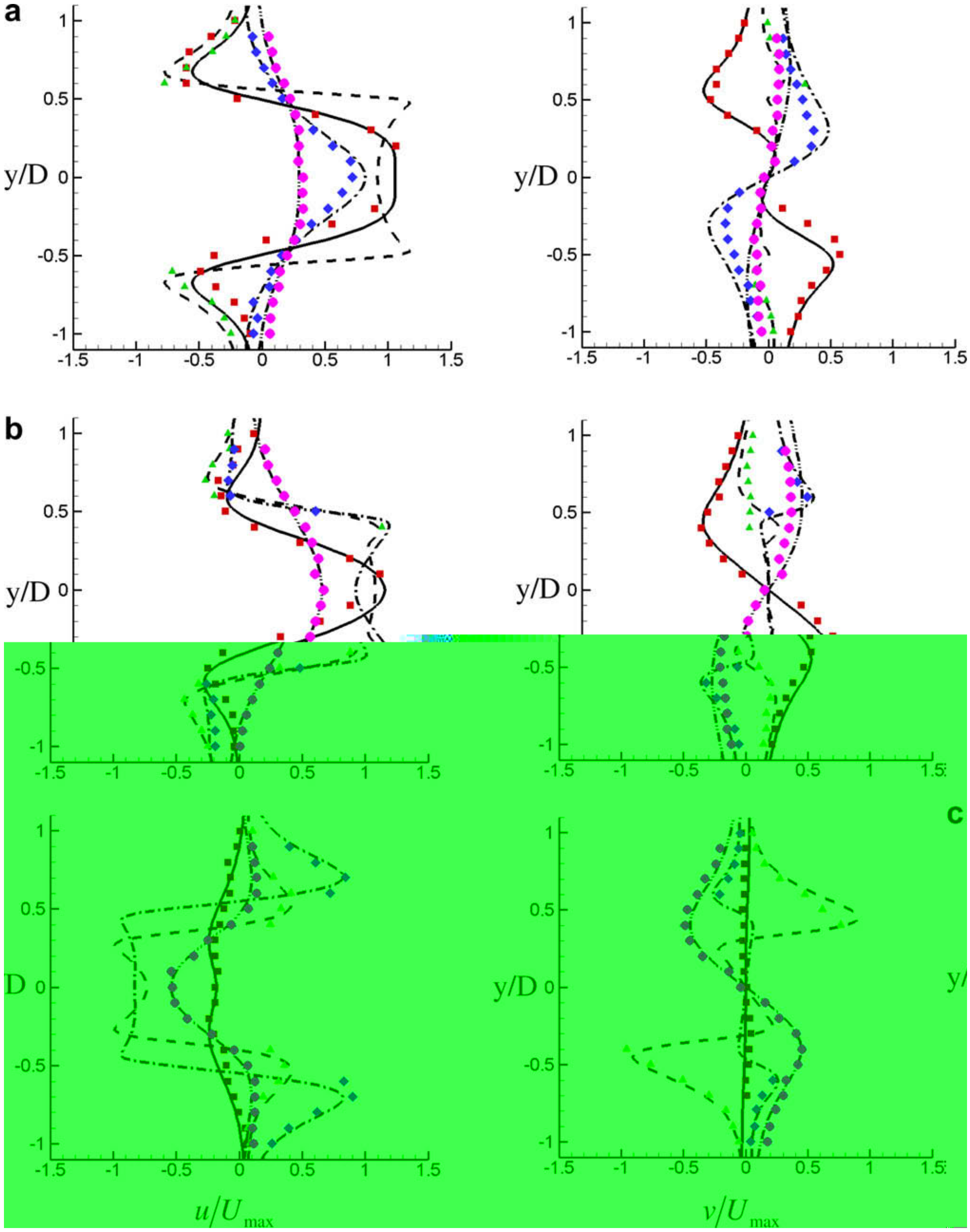

Fig. 10. Computed velocity profiles u/U_{max} and v/U_{max} respectively in the oscillation direction and transverse direction at four different *x* locations and three different phase angles ($\phi = 2\pi ft$): (a) 180°; (b) 210°; (c) 330°. Lines are the present results; symbols are the experimental results of Dütsch et al. [37]: (— and ■) at $x = -0.6D$; (— and ▲) at $x = 0.0D$; (–·– and ♦) at $x = 0.6D$; (–···– and •) at $x = 1.2D$.

3.3. In-line oscillating cylinder in a fluid at rest

The case of an in-line oscillating cylinder in a fluid at reset is studied to validate present method on solving boundary problem. And we expect a smoother solution of the forcing exerting on the moving cylinder at the early stage of the computation by applying the immersed boundary method based on vorticity formulations than that based on velocity–pressure formulations.

The two key parameters for this flow are the Reynolds number, $Re = \frac{\rho \cdot U_{\max} \cdot D}{\mu}$, where $U_{\max}$ is the maximum velocity of the cylinder, D is the diameter of the cylinder, and the Keulegan–Carpenter number, $KC = \frac{U_{\max}}{f \cdot D}$, where f is the characteristic frequency of the oscillation. The in-line motion of the cylinder is described by a simple harmonic oscillation,

$$x(t) = -A \cdot \sin(2\pi f t) \tag{63}$$

where A is the amplitude of the oscillation and $x(t)$ is the position of the cylinder in the direction of the oscillation. The two key parameters in present work are chosen as $Re = 100$ and $KC = 5$ corresponding to the experimental and numerical study of Dütsch et al. [37], the numerical studies of Yang and Balaras [38] and Choi et al. [39].

The characteristic length is chosen as the diameter of the cylinder D, the characteristic velocity is chosen as the maximum velocity of the cylinder $U_{\max}$, and the dimensionless computational domain is 20×20 with a uniform mesh size $h = 1/40$. The total computational mesh grid is 801×801. Neumann boundary condition for velocity is applied at all boundaries of the computational domain. The dimensionless time step is $\Delta t = 1.0 \times 10^{-3}$ and 138 Lagrangian points are used at the immersed boundary. In this case, the value of NF is 16 and the l_2-norm of velocity defined in Eq. (62) is smaller than 5×10^{-4}. The time history of the l_2-norm of velocity in a period of oscillation is shown in Fig. 8.

Fig. 9 shows the vorticity counter at four different phase angles (a $\sim$ d represent the phase angles 0°, 96°, 192° and 288°, respectively) in a periodic vortex shedding corresponding to the studies of Dütsch et al. [37] and Yang et al. [38].

Fig. 10 shows the computed velocity profiles in the oscillation direction and transverse direction at four different x locations ($x = -0.6D$, $0D$, $0.6D$ and $1.2D$) and three different phase angles ($\phi = 2\pi ft = [180°, 210°, 330°]$), in comparison with the experimental results of Dütsch et al. [37]. The agreement is very good.

Fig. 11 shows the predicted time history of the in-line force F_x calculated by Eq. (50) and compares with the results of Dütsch et al. [37] in a period of the oscillation. Again, a good agreement is observed. This indicates that present method can calculate the force acting on the cylinder's surface accurately.

Fig. 12 shows the predicted time history of the in-line force F_x obtained by present vorticity–velocity formulations and compares with the results obtained by the velocity–pressure formulations. Here, we use the method of velocity–pressure formulations described by Wang et al. [13] to compute the same case. At the early stage of the computation, the curve of the force obtained by present vorticity–velocity formulations is smooth as we expected while there is some oscillation of the force obtained by the velocity–pressure formulations. After some time steps, the convergence of solving the pressure field is reached and the oscillation of the force obtained by the velocity–pressure formulations is vanished. It indicates that a more accuracy solution of the coupling between the moving object and flow field is obtained by present vorticity–velocity formulations. This is the main advantage of the present IB method based on the vorticity–velocity formulations. And it can be applied to calculating the trajectories of sudden start objects and particles in multiphase flows smoothly and accurately.

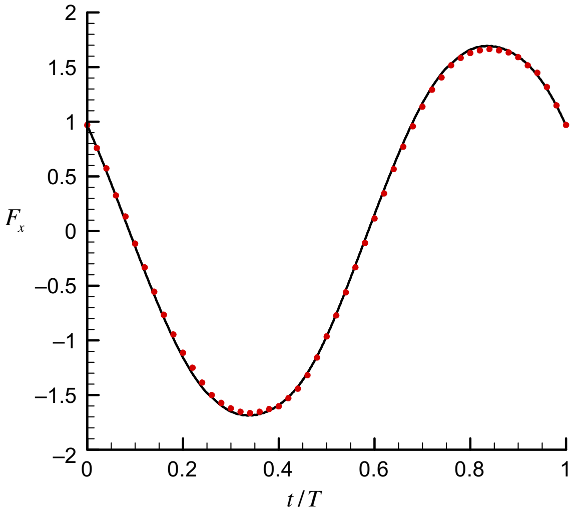

Fig. 11. Predicted time history of the in-line force F_x for a circular cylinder in-line oscillation at $Re = 100$ and $KC = 5$ in a periodic time T of the oscillation. Line represents present results and symbols represent the results of Dütsch et al. [37].

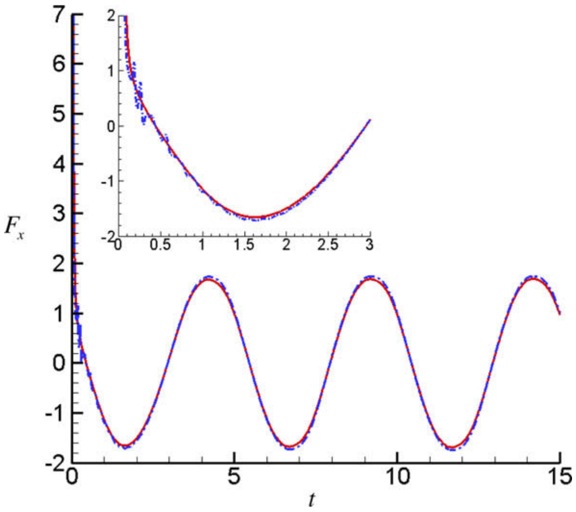

Fig. 12. Predicted time history of the in-line force F_x for a circular cylinder in-line oscillation at $Re = 100$ and $KC = 5$. The solid line (—) represents the results of present vorticity–velocity formulations and the dash dot line (-·-) represents the results of the velocity–pressure formulations.

4. Conclusions

A new immersed boundary method based on vorticity–velocity formulations for the simulation of 2D incompressible viscous flow is proposed in present work. The velocity and vorticity are, respectively, divided into two parts: one is the velocity and vorticity without the influence of the immersed boundary, and the other is the corrected velocity and vorticity derived from the influence of the immersed boundary. The corrected velocity is obtained from the concept of the direct forcing, and in present work, the multi-direct forcing proposed by Wang et al. [13] and Luo et al. [14] is applied to ensure the well satisfaction of the no-slip boundary condition at the immersed boundary. The corrected vorticity is derived from the governing equations (vorticity transport equation and velocity Poisson equations). This is the main contribution of present work.

To validate present method, three cases including decaying vortices, flow past a stationary circular cylinder and an in-line oscillating cylinder in a fluid at rest are conducted. In the case of decaying vortices, spatial second-order accuracy of present method is obtained. In the case of flow past a stationary circular cylinder, the effect of the multi-direct forcing is verified. The good agreements of the drag coefficient, the lift coefficient and the Strouhal number between present results and previous numerical and experimental results validate the accuracy of present method. And at last, the case of an in-line oscillating cylinder in a fluid at rest is studied. Again, the good agreements of the results between present work and previous numerical and experimental results indicates the validity and accuracy of present immersed boundary method based on vorticity–velocity formulations. The solution of the forcing exerting on the oscillating cylinder obtained by present method is smoother than that of the velocity–pressure formulation especially at the early stage of the computation. This is the main advantage of the present immersed boundary method based on the vorticity–velocity formulations.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Nos. 50736006 and 50776080). We are grateful to that.

References

[1] H.H. Hu, N.A. Patankar, N. Zhu, Direct numerical simulation of fluid–solid systems using the arbitrary Lagrangian Eulerian technique, J. Comput. Phys. 169 (2001) 427–462.
[2] R. Glowinski, T.W. Pan, T.I. Hesl, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (1999) 755–794.
[3] A. Prosperetti, H.N. Oguz, Physalis: a new $O(N)$ method for the numerical simulation of disperse systems: potential flow of spheres, J. Comput. Phys. 167 (2001) 196–216.
[4] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flows, J. Comput. Phys. 114 (1994) 146–159.
[5] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[6] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[7] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[8] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip boundary with an external force field, J. Comput. Phys. 105 (1993) 354–366.
[9] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, J. Comput. Phys. 123 (1996) 450–465.

[10] J. Mohd-Yusof, Combined Immersed Boundary/B-splines Method for Simulations of Flows in Complex Geometry, Annual Research Briefs, Center for Turbulence Research, 1997, pp. 317–327.

[11] E.A. 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.

[12] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys 209 (2005) 448–476.

[13] Z. Wang, J. Fan, K. Luo, Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles, Int. J. Multiphase Flow 34 (2008) 283–302.

[14] K. Luo, Z. Wang, J. Fan, K. Cen, Full-scale solutions to particle-laden flows: multiphase forcing and immersed boundary method, Phys. Rev. E 76 (2007) 066709.

[15] Z.G. Feng, E.E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (2005) 20–51.

[16] T.R. Lee, Y.S. Chang, J.B. Choi, D.W. Kim, W.K. Liu, Y.J. Kim, Immersed finite element method for rigid body motions in the incompressible Navier–Stokes flow, Comput. Methods Appl. Mech. Eng. 197 (2008) 2305–2316.

[17] J. Kim, D. Kim, Haecheon Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.

[18] C. Daives, P.W. Carpenter, A novel velocity–vorticity formulation of the Navier–Stokes equations with applications to boundary layer disturbance evolution, J. Comput. Phys. 172 (2001) 119–165.

[19] D. Russell, Z.J. Wang, A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (2003) 177–205.

[20] T.B. Gatski, Review of incompressible fluid flow computations using the vorticity–velocity formulation, Appl. Numer. Meth. 7 (1991) 227–239.

[21] X. Deng, H. Zhang, Developing high-order weighted compact nonlinear schemes, J. Comput. Phys. 165 (2000) 22–44.

[22] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.

[23] S.K. Lele, Compact finite difference scheme with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[24] Y. Zhuang, X.H. Sun, A high order fast direct solver for singular Poisson equations, J. Comput. Phys. 171 (2001) 79–94.

[25] I. Orlanski, A simple boundary condition for unbounded hyperbolic flow, J. Comput. Phys. 21 (1976) 251–269.

[26] L.L. Pauley, P. Moin, W.C. Reynolds, The structure of two-dimensional separation, J. Fluid Mech. 220 (1990) 397–411.

[27] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454–493.

[28] D.J. Triton, Experiments on the flow past a circular cylinder at low Reynolds number, J. Fluid Mech. 6 (1959) 547–567.

[29] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction–vorticity equations in irregular regions, J. Comput. Phys. 176 (2002) 231–275.

[30] A.L.F.L.E. Silva, A. Silveira-Neto, J.J.R. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, J. Comput. Phys. 189 (2003) 351–370.

[31] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.

[32] R. Clift, J.R. Grace, M.E. Weber, Bubbles, Drops, and Particles, Academic Press, New York, 1978. p. 154.

[33] 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.

[34] C.H. Williamson, Vortex dynamics in the cylinder wake, Annu. Rev. Fluid Mech. 28 (1996) 477–539.

[35] C.P. Jackson, A finite element study of the onset of vortex shedding in flow past variously shaped bodies, J. Fluid Mech. 182 (1987) 23–45.

[36] M. Provensal, C. Mathis, L. Boyer, Benard-von Karman instability: transient and forced regimes, J. Fluid Mech. 182 (1987) 1–22.

[37] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, J. Fluid Mech. 360 (1998) 249–271.

[38] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (2006) 12–40.

[39] J.I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys. 224 (2007) 757–784.

[40] D.V. Le, B.C. Khoo, K.M. Lim, An implicit-forcing immersed boundary method for simulating viscous flow in irregular domains, Comput. Methods Appl. Mech. Eng. 197 (2008) 2119–2130.

[41] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.